

SVD Compression

Let f be a given image represented as a m_r by m_c matrix. By applying the singular value decomposition (SVD) to f , we can write $f = U\Sigma V^T$, where U is an m_r by m_r orthogonal matrix ($U^T = U^{-1}$), Σ is an m_r by m_c diagonal matrix (0 except on its main diagonal) and V is a m_c by m_c orthogonal matrix ($V^T = V^{-1}$). The diagonal entries of Σ are the singular values of f , and by convention they can be ordered by decreasing magnitude,

$$\sigma_i = \Sigma_{i,i} \geq \Sigma_{i+1,i+1} = \sigma_{i+1}.$$

We can use the SVD to compress f by defining approximations f_s to f by $f_s = U\Sigma_s V^T$, where Σ_s keeps only the s largest singular values, $\sigma_1, \dots, \sigma_s$, replacing the rest with zeros. This is not necessarily the best way to compress images, but it is an interesting illustration of the SVD and that the largest singular values tend to correspond to the most important information.

By the Eckart-Young Theorem, f_s is the best rank s approximation to f in the sense of minimizing $\sum_{i,j} (f_{i,j} - g_{i,j})^2$ over all matrices g having exactly s nonzero singular values. Note that with only s nonzero singular values of f_s , it is only necessary to store the first s columns of U and V in order to represent f_s . Thus the total number of elements we need to store is $s(1 + m_r + m_c)$, which is less than $m_r m_c$ for $s < \frac{m_r m_c}{1 + m_r + m_c}$.

MATLAB: To run the accompanying code, SVD_compress.m, on the elvis test image, type

```
g = SVD_compress('elvis.bmp',r);
```

where r is the ratio of the retained singular values to the total number. For example, the result with $r = .1$ is shown in Figure 1.



Figure 1: Original image and approximation keeping $\approx 10\%$ of the singular values