

Dynamic Mode Decomposition and Video Background Detection

Hongda Li

March 15, 2021

Abstract

The objective of the paper is to demonstrate the application of Dynamic Mode Decomposition for Video Background extraction. Using the idea of modeling video as a dynamical system with low rank orthogonal modes.

1 Introduction and Overview

Two videos are given for demonstrating the usage of Dynamic Mode Decomposition (DMD). Both contains a static background and some objects moving in the foreground.

Each frame of the video is view as a snapshot of as snapshot of a high dimensional dynamical system. The assumption we make is that the evolution of system are govern by low rank mode that can be easily extracted via the use of PCA. Before feeding into the D

2 Theoretical Background

Dynamic Mode Decomposition

The premise of DMD is treating data collected from the state of the dynamical system, let it be x_i be a vector of all the state variables of the system at a given time moment i . In addition we assume that these snapshot are taken on an evenly spaced interval. Most of the theoretical description of the method can be found in book *Dynamic Mode Decomposition* by Kutz et.al[[dmd'book](#)].

The data are represented as a matrix, where each column contain the snapshots for each of state variable at each time moment:

$$\mathbf{X}_{\text{original}} = \begin{bmatrix} x_1 & x_2 & \cdots & x_{m+1} \end{bmatrix} \quad (1)$$

An all the states of the dynamical system for each of the time moment can be viewed as two matrices in the following way:

$$\mathbf{X} = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix} \quad \mathbf{X}' = \begin{bmatrix} x_2 & x_2 & \cdots & x_{m+1} \end{bmatrix} \quad (2)$$

And the idea of DMD is to approximate a matrix, assumed to be low rank, appearing in the following optimization problem:

$$\underset{\mathbf{A}}{\operatorname{argmin}} \| \mathbf{A}\mathbf{X} - \mathbf{X}' \|_F = \mathbf{A}\mathbf{X}'\mathbf{X}^\dagger \quad (3)$$

The matrix \mathbf{A} is also known as the Koopman operator, which is defined in a much general context than this paper but it's worth mentioning. Next our objective is looking for a way to approximate the eigenvalues and eigenvectors of the Koopman operator: \mathbf{A} . We also do not make any assumption about the entries in the matrix, hence, we assume that they can be complex. and the algorithm can be outlined as the following:

1. Taking the Truncated SVD on the data matrix: $U\Sigma V^H = \mathbf{X}$. The SVD of this matrix contain information about the Orthogonal Modes of the Dynamical System.
2. Then: $\mathbf{X}' = A\mathbf{X} = AU\Sigma V^H$, Implying that $AU = X'V\Sigma^{-1}$, and apply U^H on the left we have $U^H AU = \tilde{A}$. Matrix \tilde{A} is similar to matrix A , meaning that share a subset of the eigenvalues (Depending on the size of matrix U).
3. The Eigen Decomposition of matrix: $\tilde{A}W = W\Lambda$. Leading Eigenvalues in matrix Λ is the same as matrix A .
4. Consider describing the dynamics of A we have: $U\tilde{A}W = AUW = X'V\Sigma^{-1}W$ and this is call he DMD, we name it: Φ . This is a set of vectors that best describes the moments in the original system, it doesn't have to be orthogonal.
5. The evolution of the system can be model by the continous time system, approximated using the eigenvalues matrix Λ as:

$$(\Phi^\dagger u_0) \exp(\Omega t) \quad \text{Where: } \Omega = \frac{\ln(\Lambda)}{\Delta t} \quad (4)$$

Note: DMD is useful because the size of the eigenvalues in the matrix Λ tells us how much each of the eigenvectors in A governs the dynamic modes. If $|\omega_i|$ is small, then it means it's not moving much and that is likely to be the background of the video.

3 Algorithm Implementation and Development

There are 2 key parts for the algorithmic design part, The DMD itself, which is largely described in the theoretical section, and the subtraction of the video back, and that will be the focus here.

Note: To get the best results some trickeries are involved. The vide are rescaled to only include the important parts for better computational speed or inverted to get a better visual for the video foreground.

Video Background Subtraction

Algorithm 1: Algorithm 1: Background Subtraction

Input: Reconstructed with DMD: XLowRank
XSparse := XOriginal - Abs(XLowRank)
R := (XSparse < 0).*XSparse
XLowRank = R + abs(XLowRank)
XSparse = XSparse - R
Output: XSparse

The algorithm [Algorithm 1](#) Takes in the data matrix reconstructed using low-rank dynamics, and it outputs the part of the dynamics that have a lot of movement. At line 2, we subtract the static background to get the moving components. At line 3, we get all the negative value to be positive because negative value for video stream doesn't make sense, at line 4 we balance out all the negative value back into the static background by adding them, hence, at line 5, taking the difference (Adding the absolute value of the negative parts) gives us the foreground of the video.

The variable "XSparse" represent the part of the data matrix constructed by Dynamic modes with $|\omega_i| \gg 0$ and the variable "XLowRank" represents the part of the data matrix constructed by dynamic modes corresponds to $|\omega| \approx 0$.

Algorithm 2: Algorithm 2: Choosing Best Omega and Reconstruct

Input: DMD Modes: Φ , EigenValues: Λ , Initial Condition: u_0 , Delta T: Δt , Time Moments: t
 $\Omega := \ln(\text{Diag}(\Lambda))/\Delta t$
 $\omega_+, \text{Idx} = \min(\Omega)$
 $\phi_+ := \Phi(:, \text{Idx})$
 $y_0 := \phi_+^\dagger u_0$
Reconstructed = $\phi_+ y_0 \exp(\omega_+ t)$
Return: Reconstructed, Idx

Best Omega Value

[Algorithm 2](#) Looks for the continuous eigenvalues of matrix A , denotes as ω_+ such that ω_+ is as small as possible. When this condition is made true, it's highly likely that the Dynamic Modes corresponds to the ω_+ is the background of the video. And this is the key exploit of the Dynamic Mode decomposition with this particular application.

4 Computational Results

Selection of PCA Modes

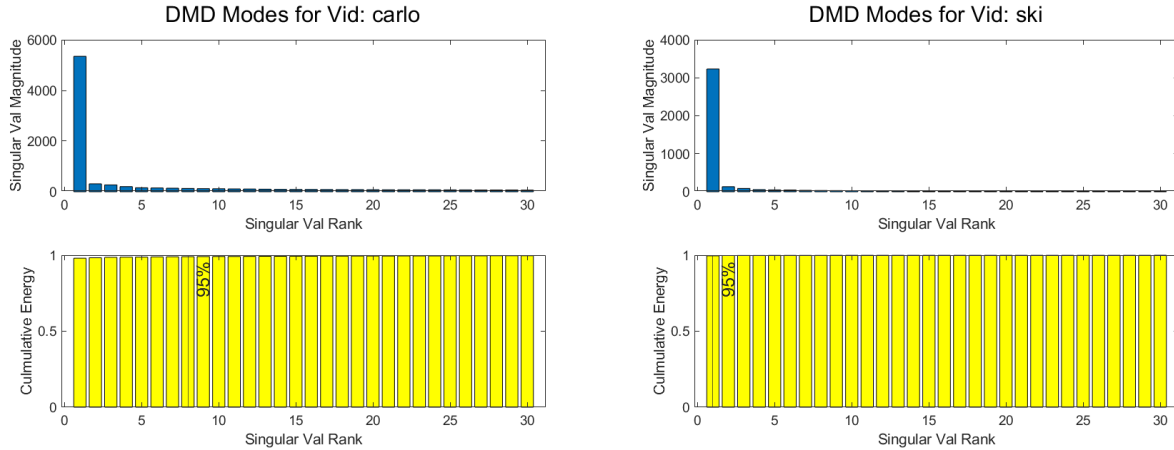


Figure 1: Fig 1: Carlo and Ski Singular Values

Selection of the PCA modes for the video uses sum of the eigenvalue squared, often referred to as the energy of the eigenvalues of the covariance matrix. In here, we select the singular value with just over 95% of the energy to for the computation of DMD.

As it can be seemed on fig:1]Fig 1, most of the the variance of the “Ski” video can be explain with the first singular values, and for the “Carlo” video, 8 Principal Components are able to explain most of the variance. This gives us the choice of how to truncate the data matrix \mathbf{X} .

Selection of Eigenvector for Dynamic Modes

The selection is simply choosing the smallest values ω_i (The diagonal of the Ω matrix).

As it can be seemed on [Fig 2](#), the dominant Principal modes corresponds to the same Orthogonal Basis. This is true because the first video has a lot of background and the moving parts are just principal modes, resulting in U matrix being a scalar, hence making A also a scalar.

However for the “Carlo” video, there many singular values, and resulting more options for choosing ω_+ for

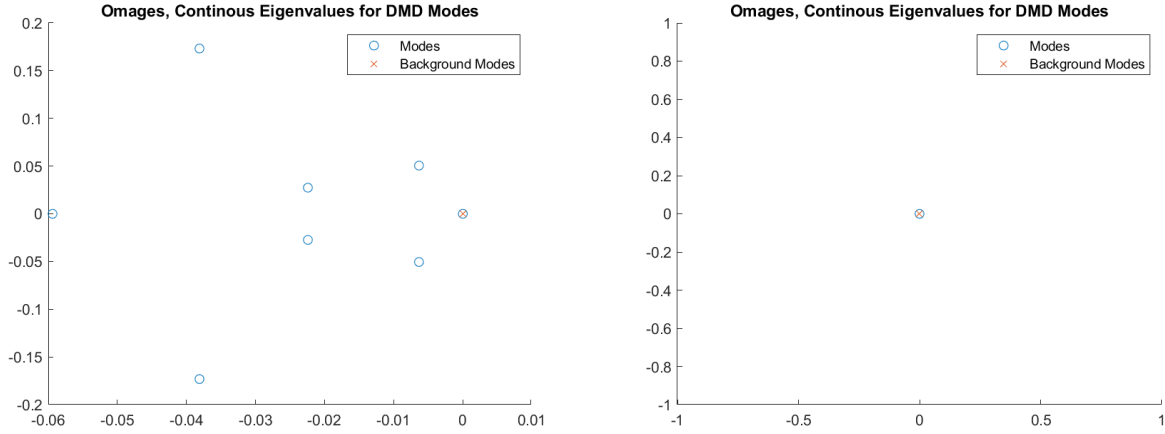


Figure 2: Fig 2: Values of Omegas

the Koopman operator, therefore, we have to choose ω_+ such that $\|\omega_+\|$ is as small as possible. In this case, the principal mode is controlled by it and it will be static.

Here we made the choice of choosing the smallest ω , but in practice, multiple eigenvalues can be chosen to get different amount of separations.

Video Original and Foreground

To extract the background, we use [Algorithm 2](#) which will take out the foreground if the data matrix reconstructed using the smallest ω is given.

The results of the separation using the smallest ω is shown in [Figure 3](#), and it can be easily seemed that the moving objects has been successfully separated from the static background, which are shown in [Figure 3](#)

5 Summary and Conclusions

In this assignment we have seen one of the applications of DMD. Some of the practical challenges involved are the big computational complexity. However, to augment the idea, it's possible truncate the video into different sections and we seek for the best reconstruction for each section of the video. In practice, the method of DMD is applicable only when PCA gives reasonable results, which means that DMD will not be robust towards noise introduced via camera shake. In addition the Koopman operator is approximated, hence it cannot extrapolate.

Appendix A MATLAB Functions

Foreground/Original: carlo



Foreground/Original: ski

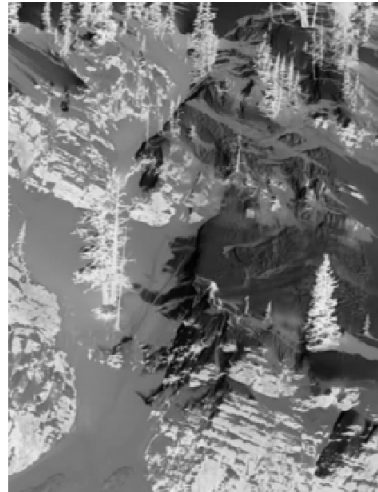
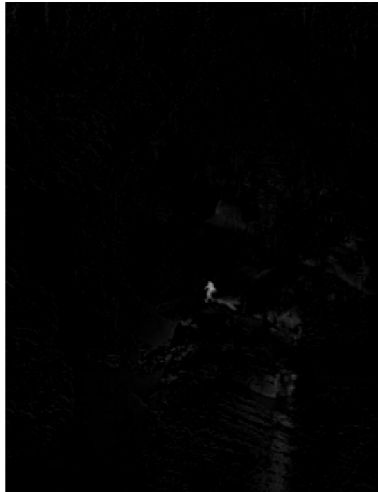


Figure 3: Fig 3: Background Extraction

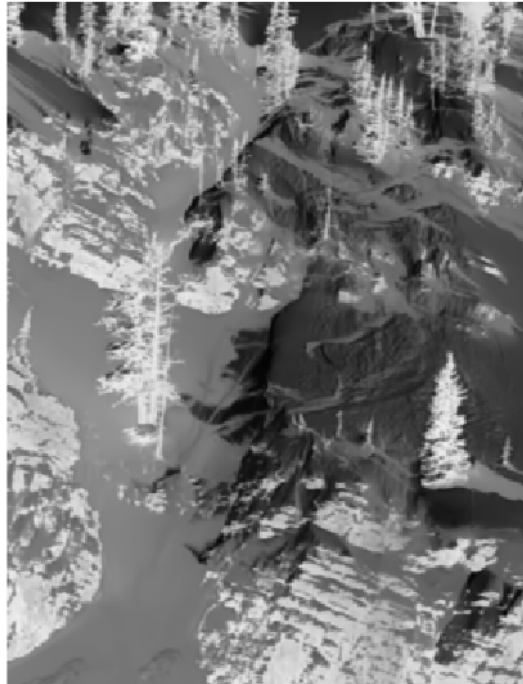


Figure 4: Figure 3: Carlo Static Background (Top), Ski Static Background (Bottom)