MODULATION TRANSFER FUNCTION

Charmaine S. Tolledo

CONTENTS

OBJECTIVES

01

Measure the MTF of different cameras using the slanted edge technique

02

Understand the relationship between MTF and resolution.

METHODOLOGY

SETUP AND IMAGE CAPTURE

- Place the camera on a stand and position the Resolution Test Chart at a fixed distance.
- Ensure parallel alignment between the camera sensor plane and the slanted edge image.
- Capture multiple images of the test chart, using burst mode if available, and average the images to minimize noise.

DATA PROCESSING

- Open the slanted edge image using MATLAB.
- Obtain intensity profiles of a line crossing the edge (LE).
- Derive the line (d(LE)/dx).
- Perform Fourier transform of this derivative (FFT[d(LE)/dx]).

FREQUENCY ANALYSIS

- Extract the modulus of the FFT of the derivative (abs(FFT[d(LE)/dx])).
- Keep only the positive half of the modulus.
- Calibrate the scale of the Fourier Transform plot using the scale bar from the image.

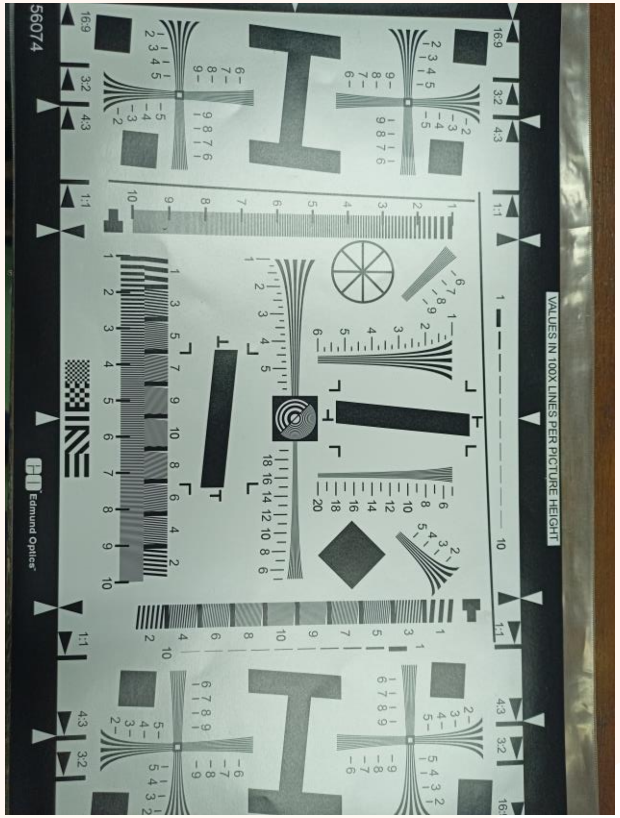
SPATIAL FREQUENCY MEASUREMENT

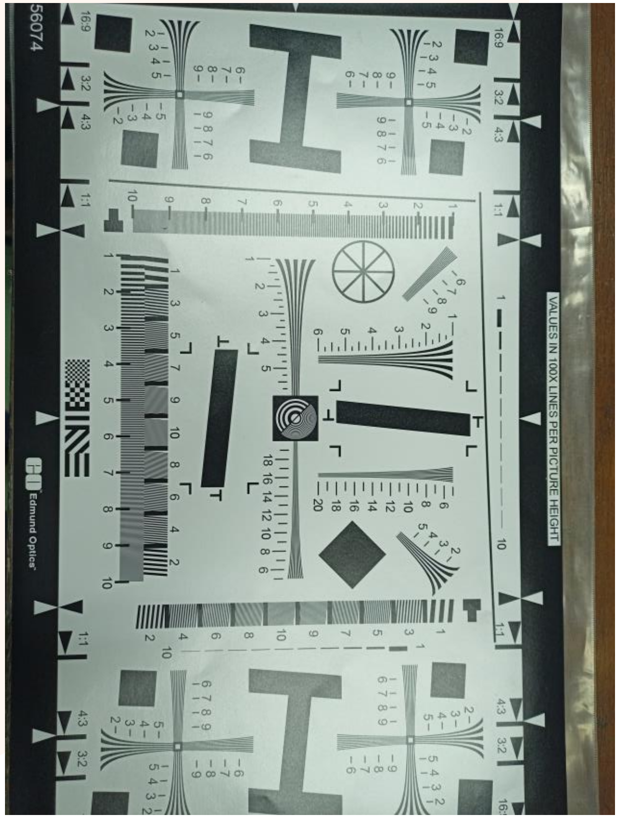
- Express spatial frequency in cycles/mm or line pairs (lp)/mm.
- Repeat steps for several lines across the edge and average the results.

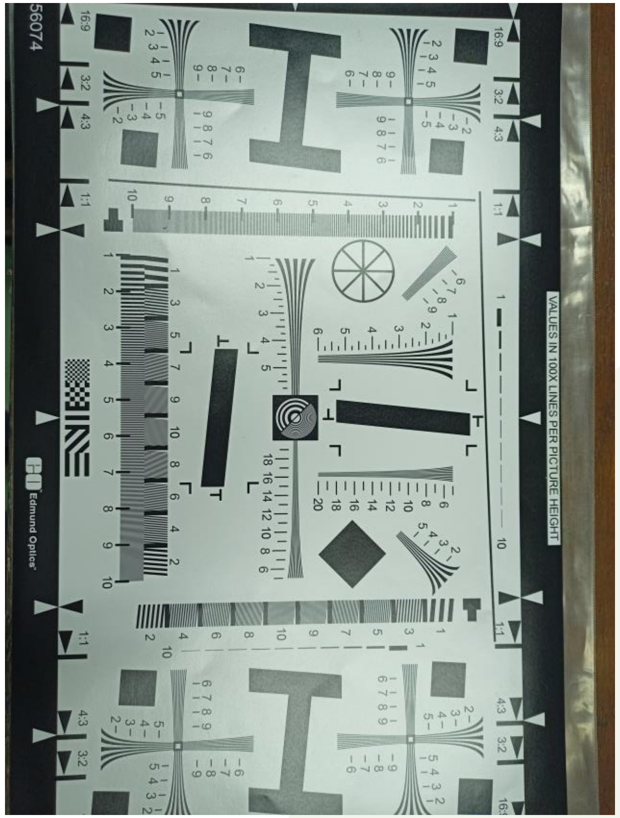
NORMALIZATION AND RESOLUTION CALCULATION

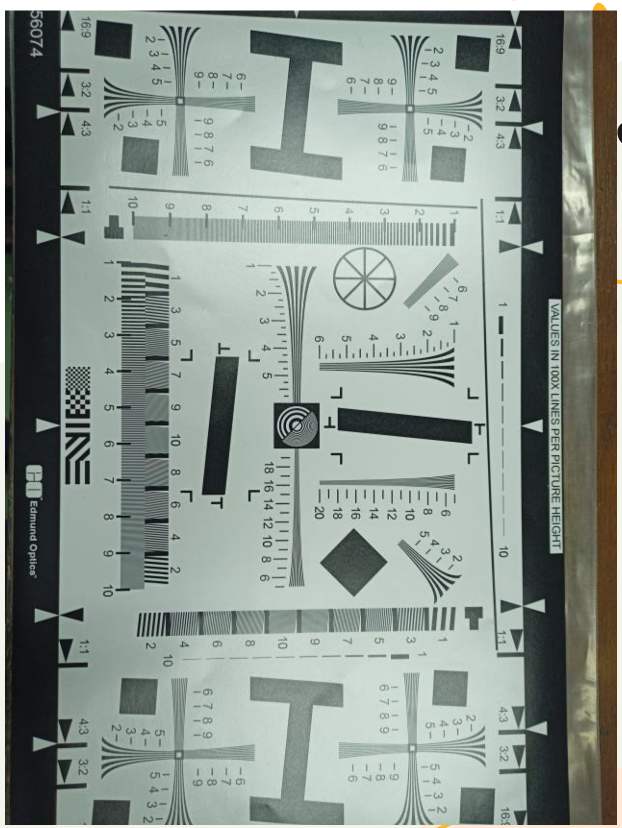
- Normalize the average MTF to 1.0.
- Identify the point where MTF is at 0.5; this represents the camera's resolution in lp/mm.
- Observe line pair images to pinpoint the stage at which the camera loses detail or experiences a phase flip.

SAMPLE IMAGES USED





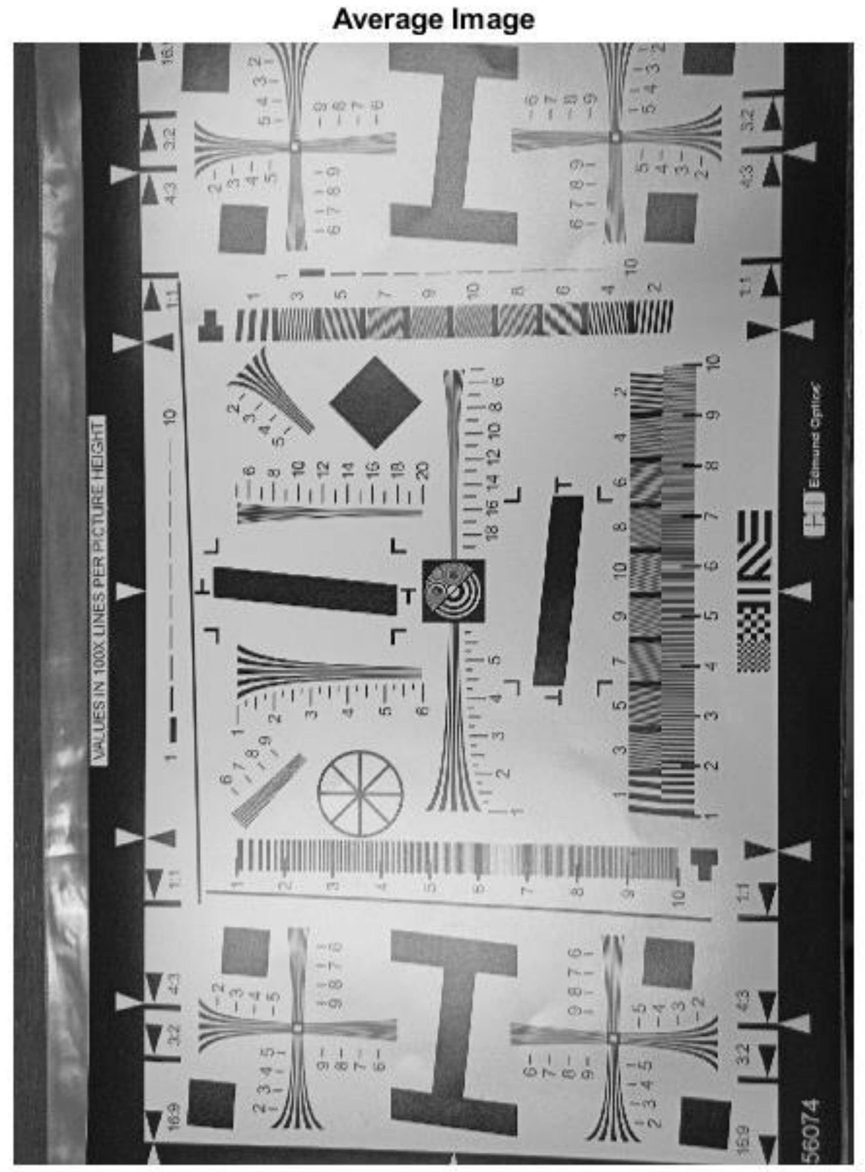


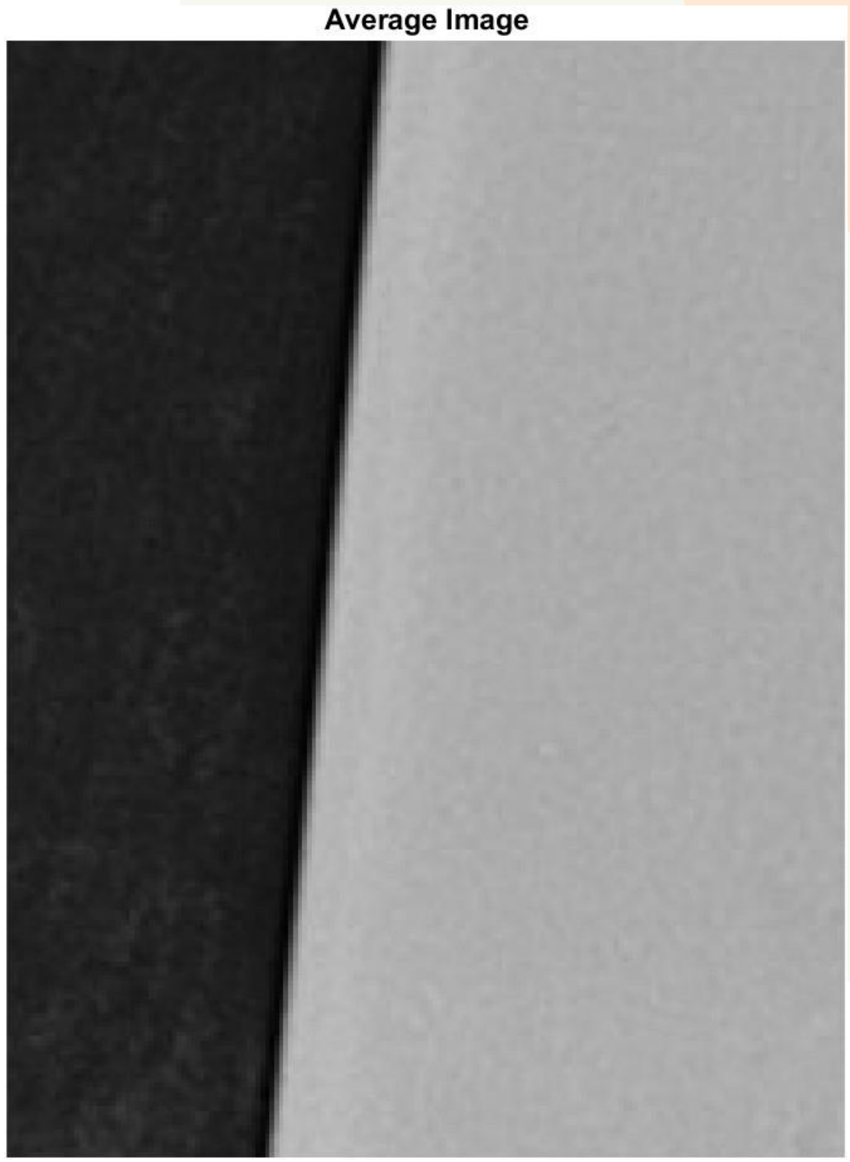


CODE AVERAGING THE 4 IMAGES

```
img_folder = 'C:\Users\User\Documents\MATLAB\167-Activity 4\MTF';
file_extension = '.jpg';
file_list = dir(fullfile(img_folder, ['*', file_extension]));
num_images = numel(file_list);

% Initializing a variable to store the total image
total_image = zeros(size(imread(fullfile(img_folder, file_list(1).name))));

% Looping through each image file
for i = 1:num_images
    image_name = file_list(i).name;
    image = imread(fullfile(img_folder, image_name));
    if size(image, 3) == 3
        image = rgb2gray(image);
    end
    total_image = total_image + double(image);
end

% Calculating the average image
average_image = uint8(total_image / num_images);
J = imrotate(average_image, 90)
```

RESULTING AVERAGE IMAGE

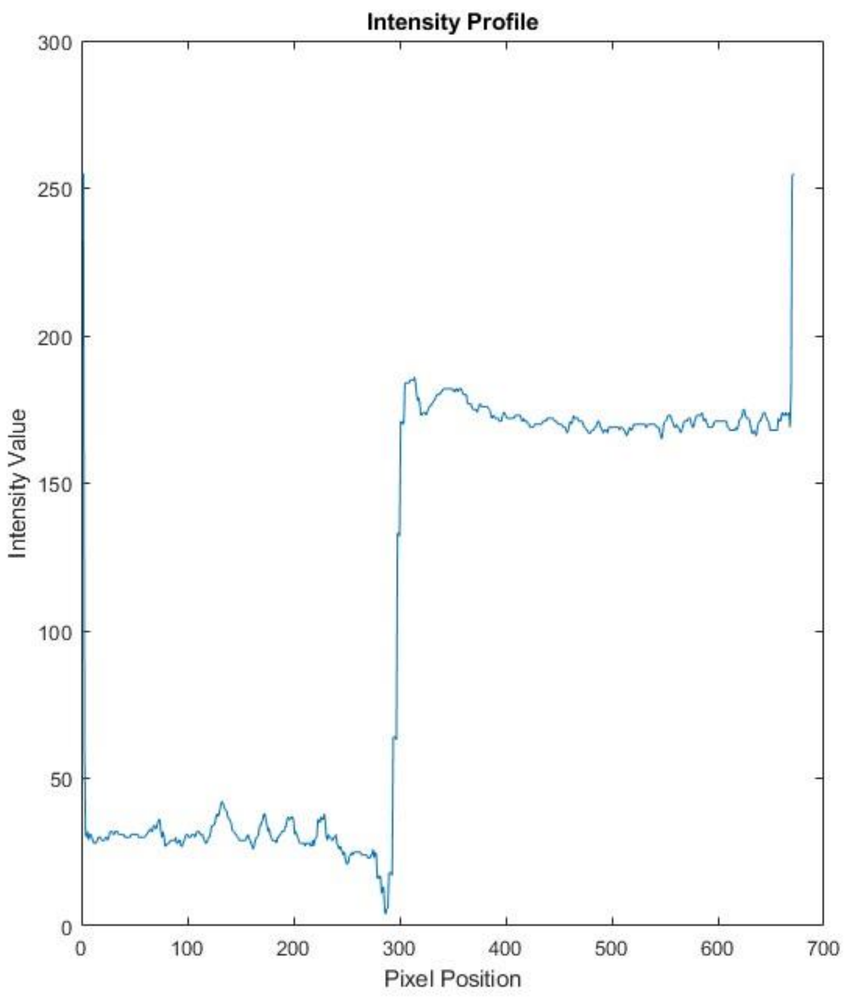




For me to be able to get the slanted image, I zoomed in the first image and saved as picture, as seen in the second photo.

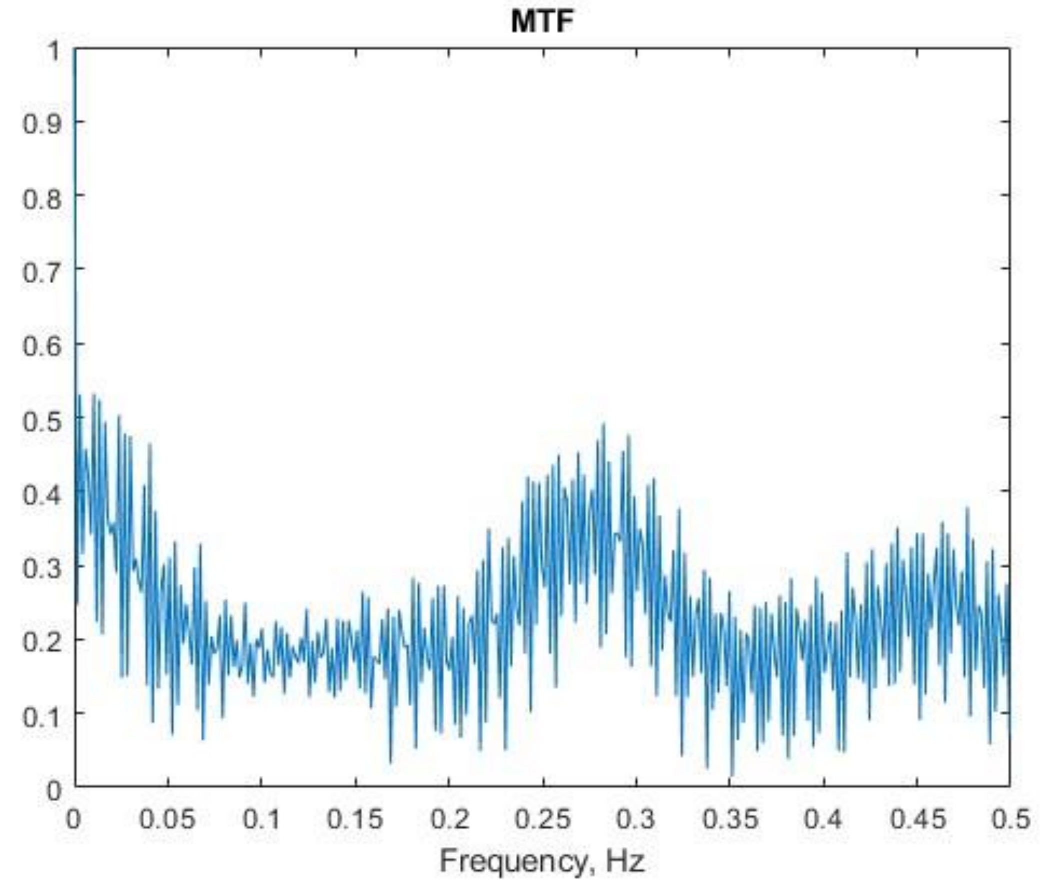
OBTAINED INTENSITY PROFILE

```
slantedImage = imread(['C:\Users\User\Documents\MATLAB\167-Activity 4\slant.jpg']);
slantedImage = rgb2gray(slantedImage);
rowIdx = 100;
% Extracting the intensity profile
intensityProfile = slantedImage(rowIdx, :);
plot(intensityProfile);
title('Intensity Profile');
xlabel('Pixel Position');
ylabel('Intensity Value');
```



MEASURING THE MTF

```
% Taking the derivative of the intensity profile
derivativeProfile = diff(intensityProfile);

% Taking the FFT of the derivative profile
FFT_derivative = fft(derivativeProfile);

% Calculating the MTF
MTF = abs(FFT_derivative) / max(abs(FFT_derivative));

% Setting up frequency axis
dx = 1;
N = length(intensityProfile) - 1;
Fmax = 1 / (2 * dx);
df = 1 / ((N - 1) * dx);
frequencyAxis = 0:df:Fmax;

K = length(frequencyAxis);

% Displaying the MTF
plot(frequencyAxis(1:K), MTF(1:K));
xlabel('Frequency, Hz');
title('MTF');
```



REFLECTION

I was not able to fully complete the objectives of this activity, but I did try my best to code and complete the activity. I faced some challenges, but I learned a lot from the experience. I think I deserve a grade of **90** for making an effort to finish this activity and I promise to make a better activity next time. Special thanks to Mark Danganan for helping me with the codes.

THANK YOU!