Description of the SVD Filter

Martin J. Gander

April 17, 1997

1 Introduction

In this brief note I explain how the singular value decomposition can be used in image compression and in image filtering. I'm giving the minimal mathematical background necessary and show some experiments. References into relevant literature are also given.

2 Mathematical Background

Every real matrix A can be decomposed into a product of three matrices

$$A = U\Sigma V^T$$

where U and V are orthogonal matrices, $U^TU = I$ and $V^TV = I$ and Σ is a diagonal matrix, $\Sigma = diag(\sigma_1, \sigma_2 \ldots)$ and σ_j denotes the j^{th} singular value of A, $\sigma_j \geq 0$. This decomposition is called the singular value decomposition (SVD) of A. Usually it is organized such that the singular values are ordered, $\sigma_1 \geq \sigma_2 \geq \sigma_3 \ldots$. I denote the column vectors of U and V by $\boldsymbol{u}_j$ and $\boldsymbol{v}_j$.

The spectral norm $||A||_2$ of a matrix A, which is a measure for the size of the entries of A, is given by the largest singular value of A,

$$||A||_2 \equiv \sigma_1.$$

If A is singular then some of the singular values are equal to zero. More precisely if A has rank r then only the first r singular values $\sigma_1, \sigma_2, \ldots, \sigma_r$ are non-zero.

3 The Singular Value Decomposition of an Image

I restrict my description of the filtering algorithm to gray scale images. Generalization to color images are possible. Every gray scale image can be thought of as a matrix A which contains for each pixel in the image a number which corresponds to the gray scale of that pixel. I assume that if two numbers in the matrix A are close in magnitude, the gray shades they represent are similar.

Computing the singular value decomposition of the image represented by A, I get

$$A = U\Sigma V^T = \begin{bmatrix} & & \\ \boldsymbol{u}_1 & \boldsymbol{u}_2 & \dots \\ & & \end{bmatrix} \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \ddots \end{bmatrix} \begin{bmatrix} & \boldsymbol{v}_1^T & \\ & \boldsymbol{v}_2^T & \\ & \vdots & \end{bmatrix}.$$

Then the image A can be written as a sum of images,

$$A = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^T + \sigma_1 \boldsymbol{u}_2 \boldsymbol{v}_2^T + \dots.$$

4 Using SVD to Compress an Image

Now since the singular values are ordered, the main contribution to the image comes from the first terms in the sum. An example is shown with the image sequence in figure 1. Even zooming into the image one can hardly notice a difference between the completely restored image using all the 512 singular values and the one restored with 200 singular values only. This suggests that the singular value decomposition could be used to compress images allowing some loss. Important topics to be addressed are the following:

- Should the SVD be applied to the whole image or to subblocks seperately ?
- The singular vectors in the compressed format consist of real numbers, while the original image consists of bytes. To store a byte one needs 8 bits, to store a real one needs 64 bits. Could one develop an algorithm which computes the SVD with less accuracy to get a better compression ? (How about fixed point arithmetic ?)
- A natural measure for the loss is the spectral norm of the error, which equals the size of the first neglected singular value. Is this a good measure for the visibility of the error ?
- If there are in fact some singular values identical to zero, either using subblocks or the whole image, then one can get lossless compression. Imagine that one part of the image is a uniformly gray wall. Then a subblock in that part of the image has rank one and therefore one has to keep only one singular value and the corresponding singular vectors, without loosing any information. This is the basic idea of Shiskins algorithm, although he is using a slightly different decomposition. The question is how often does that occur in a real example. I could imagine that in a video, where the difference of images is sent, this occurs quite frequently, especially if there is not much movement in the frame.

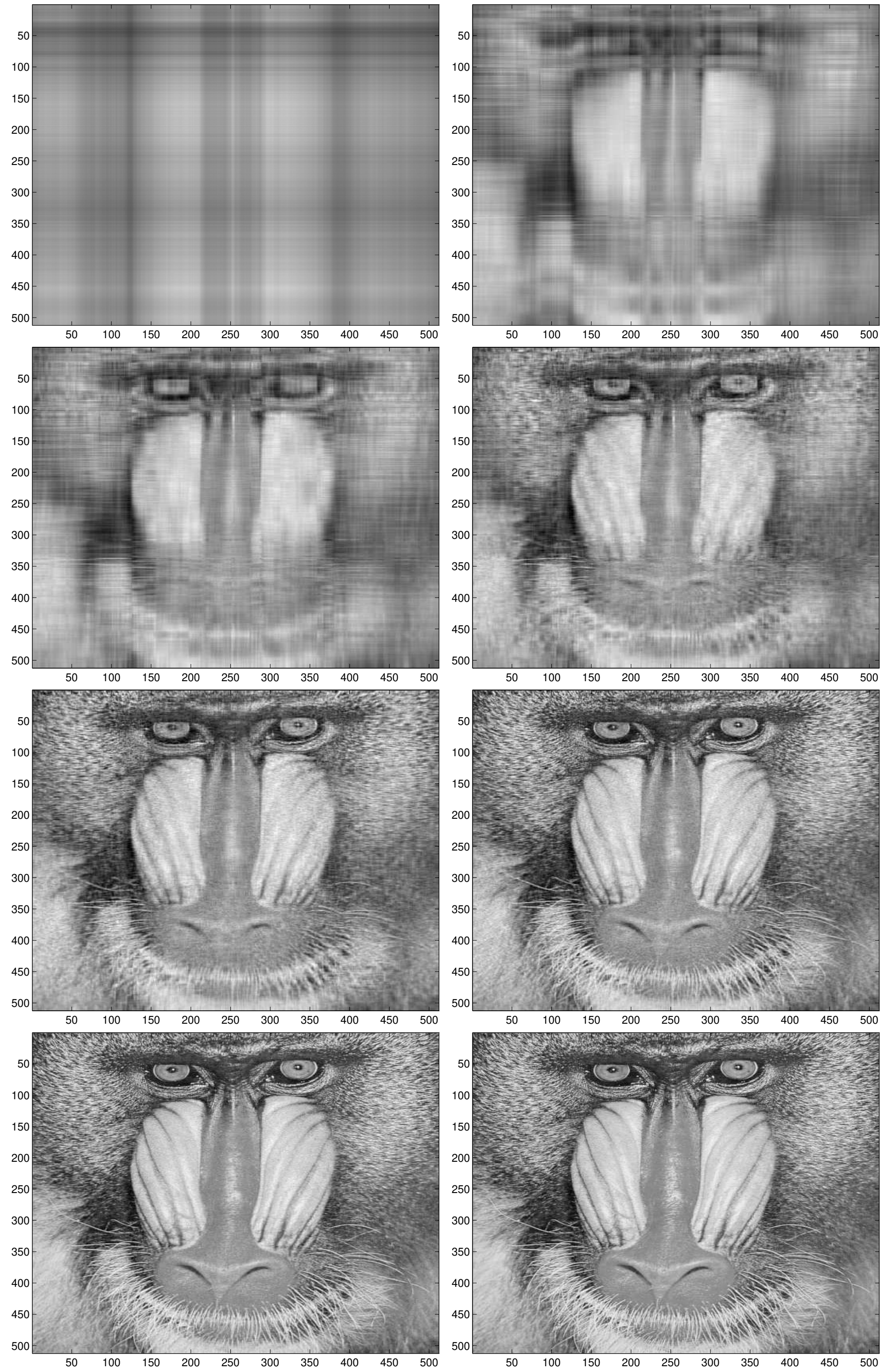

Figure 1: Reconstruction of the baboon using 1, 5, 10, 20, 50, 100, 200 and all 512 singular values

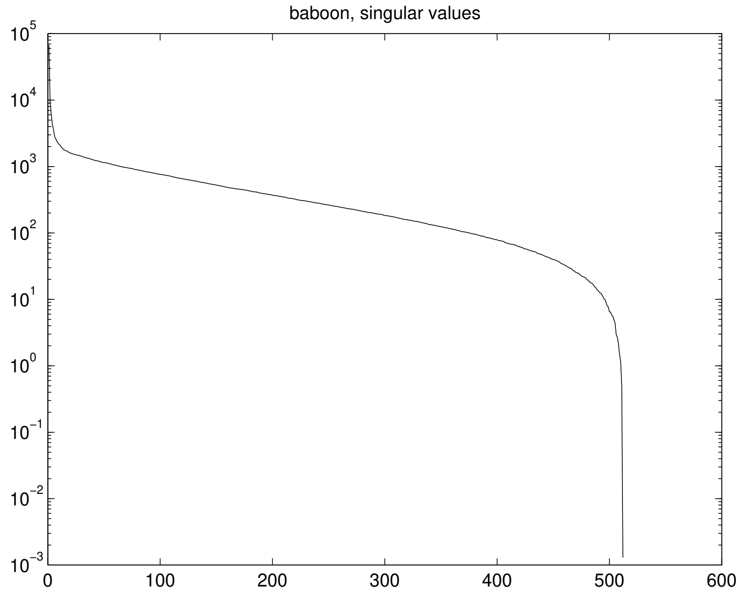

Figure 2: Decay of the singular values for the baboon picture

Figure 2 shows the decay of the size of the singular values for the example given in figure 1. The shape of the curve is typical for all the example images we tried. The first few singular values seem to contain the main contributions to the image. After that there is a plateau where the decay of the singular values is close to linear (note the logarithmic scale in the plot). Finally the last singular values decay quickly to zero.

5 Using SVD as a Filter

An interesting question is: what do the singular values on the plateau and the last few contribute to the picture ? Figure 3 shows their contribution enhanced so it becomes visible. The remaining part of the image looks like random noise. This fact has been used successfully to filter images and reach a better compression rate for standard compression algorithms by either neglecting small singular values or puting all the singular values through a nonlinear filtering function [1], [2] and [**?**]. An extended study of the application of the SVD in image processing has been done in [3]. Further indication that noise is hard to compress can be found in [4] and the references given therein. How could the SVD filter be used to enhance the performance of fractal compression algorithm ? The experiments we performed show that this is possible. However one has to accept some loss in the quality of the image.

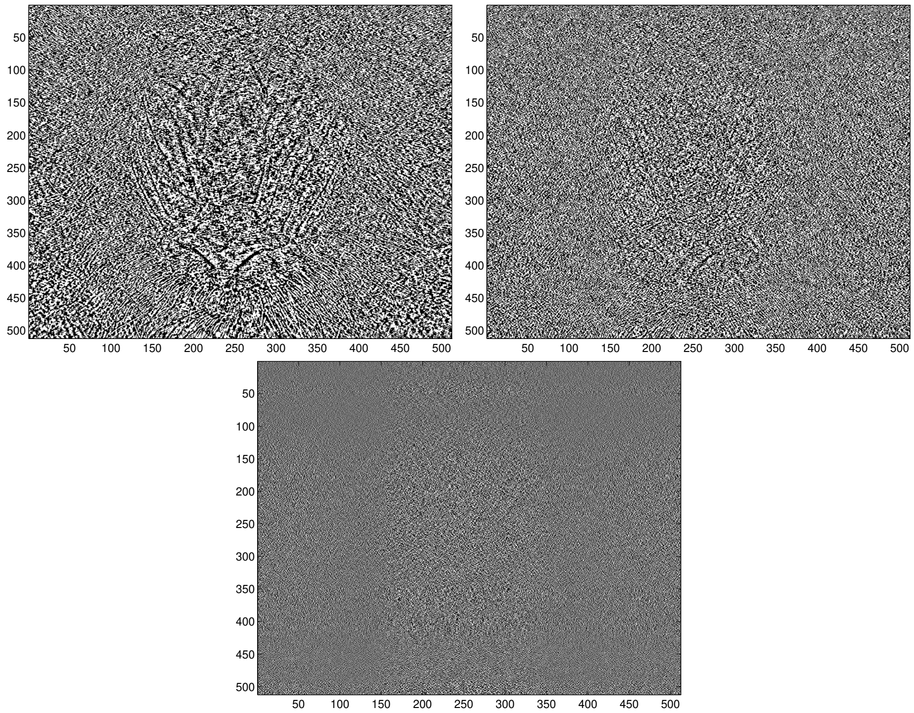

Figure 3: Remaining image without using the first 100, 200 and 400 singular values

6 Conclusion

I described the two basic ideas how to use SVD to compress or filter images. I related the SVD compression method to the method proposed by Shishkin, which is a lossless compression method. Considering SVD as a filter, I think one should compare SVD to other filtering methods, like methods based on wavelets.

References

[1] K. Konstantinides and G.S. Yovanof, Improved compression performance using SVD-based filters for still images, SPIE Vol. 2418.

[2] K. Konstantinides and G.S. Yovanof, Application of SVD based spatial filtering to video sequences, IEEE 1995.

[3] H. C. Andrews, Singular Value Decomposition and Digital Image Processing, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-24, no. 1, February 1976.

[4] B.K. Natarajan and V. Bhaskaran, Effective Nearly Lossless Compression of Digital Video Sequences via Motion-Compensated Filtering, IEEE 1994.

[5] H.C. Lee, Digital Image Noise Suppression Method Using SVD Block Transform, United States Patent Number 5010504.