

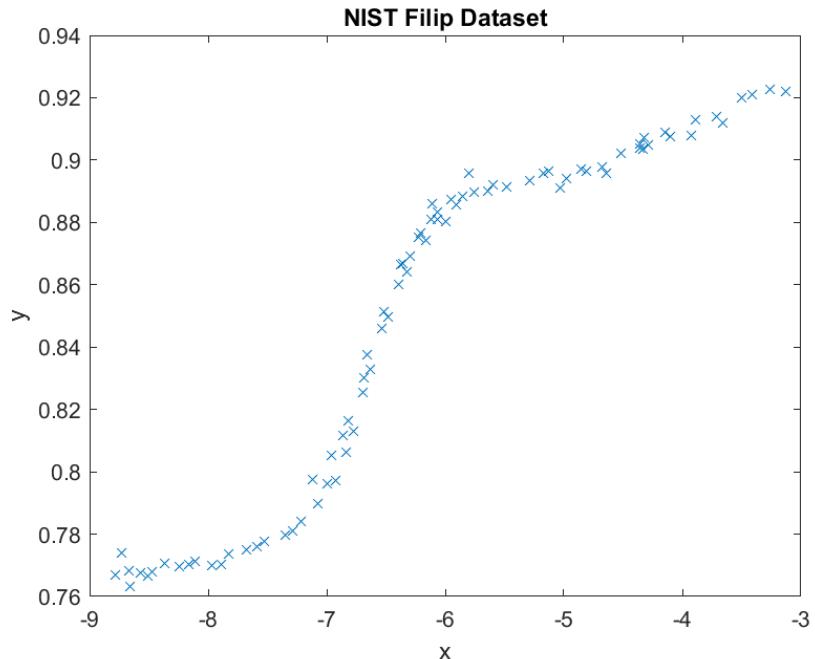
HW3  
CS 6210

Spencer Peterson

10/25/2021

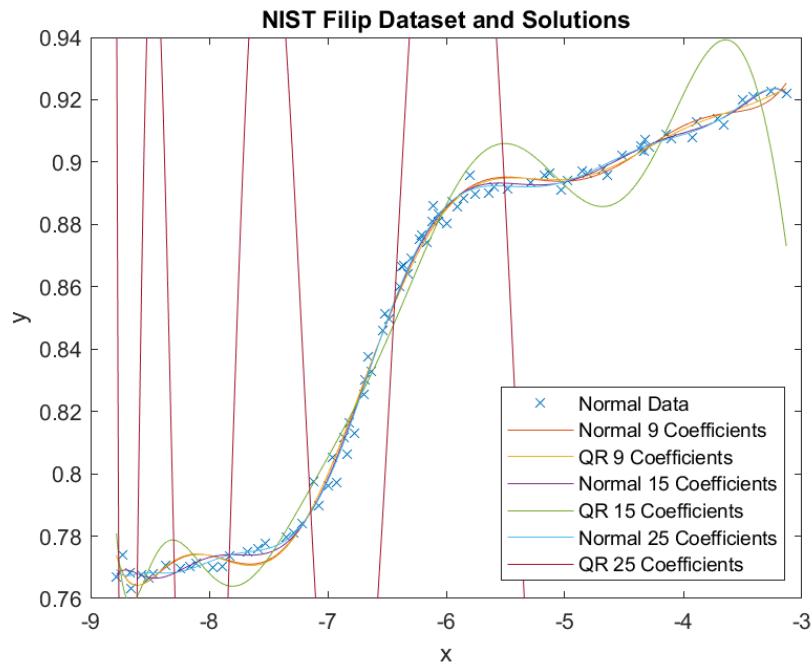
## 1 Filip Dataset

### a. Plot the Data



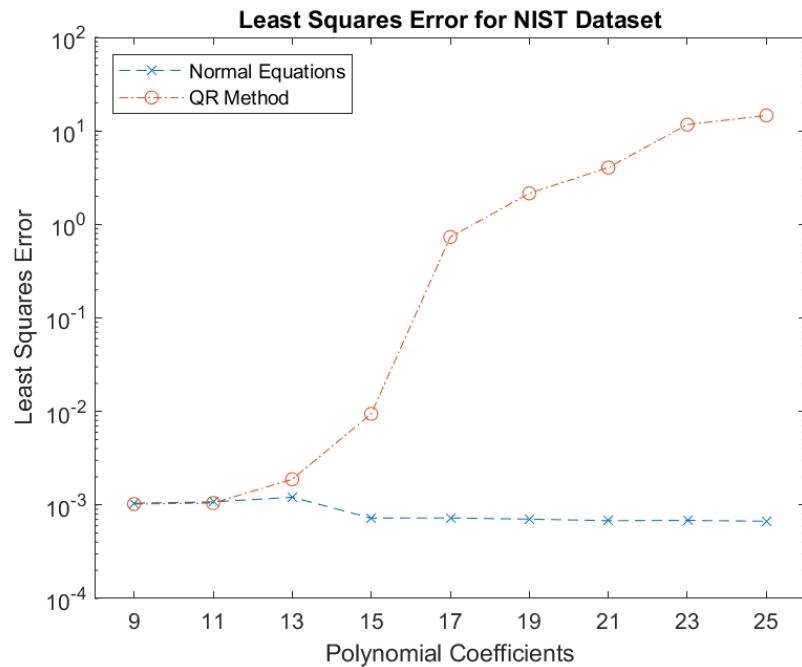
### b. Solutions

The results are shown in the image below. The normal equations tend to perform better with lots of coefficients than the QR method in this case. This is probably due to the fact that the QR decomposition is more dependent on machine precision than the normal data.



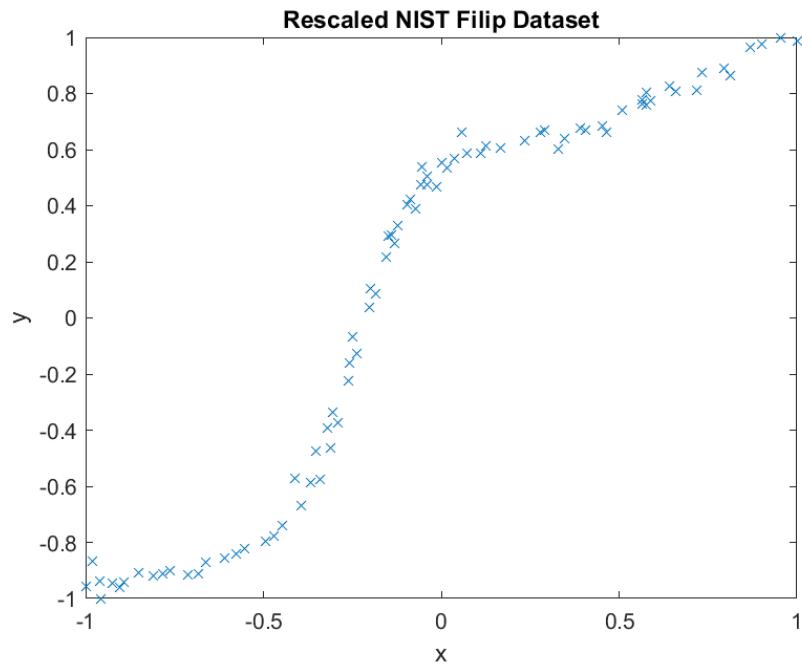
### c. Least Squares Error

The fact that the QR decomposition is less accurate is supported by the least squares errors. The results are shown in the figure below.

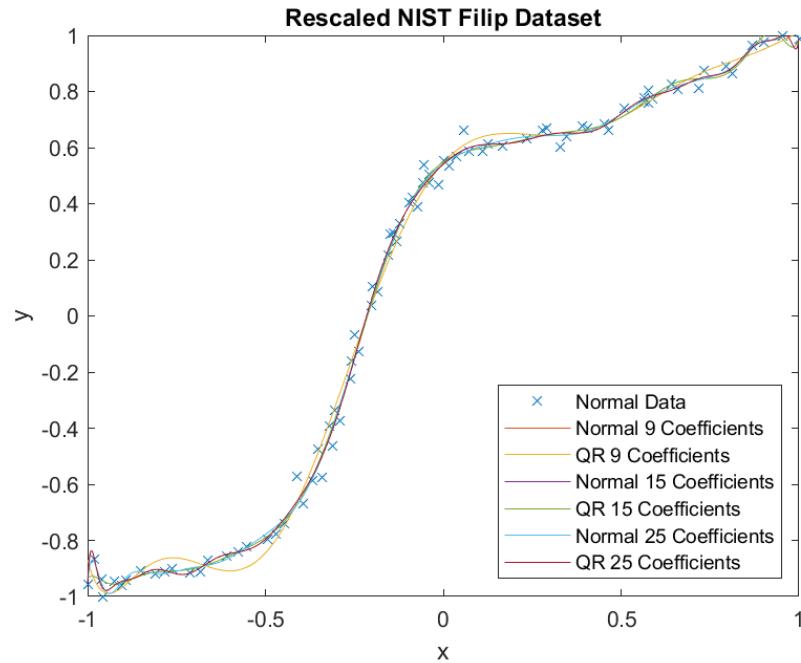


#### d. Scaling the QR Method

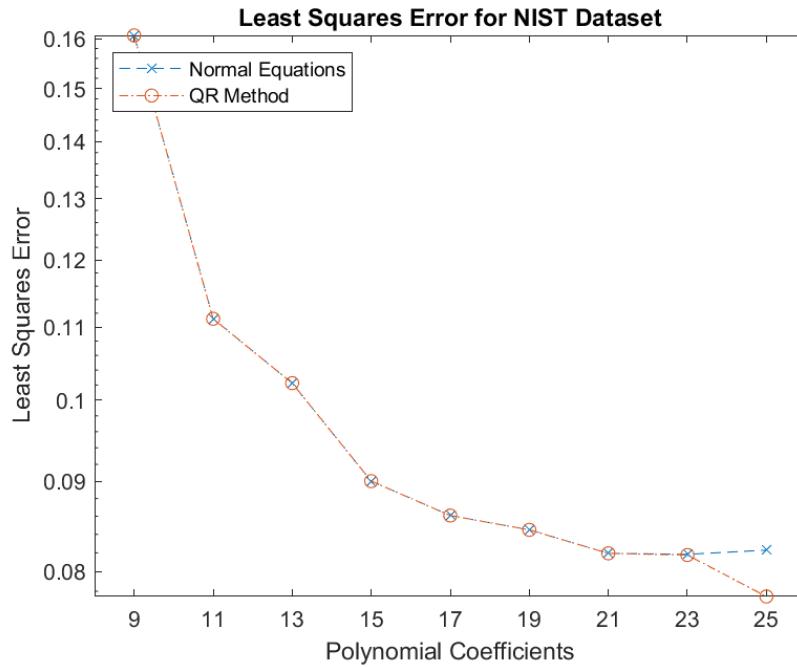
The scaled data is shown in the figure below.



The solutions are shown below for the QR solution. The scaling seems to have fixed the issue with the QR method getting more error with more coefficients. The before is shown above.



The error for the QR method and normal method is shown below:



### e. Reliable Results

The QR method provides better results if the data is well scaled. This may be an issue for some datasets, so the most reliable and robust results seems to be the normal equations.

## 2 SVD for Image Processing

### a. SVD compression

The figures for the image compression are shown below.

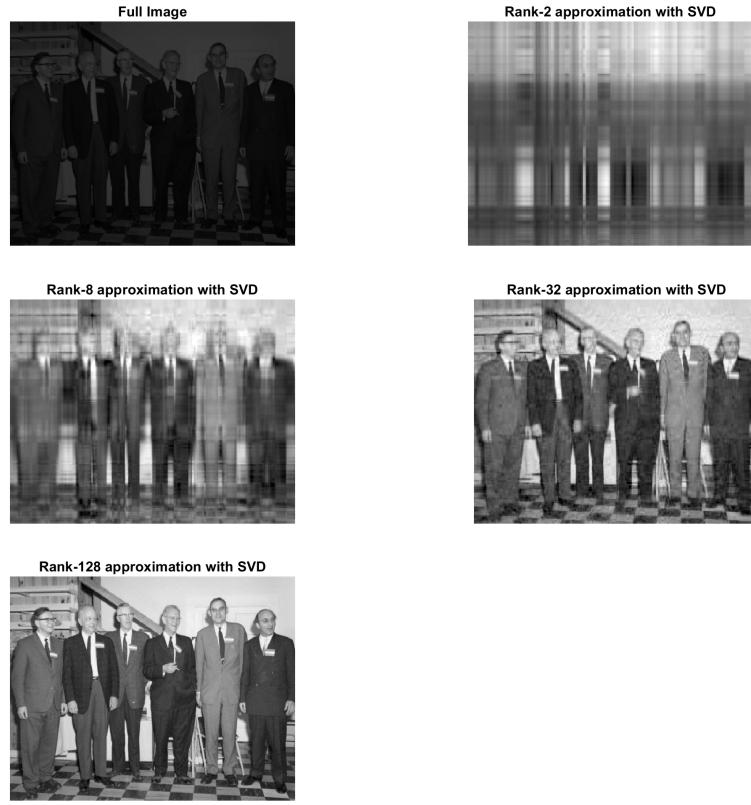


Figure 1: Gatlin Image Approximation

### b. Comments

The rank 32 approximation is the first approximation that really resembles the original image. However, the general outlines are shown even in the rank 8

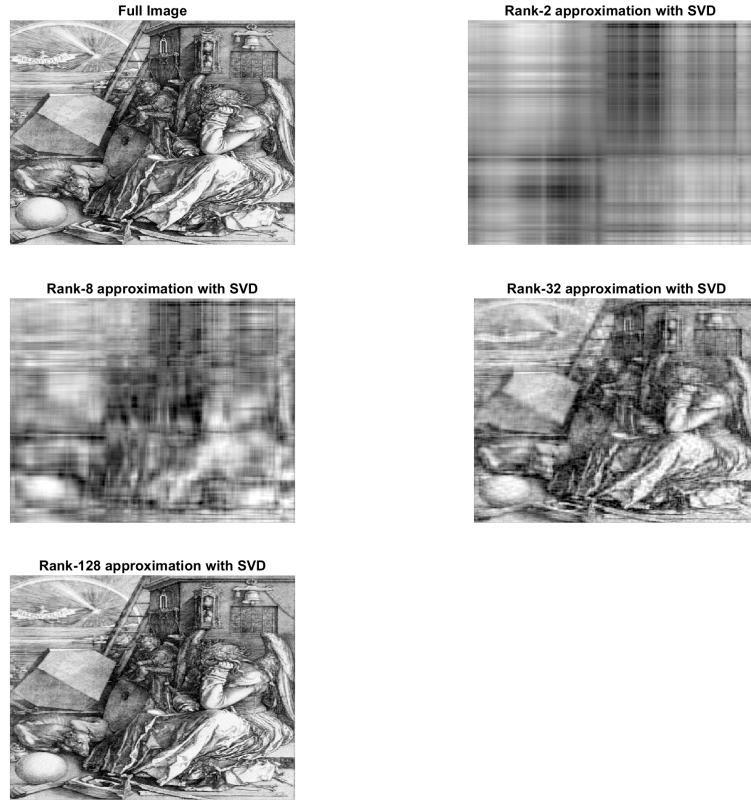


Figure 2: Durer Image Approximation

approximation. For a normal person, the rank 128 approximation is identical to the original image.

The storage is equal to  $O((M + N)r)$  where M is the height of the image in pixels, N is the width in pixels and r is the number of singular values. In this case, it is  $1120r$ .

### c. rSVD

#### i. Image Quality

The image approximations by the SVD and rSVD respectively are shown in the figures below. The rSVD approximation with the 32 rank, which earlier we said was the first recognizable image, is slightly worse for the rSVD than the SVD. However, the lower rank approximations look slightly better, especially for the Gatlin image.

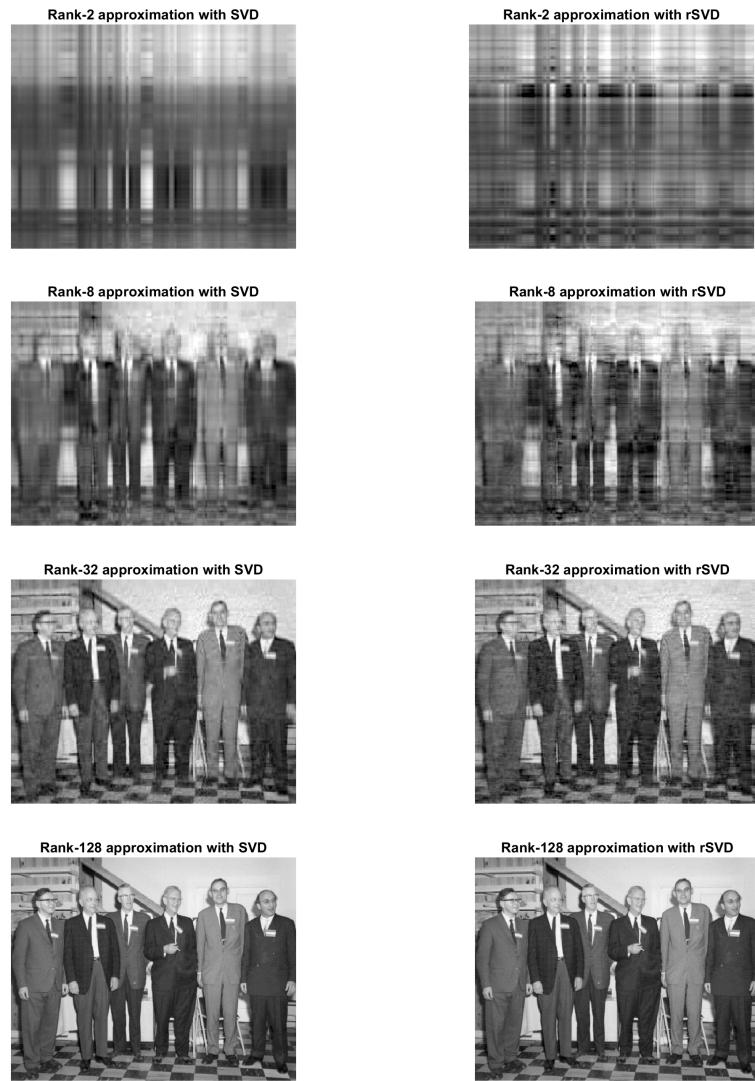


Figure 3: Gatlin, SVD vs rSVD

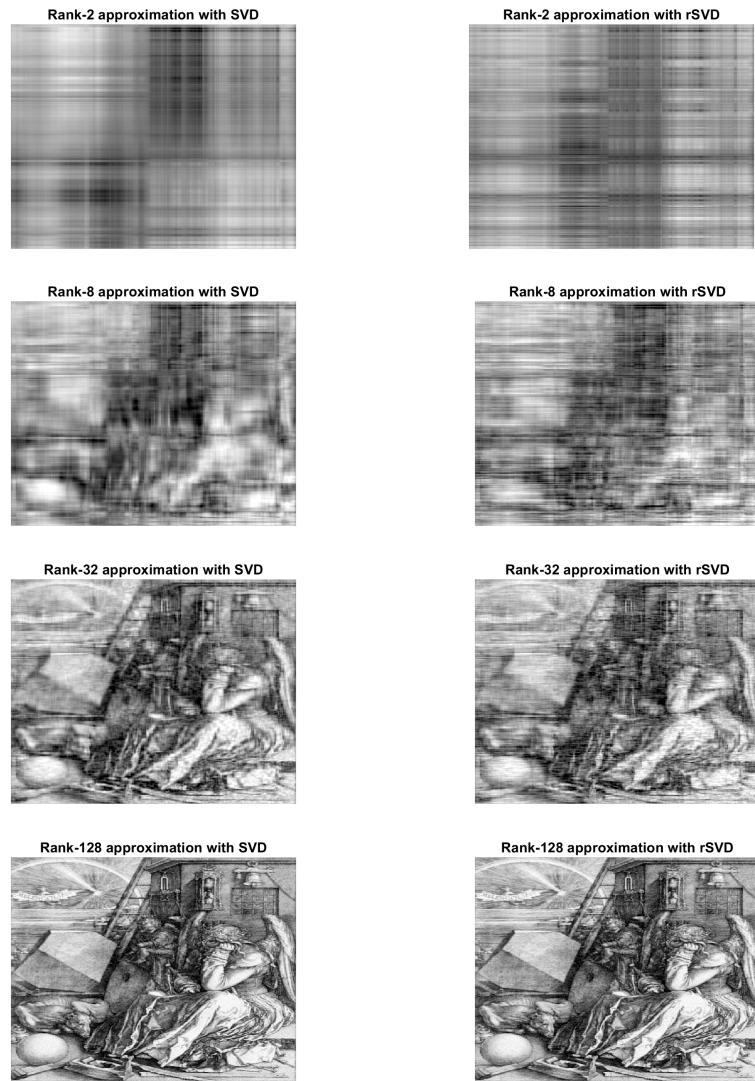
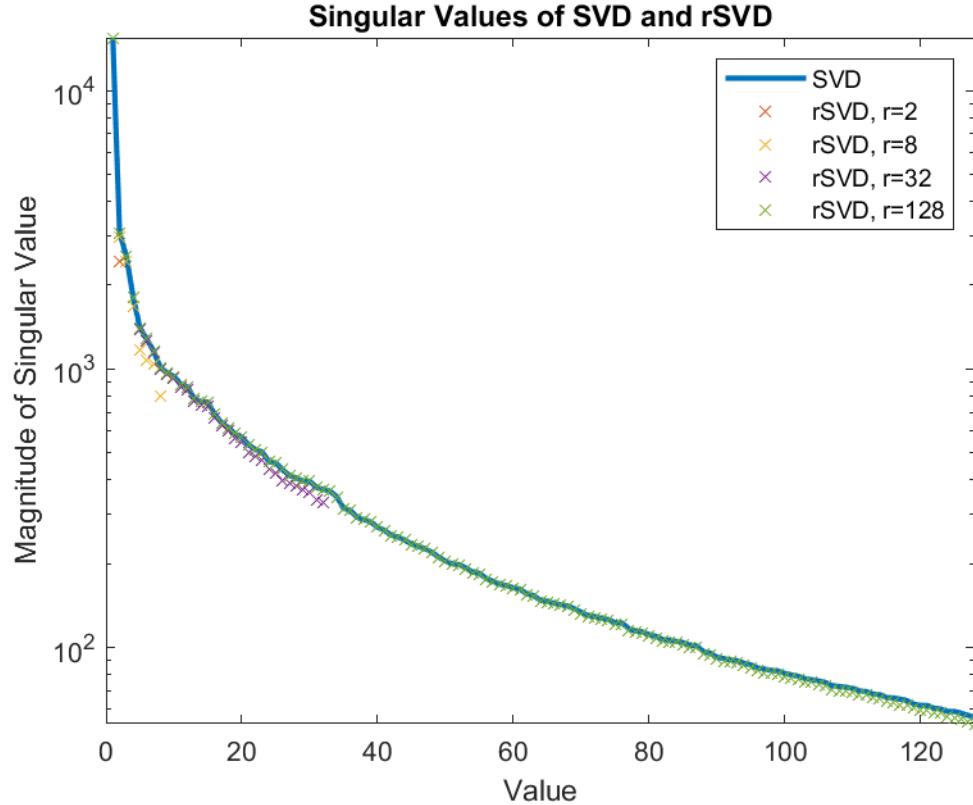


Figure 4: Durer, SVD vs rSVD

## ii. Accuracy of Singular Values

The singular values for the rSVD tend to be slightly smaller than the singular values for the SVD. However the differences are fairly slight and improve the higher rank we use.



## iii. Timing

The timing results for the SVD and different ranks of rSVD are shown in the table below. All of the rSVD times are faster than the SVD, however the rank 128 rSVD takes nearly as much time as the normal SVD and provides half of the information. So for more complete results, one might as well use the full SVD.

SVD	rSVD-2	rSVD-8	rSVD-32	rSVD-128
$0.0479 \pm 0.008$	0.0020	0.0029	0.0058	0.0417

## d. Power Method

It is difficult to see the difference the power method makes in the images as shown in figure 5. However, we have found the error between between the

singular values calculated by the full SVD and the different singular values produced by the power method and shown the results in the table below. You can see that the larger the power, the more similar the rSVD's result to that of the SVD.

	q=1	q=2	q=3
Gatlin	1.56	0.0354	0.0306
Durer	45.72	5.11	0.339

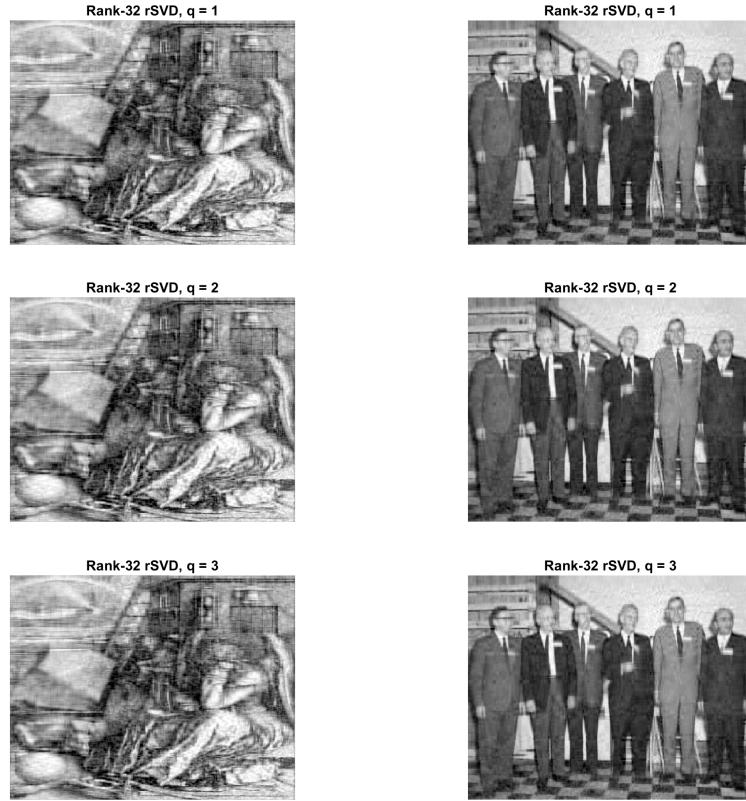


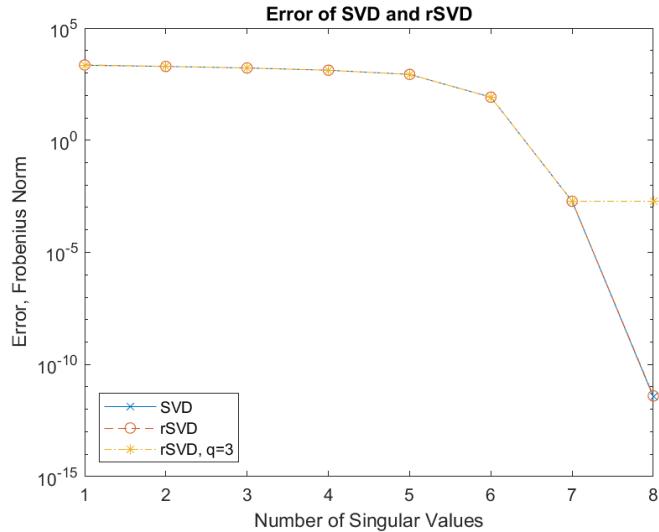
Figure 5: Power method images

The computation times are shown in the table below. The power method didn't seem to have a large effect on the timing with this size of matrix.

	q=1	q=2	q=3
Gatlin	0.106	0.0076	0.0074
Durer	0.0082	0.0058	0.0062

### 3 Matrix

We used the SVD, rSVD, and rSVD with a power method to approximate the matrix. The error, which was found as the Frobenius norm of the approximated matrix subtracted from the original matrix is shown in the image below.



All of the method performed fairly similarly. The accuracy is very bad until the using more than six singular values. This is because the Eigenvalues for this matrix are very large in magnitude. Using the eig command on the matrix returns [1030, 1020, 1020, 1000, 911, 81.3, -0.00202, -1020]. The assumption that the error is approximately equal to the next singular value doesn't hold because the singular values are very large and very close to each other. This means that we need almost all of the singular values to appropriately approximate this matrix.