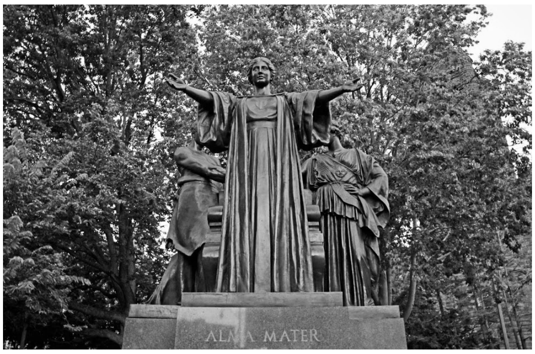
3 Image Compression

We have a grayscale picture that is $m \times n$ pixels in size:



Each pixel is a shade of gray from 0 (black) to 255 (white). This gives an $m \times n$ matrix A. Each entry of A is one pixel of the image; that entry is some integer from 0 to 255, giving the brightness of that pixel.

Encoding a large picture takes up a lot of space:

- Say our picture is 625×960 pixels.
- Each pixel has 256 possible values. (This takes up exactly 8 bits, or 1 byte, of memory on a computer).
- The whole picture requires $625 \times 960 = 600000$ bytes, so 600 kB.
- That is not so bad, but color pictures from your personal camera would be about 30MB each without any compression... and that quickly adds up.

Question. Can we do better?

Use singular value decomposition of A:

$$A = \begin{bmatrix} | & & | \\ \mathbf{u}_1 & \dots & \mathbf{u}_{625} \\ | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 & \\ 0 & \sigma_2 & \\ & & \ddots \end{bmatrix} \begin{bmatrix} - & \mathbf{v}_1^T & - \\ & \vdots & \\ - & \mathbf{v}_{960}^T & - \end{bmatrix}$$

Recall we can rewrite this as

$$A = \mathbf{u}_1\sigma_1\mathbf{v}_1^T + \mathbf{u}_2\sigma_2\mathbf{v}_2^T + \ldots + \mathbf{u}_r\sigma_r\mathbf{v}_r^T$$

For most pictures $r = 625$, the maximal rank of A.

Idea. Throw away the term $\mathbf{u}_i\sigma_i\mathbf{v}_i^T$ when σ_i is small. If $k \leq r$, define

$$A_k = \mathbf{u}_1\sigma_1\mathbf{v}_1^T + \mathbf{u}_2\sigma_2\mathbf{v}_2^T + \ldots + \mathbf{u}_k\sigma_k\mathbf{v}_k^T$$

The matrix A_k is very close to the matrix A, if $\sigma_{k+1}, \ldots, \sigma_r$ are small.