KENNESAW STATE
UNIVERSITY

CS 7367
MACHINE VISION

ASSIGNMENT 2

IMAGE RESOLUTION

INSTRUCTOR

Dr. Sanghoon Lee

Your Name: Adam Kangiser
KSU ID: 000681701

1. ABSTRACT

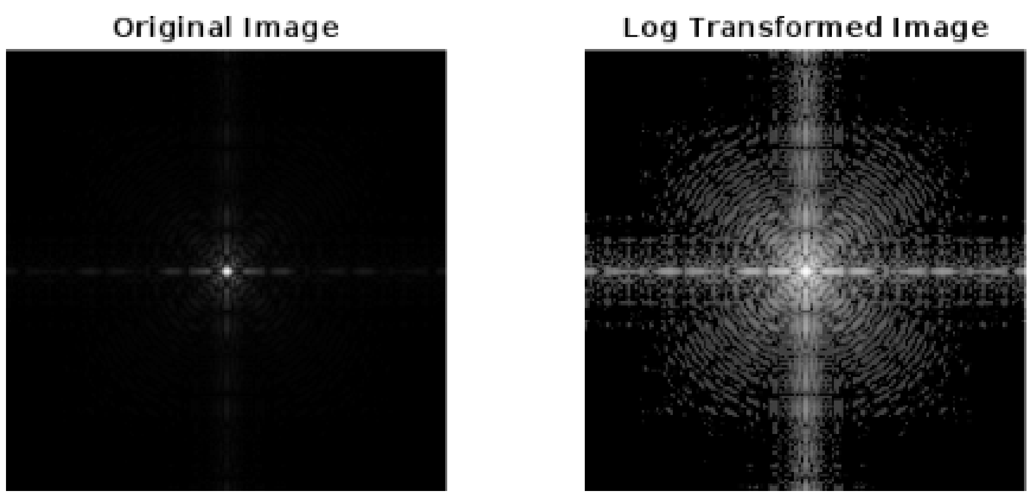
This paper focuses on three techniques used in grayscale image processing; Logarithmic Transformation, Power law Transformation and Histogram Equalization. With the help of MATLAB we apply these methods to enhance the 'fourierspectrum.pgm' image and the 'banker.jpeg' image. Logarithmic Transformation proves effective in enhancing low intensity details by stretching values. Power law Transformation, which relies on an exponent provides an approach for either enhancing or suppressing features. Histogram Equalization redistributes pixel intensities to improve contrast in the image. Our evaluation includes metrics like mean and standard deviation to measure the effectiveness of each technique. The paper concludes by comparing the advantages and limitations of these methods serving as a guide, for both professionals and researchers working in image processing.

2. TEST RESULTS

<u>Logarithmic Transformation:</u>

In the Logarithmic Transformation we applied a scaling c` which was calculated based on the maximum pixel intensity. We used MATLAB to implement this transformation, on each pixel of the image `'fourierspectrum.pgm'` using the function s = c x log(1+r).

When we applied the Logarithmic Transformation it resulted in enhancing low intensity values thereby making darker regions more distinguishable. The output image displayed improved details in areas that were previously difficult to see due to intensity. However higher intensity values remained largely unchanged preserving the structure of the image.



$$c = 255 / \log(256)$$

Figure 1: Original image (left) and log-transformed image (right) with c = 255/log(256).

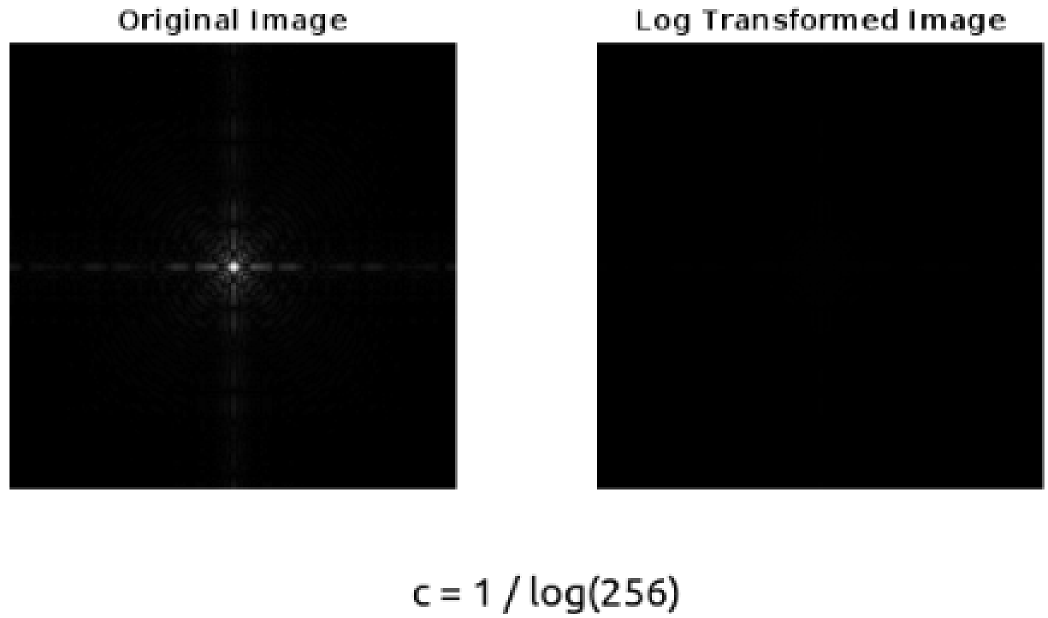


Figure 2: Original image (left) and log-transformed image (right) with c = 1/log(256).

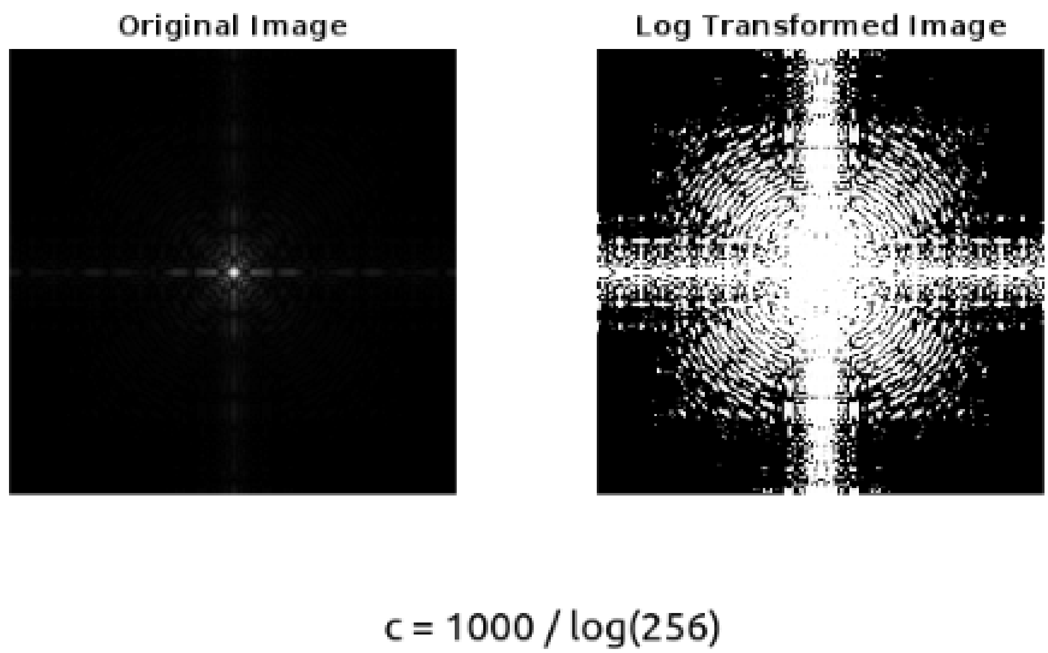


Figure 3: Original image (left) and log-transformed image (right) with c = 1000/log(256).

Power-law Transformation:

For the Power law Transformation we selected gamma values of 0.3 and 3 as the exponents. To execute the transformation we used the formula s = 255 x (r/255)^gamma.

When applied this transformation enhances contrast, for mid high intensity values making certain features more prominent. Depending on the chosen gamma value this transformation allows for flexibility in highlighting or reducing features in an image.

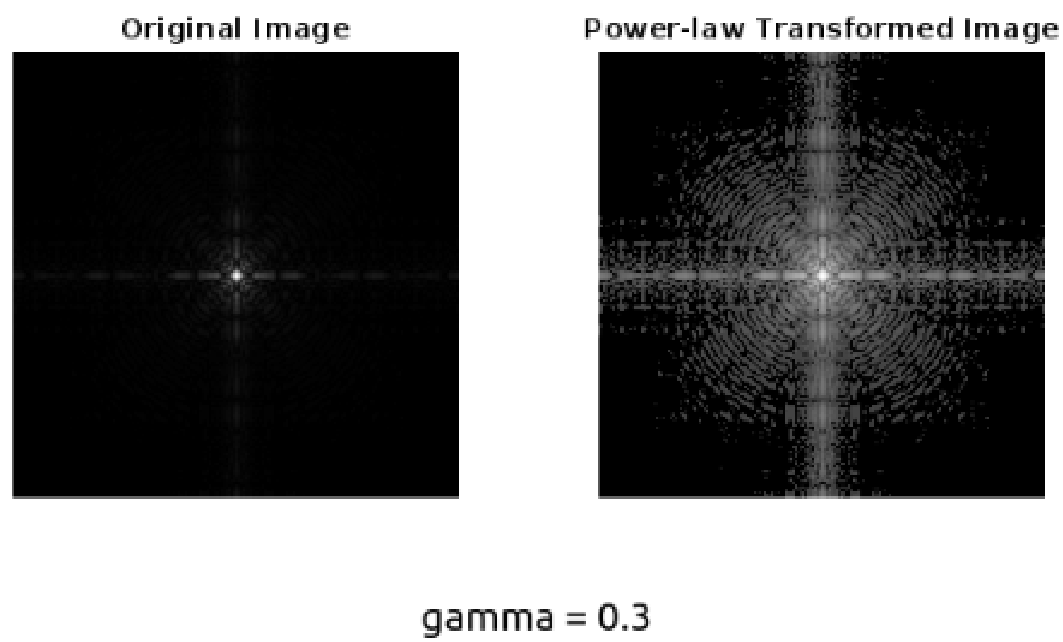


Figure 4: Original image (left) and power-law-transformed image (right) with gamma = 0.3.

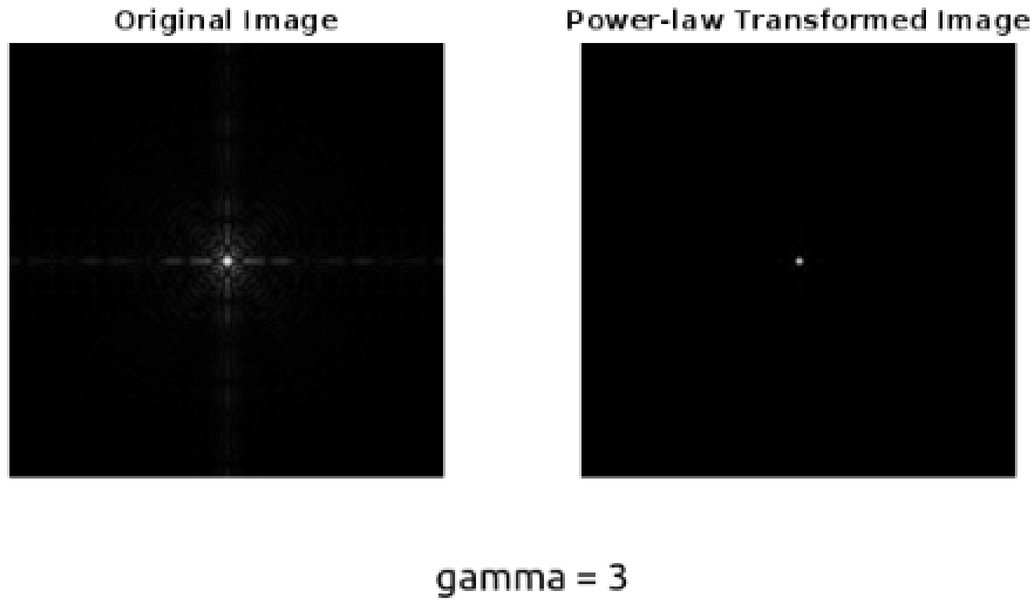


Figure 5: Original image (left) and power-law-transformed image (right) with gamma = 3.

Histogram Equalization:

I used a technique called Histogram Equalization to distribute the intensities in the image. To achieve this I utilized MATLABs built-in function called `histeq`. After applying Histogram Equalization I calculated the standard deviation of the intensities both before and after the process.

The application of Histogram Equalization significantly enhanced the contrast of the image. The analysis of metrics indicated that there was an increase in the deviation, after equalization, which implies a distribution of pixel intensities. On the other hand the mean value remained relatively consistent suggesting that the overall brightness level was maintained throughout.

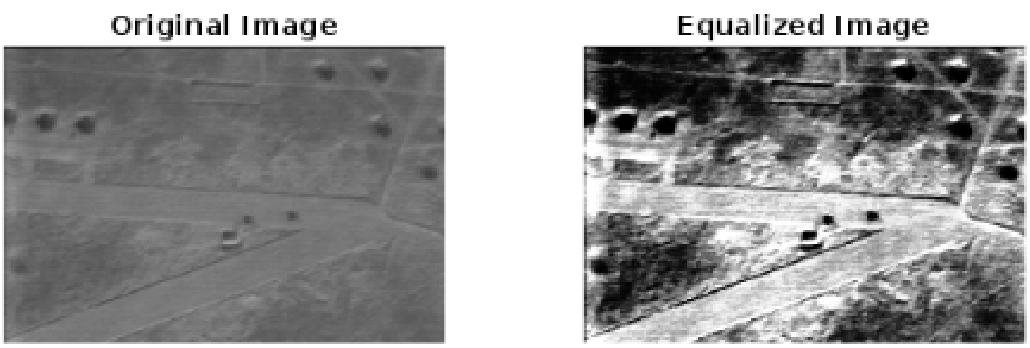


Figure 6: Original image (left) and power-law-transformed image (right) with gamma = 0.3.

3. Discussion

Working on this project has been an experience providing valuable insights into the practicality and limitations of various techniques used for enhancing grayscale images. One important lesson I learned was the significance of choosing transformations depending on the characteristics of an image. For example I found that the Logarithmic Transformation worked wonders in highlighting details in areas while the Power law Transformation offered flexibility for adjusting contrast in different scenarios. On the other hand Histogram Equalization proved to be a technique for improving overall contrast though it may not always be suitable for specialized tasks.

Course this project did come with its share of challenges. Initially I faced some issues with data types and matrix operations in MATLAB. Thankfully those were eventually resolved. It is worth mentioning that one limitation of this study was our reliance on a test image, which may not fully encompass the range of applicability for each method. In work it would be beneficial to test these algorithms across a set of images and even consider integrating machine learning techniques to automate the selection of optimal transformations based on image characteristics. Additionally exploring real world applications like imaging or satellite image processing could help validate the effectiveness of these transformations, within domains.

This project acts as a resource, for individuals looking to comprehend or apply grayscale image transformations. It also opens the door for research and exploration, in this particular field.

4. CODES

4.1 Code for Log Transformation

```
% Name: Adam Kangiser
% KSU Number: 000681701
% Assignment 2
% Read the image
img = imread('fourierspectrum.pgm');
img = double(img); % Convert image to double precision
% Set the scaling constant
c = 1000 / log(256);
% Perform log transformation
log_transformed_img = c * log(1 + img);
% Convert back to uint8
log_transformed_img = uint8(log_transformed_img);
% Display images
subplot(1, 2, 1), imshow(uint8(img)), title('Original Image');
subplot(1, 2, 2), imshow(log_transformed_img), title('Log Transformed Image');
```

••

4.2 Code for Power-law Transformation

```
% Name: Adam Kangiser
% KSU Number: 000681701
% Assignment 2
% Read the image
img = imread('fourierspectrum.pgm');
img = double(img); % Convert image to double precision
% Set gamma
gamma = .3;
% Perform power-law transformation
power_law_transformed_img = 255 * ((img / 255) .^ gamma);
% Convert back to uint8
power_law_transformed_img = uint8(power_law_transformed_img);
% Display images
figure;
subplot(1, 2, 1), imshow(uint8(img)), title('Original Image');
subplot(1, 2, 2), imshow(power_law_transformed_img), title('Power-law Transformed
Image');
```

••

4.3 Function for Histogram Equalization

```
% Name: Adam Kangiser
% KSU Number: 000681701
% Assignment 2
% Read the image
img = imread('banker.jpeg');
img = double(img); % Convert image to double precision
% Perform histogram equalization
```

```
equalized_img = histeq(uint8(img));
% Convert equalized_img to double for statistical calculations
equalized_img_double = double(equalized_img);
% Calculate mean and standard deviation
original_mean = mean(img(:));
original_std = std(img(:));
equalized_mean = mean(equalized_img_double(:));
equalized_std = std(equalized_img_double(:));
% Display original and equalized image
figure;
subplot(1, 2, 1), imshow(uint8(img)), title('Original Image');
subplot(1, 2, 2), imshow(equalized_img), title('Equalized Image');
```