Background Subtraction in Video Streams

Mingzhong Deng

March 15, 2019

Abstract

This project uses singular value decomposition (SVD) and dynamic mode decomposition (DMD) to analyze videos of objects in motion and try to seperate the background and foreground. The moving objects in three videos are successfully captured by DMD.

1 Introduction and Background

In this project, I analyze three video files with moving objects with the Sigular Value Decomposition (SVD) technique and Dynamic Mode Decomposition (DMD) algorithm in order to subtract the background of each of the three videos. The project uses SVD for reducing the dimension of the videos by finding the highest modes to preserve the information after reducing the dimension. DMD extends the power of SVD data from the dynamic systems. Subtracting background of videos of moving objects is a good analysis because the time dynamic is known ahead of time.

The three videos includes:

- Pouring Tea into two glass cups. In the video, I purposely put a dear figure on the side of two glass cups to test out if DMD is able to extract the nonmoving dear from the original video.
- Lecture Video by Professor Kutz, with a whiteboard behind Professor Kutz and a couple of students sitting between the position of Professor Kutz and the camera.
- Fruits rolling on the floor. Multiple kinds of fruits, lemon, orange, lime, avacado, and apple are rolled from the left to right side of the camera without making a stop.

2 Theoretical Background

Dynamic Mode Decomposition is a powerful techinique for the discovery of dynamical systems from high-dimensional data. It is originated as a method to decompose complex flows into a simple representation based on spatio-temporal coherent structures. It could be used as an combination of spatial dimensionality-reduction techniques. It could be used for three major tasks: diagnostics, state estimation and future state prediction, and control. In practice, when the state dimension n is large, the matrix A may be intractable to analyze directly. Instead, DMD circumvents the eigendecomposition of A by considering a rank-reduced representation in terms of a POD-projected matrix $\widetilde{A}$. The following is the algorithm for DMD:

- First, take the singular value decomposition (SVD) of X.

$$X = U\Sigma V^*$$

- The matrix A from may be obtained by using the pseudo-inverse of X obtained via the SVD.

$$A = X'V\Sigma^{-1}U^*$$

- Compute the eigendecomposition of $\widetilde{A}$.

$$\widetilde{A}W = W\Lambda$$

  where columns of W are eigenvectors and Λ is a diagonal matrix containing the corresponding eigenvalues λ_k.

- Finally, we may reconstruct the eigendecomposition of A from W and Λ. In particular, the eigenvalues of A are given by Λ and the eigenvectors of A (DMD modes) are given by columns of Φ.

$$\Phi = X'V\Sigma^{-1}W$$

3 Algorithm Implementation and Development

The algorithm for four videos with different length is the same in the project.

- Read video files, extract the data, construct a matrix for the data.
- Convert the data from RGB colors to grayscale.
- Perform the DMD algorithm on the grayscale video matrix.
- Extract the background matrix from the result of DMD matrix.
- Extract the foreground matrix from background and original video matrix.
- Plot the three timeframes of each of the three matrix to see the difference.

4 Computational Results

4.1 Pouring Tea Video

Figure 1 shows the singular values from the tea pouring video, we can see that the first mode contributes around 46% to the actual video, and the second highest mode contributes only around 3%.

Figure 2 shows the pictures of the original video (top), the background (middle), and the foreground (bottom) of the video in three different time frames. Since the shape of two cups and the dear model did not move at all during the entire video, they are being captured as part of the background. During the video, the tea bottle is moved a couple times for pouring tea in different cups, but they need to stay still in a certain position for pouring, so the shape of the tea bottle is captured in both of the background and foreground. This is the same for the height of the actual tea. Only part of the tea is captured because it was staying still while the tea bottle is pouring tea in the other cup. Overall, the DMD algorithm successfully captured the movement in the video and it is a great tool for background extraction if the movement is more consistant. It could work better if the tea pouring action is more consistant on one of the two cups.

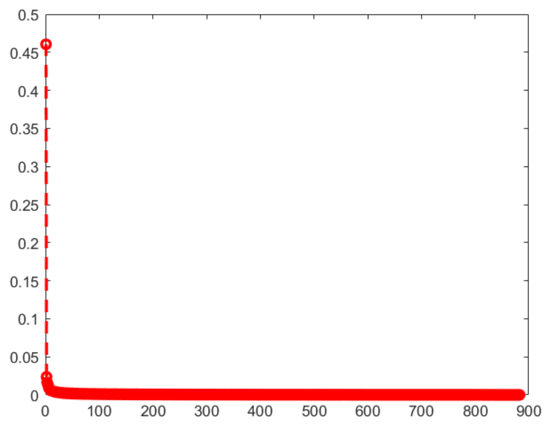

Figure 1: Singular value proportion of the Tea Pouring Video data.

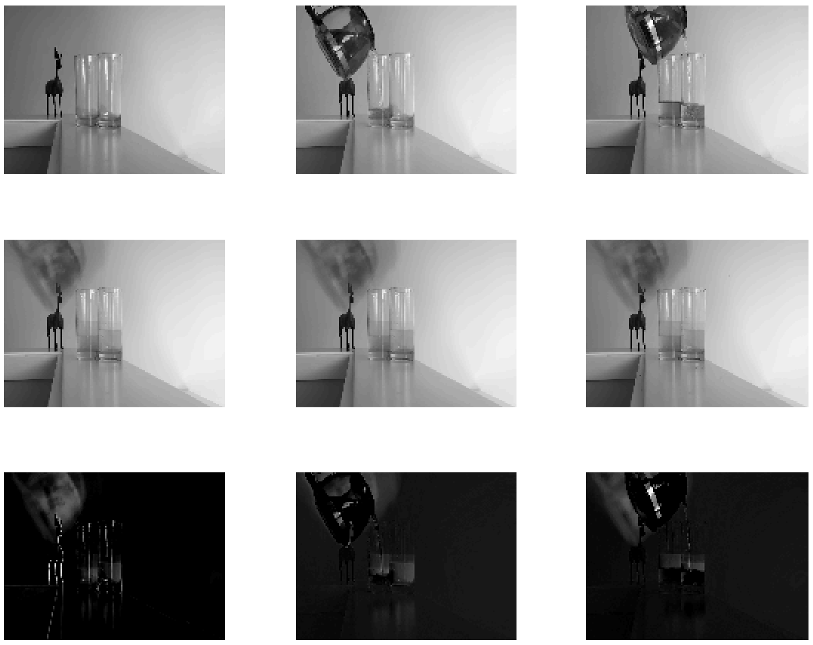

Figure 2: Frame 20, 120, 240 of the original video (Top), background video (Middle), and foreground video (Bottom).

4.2 Professor Kutz Lecture Video

Figure 3 shows the singular values from the lecture video, we can see that the first mode contributes around 38% to the actual video, and the second highest mode contributes only around 5%. Figure 4 shows the pictures of the original video (top), the background (middle), and the foreground (bottom) of the video in three different time frames.

Since the shape of two student and the whiteboard did not move at all during the entire video, they are being captured as part of the background. During the video, Professor Kutz is moving a couple times for his lecture, but he did stand in the same position for some time, so the shape of him is captured in both of the background and foreground. In the lower right picture, we can see that Professor Kutz's hand and tie is shown clearly, which means he is in movement and the DMD algorithm captured that successfully.

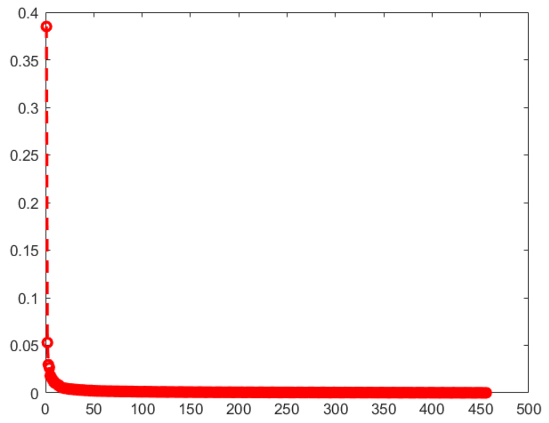

Figure 3: Singular value proportion of the Lecture Video data.

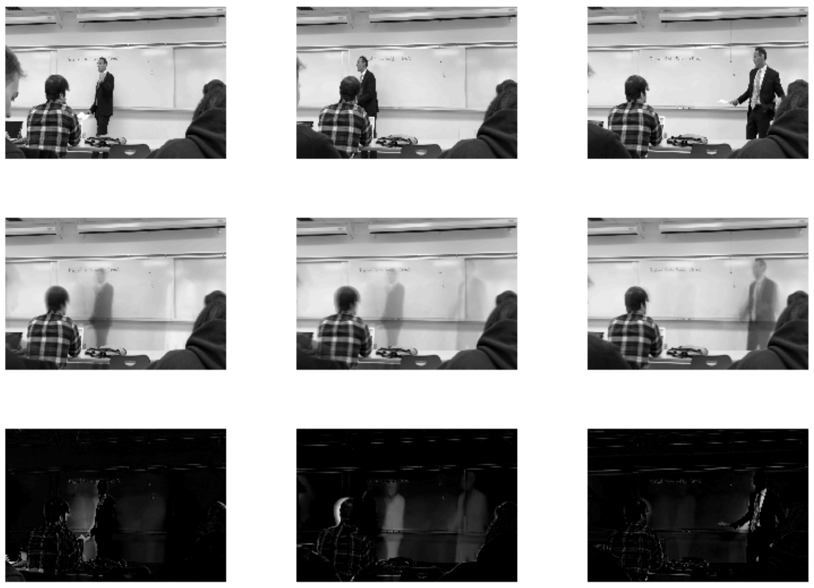

Figure 4: Frame 20, 360, 640 of the original video (Top), background video (Middle), and foreground video (Bottom).

4.3 Fruits Rolling Video

Figure 5 shows the singular values from the rolling fruit video, we can see that the first mode contributes around 57% to the actual video, and the second highest mode contributes only around 3%. Figure 6 shows the pictures of the original video (top), the background (middle), and the foreground (bottom) of the video in three different time frames.

Since the shape and outline of the floor did not move at all during the entire video, they are being captured as part of the background clearly. During the video, different fruits is rolled from the left to the right and did not stop at all in the video, so the shapes of different fruits is never captured in the background movie and successfully captured in the foreground video. In the bottom three picture, we can clearly see the lemon, line and part of the avacado, which means the DMD algorithm is useful to capture movement and remove background in data that is constantly moving.

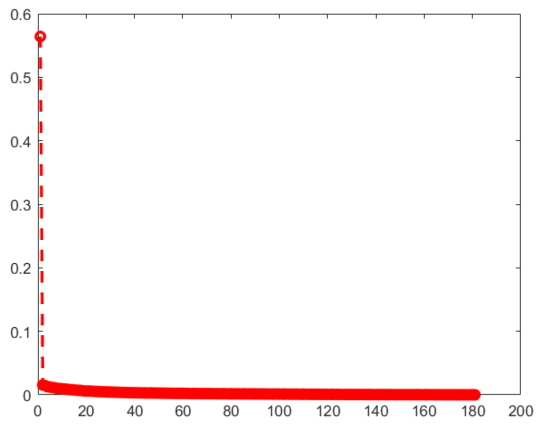

Figure 5: Singular value proportion of the Fruits on Floor Video data.

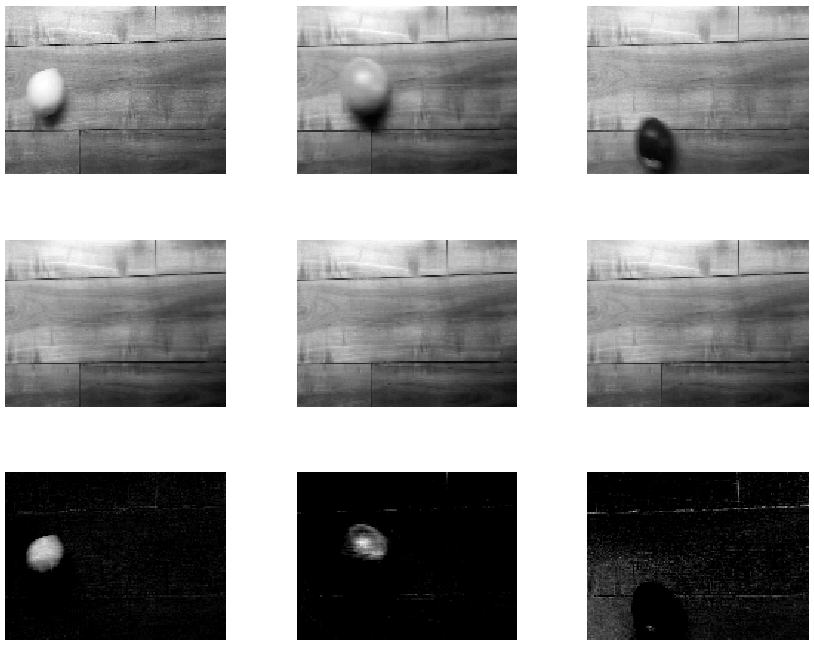

Figure 6: Frame 20, 80, 150 of the original video (Top), background video (Middle), and foreground video (Bottom).

5 Summary and Conclusions

Based on the graphical result of the three testing videos, using the DMD algorithm to seperate background and foreground is successful. DMD does a fantastic job with the objects that are constantly moving. Eventhough the DMD algorithm gives somewhat inaccurate results for motions that are not consistently moving, the overall motion can be captured.

6 References

Data-Driven Modeling & Scientific Computation
Dynamic Mode Decomposition: An Introduction

Appendix A - MATLAB functions

- `imagesc`: imagesc(C) displays the data in array C as an image that uses the full range of colors in the colormap.
- `svd`: [U,S,V] = svd(A) performs a singular value decomposition of matrix A, such that $A = U * S * V'$.
- `uint8`: uint8(X) converts the values in X to type uint8.

Appendix B - MATLAB code

```
clear all; close all; clc

v = VideoReader('floor.mp4');
video = read(v);
frames = get(v, 'numberOfFrames');
duration = get(v, 'Duration');

%%
for i = 1:frames
    video_mat2 = rgb2gray(video(:,:,:,i));
    video_reshape = reshape(video_mat2, [], 1);
    video_mat(:,i) = double(video_reshape);
end
%%
X = video_mat;
X1 = X(:,1:end-1);
X2 = X(:,2:end);
r = 2;
dt = duration/frames;
[Phi, omega, lambda, b, Xdmd, S] = DMD(X1,X2,r,dt);
%%
video_full = video;
video_back = uint8(Xdmd);
video_fore = uint8(X(:,1:frames-1) - Xdmd);
for i = 1:frames-1
    video_back_reshape(:,:,:,i) = reshape(video_back(:,i),544,960);
    video_fore_reshape(:,:,:,i) = reshape(video_fore(:,i),544,960);
end

%%
s = diag(S)/sum(diag(S));
figure(1)
plot(s,'ro--','Linewidth',[2]);

%%

figure(2)
subplot(3,3,1), imagesc(rgb2gray(video_full(:,:,:,20))),
colormap(gray), axis off
subplot(3,3,2), imagesc(rgb2gray(video_full(:,:,:,150))),
colormap(gray), axis off
subplot(3,3,3), imagesc(rgb2gray(video_full(:,:,:,350))),
colormap(gray), axis off
subplot(3,3,4), imagesc(video_back_reshape(:,:,:,20)),
colormap(gray), axis off
subplot(3,3,5), imagesc(video_back_reshape(:,:,:,150)),
colormap(gray), axis off
subplot(3,3,6), imagesc(video_back_reshape(:,:,:,350)),
colormap(gray), axis off
subplot(3,3,7), imagesc(video_fore_reshape(:,:,:,20)),
colormap(gray), axis off
subplot(3,3,8), imagesc(video_fore_reshape(:,:,:,150)),
colormap(gray), axis off
subplot(3,3,9), imagesc(video_fore_reshape(:,:,:,350)),
colormap(gray), axis off

function [Phi, omega, lambda, b, Xdmd, S] = DMD(X1,X2,r,dt)
%% DMD
[U,S,V] = svd(X1, 'econ');
% r = min(r, size(U,2));

U_r = U(:, 1:r); % truncate to rank-r
S_r = S(1:r, 1:r);
V_r = V(:, 1:r);
```

```
Atilde = U_r' * X2 * V_r / S_r; % low-rank dynamics
[W_r,D] = eig(Atilde);
Phi = X2 * V_r / S_r * W_r; % DMD modes

lambda = diag(D);       % discrete-time eigenvalues
omega = log(lambda)/dt; % continuous-time eigenvalues
%% Compute DMD mode amplitudes b
x1 = X1(:, 1);
b = Phi\x1;
%% DMD reconstruction
mm1 = size(X1, 2); % mm1 = m - 1
time_dynamics = zeros(r, mm1);
t = (0:mm1-1)*dt; % time vector
for iter = 1:mm1
    time_dynamics(:,iter) = (b.*exp(omega*t(iter)));
end
Xdmd = Phi*time_dynamics;
end
```