Dynamic Mode Decomposition Background Foreground Seperation

Leif Wesche

3-1-2018

Abstract: In this assignment, dynamic mode decomposition was used to separate the background and foreground of two separate videos. The background and foreground footage was reconstructed by evaluating the frequencies associated with each dynamic mode. The background was reconstructed using only the modes that were associated with lowest frequencies, and the foreground was reconstructed from high frequency modes. The first of the two videos tested contained less movement than the second video tested. As a result, it was seen that the background reconstruction was clearer in the first video, while the foreground reconstruction was clearer in the second video.

I. Introduction and Overview

The purpose of this assignment was to separate the background and foreground of several videos using the dynamic mode decomposition (DMD) method. Two videos were recorded and tested, and each video was decomposed using DMD. The first video was footage of a persons slowly moving in front of a clock. The second video consisted of footage of a person playing with a kendama, with much more movement than the first video. The DMD frequency spectrum was analyzed, and modes with the lowest frequencies were used to reconstruct the background of each frame of the videos. The foreground was reconstructed using the high frequency modes.

II. Theoretical Background

Dynamic mode decomposition is used to separate data sets that are continuous in time into a set of spacial modes. These modes are called dynamic because each mode oscillates, decays, or grows in time. By analyzing the rate of change of each of these modes, the modes associated with the background of a video can be deduced and used to reconstruct the background portion of a video. Since the background in the videos tested didn't change much, the dynamic modes associated with them often had very low frequencies, close to zero. By setting a threshold frequency value and reconstructing the video from only modes with frequencies below the threshold value, the background of a video can be reproduced. Similarly, the foreground was reproduced by reconstructing the videos from high frequency modes.

In the case of video decomposition, each frame can be thought of as a snapshot in time. DMD is performed by organizing each frame into column vectors of two data matrices. The first data matrix, X, consists of the first video frame through the second to last video frame. The second data matrix, X', consists of the second data frame through the last frame. It is supposed that theoretically, the two data matrices can be related through a single linear operator A as shown below in Equation 1.

$$X' = AX \tag{1}$$

Although the actual frame by frame evolution of each video is nearly guaranteed to be a non linear process, A is a large enough linear operator that it can be used to approximate the non linear behavior of the video dynamics. In other words, the eigen vectors and eigen values of A can be used to approximate how the data in X changes in a later times-step X'.

Calculating A itself is difficult because of the size of the matrix. Even using standard resolution video data, the matrix A easily approaches the size of over a million square data points. So instead, dynamic mode decomposition calculates the eigen values and eigen vectors of a closely related matrix, $\tilde{A}$.

To compute $\tilde{A}$, first, the data set X is decomposed using singular value decomposition, yielding U, Σ, and V matrices. Instead of computing A directly, $\tilde{A}$ is calculated, which is A projected onto the orthogonal basis U. Using the X', U, Σ, and V matrices, $\tilde{A}$ is calculated as shown below in Equation 2

$$\tilde{A} = U * X' * V * \Sigma^{-1} \tag{2}$$

Next, an eigen decomposition is performed on $\tilde{A}$, which yields the eigen vectors and eigen values of $\tilde{A}$, W and Λ respectively. Since $\tilde{A}$ is just a projection of A, the eigen values of the two matrices are the same. The matrix ϕ is calculated, which reconstructs the full high dimensional matrix of eigen values associated with A. ϕ is calculated as shown in Equation 3 below.

$$\phi = X' * V * \Sigma^{-1} * W \tag{3}$$

The DMD frequencies, corresponding to the eigenvalues of A, were calculated by taking the natural log of the diagonal eigenvalue matrix, Λ. Since the frequencies that were close to zero were thought to be the frequencies that corresponded to the background, the eigen values corresponding to these low frequencies were used to reconstruct the background. The eigen values corresponding to the remaining higher frequencies were used to reconstruct the foreground. The videos were reconstructed using Equation 4 shown below, where X_i represents a single reconstructed frame, Λ_T is the truncated list of eigen values, and b is an initial condition vector.

$$X_i = \phi \Lambda_T^t * b \tag{4}$$

III. Algorithm Implementation and Development

The videos were first loaded into Matlab, converted to gray scale, converted to doubles, and reduced in resolution to speed up processing times. Both videos tested were imported at resolutions of 1920x1080, and reduced to a tenth of their original resolution. X was constructed by organizing each frame into column vectors and storing the each frame except the final frame. X' was constructed by storing each frame except the first frame.

DMD was performed by first decomposing X using Matlab's "svd" command and calculating $\tilde{A}$ according to Equation 2. Then, $\tilde{A}$ was decomposed using eigen value decomposition to determine W, the eigen vectors of $\tilde{A}$, and Λ, the diagonal eigen values of $\tilde{A}$. Using these eigen vectors and values of $\tilde{A}$, ϕ was constructed according to Equation 3.

First, the original video was reconstructed from the dynamic modes using Equation 4. The initial value term b was calculated by solving the equation $\phi = bX_1$, where X_1 is the first frame in column form.

Next, the mode frequencies corresponding to each eigen value were calculated by taking the natural log of each eigen value and dividing the frequencies by the time between each step. The eigen values Λ were plotted, along with the frequencies corresponding to each eigen value. A cutoff frequency in the range of 0.5 to 1 was established and verified later by testing how well the foreground and background were reconstructed. To reconstruct the background, the diagonal eigen value matrix Λ was truncated by setting any values that corresponded to frequencies greater than the cutoff frequency to zero, then reconstructing each frame using Equation 4. The foreground was reconstructed the same way, but instead setting eigen values with frequencies lower than the cutoff frequency were set equal to zero.

IV. Computational Results

Reconstructing the entire video from the DMD modes worked well using Equation 4. The original video pixel values and the values obtained from reconstructed DMD modes were nearly identical, and when played side by side the two videos were indistinguishable.

Next, the eigen values and mode frequencies were plotted for each data set, and the diagonal eigen value matrix was truncated according to the range of mode frequencies to separate the background and foreground of each video. The eigen values and mode frequencies for Test 1 are shown below in Figure 1. The eigen values and mode frequencies for Test 2 are shown in Figure 4 in the appendices.

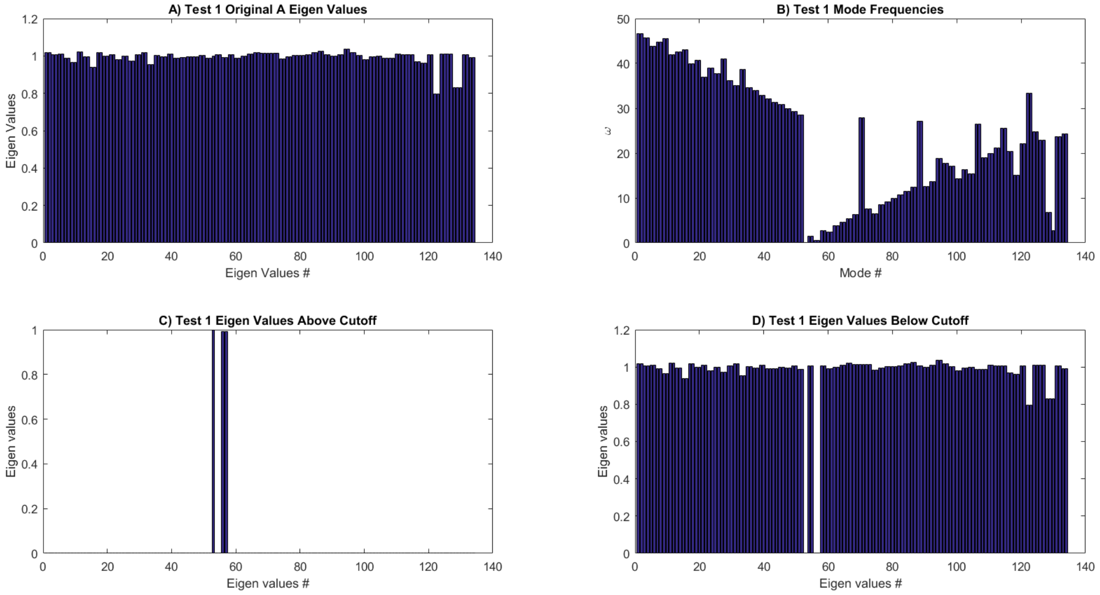

Figure 1: Test 1 eigen values, mode frequencies, and truncated eigen values.

Figure 1A shows the original range of eignen values for the first video. The mode frequencies associated with the set of eigen values were obtained by taking the natural log of the eigen values and dividing the resultant number by the time in between each frame. The mode frequencies are shown in Figure 1B, and upon inspection it is clear that a number of the frequencies are very close to zero, while others are much higher. By inspection of the graph, a cutoff frequency was established at $\omega = 0.55$ for Test 1. This value was verified by reconstructing the videos with a range of cutoff frequency values to determine which cutoff frequency yielded the best results. The cutoff frequency determined for Test 2 was $\omega = 0.1$

To reconstruct the background, a new diagonal matrix of eigen vectors was constructed by setting all eigen vectors that correspond to frequencies higher than the natural frequency to zero. The eigen values used to construct the background in Test 1 are shown in Figure 1C. To reconstruct the foreground, a new diagonal matrix of eigen vectors was constructed by setting all eigen values with frequencies greater than the cutoff frequency to zero, as shown for Test 1 in Figure 1D. Each new eigen value matrix was the same size as the original eigen value matrix. Equation 4 was used to reconstruct the background and foreground using their respecitve diagonal eigen value matrices. The results of the foreground and background reconstruction are shown in Figure 2 below.

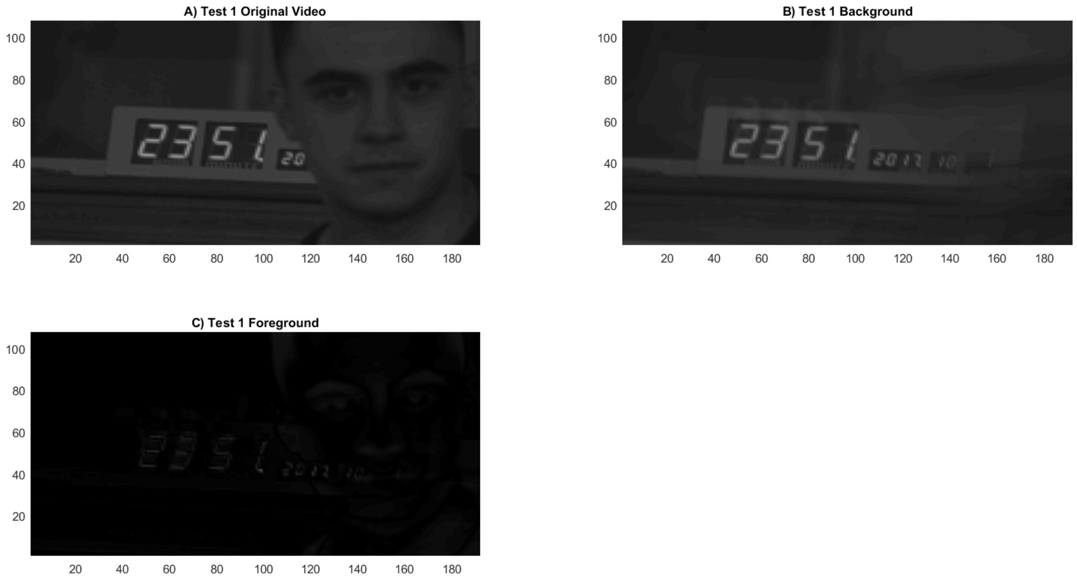

Figure 2: Test 1 original, background, and foreground videos.

Figure 2 compares a single frame from the same point in the original Test 1 video, background reconstruction, and foreground reconstruction. Each gray scale video used a standard pixel brightness intensity in the range of 0 to 255. The original video, shown in Figure 2A, consisted of about 5 seconds of the background, then the person slowly moving their head into the view of the camera from left to right for another 4 seconds.

The background reconstruction, shown in Figure 2B, worked very well. As the person moved his head into the frame, a slight blur could be seen in place of the persons face, but the background behind it was clearly distinguishable. At the frame shown, the persons whole head is in the frame in the original video, yet only a blur is shown in the reconstruction. The reconstructed video even clearly shows details on the alarm clock which are obscured by the persons head in the original video.

The reconstruction of the foreground did not work nearly as well as the background. As shown in Figure 2C, most of the screen was dark. Some motion of the mans head and a small blinking light on the clock is visible in the video, but overall the reconstruction was poor. The cutoff frequency was adjusted, and I experimented with using two different frequencies for the upper and lower cutoff frequencies, but the foreground reconstruction of Test 1 never improved much. This poor foreground reconstruction could be due to the fact that over half of the video consists of just the background, without no real foreground movement until over half way through the video.

Figure 3 compares a single frame from the same point in time in the original Test 2 video, the background reconstruction, and the foreground reconstruction. Test 2 used a still video which showed a man sitting on a couch playing with a kendama and moving around. A frame from the original video is shown in Figure 3A.

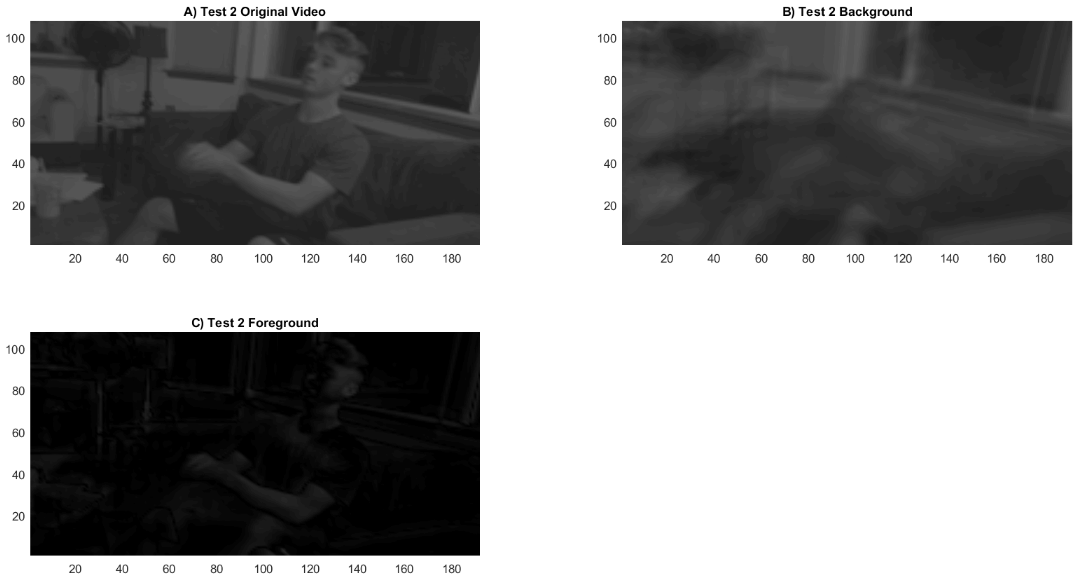

Figure 3: Test 2 original, background, and foreground videos.

The background reconstruction in Test 2, seen in Figure 3B was not as crisp as the reconstruction in Test 1. The mans body, which did not move as much as his arms or head but moved less than the rest of the surroundings, was only partially reconstructed in both the background and foreground reconstructions. The rest of the room reconstructed well though, and the area obscured by the mans arms and legs in the original video is somewhat visible in the background reconstruction.

The foreground reconstruction, shown in Figure 3C, was much more accurate in test 2 than in Test 1. In test 2, the arms, legs, and head of the man are much more visible than the surroundings. The foreground reconstructed likely worked better in test 2 because the video used for test 2 consisted of much more fast and sharp fast motion, compared to the slow motion in test 1. This fast motion could have resulted in more heavily weighted high frequency dynamic modes in test 2, compared to the high frequency dynamic modes in test 1.

V. Summary and Conclusions

Dynamic mode decomposition was used to successfully isolate the foreground and background videos in both tests. It was found that in test 1, where the video consisting of slow movement and long background shots was decomposed, the DMD algorithm did a much better job reconstructing the background than the foreground. In test 2, where the video decomposed contained more fast movement, the foreground was reconstructed better than in test 1, and the background was reconstructed slightly worse. This was likely due to the fact that the slow moving, low frequency dynamic modes in test 1 were weighted more heavily than the low frequency modes in test 2, so the background reconstruction was much more clear. The foreground reconstruction in test 2 likely improved because the faster high frequency dynamic modes were weighted more heavily in the case of a video with more fast motion.

VI. Appendix

Appendix A: Additional Figures

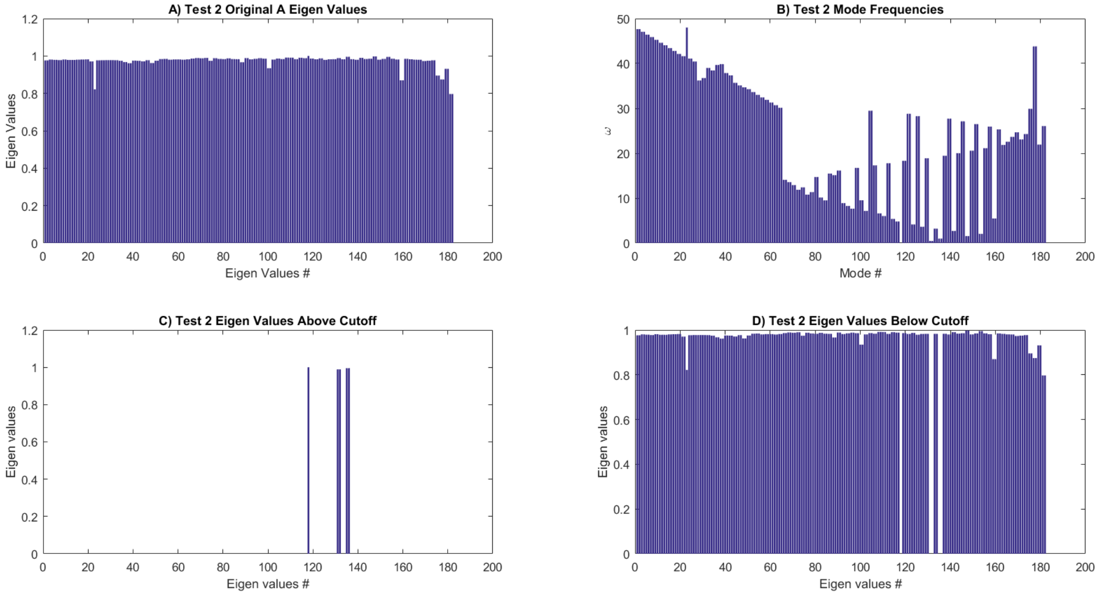

Figure 4: Test 2 eigen values, mode frequencies, and truncated eigen values.