THE UNIVERSITY OF ARIZONA®

An Application of Linear Algebra to Image Compression

Paul Dostert

July 2, 2009

Image Compression

There are hundreds of ways to compress images. Some basic ways use singular value decomposition

Suppose we have an 9 megapixel gray-scale image, which is 3000×3000 pixels (a 3000×3000 matrix). For each pixel, we have some level of black and white, given by some integer between 0 and 255. Each of these integers (and hence each pixel) requires approximately 1 byte to store, resulting in an approximately 8.6 Mb image.

A color image usually has three components, a red, a green, and a blue (RGB). EACH of these is represented by a matrix, so storing color images takes three times the space (25.8 Mb).

We will look at compressing this image through computing the singular value decomposition (SVD).

Singular Value Decomposition (SVD)

Any nonzero real $m \times n$ matrix A with rank $r > 0$ can be factored as $A = P\Sigma Q^T$ with P an $m \times r$ matrix with orthonormal columns, $\Sigma = diag\,(\sigma_1, \ldots, \sigma_r)$ and Q^T an $r \times n$ matrix with orthonormal rows. This factorization is called the ***singular value decomposition (SVD)*** .

This is directly related to the spectral theorem which states that if B is a symmetric matrix ($B^T = B$) then we can write $B = U\Lambda U^T$ where Λ is a diagonal matrix of eigenvalues and U is an orthonormal matrix of eigenvectors.

To see the relationship, notice:

$$\begin{aligned} A^TA &= Q\Sigma^T P^T P\Sigma Q^T = Q\Sigma^2 Q^T \\ AA^T &= P\Sigma Q^T Q\Sigma^T P^T = P\Sigma^2 P^T \end{aligned}$$

These are both spectral decompositions, hence the σ_i are the positive square roots of the eigenvalues of A^TA. In the SVD, the matrices are rearranged so that $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$.

Reducing the SVD

Using the SVD we can write an $n \times n$ invertible matrix A as:

$$\begin{aligned} A &= P\Sigma Q^T = (\mathbf{p_1},\, \mathbf{p_2}, \ldots, \mathbf{p_n}) \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & \sigma_n \end{pmatrix} \begin{pmatrix} \mathbf{q_1}^T \\ \mathbf{q_2}^T \\ \vdots \\ \mathbf{q_n}^T \end{pmatrix} \\ &= \mathbf{p_1}\sigma_1\mathbf{q_1}^T + \mathbf{p_2}\sigma_2\mathbf{q_2}^T + \cdots + \mathbf{p_n}\sigma_n\mathbf{q_n}^T \end{aligned}$$

Since $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$ the terms $\mathbf{p_i}\sigma_i\mathbf{q_i}^T$ with small i contribute most to the sum, and hence contain the most information about the image. Keeping only some of these terms may result in a lower image quality, but lower storage size. This process is sometimes called Principal Component Analysis (PCA).

Issues in using PCA

- A requires n^2 elements. P and Q require n^2 elements each and Σ requires n elements. Storing the full SVD then requires $2n^2 + n$ elements.
- Keeping 1 term in the SVD, $\mathbf{p_1}\sigma_{11}\mathbf{q_1}^T$, requires only $2n + 1$ elements.
- If we keep $k \approx \frac{n}{2}$ terms, then storing the reduced SVD and the original matrix are approximately the same.
- The P and Q are normalized (each $\mathbf{p_i}, \mathbf{q_i}^T$ has norm 1) so the error in the reduced SVD is given by only the σ values:

$$\text{Error} = 1 - \frac{\sum_{i=1}^{k} \sigma_i}{\sum_{i=1}^{n} \sigma_i}$$

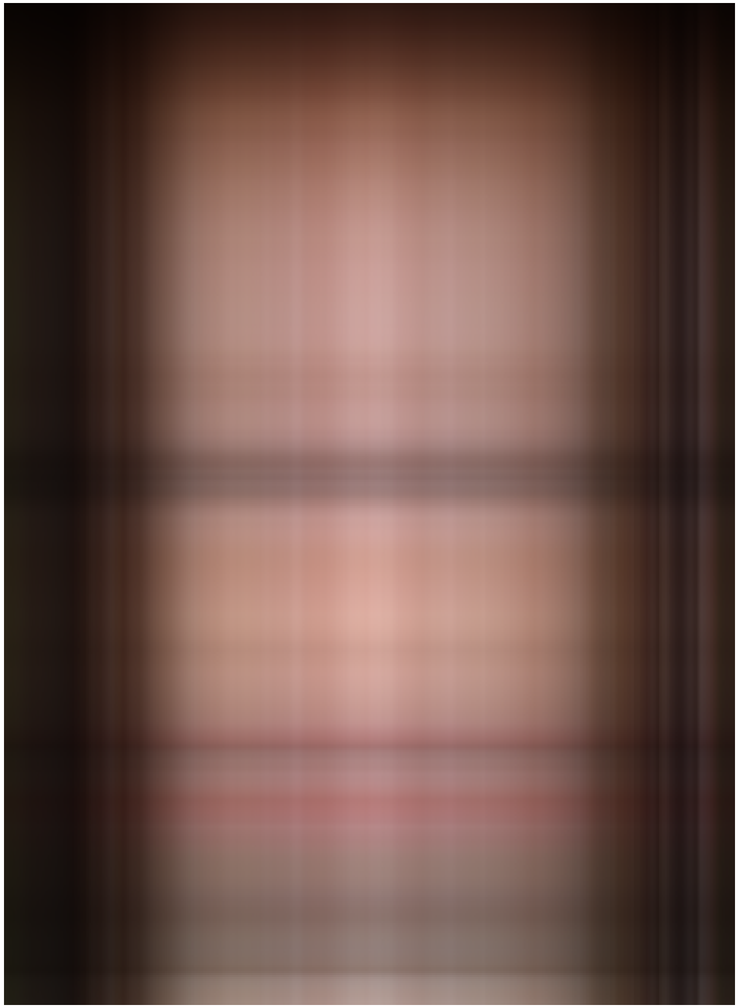
- Color images are often in RGB (red, green, blue) where each color is specified by 0 to 255. This gives us three matrices. The reduced SVD can be computed on all three separately or together.

Examples: 1 Term

The following is a 500×500 image. The reduced SVD was applied equally to each color:

Original

Using 1 terms



Examples: 3 Terms

The following is a 500×500 image. The reduced SVD was applied equally to each color:

Original

Using 3 terms

Examples: 5 Terms

The following is a 500×500 image. The reduced SVD was applied equally to each color:

Original

Using 5 terms

Examples: 10 Terms

The following is a 500×500 image. The reduced SVD was applied equally to each color:

Original

Using 10 terms

Examples: 20 Terms

The following is a 500×500 image. The reduced SVD was applied equally to each color:

Original

Using 20 terms

Examples: 30 Terms

The following is a 500×500 image. The reduced SVD was applied equally to each color:

Original

Using 30 terms

Examples: 40 Terms

The following is a 500×500 image. The reduced SVD was applied equally to each color:

Original

Using 40 terms

Examples: 50 Terms

The following is a 500×500 image. The reduced SVD was applied equally to each color:

Original

Using 50 terms

Examples: 75 Terms

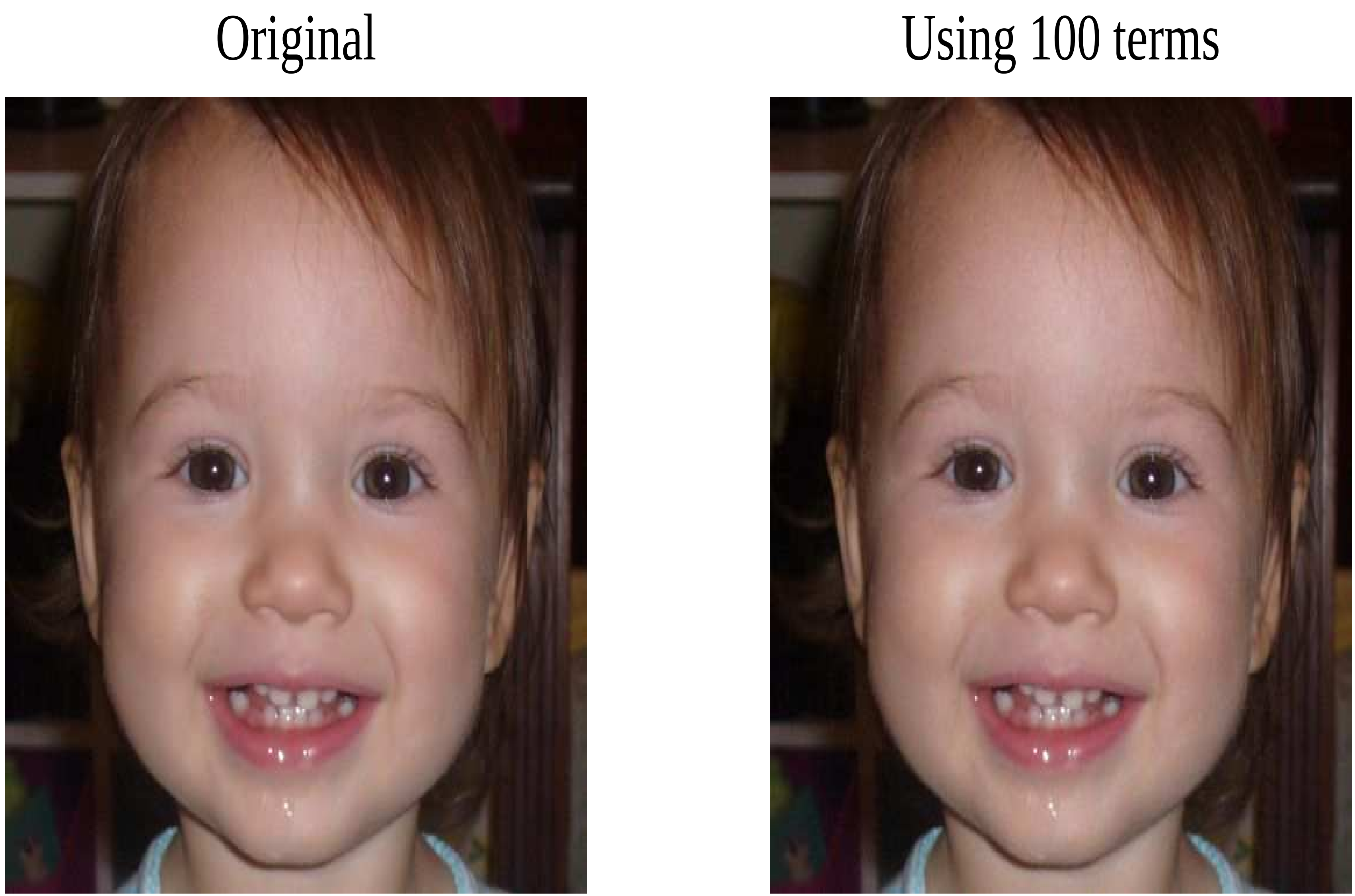
The following is a 500×500 image. The reduced SVD was applied equally to each color:

Original

Using 75 terms

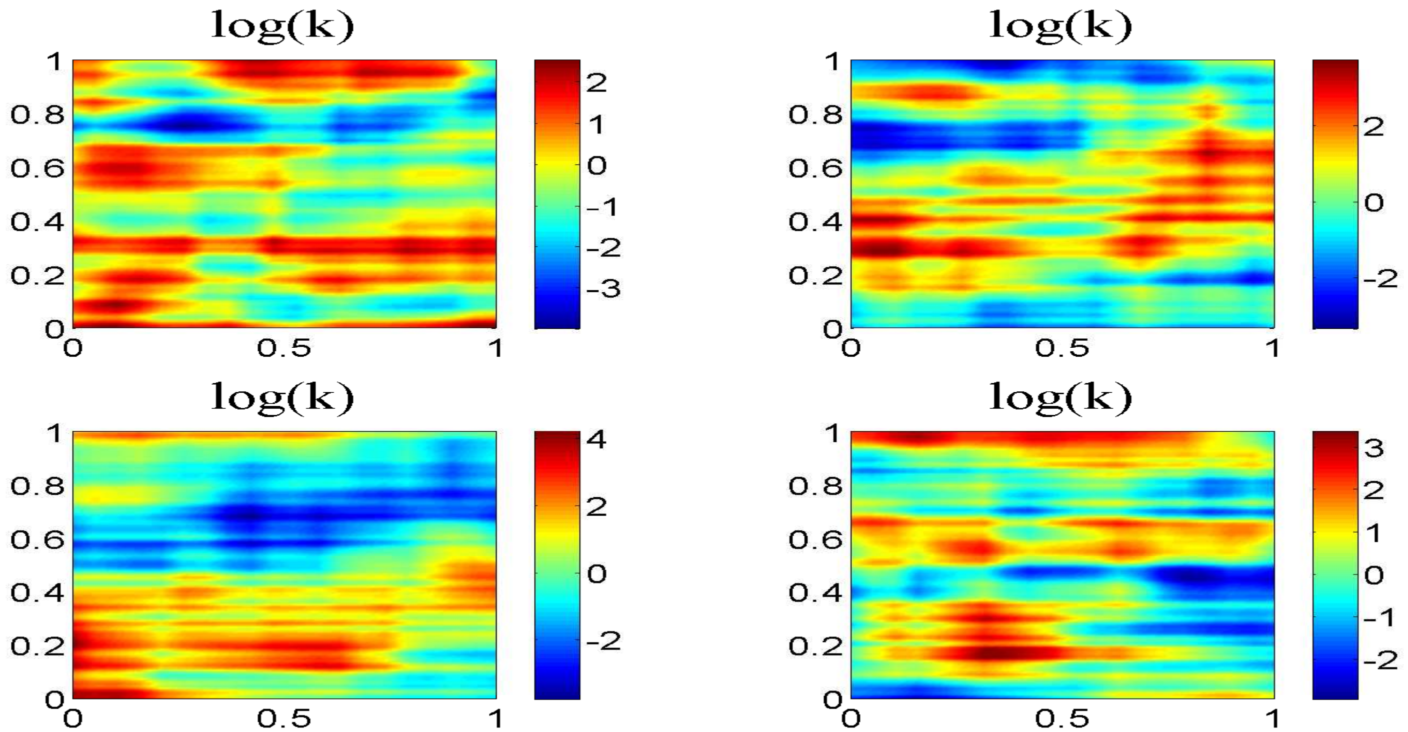
Examples: 100 Terms

The following is a 500×500 image. The reduced SVD was applied equally to each color:



Results & Other Applications

- $k = 100$ gives a fairly accurate reproduction, with 7.53% error.
- The reduced SVD stores $k\,(2n+1) = 100 \cdot (1001)$ numbers, $\approx 40\%$ of the original image size.
- Many uses besides image compression, such as parameterizing possible permeability profiles for underground reservoirs.



- Moral of the story: *take more linear algebra and numerical analysis.* There are hundreds of fun applications!