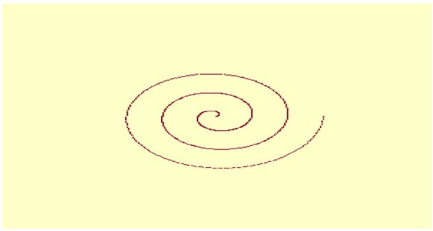
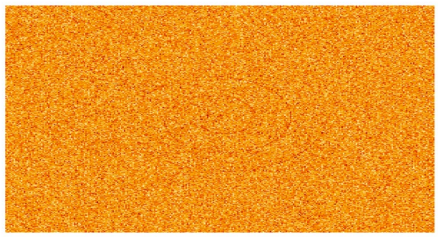
Image Recovery

Zach Fuller

This project's purpose is to demonstrate how to improve the resolution of a "noisy" image using singular value decomposition (SVD) and thresholding. The code, with comments, can be found in "Image Recovery.R"

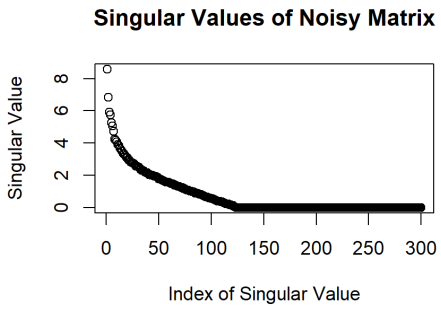
To start, I generate a simple image and add random Gaussian noise.

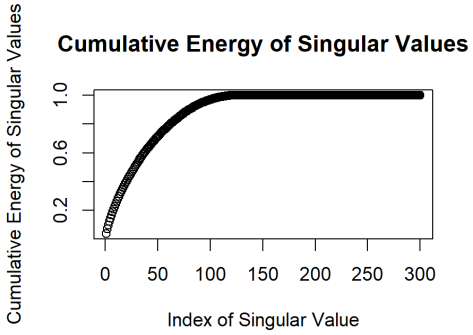




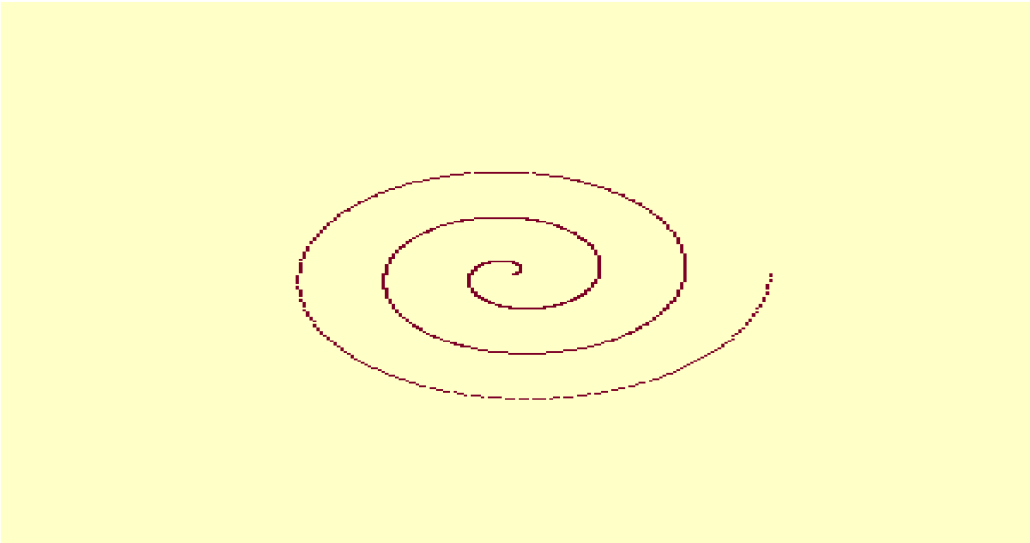
If you look closely, you can still see the spiral from the original image, but it is much less clear.

To recover the original image from the noisy image, I start by finding the SVD of the image matrix. The goal is to find a sum of singular values that captures most of the "energy".



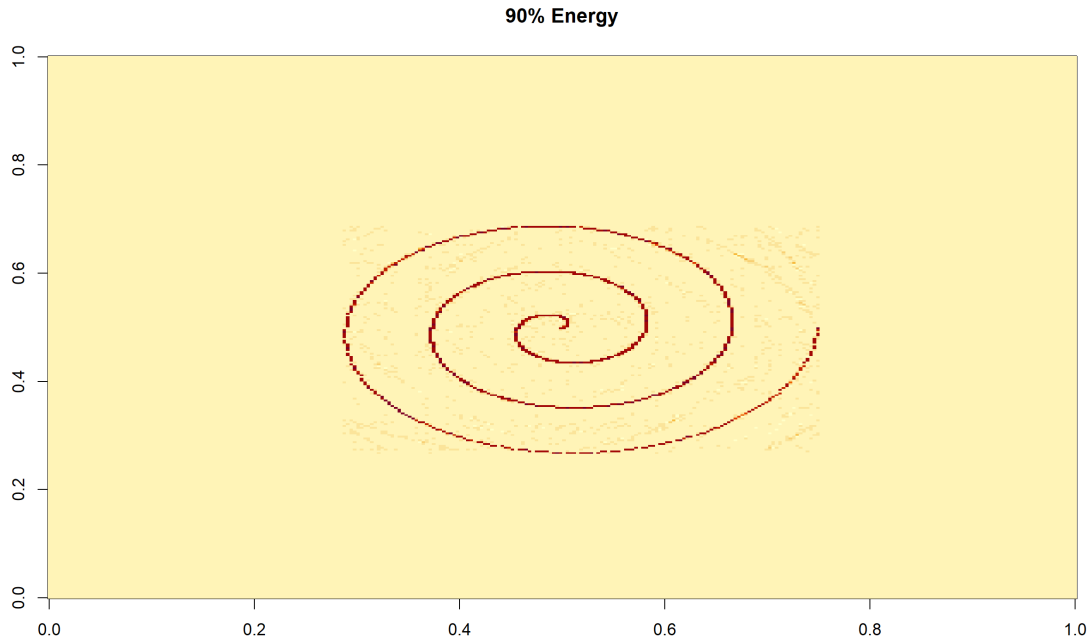


To start, look at an image reconstructed from the full SVD.



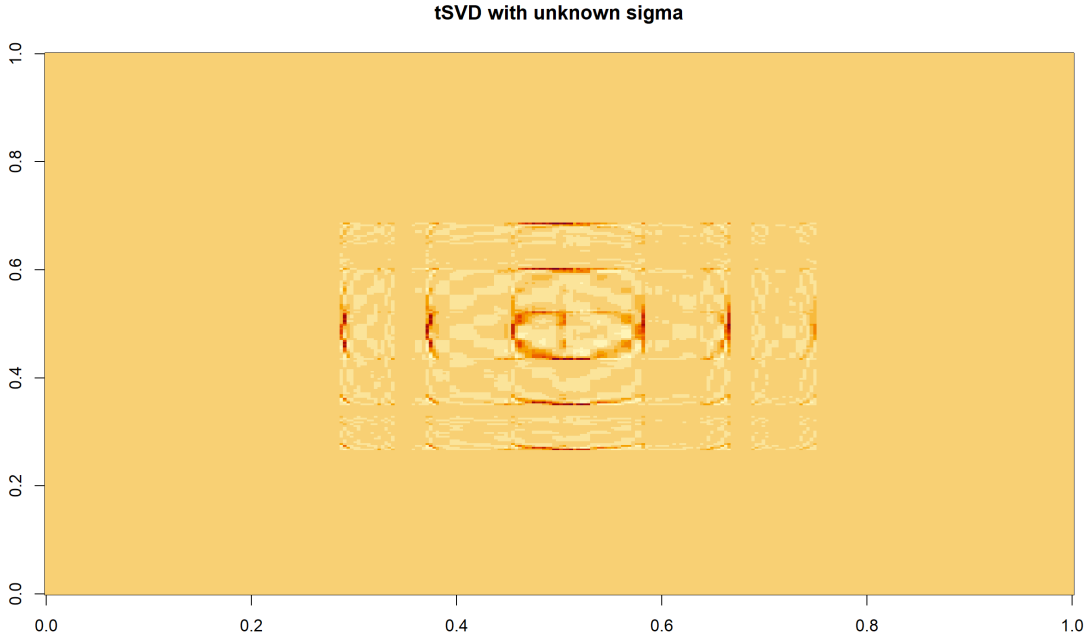
To my sight, this is indistinguishable from the original image. Now, why should I care to look at filtering out SVDs? In this example, I generated the original image with a 300x300 matrix of values. For a typical photograph, you are likely looking at something larger. To save on storage, or computation time, you might want to see if you can get away with fewer singular values.

When setting a threshold to retain 90% of the energy, we still have some residual noise, but you can tell what the original image was. The nice thing about this is that I went from 300 singular values to 81, so less information was needed.

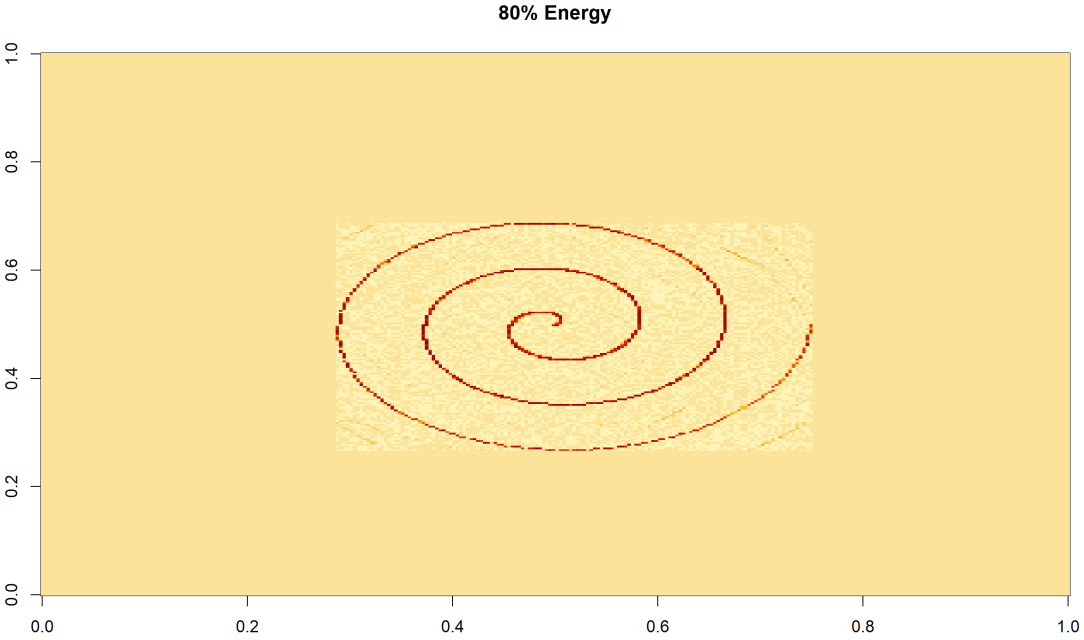


An alternative approach, based on the threshold suggested by Davish and Donoho (2013), ends up leaving me with just 4 singular values.

The image isn't quite clear in this case, but enough information is retained that you could possibly guess at the underlying image.



Just for fun, see what 80% energy looks like. For 80% energy, only 62 singular values are needed.



It’s quite interesting to see that an image can be recovered from a image that has been obfuscated with noise! We can also see that thresholds can be set to save on storage, while still retaining information needed to reconstruct the image.