Table of Contents

Image Compression using Singular Value Decomposition (SVD)

This script demonstrates how SVD can be used to compress a digital image. The core idea is that any image can be represented as a matrix, and SVD allows us to find the best lower-rank approximation of that matrix. These lower-rank approximations require less data to store, effectively compressing the image.

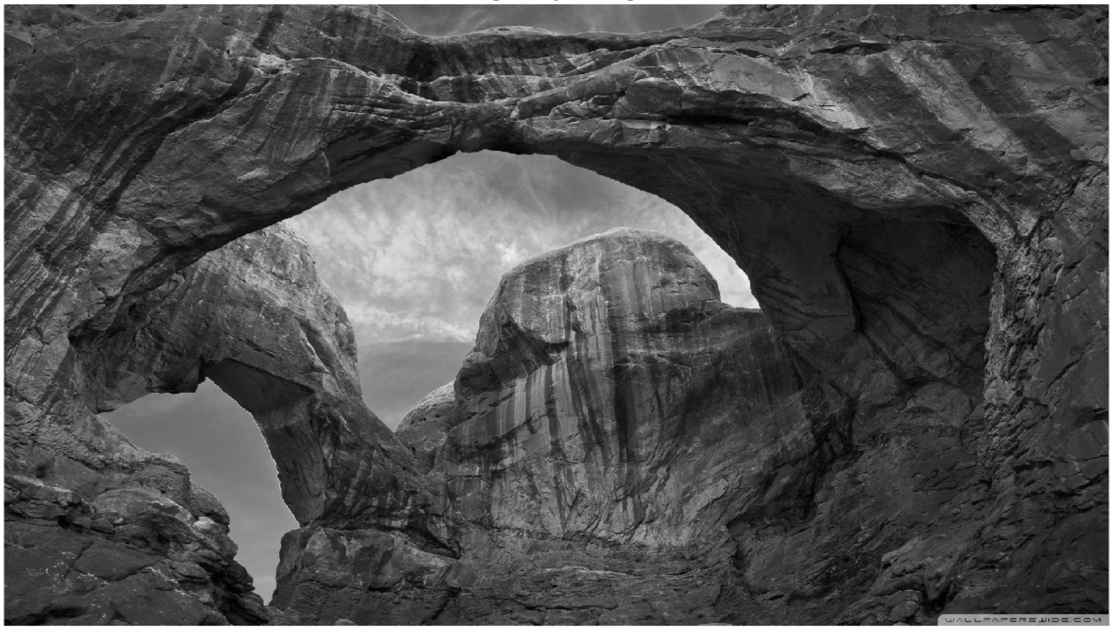
This script performs the following steps: 1. Loads an image and converts it to a grayscale matrix. 2. Computes a low-rank (compressed) version of the image. 3. Analyzes the compression error as a function of the rank.

```
clear; clc; close all;
```

Section 1: Load and Prepare the Image

```
rgb_image = imread('arches.bmp');
gray_image = rgb2gray(rgb_image);
figure('Name', 'Original Image');
imshow(gray_image);
title('Original Grayscale Image');
drawnow
```

Original Grayscale Image



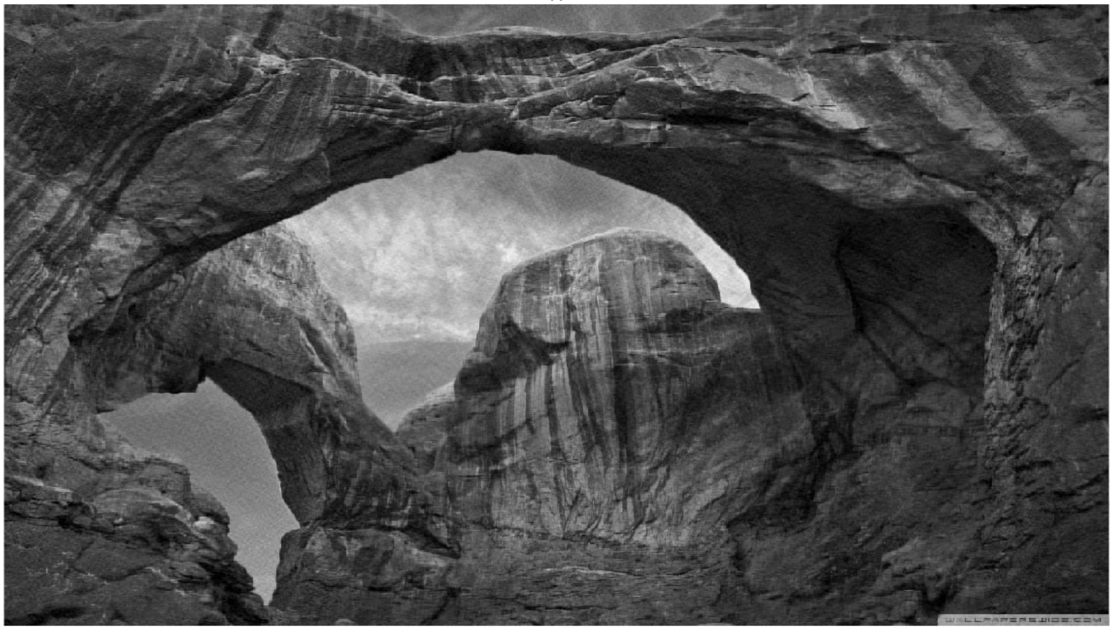
Section 2: Compressing the Image with a Low-Rank Approximation

The best rank-k approximation is found by keeping only the first k singular values and vectors of an SVD of a matrix

```
rank_to_show = 150;
compressed_image = BestApprox(gray_image, rank_to_show);

figure('Name', 'Compressed Image');
imshow(compressed_image);
title(['Rank-' num2str(rank_to_show) ' Approximation']);
drawnow
```

Rank-150 Approximation



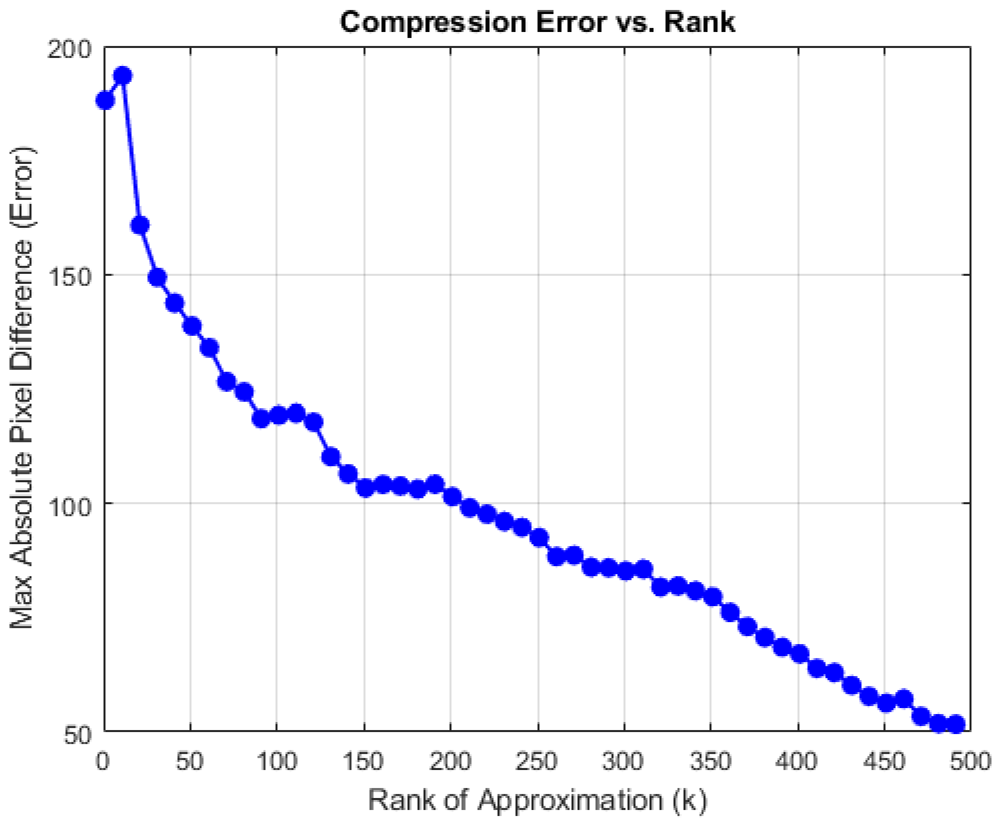
Section 3: Analyzing Compression Error

We can measure the image quality loss computing the largest single pixel difference between the original and the compressed version

```
ranks_to_test = 1:10:500;
errors = dApprox(gray_image, ranks_to_test);
```

Now we plot the error against the rank. As expected, the error decreases as we use a higher rank

```
figure('Name', 'Error Plot');
plot(ranks_to_test, errors, 'b-o', 'LineWidth', 1.5, 'MarkerFaceColor', 'b');
grid on;
title('Compression Error vs. Rank');
xlabel('Rank of Approximation (k)');
ylabel('Max Absolute Pixel Difference (Error)');
drawnow
```



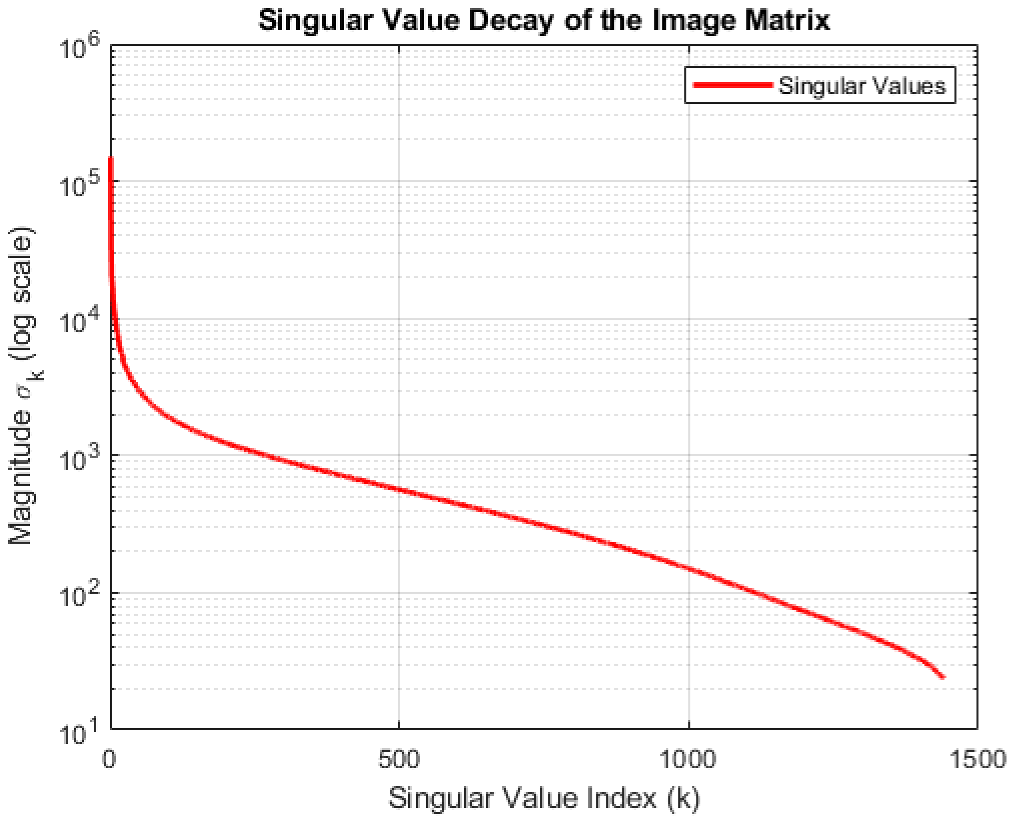
Section 4: Analysis of Singular Values

The magnitude of the singular values tells us how much information is captured by each rank

```
[~, S, ~] = svd(double(gray_image));
singular_values = diag(S);
```

We plot the singular values on a logarithmic scale to better visualize their decay

```
figure('Name', 'Singular Values');
semilogy(singular_values, 'r-', 'LineWidth', 2);
grid on;
title('Singular Value Decay of the Image Matrix');
xlabel('Singular Value Index (k)');
ylabel('Magnitude \sigma_k (log scale)');
legend('Singular Values');
drawnow
```



Published with MATLAB® R2024b