Appearance-based Object Classification

Sharif Anani
South Dakota School
of Mines & Technology
501 E. St. Joseph St.
Sharif.Anani@mines.sdsmt.edu

Abstract

In this project, we are to use eigenvalue decomposition to aid in the classification an object. The final goal is to be able to differentiate the object from other object, and to be able to calculate the orientation at which the object is, using a set of training and test images.

1. Introduction

Object classification is one of the very popular areas of computer vision; it can be a very effective tool in production lines, autonomous vehicles, and artificial intelligence. To further deepen our understanding as students studying the topic, we are tasked with creating an appearance based object classification pipeline.
The first step of the project is to use the provided training images to create manifolds for different objects. Each individual manifold will be used to approximate the orientation of an object in a test image. Another manifold will be created that encompasses all the known object, that manifold will be used to differentiate between objects and help classify an anonymous object as one those in the training image sets.
Calculating the manifolds and subspaces will be done in two ways: once through Singular Value Decomposition (SVD), and once through the Fast Fourier Transform (FFT). The two approaches will be compared in different aspects.

2. Computing and Analyzing Eigenspaces

Two types of eigenspaces need to be computed. The eigenspaces need to be computed in a fashion that optimizes image representation and image distinction. Going through the optimization problem by minimizing the expected value of the error between the reconstructed image from the eigenspace and the actual image, and minimizing the expected difference between the squared distance between a random image with the same statistics as the set and a random training image. Both these criteria lead to that the optimal basis is the one comprised of the eigenvectors of the sample covariance matrix of the set.
The eigenvectors are computed using SVD, where the vectors of interest are the left singular vectors.

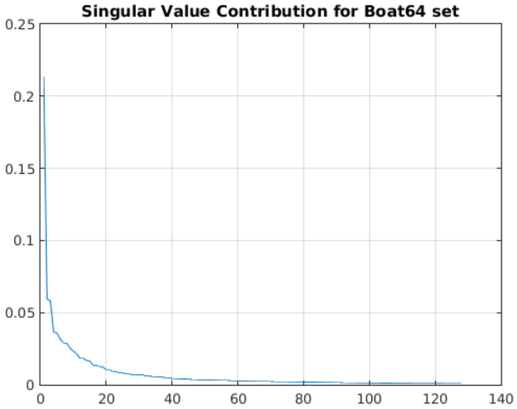


Figure 1. Singular contributions for `Boat64` set of training images

2.1. Choosing the Number of Eignvectors to Use

The purpose of these computations is to be able to recognize the orientation of an object without retaining all the data from the training set, i.e, constructing a manifold using the most dominant eigenvectors and using that as the compass that tells us the orientation of the test image.
To have a better idea about how many vectors to use, we can look at our singular values; the higher the magnitude of the singular value, the more information it retains about the set of which it came from.
We can judge by examining a plot of each singular value devided by the summation of all singular values, as this tells us it's relative weight in the set. For the `Boat64`, `Cabinet64`, and `Cup64` training sets, those plots are illustrated in Figures 1, 2, and 3

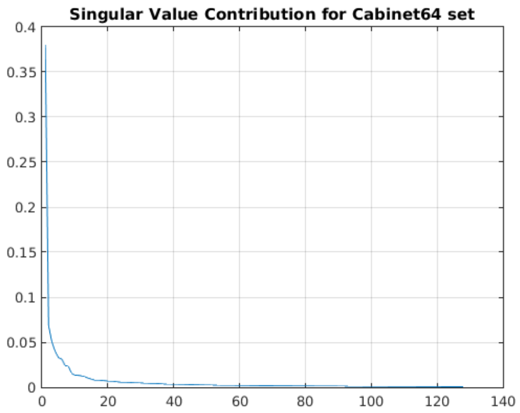


Figure 2. Singular contributions for `Cabinet64` set of training images

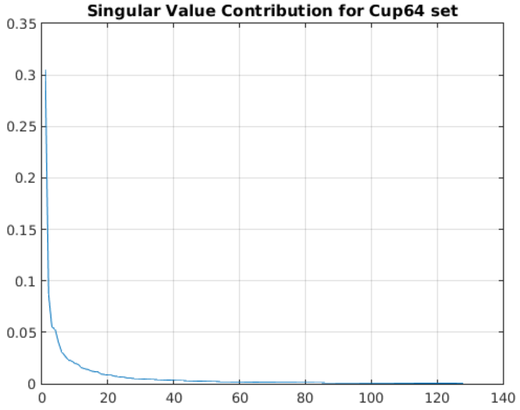


Figure 3. Singular contributions for `Cup64` set of training images

Once the slope starts to become horizontal, singular values are no longer significantly contributing to making the restored information distinct, and thus can be safely dropped while still preserving most of the information.
While this metric is intuitive and easy to understand, a better one to choose is the energy recovery ratio. The energy recovery ratio is a metric that expresses what percentage of the energy is restored using k singular values. The energy recovery ratio is calculated using the following formula:

$$\rho(X, u_k) = \frac{\sum_{i=1}^{i=k} ||u_i^T * X||_F^2}{||X||_F^2} \tag{1}$$

where X is the collection of unbiased images and u_i is the i_{th} singular/eigenvector. This metric was used for all results displayed in this paper. To see how the metric is used, we can plot the energy recovery function for the a few sets and examine those functions. These function can be helpful in telling us the number of eigenimages needed to adequately represent the set. The tolerance used in this paper is $\mu = 90\%$.

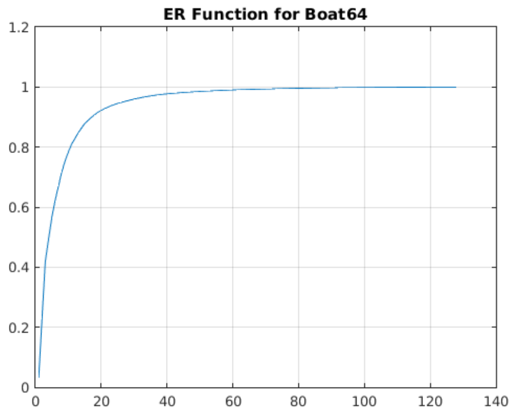


Figure 4. $\rho(X, u_k)s$ for `Boat64`

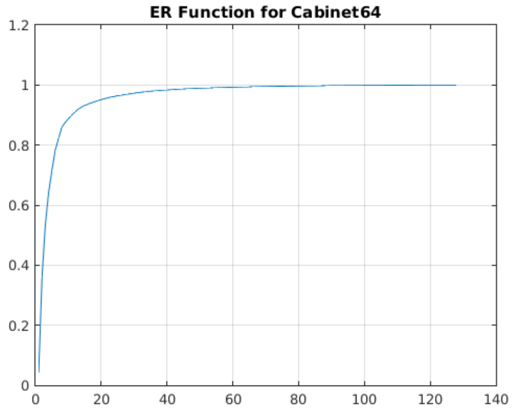


Figure 5. $\rho(X, u_k)s$ for `Cabinet64`

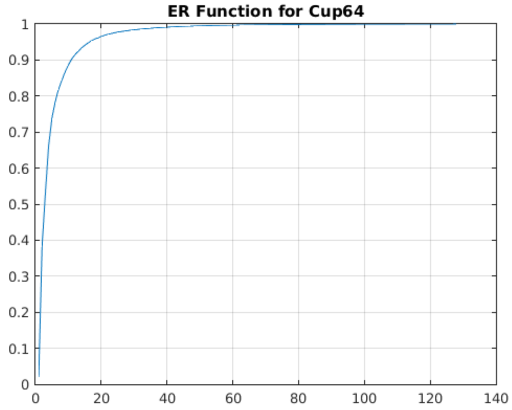


Figure 6. $\rho(X, u_k)s$ for `Cup64`

As part of the project, a test run was done on all the image sets to examine the number of eigenimages required to recover 80, 90, and 95% of the energy of the set. The results are shown in Table 1

Table 1. Table showing number of eigenimages needed to recover 80,90, and 95% of the energy after reconstructing

Image Set Name	k for 80,90, and 95% ER
ImagedataBoat64.mat	[11,18,27]
ImagedataCabinet64.mat	[7,12,20]
ImagedataCar64.mat	[6,14,28]
ImagedataChair64.mat	[7,15,26]
ImagedataCup64.mat	[7,11,17]
ImagedataFlashlight64.mat	[7,14,23]
ImagedataHandle64.mat	[6,9,15]
ImagedataHoseReel64.mat	[15,25,36]
ImagedataKeyboard64.mat	[18,28,41]
ImagedataLED64.mat	[3,5,7]
ImagedataLight164.mat	[10,19,30]
ImagedataLight264.mat	[11,19,32]
ImagedataMug64.mat	[5,9,14]
ImagedataScooter64.mat	[12,22,34]
ImagedataSprayBottle64.mat	[8,13,19]
ImagedataStapler64.mat	[6,13,25]
ImagedataTrash64.mat	[7,14,25]
Imagedataibook0164.mat	[10,20,35]
Imagedataimac0464.mat	[11,18,27]
Imagedataimac9864.mat	[14,24,37]

2.2. Computing the Manifolds

After deciding the number of eigenvectors required to accurately represent the set, we are to compute the n-dimensional manifold that represents a set. The manifold is the projection of the training images on the eigenspace, i.e, the representation of the image as a point in the new eigenspace. The new point representing each image can be computed by:

$$p = f_t^T \phi_j, j = 1...n \tag{2}$$

Where:

1. f_t^T is the transpose of the t^{th} image in the set
2. ϕ_j is the j^{th} eigenvector
3. n is the number of eigen/singular vectors used.

To illustrate what the manifolds look like and what they mean, we can choose to use three eigenvectors, and thus, this will give us a three dimensional manifold. Because the angle of rotation is only varying by moving along a latitude line, the manifold will be a line in three dimensions. Three dimensional manifolds might not be very useful in recovering information, as the first three singular vectors may not contribute enough of the total energy. Three 3-dimensional manifolds are shown in Figures 7, 8, and 9.

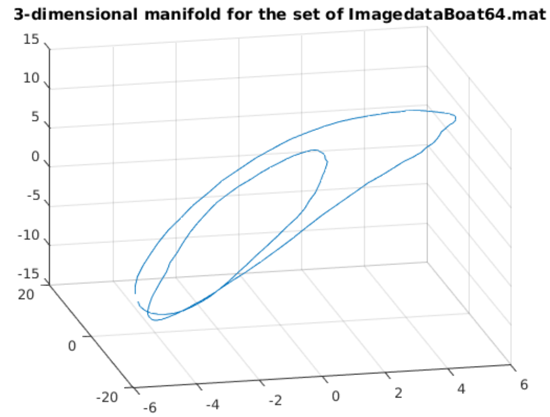


Figure 7. Three dimensional manifold for the `Boat64` set of training images

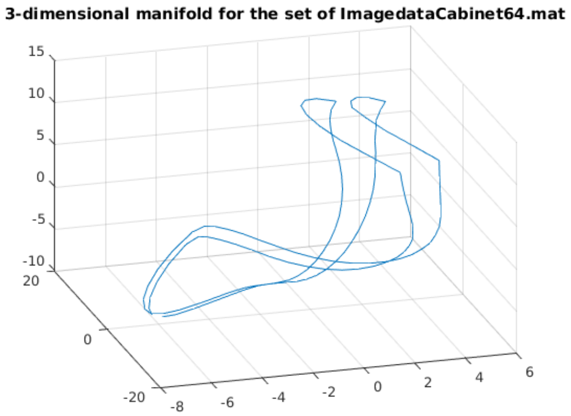


Figure 8. Three dimensional manifold for the `Cabinet64` set of training images

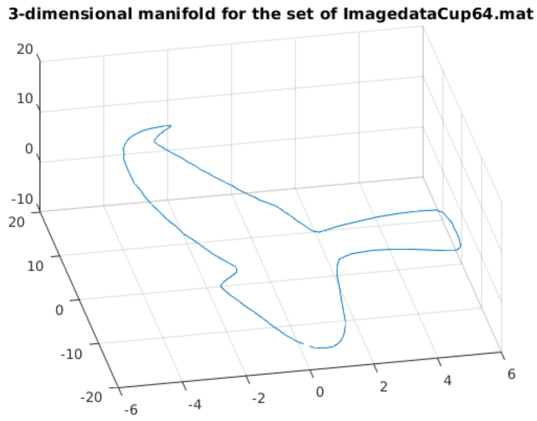


Figure 9. Three dimensional manifold for the `Cup64` set of training images

Examining the 3-dimensional manifolds shows that while points in the subspace seem to show good separation in the `Boat64` set of images, the `Cabinet64` image set has a 3-dimensional manifold that looks like two identical curves slightly translated, which means that if the image of

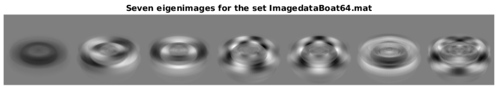


Figure 10. Eigenimages for the `Boat64` training set

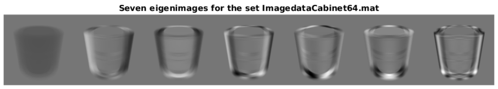


Figure 11. Eigenimages for the `Cabinet64` training set

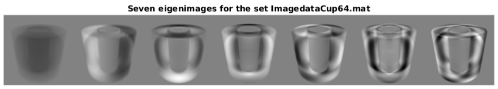


Figure 12. Eigenimages for the `Cup64` training set

the object was lowered to three dimensions, it looks very similar from all sides, making distinguishing orientation a difficult task if done in three dimensions.

2.3. Global Manifolds and Subspaces

The global manifold was computed two different sets of singular values:

1. Computing the singular vectors of all the eigenspaces concatenated

2. Computing the eigenvectors of the sample correlation matrix of all the image sets concatenated

The two methods were done for comparison, and the following was the conclusion: Using the local eigenspaces to compute a global one requires 151 singular values to achieve 90% of the energy. So the appropriate number of eigenvectors is slightly large to store.
Using all the training images from all the training sets to compute a global manifold and subspace is computationally expensive; the number of images to be processed is considerably large. However, the number of eigenvectors needed to recover 90% of the energy is 35. This means that for a live task of identifying objects, the second approach can be faster and more practical as the dimension of the subspace is $35 < 151$, which results in a considerable saving in computation time.

2.4. Eigenimage Gallery

In this section, a collection of eigenimages are shown from the control sets used to display the results, and from the global space computed using a concatenation of all the provided training images.

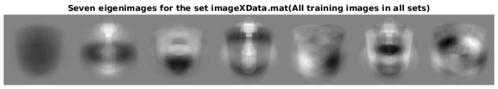
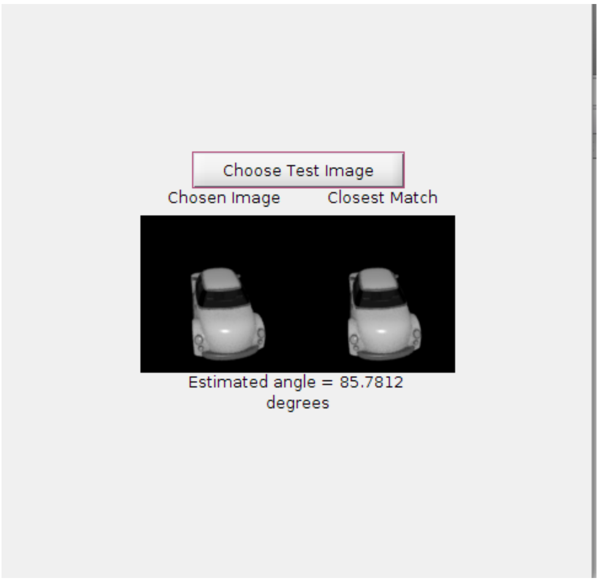


Figure 13. Eigenimages for the combination of all training set (related to recognition before orientation detection)



Estimated angle = 85.7812
degrees

Figure 14. GUI for matching objects

3. Matching Test Images to the Training Set

To match test images to the training set, a GUI was developed where the user clicks a browse button, navigates to the test images, and the matching is done on the global manifold first. Depending on where the match happens in the global manifold, a local manifold is chosen, and then the object is matched in that manifold. Figure 17 shows an example result.

4. Using Fourier Techniques to Estimate Eigenimages

As developed in [1], both the energy recovery ratio and the singular vectors/singular basis can be computed in the Fourier space with a significantly smaller computational expense. The approach exploits the circulant nature of the data taken under a controlled environment and proves that using DFT matrices, an energy recovery ratio for a circulant set can be computed and will always satisfy the actual energy recovery ratio computed using the data from the SVD.
The algorithm on page 8 of [1] has been implemented in order to estimate the eigenimages, the following results show both the estimates and the actual eigenimages computed for the project.

The main reason the eigenimages deviate after some rotation is because the rotation is not in two dimensions, rota-

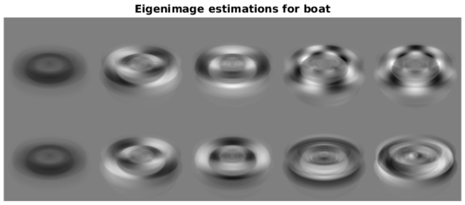


Figure 15. Actual (top) and estimate (bottom) eigenimages for the boat set

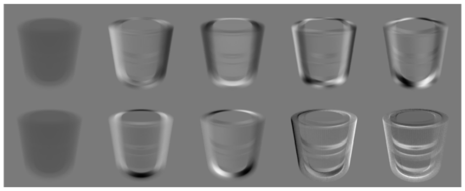


Figure 16. Actual (top) and estimate (bottom) eigenimages for the cabinet set

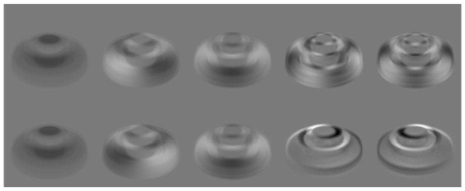


Figure 17. Actual (top) and estimate (bottom) eigenimages for the car set

tions of 3d objects from the views we are testing with result in a response dominated by a harmonic, but purely sinusoidal.

References

[1] C.-Y. Chang, A. Maciejewski, and V. Balakrishnan. Fast eigenspace decomposition of correlated images. *Image Processing, IEEE Transactions on*, 9(11):1937–1949, Nov 2000.