Study of the Principal Component Analysis

Peter Schultz - Feb 26 2019

I. Introduction and Overview

The singular value decomposition is an extremely powerful tool in linear algebra. It allows the factorization of a matrix into three constitutive components.

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

This important equation is used in many applications, and one such application will be studied herein.

II. Theoretical Background

Any matrix can be represented by the above equation. This representation can be explained by the effect of multiplying some vector or matrix by **A**. The two components, **U** and **V*** are responsible for the rotation of said vector or matrix, and **Σ** is responsible for its stretching.

Another explanation is that the columns of **U** span the row-space of **A**, while the rows of **V*** span the column-space of **A**. If **A** contains redundant information, it will be removed in the SVD. The **Σ** matrix is a diagonal matrix consisting of a number of so-called singular values, which represent the relative magnitude or importance of the basis vectors contained in **U** and **V***. The singular values are in fact the square roots of the nonzero eigenvalues of **A*A** or **AA***, which have the same eigenvalues. The number of singular values correspond to the number of dimensions spanned in **A**. Any redundant vectors in these two matrices will end up being multiplied by zeros in the **Σ** matrix. This component of the SVD is vital to the application studied herein, which is the principal component analysis (PCA). In this example, the PCA is used to identify and remove redundancies present in the movement of a spring-mass system recorded using three cameras from different angles. There are four cases:

1. Ideal Case, where the mass is displaced in only the z direction.
2. Noisy case, where the mass is displaced in the z direction but the camera is shaking.
3. Horizontal displacement, where the mass is displaced perpendicular to the ground (z direction) and horizontally.
4. Horizontal displacement and rotation, where the mass is displaced in z, horizontally, and rotated, which makes tracking the mass difficult.

III. Algorithm Implementation and Development

In order to make it easier to track the mass in the video recordings, a flashlight was placed on top of the paint can, which is attached to a spring. This allows it to be tracked by finding the maximum pixel intensity in each frame of the recording from each camera view. In general, the max pixel intensity is 255, and the source is the flashlight. However, in some cases, another component of the frame will have the highest magnitude; for this reason, parts of the frame from each view are cropped out to achieve cleaner tracking of the mass in the system.

The position of the mass at every time point from each of the three cameras is saved into a matrix **X** which has the form,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_a \\ \mathbf{y}_a \\ \mathbf{x}_b \\ \mathbf{y}_b \\ \mathbf{x}_c \\ \mathbf{y}_c \end{bmatrix}$$

where x_a, y_a represent the x and y coordinates of the first camera, b for the second camera, and c for the third. This results in a matrix of dimensions [6 x N] where N is the number of frames in the shortest recording of the spring-mass system. This matrix will serve as the **A** matrix described in section II. The SVD of this matrix is taken to obtain u, s, and v, which represent **U**, **Σ**, and **V*** respectively. The final step is to project **X** onto this newly created space spanned by the orthonormal basis vectors contained in u. This is done by this computation: $Y = u^T * X$, where u^T represents the non-conjugate transpose of u and **Y** represents the projection of the data contained within **X** onto the *ideal* orthonormal basis vectors for this dataset. The resulting **Y** matrices are denoted the principal components projection, and are shown and discussed in section IV.

IV. Computational Results

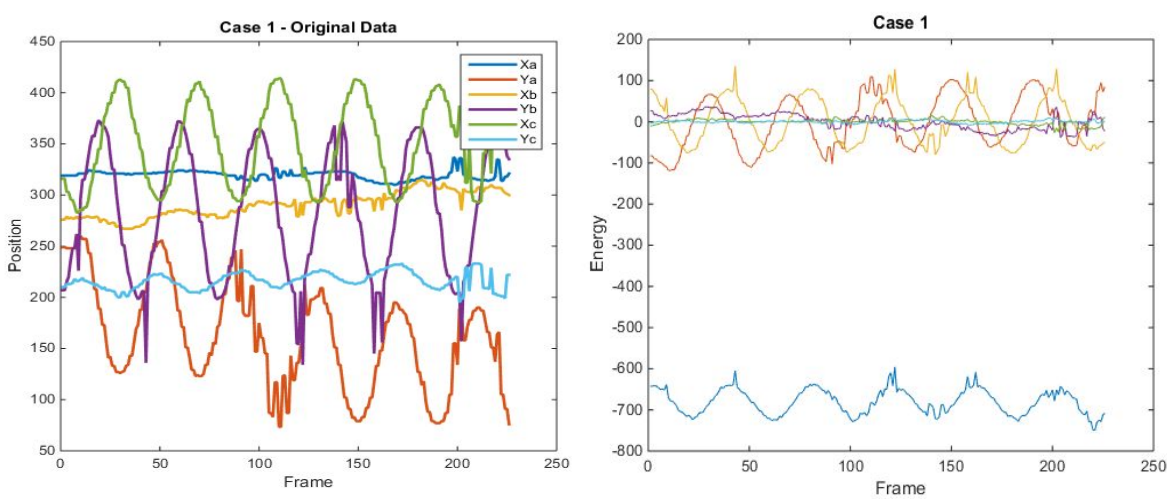

Figure 1: (Left) Sample mass position data, (right) principal components projection for ideal case.

In figure 1, the position data for the mass system is shown on the left for reference, but will not be shown for the 3 following cases. The highest energy component of the projection (at the bottom of the right graph) corresponds to the oscillation of the paint can perpendicular to the ground. There are two other sinusoidal oscillations near zero in the figure above. If the main signal represented is in the y direction, I believe that these two correspond to the oscillations in the x and z directions, however small the magnitudes. The same is true in the figures 2, 3, and 4.

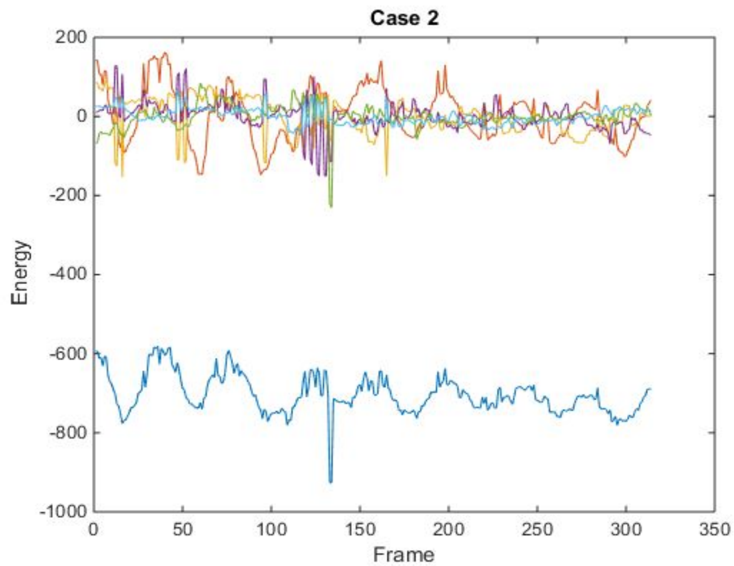

Figure 2: Principal components projection for noisy case.

Shown above is the principal component projection for the noisy case of data. The highest energy component has a clear sinusoidal trend, but is also quite noisy. This demonstrates that PCA has the ability to pick out important components of data even with noisy conditions.

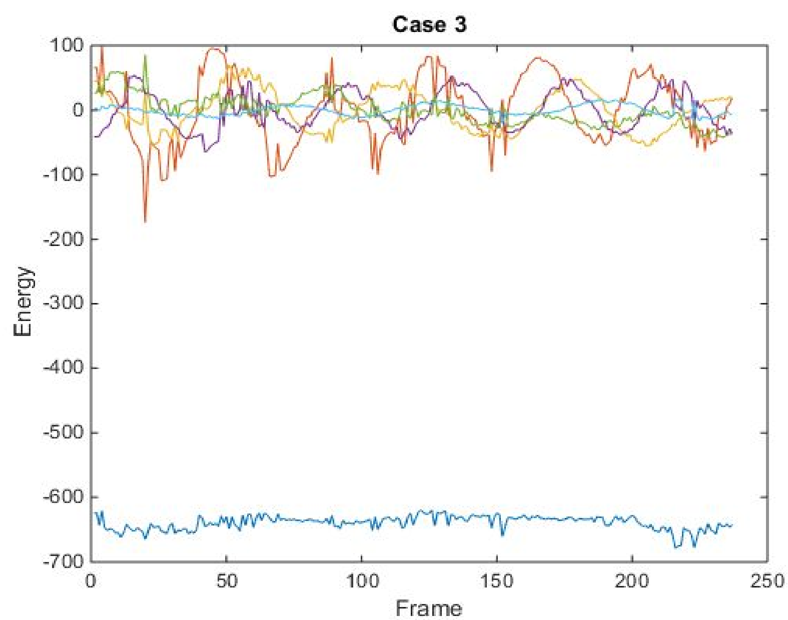

Figure 3: Principal components projection for horizontal displacement case.

Figure 3 shows the principal component projection for the third case, which is the case with vertical and horizontal displacement. Due to some problem in the code which I cannot identify, the highest energy component identified is noise. The expected high energy component is the vertical oscillation, with a secondary component corresponding to horizontal oscillation.

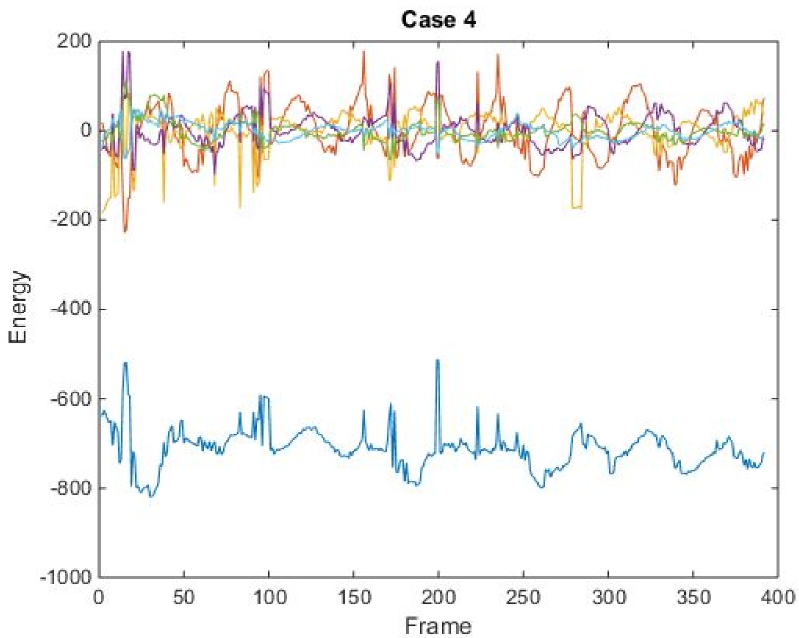

Figure 4: Principal components projection for horizontal displacement and rotation case.

Figure 4 above shows a highly non-uniform projection with highest energy. I hypothesize that the random spikes are caused by the rotation of the paint can. Once the flashlight is not facing any of the cameras, its position can no longer be tracked until it rotates such that the flashlight is once again in view. For those brief periods without view of the flashlight, the position extraction portion of the algorithm will choose a random location in the field of view with the next highest pixel intensity as the location of the paint can.

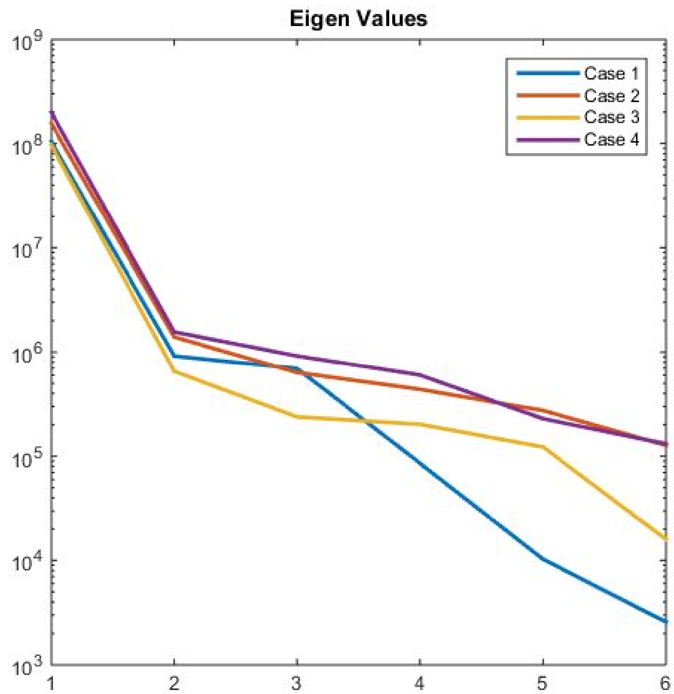

Figure 5: Eigenvalues from each case.

Lastly, figure 5 above illustrates the eigenvalues of **X*X**, which are the singular values of **X** squared. They demonstrate the relative importance of each of the 6 orthonormal vectors contained in the principal component decomposition in descending order. The first and highest value for each of the cases corresponds to the vertical oscillation of the paint can. This indicates that this basis vector contains the most energy compared to the basis with the next most energy by about two orders of magnitude in each of the cases.

V. Summary and Conclusions

This study highlights a significant conclusion. Despite the fact that the data entered into the SVD has no physical context whatsoever, the algorithm is able to identify the orientation of oscillation, and subsequently the ideal axes/basis vectors for describing the data. This is but one application that shows the usefulness of the SVD and PCA; indeed, the potential of these tools for problem solving in different applications is enormous.

Appendix A MATLAB functions used

svd(X, 'econ'): The svd command computes u, s, and v, and the 'econ' option only produces the first m columns of v if m<n, and s is size [m x m].
diag(lambda): Outputs the main diagonal of a matrix in column form.
flipud(frame): flips the input matrix in the up-down direction.
semilogy(lambda): plots the input vector with the y axis in log format.