AMATH482
HW5 Background Subtraction in Video Streams
Dean Wang

1. Abstract
By using Dynamic Mode Decomposition method on two video clips, we need separate them to foreground and background output.

2. Introduction and Overview
We use Dynamic Mode Decomposition (MDM) to downgrade to the ow-dimensionality in experimental data. Even though we learn POD before, MDM do not have to rely on a given set of governing equation. The unique point of the method is that can solve the condition of not necessarily orthogonal. Oscillations, decay, and growth are the only happened in time.

3. Theoretical Background
3.1 Setup

We first define N is the number of spatial points saved per unit time snapshot and M is the number of snapshots taken. The data is collected regularly. $t_{m+1} = t_m + \Delta t$. The snapshots are: $U_{(x,t_m)} = \begin{bmatrix} U_{(x_1,t_m)} \\ U_{(x_2,t_m)} \\ \vdots \\ U_{(x_n,t_m)} \end{bmatrix}$

3.2 The Koopman Operator

We suppose that Koopman Operator A is a linear time independent operator, and it can collect the data from t_j to t_{j+1}. $x_{j+1} = Ax_j$. x_j is N-dimensional vector collected at time j.

3.3 Dynamic Mode Decomposition

The best Koopman Operator to represent the data collected we use x_j to represent snapshot of the data at time t_j: $X_1^{M-1} = [x_1 \quad x_2 \quad x_2 \quad \cdots \quad x_{M-1}]$. We need to remember that x_M wasn't included in our Krylov basis, we need to add the residual "r" in the function. Then, we use SVD to rewrite the function: $X_2^M = AU\sum V^* + re_{M-1}^T$. Finally, multiplying $U * r = 0$ on the left and multiplying V and then $\sum^{-1}$ on the right to get: $U^* X_2^M = U^* AU\sum V^*$ and $U^* AU = U^* X_2^M V\sum^{-1}$

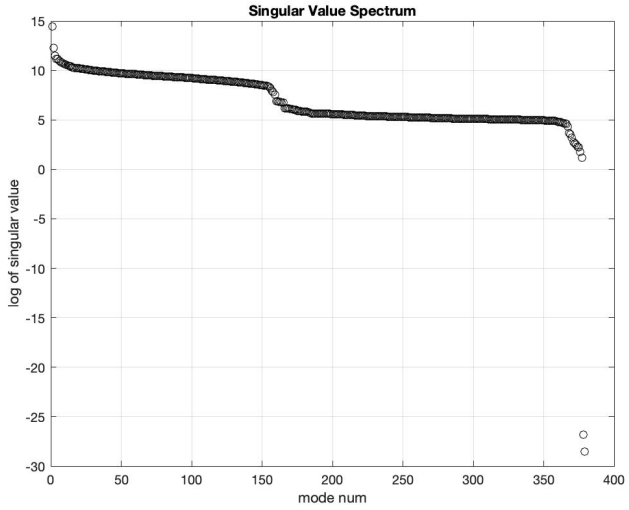
3.4 Follow the guideline from HW5 PDF, we will finally get the function: $X = X_{DMD}^{Low-Rank} + X_{DMD}^{Sparse}$

4. Algorithm Implementation and Development
I first load the video. Then, I change the file path here to choose two different videos to analyze. I turn each frame into a row of vector and use SVD analysis to analyze the original X matrix.

Then, I use the function DMD. I first clear some variable since the memory would not be enough and define the time between each frame. Moreover, I compute eigenvalues + eigenvectors, extract eigenvalues, and eigenvalues in exp form. In addition, I clear some variable since the memory would not be enough. Finally, by using pseudoinverse, I get initial condition and use DMD to reconstruct the X from initial condition.

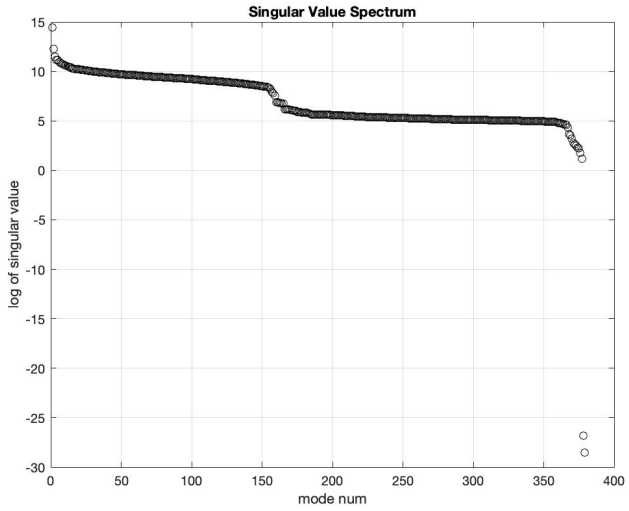
Then, going back to the main code, I construct X_sparse in low rank as the HW5 guidline instructed and plot of original video frame, foreground, and background with selected frame equal to 200. Finally, showing result plot of each frame.

5. Computational Result

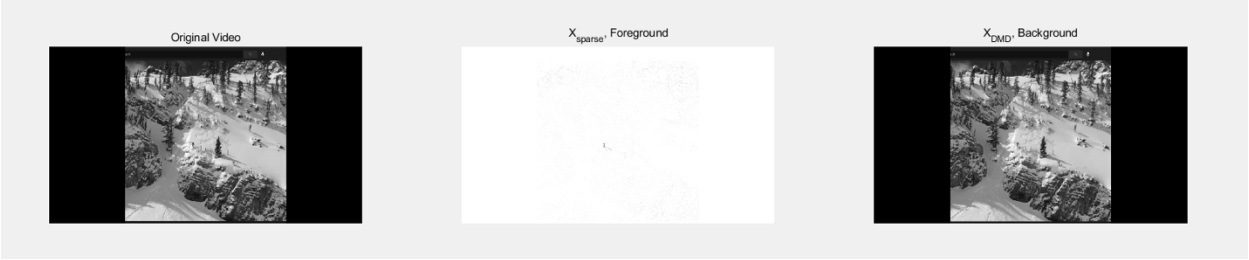


Singular Value Spectrum

According to the graph, we could realize that the result of Background is better than that of Foreground.



Singular Value Spectrum



According to the graph, we could realize that the result of Background is kind of clear. However, it's white for the Foreground. I guess the reason should be moving object is super small, which cause the result.

6. Summary and Conclusions

Dynamic Mode Decomposition is a good method in the analyzing two videos. Most of the movement is captured by the Background. In the car example, the unclear in the Foreground is because of the high speed, and the white in the second case is because of the smaller size of the moving objective than of whole background.

Appendix1

DMD.m: use DMD to reconstruct the X from initial condition

Appendix2

1. HW5.m

```
clear
clc

filepath1='ski_drop_low.mp4';
filepath2='monte_carlo_low.mp4';
videodata=VideoReader(filepath2);
%change the filepath here to choose two different video to analyse
Nnum=videodata.Height*videodata.Width;%number of frames

%%

X=zeros(Nnum,videodata.NumFrames);
for i= 1:v
    ideodata.NumFrames


    I=read(videodata,i);
    I=rgb2gray(I);
%     subplot(2,1,1)
%     imshow(I)
%     subplot(2,1,2)
%     imshow(I_low/max(max(I_low)))
%     pause(0.0001)
    uxt=reshape(I,Nnum,1);
    X(:,i)=uxt;

end
disp('finish constructing X matrix')
%%
[~ ,S0 ,~]=svd(X,'econ');
singulvalues=diag(S0);
figure(1)
plot(log(singulvalues),'ko')
xlabel('mode num')
ylabel('log of singular value')
title('Singular Value Spectrum')
grid on
lowrank=find(log(singulvalues)>4,1,'last');
disp('finish SVD analysis of X matrix')
%%
%DMD
[X_DMD,X1]=DMD(X,videodata,lowrank);
disp('finish DMD analysis')
%%
```

```
X_sparse=X1-abs(X_DMD);
R = X_sparse.*(X_sparse<0);
X_sparse = X_sparse-R;

%%
sf=200;
I=X(:,sf);
I=reshape(I,540,960);
figure()
subplot(1,3,1)
imshow(uint8(I))
title('Original Video')

I=X_sparse(:,sf);
I=reshape(I,540,960);
subplot(1,3,2)
imshow(imcomplement(uint8(I)*2))
%increase the lightness of foreground by *2
%reverse the image to make the forground image more clear
%as the figure is a dot in ski video
title('X_{sparse}, Foreground')

I=X_DMD(:,sf);
I=reshape(I,540,960);
subplot(1,3,3)
imshow(uint8(I))
title('X_{DMD}, Background')
%%
disp('showing result please wait')
figure()
for i =1:videodata.NumFrames-1

    I=X(:,i);
    I=reshape(I,540,960);
    subplot(1,3,1)
    imshow(uint8(I))
    drawnow
    title('Original Video')
    I=X_sparse(:,i);
    I=reshape(I,540,960);
    subplot(1,3,2)
%     imshow(uint8(I)*1.3)
    imshow(imcomplement(uint8(I)*2))
    drawnow
    title('X_{sparse}, Foreground')
    I=X_DMD(:,i);
    I=reshape(I,540,960);
    subplot(1,3,3)
    imshow(uint8(I))
    drawnow
    title('X_{DMD}, Background')
    disp(['showing frame ',num2str(i),' / ',num2str(videodata.NumFrames)])
end
```

2. DMD.m

```
function [X_DMD,X1]=DMD(X,videodata,lowrank)
clearvars -except X lowrank videodata singulvalues
```

```
%DMD
X1 = X(:,1:end-1);
X2 = X(:,2:end);

dt=videodata.Duration/videodata.NumFrames;

% lowrank=360;

[U, Sigma, V] = svd(X1,'econ');
Ulr=U(:,1:lowrank);
Slr=Sigma(1:lowrank,1:lowrank);
Vlr=V(:,1:lowrank);

S = Ulr' * X2 * Vlr*diag(1./diag(Slr));

[eV, D] = eig(S); % compute eigenvalues + eigenvectors
mu = diag(D); % extract eigenvalues
omega = log(mu)/dt;% eigenvalues in exp form

clearvars -except X lowrank videodata singulvalues X1 X2 Vlr Slr eV omega dt
Nnum
%clear some variable since the memery would not be enough
Phi = X2*Vlr/Slr*eV;

y0=Phi\X1(:,1);

X_DMD=zeros(lowrank,videodata.NumFrames-1);

for i=1:videodata.NumFrames-1
   X_DMD(:,i)=y0.*exp(omega*dt*(i-1));
end

X_DMD=Phi*X_DMD;

end
```