Image Compression Project

Note: All code used in the project must be written by you and attached to the project file. Provide code in the pdf only in Matlab.

We will be performing image compression.

1. (**IMPORTING AND PLAYING AROUND**) Import the following "sunflower" image into a matrix, see Figure 1. When you import the jpeg, it will be encoded in 8-bit numbers (minimum 0 and maximum 255). Also, the image will have red, green, and blue channels, so it will be an $853 \times 640 \times 3$ array, lets call it A. We can extract the R channel by taking $A(1:853, 1:640, 1)$, the G channel by $A(1:853, 1:640, 2)$ and the B channel by $A(1:853, 1:640, 1)$.

   Note: If you use double the colors are normalized out of 1. If you use uint8 they are normalized out of 255.

   (a) Change the contrast of the image. That is, make the dark more dark and the light more light. Try using a hyperbolic tangent function. In order to evaluate the hyperbolic tangent function, use an interpolating polynomial of degree 10 with Chebyshev nodes. *Hint: the values at the* 11 *Chebyshev nodes can be computed as you wish.*

   Hint: to visualize in Matlab, create a figure and show the image matrix S by using:

   figure

   image(S);

   (b) Perform, say, a 40° rotation of the image and explain the two-dimensional interpolation you used.

2. (**IMAGE COMPRESSION**) Now, we will perform image compression via a singular value decomposition. A singular value decomposition

Figure 1: Sunflower

of a matrix $A = U\Sigma V^T$ always exists for any $m \times n$ matrix A. The matrix Σ is an $m \times n$ matrix that is all zeros, except the diagonal which consists of the square roots of the eigenvalues of $A^T A$. The columns v_j of V are the unit eigenvectors of $A^T A$. The columns of U can be derived by computing Av_j and then normalizing. Our task, therefore, is to compute the eigenvalues and eigenvectors of $A^T A$. We will not compute a full SVD, but build it up slowly and stop when we have recovered a good approximation to A.

(a) Compute the largest eigenvalue and eigenvector pair using the power method. Why is the power method guaranteed to work for $A^T A$ for this image? What is the rate of convergence of the power method?

(b) Compute the second largest eigenvalue and eigenvector pair using the inverse power method. Do not compute the inverse of a matrix here. Explain why the Gauss-Seidel method will not work here. Solve this instead via another method we learned in class.

(c) Show that the SVD can be written as a generalized spectral decomposition: $A = \sum_{i=1}^{n} \sqrt{\lambda_i} u_i v_i^T$, where λ_i is the ith largest eigenvalue of $A^T A$, v_i is its corresponding unit eigenvector, and $u_i = \frac{Av_i}{\|Av_i\|_2}$. In image compression, we will truncate the sum: $\tilde{A} = \sum_{i=1}^{K} \sqrt{\lambda_i} u_i v_i^T$. Find the value of K for which $\tilde{A}$ begins to resemble a sunflower. Find the value of K for which $\tilde{A}$ is a "good" quality image.

(d) Why is it that truncating the SVD in this way is a good idea for image compression? *Hint: read about the lowest-rank approximation of a matrix.*