

**FACIAL RECOGNITION:
USING EXPERIMENTAL COLOR
IMAGE PROCESSING
TECHNIQUES**

by

ISAAC TREVINO, B.S.E.E.

THESIS
Presented to the Graduate Faculty of
The University of Texas at San Antonio
in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

COMMITTEE MEMBERS:
Artyom Grigoryan, Ph.D., Chair
Yongcan Cao, Ph.D.,
Ram Krishnan, Ph.D.

THE UNIVERSITY OF TEXAS AT SAN ANTONIO
College of Engineering
Department of Electrical and Computer Engineering
May 2020

Copyright 2020 Isaac Trevino
All Rights Reserved

DEDICATION

This thesis is dedicated to my family whom have been there for me emotionally and financially. Thank you for your moral support and the love you have provided me during my time in college. No matter the cost you have been there when I needed you most.

ACKNOWLEDGEMENTS

I'd like to thank Dr. Artyom Grigoryan for academic assistance and for supervision of this thesis. I'd also like to thank Dr. Lyle R. Hood for mentorship and council in my last academic year. Acknowledging the University of Texas San Antonio College of Engineering for providing the resources for me to progress in my degree plan.

May 2020

**FACIAL RECOGNITION:
USING EXPERIMENTAL COLOR
IMAGE PROCESSING
TECHNIQUES**

Isaac Trevino, M.S.
The University of Texas at San Antonio, 2020

Supervising Professor: Artyom Grigoryan, Ph.D.

Facial Recognition is a developing topic as systems are improved almost daily, whether such systems benefit the public or if security outweighs popular opinion. Regardless, most facial recognition systems rely on sophisticated neural networks which use training data to produce accurate models for application [1]. However, with drawbacks for certain models such as relying on grayscale images rather than color images introduces disadvantages. This Thesis will be a general overview of facial recognition systems using color image processing techniques to improve current pattern recognition methods and understanding the potential of such systems in the near-future. The use of color data could be used as features to be classified in pattern recognition models by introducing image enhancement and color transformation methods. For most security surveillance cctv-cameras are still the main source of population identification converting most images to gray scale in the process. With state-of-the-art biometrics such as thermal imaging, three-dimensional facial modeling will be supplanted without color image processing. This happens to be a growing concern where accuracy does benefit popular opinion, and where all possibilities should be considered.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iv
ABSTRACT	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
CHAPTER 1: FACIAL RECOGNITION SYSTEM.....	1
1.1 Applications	2
1.2 NIST FERET Database.....	2
1.3 Cropping Image.....	3
CHAPTER 2: IMAGE ENHANCEMENT.....	4
2.1 Translational Block Filter in Frequency Domain.....	7
2.2 Gradient-Based Histogram Equalization.....	11
2.3 Results	17
CHAPTER 3: COLOR TRANSFORMATIONS.....	19
3.1 Quaternion Color Transform.....	19
3.2 Quantum Image.....	23
CHAPTER 4: FEATURE EXTRACTION.....	31
4.1 Local Binary Pattern	31
4.2 Color Concentration	35

CHAPTER 5: PATTERN RECOGNITION	35
5.1 Correlation Coefficient	35
5.2 Minimum Distance of Kernel Distribution	35
CHAPTER 6: CONCLUSION	38
6.1 1 Results.....	38
6.2 Improvements.....	40
CHAPTER 7: OTHER WORKS	40
7.1 Simulated Three-Dimensional Facial Construction	40
7.2 Range Imaging for Facial Analysis.....	41
CHAPTER 8: FURTHER RESEARCH	42
8.1 Normal Arbitrary Superposition State Three Component	42
8.2 Potential of FR	43
APPENDICIES	45
Appendix A	45
Appendix B	45
Appendix C	48
Appendix D	69
References.....	80
VITA	86

LIST OF TABLES

Table 2.1: Overall Average of Enhancement Measure of all 32 FERET Database Images	18
Table 6.1: Correlation Coefficient & Kernal Distribution Results.....	39
Table 6.2: Enhanced Images using Correlation and Kernal Distribution	40
Table A.1: Original Image Enhancement Measurements	69
Table A.2: Histeq Image Enhancement Measurement	70
Table A.3: Spatial Filter Enhancement Measurement	71
Table A.4: GBHE (1) Enhancement Measurement	72
Table A.5: GBHE (2) Enhancement Measurement	73
Table A.6: GBHE (3) Enhancement Measurement	74
Table A.7: GBHE (4) Enhancement Measurement	75
Table A.8: 2x2 Quaternion Image GrRGB Enhancement Measurement	76
Table A.9: 2x2 Quaternion RGBY Enhancement Measure.....	77
Table A.10: 3x3 Quaternion Image RGB Enhancement Measurement.....	78
Table A.11: 3x3 Quaternion Image CMY Enhancement Measurement.....	79

LIST OF FIGURES

Figure 1.1: Facial Recognition Flow	2
Figure 1.2: FERET Database [301-B, 307-B, 310-A, 302-A]	3
Figure 1.3: Example: (left) Mask, (middle) Original Image, (right) Mask-Cropped Image	3
Figure 2.1: Diagram of Kernel Operation in Fourier Domain of image.....	8
Figure 2.2: Montage images of three Shifted Fourier Transform faces, (top left) RGB Fourier Transform of Face Image, Using only the red color space of the transformed image (middle left) Binarized images with threshold 0.1, 0.2, 0.3, (bottom left) Binarized images with threshold 0.4, 0.5, 0.6, (top right) translational filter with $\beta=0.6$ and $\alpha=0.9,0.8,0.7$, (middle right) translational filter with $\beta=0.6$ and $\alpha=0.6,0.5,0.4$, (bottom right) translational filter with $\beta=0.6$ and $\alpha=0.3,0.2,0.1$	9
Figure 2.3: After using the translation convolutional filter with parameters $\alpha=0.5$, $\beta=0.8$ (top left) Original Histogram image, (top right) shifted Fourier transformed face, (bottom left) inversed transformed face after translation filter, (bottom right) Fourier transformed face.	10
Figure 2.4: Color images after processing each color space (RGB), (left) original image, (right) translational filtered image.	11
Figure 2.5: MATLAB's Histogram Equalization	12
Figure 2.6: Enhanced Faces (FERET 309-B, 306-B, 302-A) (Left) Original, (Middle) MATLAB's Histogram Equalization, (Right) Gradient-Based Histogram Equalization.....	12
Figure 2.7: Gradient-Based Histogram Equalization.....	12

Figure 2.8: GBHE of Face (FERET 309-B), A) Original, B) Histogram Equalization, C) GBHE using gradient (1), D) GBHE using gradient (2), E) GBHE using gradient (3), F) GBHE using gradient (4).....	14
Figure 2.9: GBHE of Face (FERET 302-A), A) Original, B) Histogram Equalization, C) GBHE using gradient (1), D) GBHE using gradient (2), E) GBHE using gradient (3), F) GBHE using gradient (4).....	15
Figure 2.10: GBHE of Face (FERET 306-B), A) Original, B) Histogram Equalization, C) GBHE using gradient (1), D) GBHE using gradient (2), E) GBHE using gradient (3), F) GBHE using gradient (4).....	16
Figure 2.11: Enhanced Images using Quaternion Enhancement Filter and Gradient Based Histogram Equalization.	17
Figure 2.12: Using Michelson Visibility Images for the above images in Figure 2.11	17
Figure 2.13: Parameterized Agaian-Michelson Entropy Enhancement Measure	18
Figure 2.14: Grigoryan-Agaian Enhancement Measurement	18
Figure 3.1: (left) 3x3 quaternion map, (right) 2x2 quaternion map.....	19
Figure 3.2: 2x2 mapped Quaternion image using colors gray, red, green, blue. (left) Quaternion image (FERET 314-B), (right) RGB plot of colors used for the image.	20
Figure 3.3: 3x3 mapped Quaternion image using colors red, green, blue. (left) Quaternion image (FERET 314-B), (right) RGB plot of colors used for the image.	21
Figure 3.4: 2x2 mapped Quaternion image using colors red, green, blue, yellow. (left) Quaternion image (FERET 314-B), (right) RGB plot of colors used for the image.	21
Figure 3.5: 2x2 mapped Quaternion image using colors cyan, magenta, yellow. (left) Quaternion image (FERET 314-B), (right) RGB plot of colors used for the image.	22

Figure 3.6: Fourier Transform of GrRGB Quaternion Images (left) Red Color Space, (middle) Green Color Space, (right) Blue Color Space.....	22
Figure 3.7: Fourier Transform of RGB Quaternion Images (left) Red Color Space, (middle) Green Color Space, (right) Blue Color Space.....	23
Figure 3.8: Bloch Sphere	25
Figure 3.9: 2x2 image, n = 1	26
Figure 3.10: Quantum Circuit divided by pixel, (top) i = 0, (bottom) i = 1	27
Figure 3.11: Quantum Circuit divided by pixel, (top) i = 2, (bottom) i = 3.	28
Figure 3.12: Histogram of count executions with probability measurements.	29
Figure 3.13: 4x4 Images used for NASS representation.....	30
Figure 3.14: NASS Histogram of 4x4 image.....	31
Figure 4.1: Binary Rotation of LBP process for Neighboring pixels f_{xy}	32
Figure 4.2: Local Binary Pattern and Uniform LBP of images 300-B,305-B,310-B (left) Original image, (second to left) Local Binary Pattern, (middle) Histogram of Local Binary Pattern, (second to right) Uniform Local Binary Pattern, (right) Histogram of Uniform Local Binary Pattern.	34
Figure 5.1: Kernel distribution of 3 images, 305-A/B, 306-A.....	37
Figure 5.2: Kernel Distribution Adjustment methods, (Top A) Two image kernel distribution, (Middle B) Center Mean Target Adjustment, (Bottom C) Maximum Peak Alignment of two images.	38
Figure 7.1: 3-D Morphable Model [7]	41
Figure 7.2: Example of facial measurement using range imaging.....	42

Figure 8.1: Original Image, (middle) filtered image $\alpha=0.9, \beta=0.3$, (right) filtered image $\alpha=0.9, \beta=0.1$ 44

Figure 8.2: Histogram and FFT of Images (top) Original Image, (middle) Filtered Image $\alpha=0.9, \beta=0.3$, (right) Filtered Image $\alpha=0.9, \beta=0.1$ 44

CHAPTER 1: FACIAL RECOGNITION SYSTEM

Facial Recognition (FR) is an important area in the computer vision discipline, with improving strides in the field we are to better understand key information of facial patterns to improve current models and accuracy. There are currently multiple variations of FR systems available where most of the research is produced by experts in the field as well as private and public sectors. Most FR systems used today by government and security agencies are known to be the most up to date systems used today, however with a growing concern of noticeable error from reputable systems one may conclude that current FR systems are not ready for deployment.

Other than security applications computer vision is being used in the consumer market. Walking into a shopping mall or a grocery store to detect customers or items could help companies determine consumer trends and speed up the checkout process. This could be used to analyze the company's traffic to sales ratio determining profit, moreover creating a smart profile with the shopping center would make identification easier and even tailor your experiences to your preferences when you arrive. The applications of computer vision are limitless where accuracy is deemed extremely important, the necessity to increase feature classification for machine learning algorithms leads us to the lack of color image processing. Most processing is done using average intensity level from grayscale images.

The focus of this thesis will be the pre-, mid-, and post- processing of a base facial recognition model. This thesis will cover image enhancement, color transformation, feature, and pattern recognition. Using the FERET Database we are able to construct a basic model for facial recognition which is discussed in the next section.

1.1 Applications

This Facial recognition system will focus on two-dimensional images which are cropped to an ellipsoidal shape. This allows the researcher to focus on the image processing algorithm rather than the computer vision aspect of object detection. Figure 1 below shows the flow processes of the FR system: Cropping of the subject's image, Enhancement of the Face, Color Transformation, Feature Extraction, and the Detection (Pattern Recognition) algorithm.

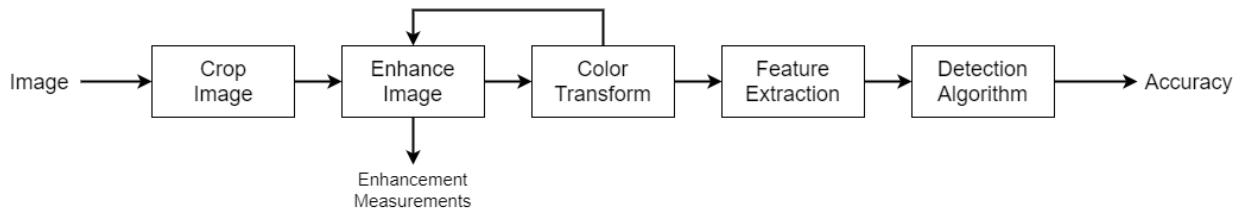


Figure 1.1: Facial Recognition Flow

For a more instructive flow of this paper we will not include the cropping of the image as a chapter instead discuss this process as a section within this chapter.

1.2 NIST FERET Database

The subjects used for this research thesis were acquired by the National Institute of Standards and Technology (NIST) Face Recognition Technology (FERET) database [2], [3]. For this paper we will use 32 images (subjects 300-316) of the database to analyze and test pattern recognition. This database was selected for the quality of the images as well as the color availability. The database shown in figure 1.2 consist of two images per subject with different facial expression, this allows us to find a unique comparison among the remaining 31 images.

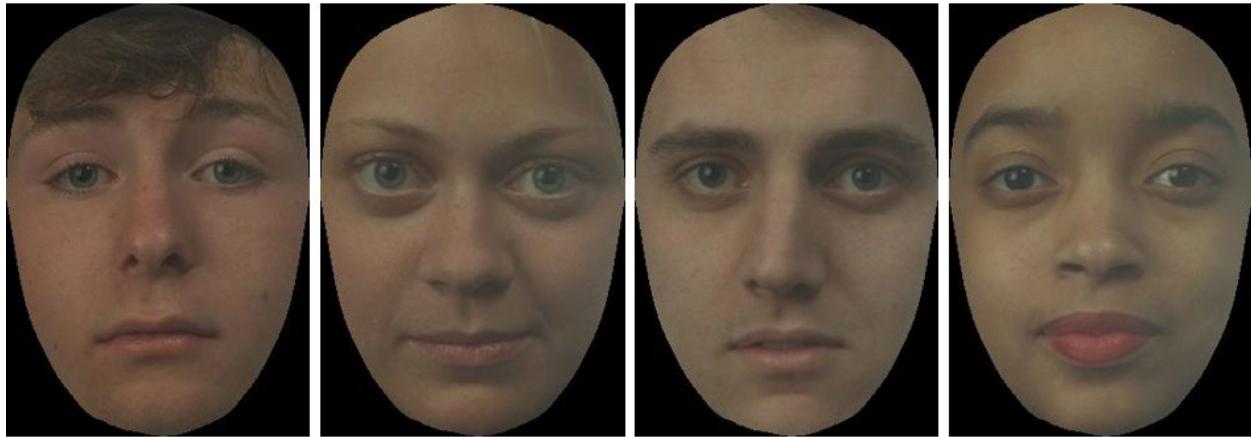


Figure 1.2: FERET Database [301-B, 307-B, 310-A, 302-A]

1.3 Cropping Image

The Cropping of the image is done using MATLAB [4] computer vision tool box as well as linear algebra arithmetic. MATLAB's Face detection function was used which utilizes Viola-Jones algorithms to detect our subjects faces, Using Haar and Local Binary Patterns (LBP) models to encode facial features. The leftmost image in figure 1.3 is the mask we used for the cropping procedure, position was determined by the MATLAB function mentioned above [5]. The FERET faces where converted to JPG after the crop-masking process for image processing.



Figure 1.3: Example: (left) Mask, (middle) Original Image, (right) Mask-Cropped Image

Throughout the remainder of this thesis, we will only mention the mask-cropped image as *the image* or *the face* for the remainder of this thesis.

CHAPTER 2: IMAGE ENHANCEMENT

Image enhancement techniques improve the image's visual quality by applying different image gradients as well as using edge detection algorithms. Focusing on these gradient images allows for a contrast balance leaving room for pattern recognition.

Although color tone in surveillance systems usually are inconsistent and may cause miscalculations due to lighting leaving the recognition system noncredible for practical applications. In order to prevent this from occurring a marker must be used. For example, having a marker or object placed around the surveillance area may be used as a standard of color identification, thus allowing the system to adjust and appropriately identify the current color tone of the time of surveillance.

By using color as a feature in classification we must consider modifying the image so the concentration of color for the image is distinguishable. For color measuring of a face we may use the histogram of the image or sectional areas depending on the feature classification method. More features may still need to be considered to work cooperatively instead of a single feature classification, for example using the color feature alongside edge detection may improve overall recognition results.

Each of the images enhancement method is measured using an estimation measurement; Estimation Enhancement Measurement (Entropy/Nonentropic used to measure the change in Intensity on the Logarithm scale), Agaian-Michelson (Entropy/Nonentropic used to measure the Michelson ratio over a natural logarithmic characterization of blocks [k:k1,l:k2]) Enhancement Measure (Parametrized/Unparametrized), Panetta-Agaian Enhancement Measure (utilizing the

center pixel of each block to produce a measurement enhancement estimation), and Mean-Square-Root Measure (Similar to the Panetta-Agaian estimation, the center pixel is used with the comparison to the mean moreover the square root is taken by the entire summation) of Enhancement [6, 7].

Grigoryan-Agaian Enhancement Measurement

$$GAEM = \frac{1}{k_1 k_2} \sum_{k=1}^{k_1} \sum_{l=1}^{k_2} \arctan \left[1 + \frac{|center_{k,l}(f) - \alpha mean(f)|}{f_0 + center_{k,l}(f)} \right]$$

Equation 2.1: Grigoryan-Agaian Enhancement Measurement

Estimation Enhancement Measurement

$$EME = \frac{1}{k_1 k_2} \sum_{k=1}^{k_1} \sum_{l=1}^{k_2} 20 \ln \left[\frac{\max_{k,l}(f)}{\min_{k,l}(f)} \right]$$

Equation 2.2: Estimated Enhancement Measurement

Estimation Enhancement Entropy Measurement

$$EMEE = \frac{1}{k_1 k_2} \sum_{k=1}^{k_1} \sum_{l=1}^{k_2} \left[\frac{\max_{k,l}(f)}{\min_{k,l}(f)} \right] \ln \left[\frac{\max_{k,l}(f)}{\min_{k,l}(f)} \right]$$

Equation 2.3: Estimation Enhancement Entropy Measurement

Agaian-Michelson Enhancement Measure

$$AME = \frac{-1}{k_1 k_2} \sum_{k=1}^{k_1} \sum_{l=1}^{k_2} 20 \ln \left[\frac{\max_{k,l}(f) - \min_{k,l}(f)}{\max_{k,l}(f) + \min_{k,l}(f)} \right]$$

Equation 2.4: Agaian-Michelson Enhancement Measure

Agaian-Michelson Entropy Enhancement Measure

$$AMEE = \frac{-1}{k_1 k_2} \sum_{k=1}^{k_1} \sum_{l=1}^{k_2} \left[\frac{\max(f) - \min(f)}{\max(f) + \min(f)} \right] \ln \left[\frac{\max(f) - \min(f)}{\max(f) + \min(f)} \right]$$

Equation 2.5: Agaian-Michelson Entropy Enhancement Measure

Parametrized Agaian-Michelson Entropy Enhancement Measure

$$PAMEE = \frac{-1}{k_1 k_2} \sum_{k=1}^{k_1} \sum_{l=1}^{k_2} \alpha \left[\frac{\max(f) - \min(f)}{\max(f) + \min(f)} \right]^\alpha \ln \left[\frac{\max(f) - \min(f)}{\max(f) + \min(f)} \right]$$

Equation 2.6: Parametrized Agaian-Michelson Entropy Enhancement Measure

The second derivative, or Panetta-Agaian Enhancement Measure

$$PAME = \frac{-1}{k_1 k_2} \sum_{k=1}^{k_1} \sum_{l=1}^{k_2} 20 \ln \left[\frac{|\max(f) - 2\text{center}_{k,l}(f) + \min(f)|}{\max(f) + 2\text{center}_{k,l}(f) + \min(f)} \right]$$

Equation 2.7: Panetta-Agaian Enhancement Measure

Mean-Square-Root Measure of Enhancement Measure

$$MSRME = \frac{1}{k_1 k_2} \sqrt{\sum_{k=1}^{k_1} \sum_{l=1}^{k_2} \frac{|\ln|\text{center}_{k,l}(f) - \mu_{k,l}(f)||}{\ln|\text{center}_{k,l}(f) + \mu_{k,l}(f)|}}$$

Equation 2.8: Mean-Square-Root Measure of Enhancement Measure

Using the mentioned equation above we may estimate the quality of the enhancement methods described in the following sections. The enhancement measurements may be described as the intensity ratios blocks sizes [1:k1,1:k2] which are characterized by a natural logarithm. The measurements are displayed in the Appendix of this paper.

2.1 Translational Block Filter in Frequency Domain

This translational convolution filter applies a kernel operator to a translational section in the Fourier Domain. This kernel operator exponentially reduces the intensity of sparse pixels that fall below a threshold, α . [8] [9]

$$\tilde{\mathcal{F}}(u, v) = \begin{cases} \sum_{\substack{1 \leq i+1 \leq k_1 \\ 1 \leq j+1 \leq k_2}} \sum_{\substack{1 \leq u \leq L_1 \\ 1 \leq v \leq L_2}} \mathcal{F}(u + (L_1 i), v + (L_2 j)) \zeta(u, v), & \text{if } \mu < \alpha \\ \mathcal{F}(u + (L_1 i), v + (L_2 j)), & \text{if } \mu \geq \alpha \end{cases}$$

Equation 2.9: Translation Block Filter

Where kernel $\zeta(u, v)$ represents

$$\zeta(u, v) = \frac{e^b - 1}{e^1 - 1}$$

Equation 2.10: Translation Block Filter Kernel

Where $k_{1,2}$ and $L_{1,2}$ represents the Fourier transformed image divided by $k_1 \times k_2$ blocks of size $L_1 \times L_2$ as shown in figure 2.4. This function iterates a kernal of variable α proportional to the mean of the transformed image to the sectional area.

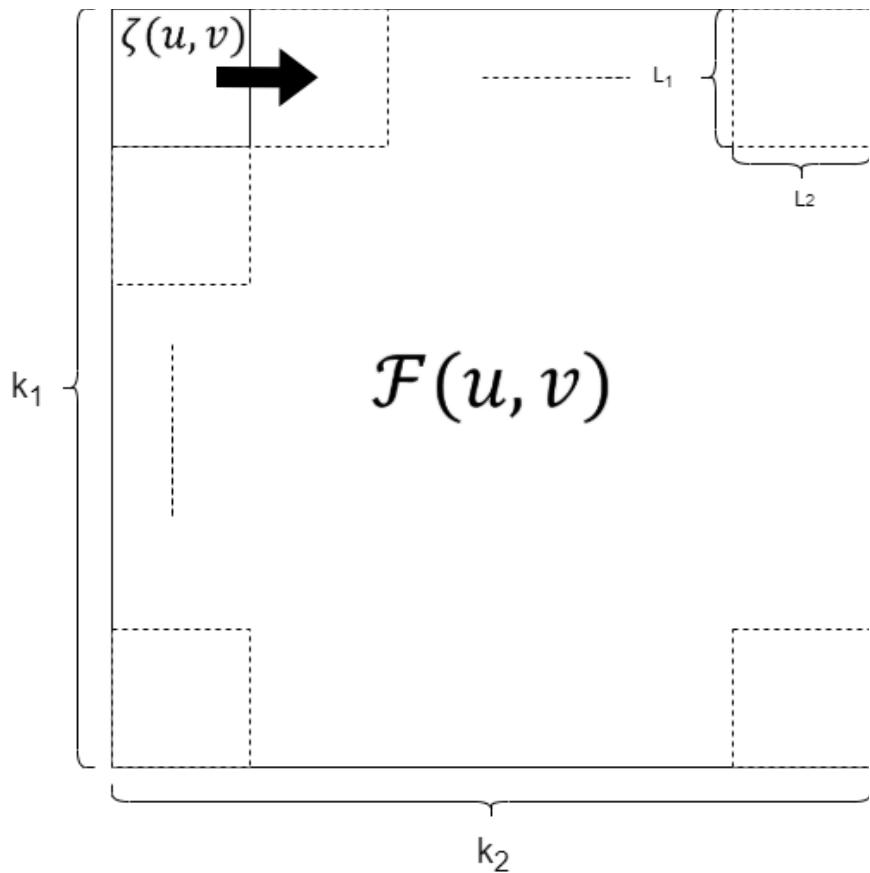


Figure 2.1: Diagram of Kernel Operation in Fourier Domain of image.

This operation is similar to the regular convolution or coefficient operation except instead of iterating over each pixel the kernel is translated over to the next sectional block on the first row and so on. This allows for a faster runtime in algorithm computation as well as introduces a nonconventional method for filtering.

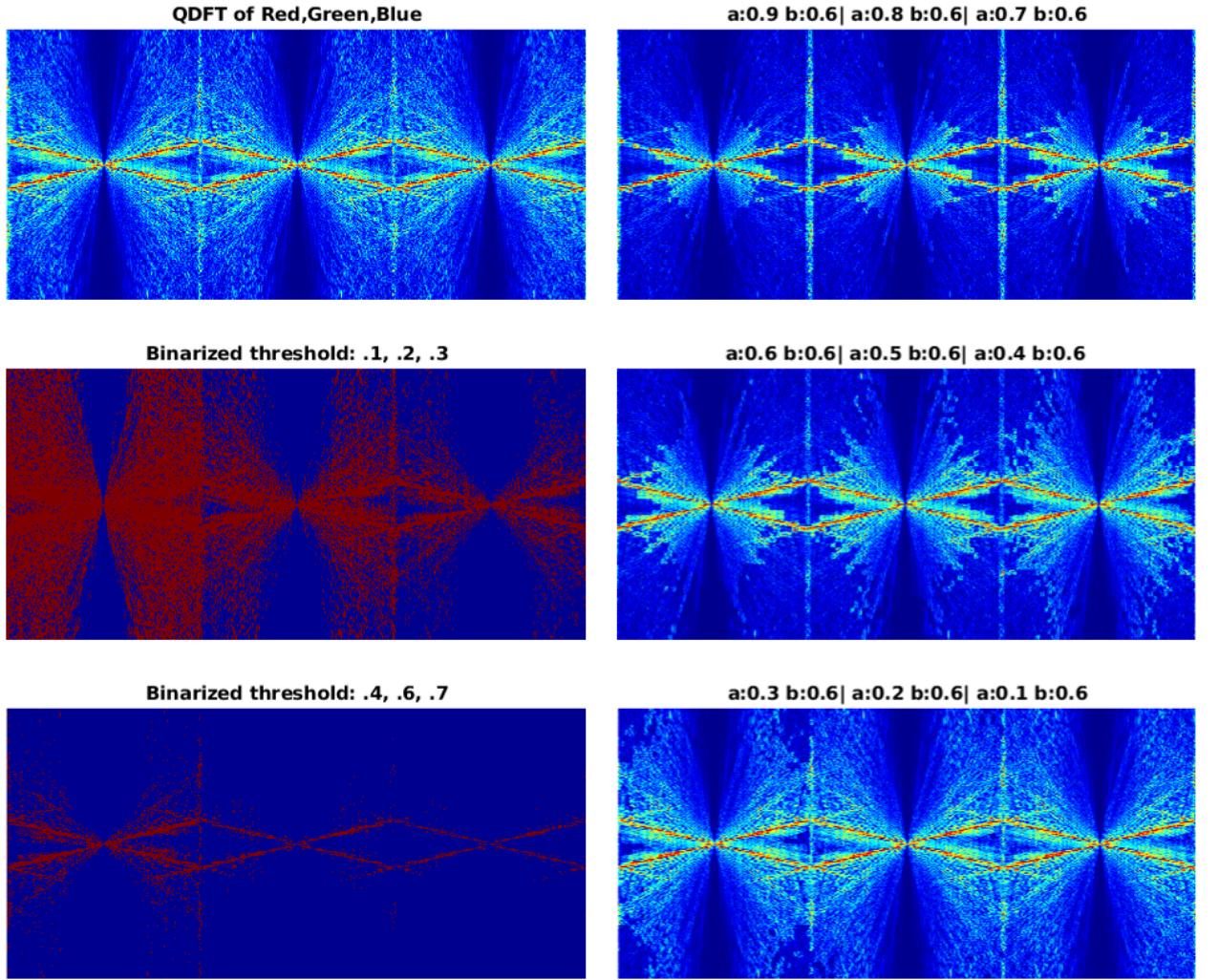


Figure 2.2: Montage images of three Shifted Fourier Transform faces, (top left) RGB Fourier Transform of Face Image, Using only the red color space of the transformed image (middle left) Binarized images with threshold 0.1, 0.2, 0.3, (bottom left) Binarized images with threshold 0.4, 0.5, 0.6, (top right) translational filter with $\beta=0.6$ and $\alpha=0.9,0.8,0.7$, (middle right) translational filter with $\beta=0.6$ and $\alpha=0.6,0.5,0.4$, (bottom right) translational filter with $\beta=0.6$ and $\alpha=0.3,0.2,0.1$.

The second column in figure 2.6 shows the filter being applied with α varying from 0.9 to 0.1. The images shows pixelization around the edges reducing the overall transformed image exponentially. The concentration of intensity is focused in the center whereas the outer region is reduced, this leaves the dense pixels and decreases the sparse pixels.

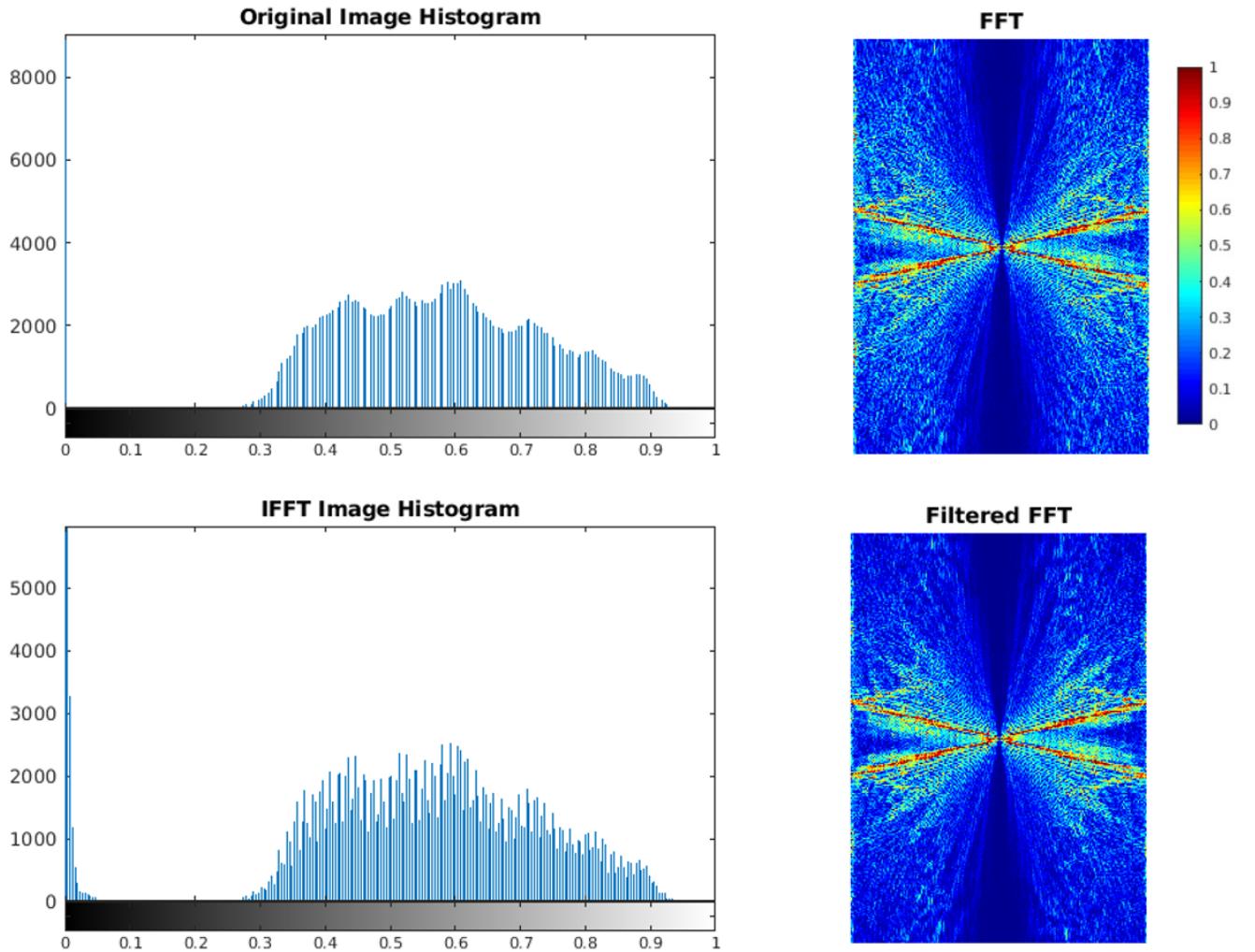


Figure 2.3: After using the translation convolutional filter with parameters $\alpha=0.5$, $\beta=0.8$ (top left) Original Histogram image, (top right) shifted Fourier transformed face, (bottom left) inversed transformed face after translation filter, (bottom right) Fourier transformed face.

The figure above demonstrates the before and after of the translational filter. By alleviating the bulk of sparse filters towards a lower intensity value we are better able to

concentrate a broader use of pixel values versus discrete values shown in the original histogram.



Figure 2.4: Color images after processing each color space (RGB), (left) original image, (right) translational filtered image.

Figure 2.10 shows the comparison of processed images. Because there is no modification to the dense pixels, were the primary of data is located, we are able to see minimal change upon reviewing the translation filtered image.

2.2 Gradient-Based Histogram Equalization

This method introduces a Gradient-Based Histogram Equalization (GB-HE) that focus on a Laplacian gradient and an alpha coefficient. This method entails using a convolutional isotropic derivative operator (kernel) also known as a Laplacian Gradient ∇^2 used to sharpen the image $f(x, y)$, so we may obtain $\nabla^2 f(x, y)$. Following the procedure proposed by Grigoryan-Agaian we may see how this enhancement process is done [10].

$$Y(x, y) = \alpha HE(f(x, y) - X_s) + X_s$$

Equation 2.11: Gradient Based Histogram Equalization

Where X_s is the convolution of the original image and the kernel:

$$X_S = f(x, y) * G$$

Equation 2.12: Convolution of the Original Image

and the kernel G being:

$$\begin{array}{l} G = \frac{1}{4} \begin{bmatrix} 1 & 0 & 1 \\ 0 & -4 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad (1) \\ G = \frac{1}{4} \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (2) \\ G = \frac{1}{8} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (3) \\ G = \frac{1}{8} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \quad (4) \end{array}$$

Equation 2.13: Kernels used for gradient convolution

Producing the following images at $\alpha = 0.7$.

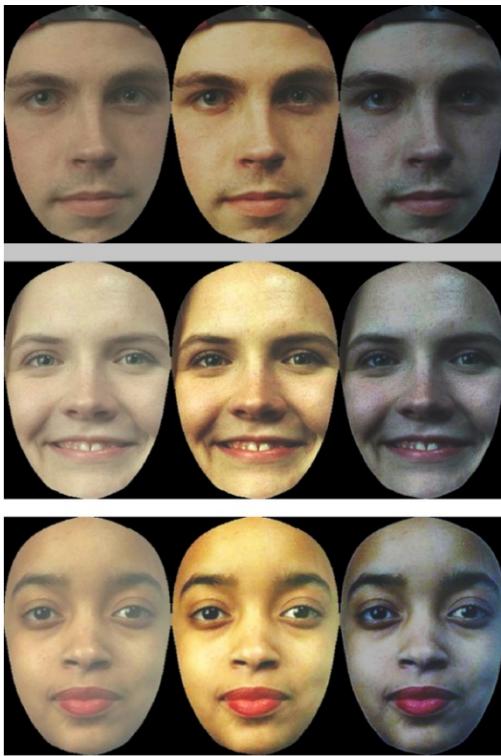


Figure 2.6: Enhanced Faces (FERET 309-B, 306-B, 302-A) (Left) Original, (Middle) MATLAB's Histogram Equalization, (Right) Gradient-Based Histogram Equalization

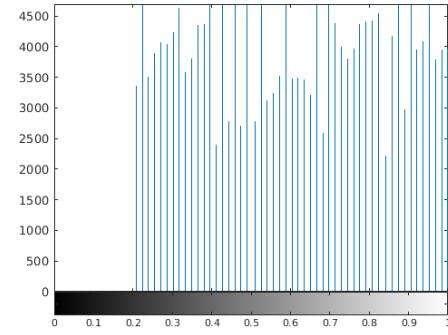


Figure 2.5: MATLAB's Histogram Equalization

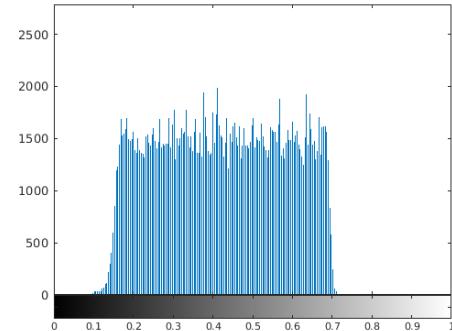


Figure 2.7: Gradient-Based Histogram Equalization

The other gradient images were not included in figure 2.1 due to the similar visual quality image of the GB-HE produced. With a standard deviation of 0.32 for Histogram

Equalization and 0.22 for GB-HE we may infer a condensed concentration from the output image $Y(x, y)$. Moreover, with a contrast comparison from MATLAB's Histogram Equalization we notice a spaced pixel color use versus the GB-HE condence color use centered around 0.4 due to α being 0.7.

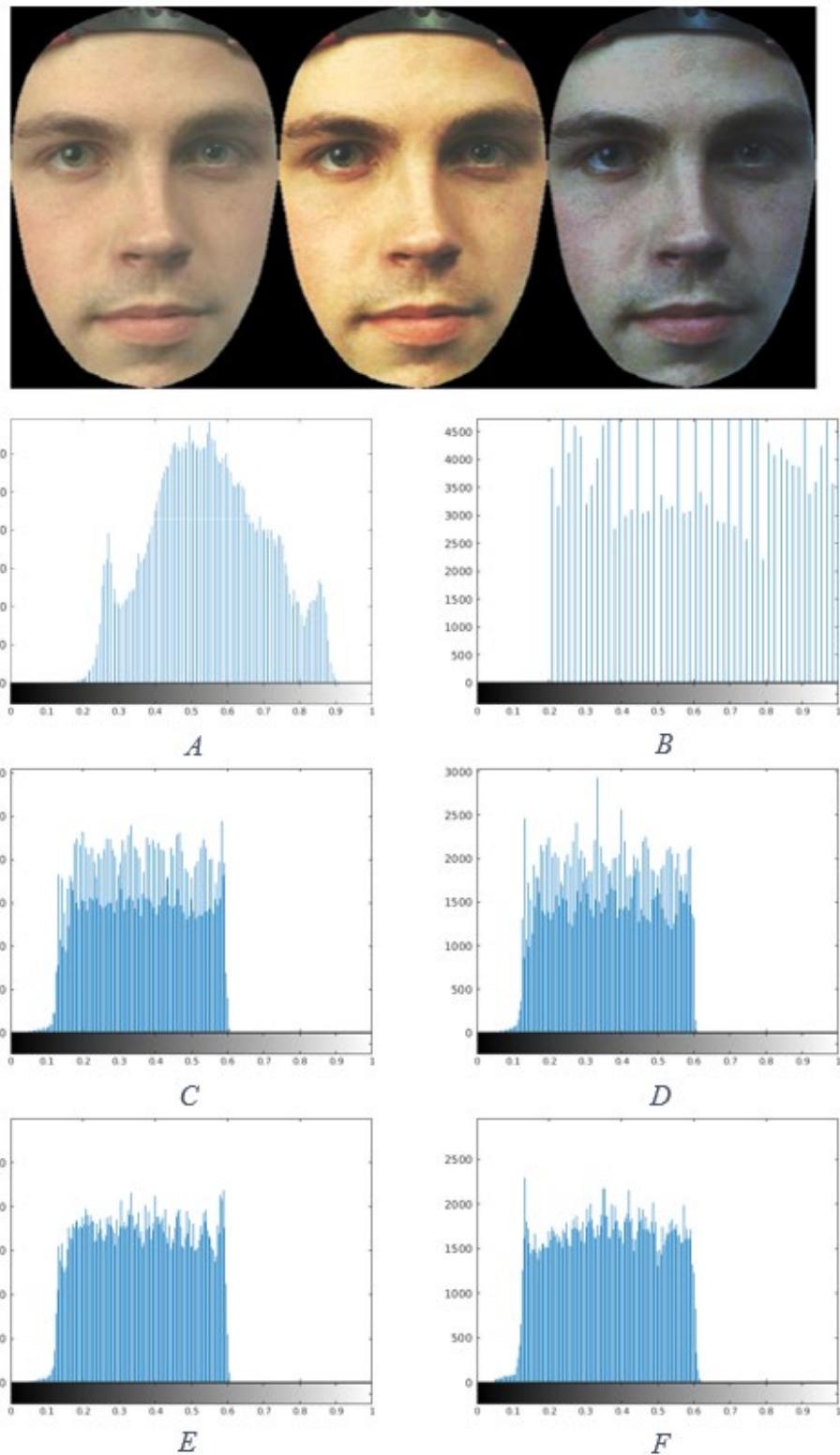


Figure 2.8: GBHE of Face (FERET 309-B), A) Original, B) Histogram Equalization, C) GBHE using gradient (1), D) GBHE using gradient (2), E) GBHE using gradient (3), F) GBHE using gradient (4).

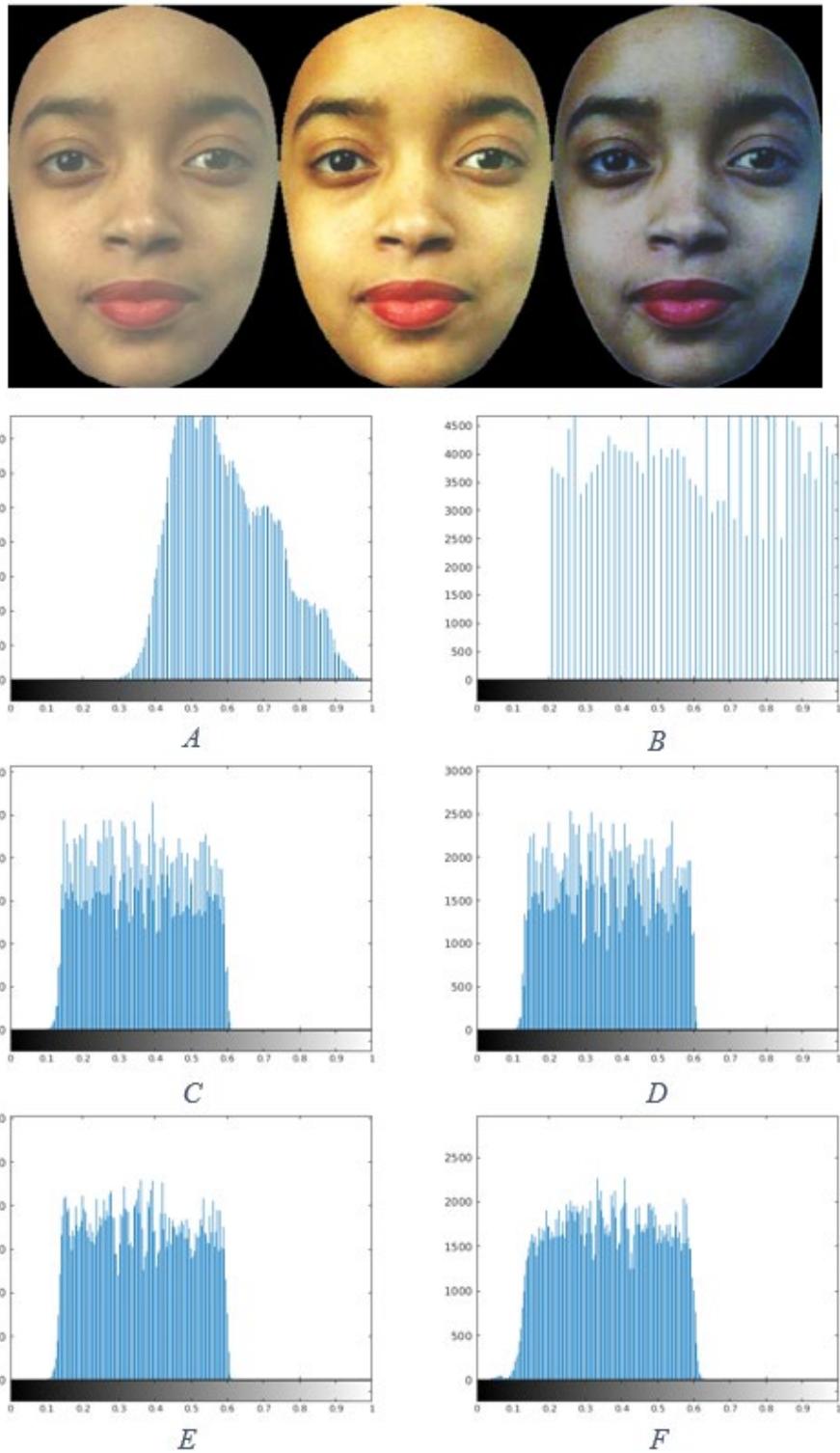


Figure 2.9: GBHE of Face (FERET 302-A), A) Original, B) Histogram Equalization, C) GBHE using gradient (1), D) GBHE using gradient (2), E) GBHE using gradient (3), F) GBHE using gradient (4).

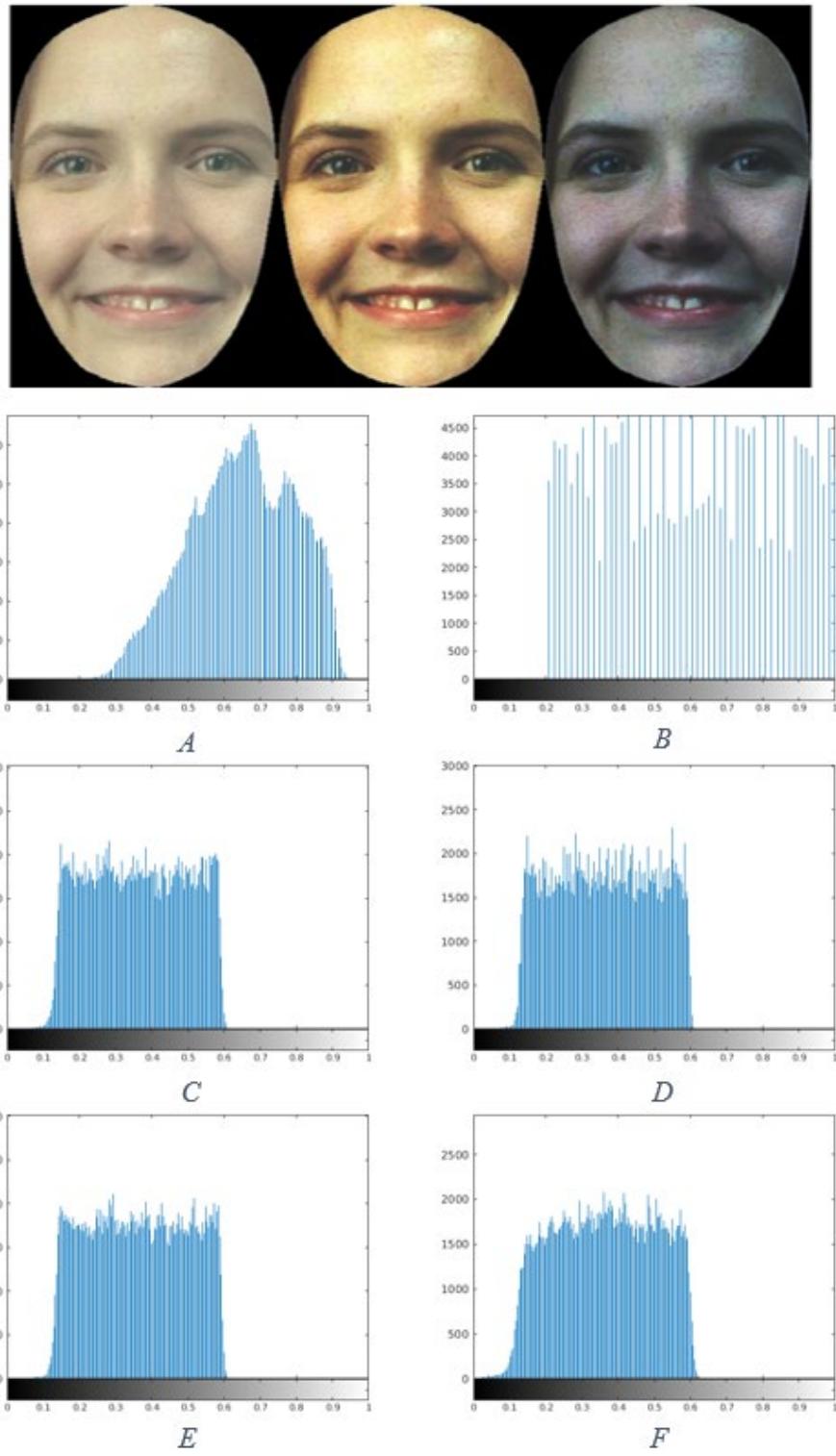


Figure 2.10: GBHE of Face (FERET 306-B), A) Original, B) Histogram Equalization, C) GBHE using gradient (1), D) GBHE using gradient (2), E) GBHE using gradient (3), F) GBHE using gradient (4).

2.3 Results

Using the two Image enhancement Gradient Based Histogram Equalization and Quaternion Translation Block Filter we can compare the enhancement quality. [11]

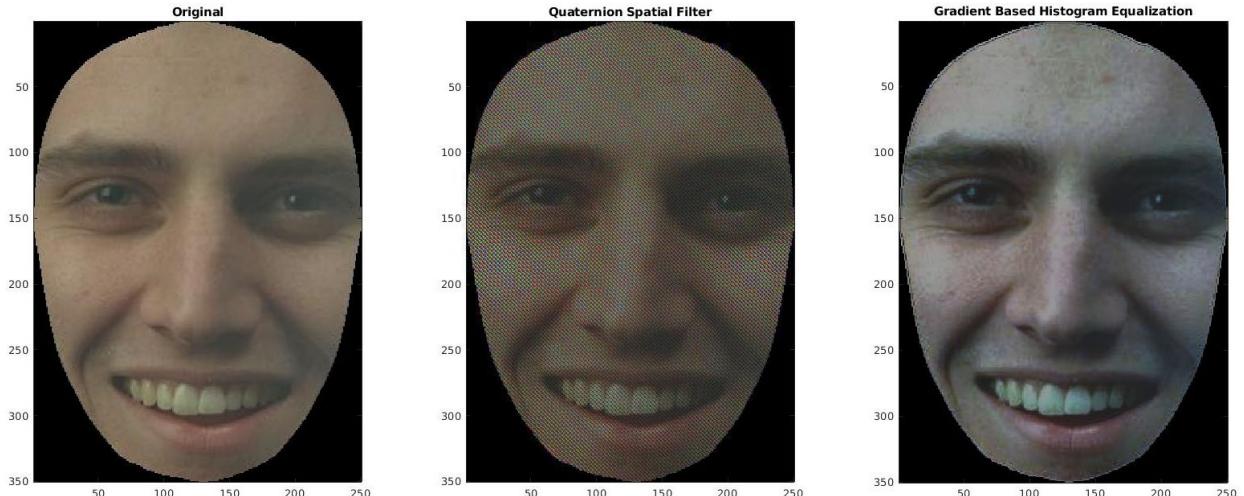


Figure 2.11: Enhanced Images using Quaternion Enhancement Filter and Gradient Based Histogram Equalization.

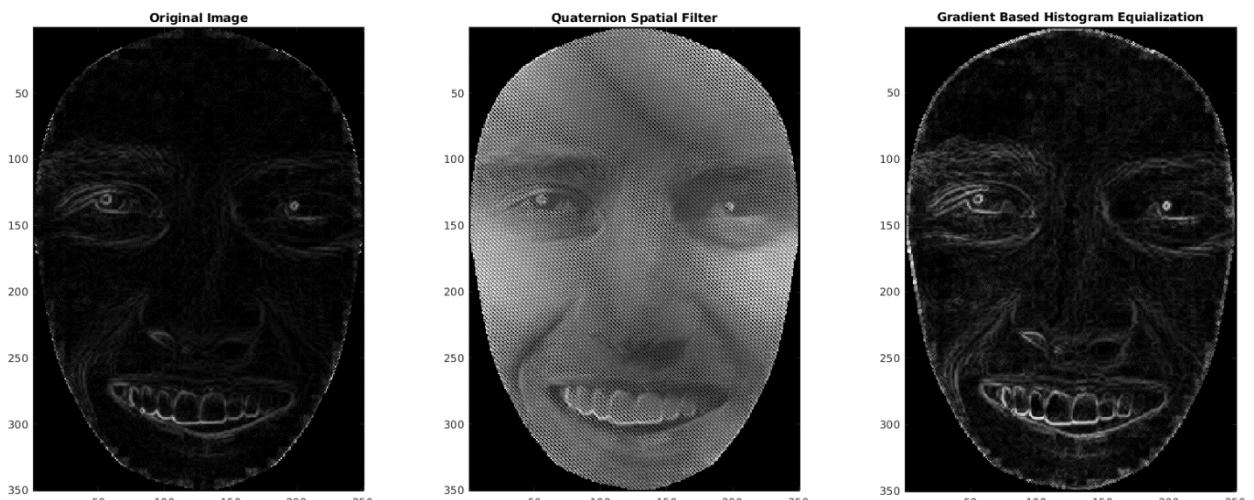


Figure 2.12: Using Michelson Visibility Images for the above images in Figure 2.11

Measurement	Original	Quaternion Spatial Filter	GBHE
EME	3.42	11.98	6.80
EMEE	1.29	108.84	11.80

AME	20.85	2.81	14.98
AMEE	0.19	0.10	0.21
PAMEE $\alpha=0.7$	0.13	0.07	0.15
PAME	9.25	1.93	7.40
MSRME	1.54	0.92	1.45
GAEM $\alpha=0.7$	1.04	1.04	1.04

Table 2.1: Overall Average of Enhancement Measure of all 32 FERET Database Images.

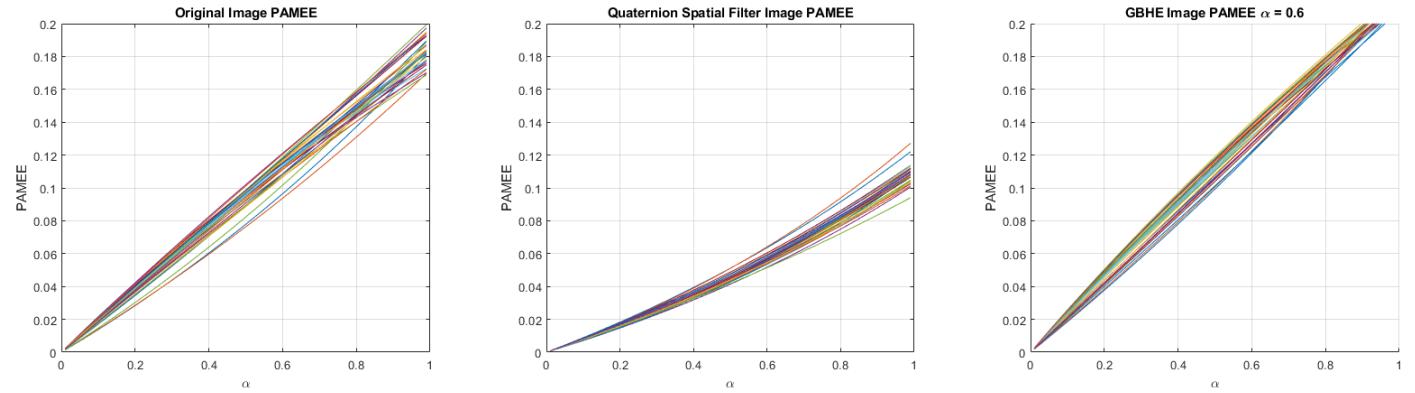


Figure 2.13: Parameterized Agaian-Michelson Entropy Enhancement Measure

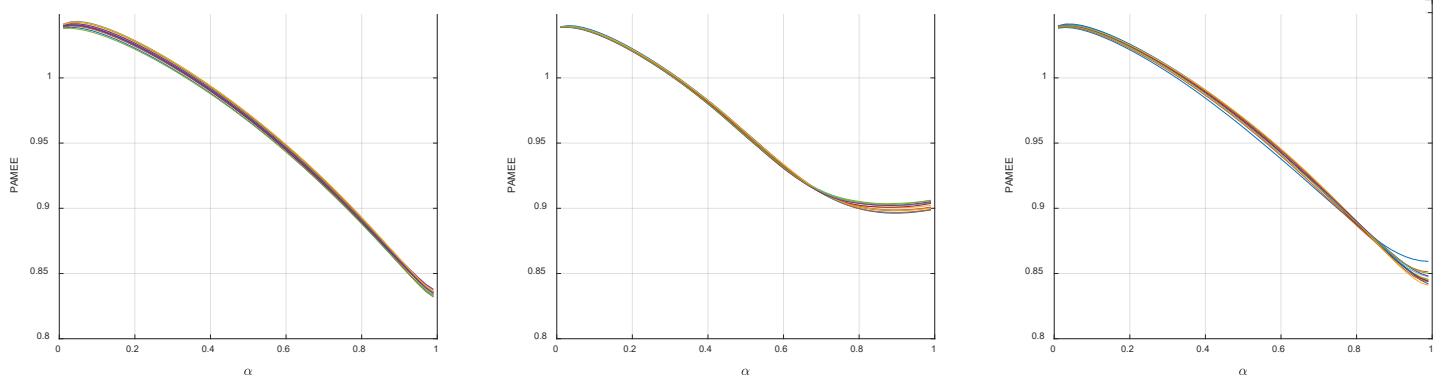


Figure 2.14: Grigoryan-Agaian Enhancement Measurement

Displaying overall increase in EME and Entropy on both enhancement transforms, The AME and PAEM Measurements demonstrate a decline in estimation of the enhancement process, while the Mean-Root-Square Measure appears to decrease as the enhancement process progresses.

CHAPTER 3: COLOR TRANSFORMATIONS

Color Transformation is an important part of any FR system which utilizes color images.

Current Enhancement are performed in the spatial and frequency domain to improve the image quality.

3.1 Quaternion Color Transform

Quaternion Color transform is a method in which RGB Color spaces are separated into a quaternion number of color modes [12]. The image is mapped into a 2x2 or 3x3 quaternion image. This transformation may be used to be filtered or enhanced, in our case the quaternion images were used to demonstrate variation in distribution density functions as well as applying post-enhancement techniques. The quaternion number, $Q = a + bi + cj + dk$, where a, b, c , and d represent different color modes. These quaternion numbers may be mapped onto an image to produce a quaternion image. Figure 3.1 demonstrates the procedure, the image on the left shows a 3x3 mapped image in which RGB, BRG, GBR form a 3x3 matrix, these color modes are iterated over the entire image vertically and horizontally. As for the right of figure 3.1 the image is mapped into a 2x2 image where GR-GB is a matrix similar to the 3x3 map spread over the image vertically and horizontally.

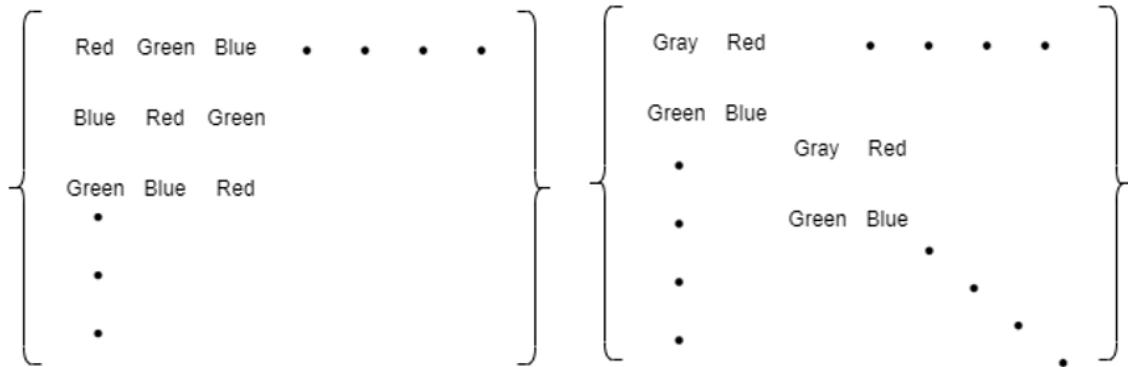


Figure 3.1: (left) 3x3 quaternion map, (right) 2x2 quaternion map.

The following images (Figure 3.2-5) show the quaternion transformed image showing a zoomed portion demonstrating the pixel map, as well as an RGB plot showing the use of the colors red, green, and blue used. Note the RGB plots overlap for colors other than red, green, and blue.
 [13, 14]

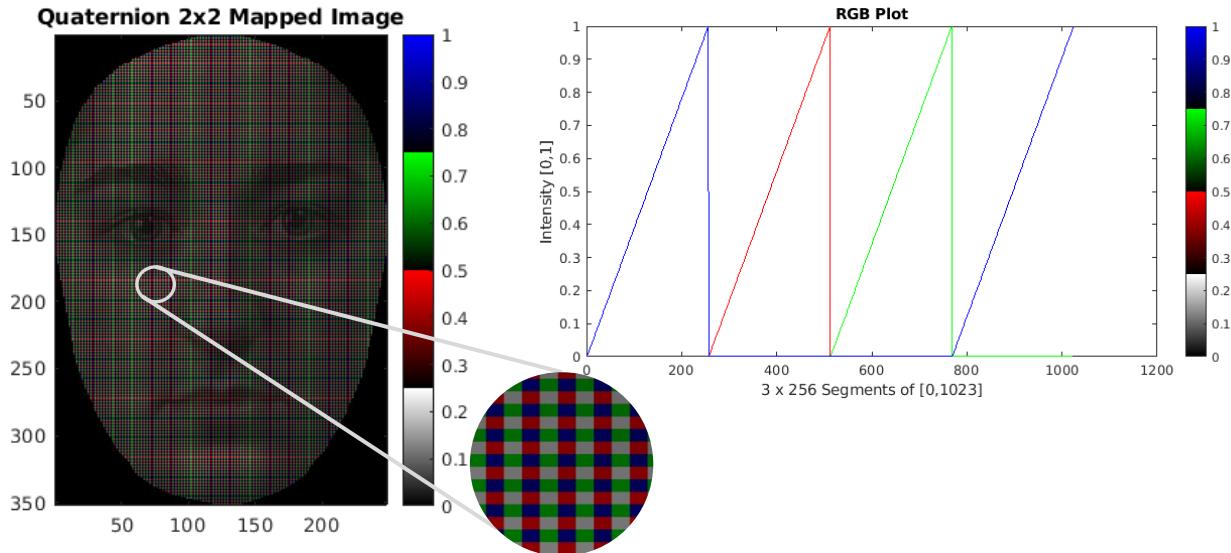


Figure 3.2: 2x2 mapped Quaternion image using colors gray, red, green, blue. (left) Quaternion image (FERET 314-B), (right) RGB plot of colors used for the image.

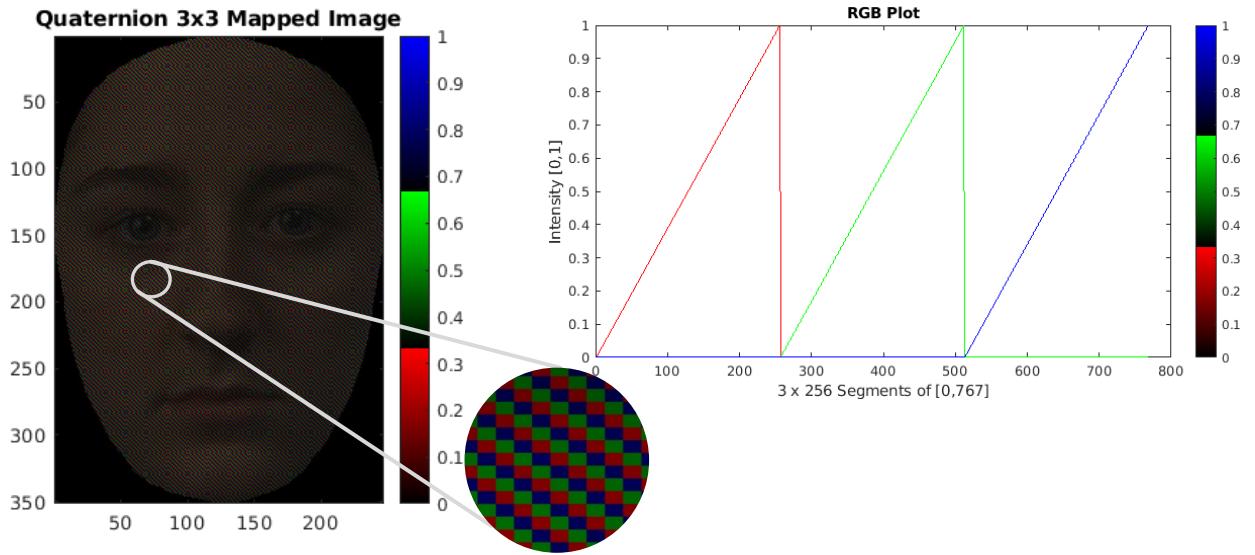


Figure 3.3: 3x3 mapped Quaternion image using colors red, green, blue. (left) Quaternion image (FERET 314-B), (right) RGB plot of colors used for the image.

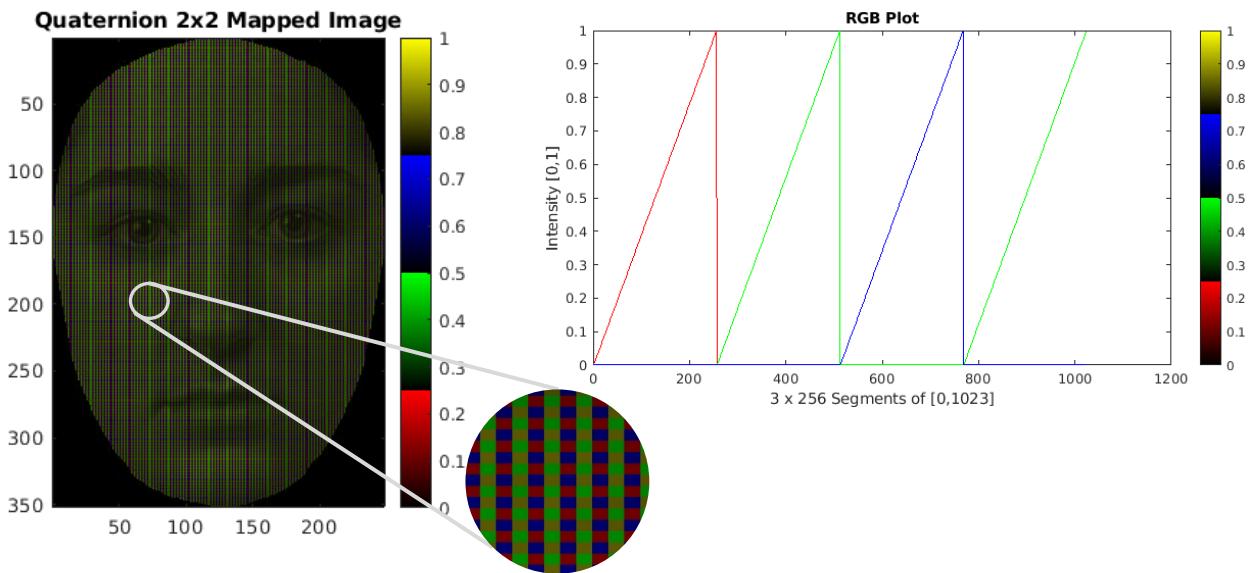


Figure 3.4: 2x2 mapped Quaternion image using colors red, green, blue, yellow. (left) Quaternion image (FERET 314-B), (right) RGB plot of colors used for the image.

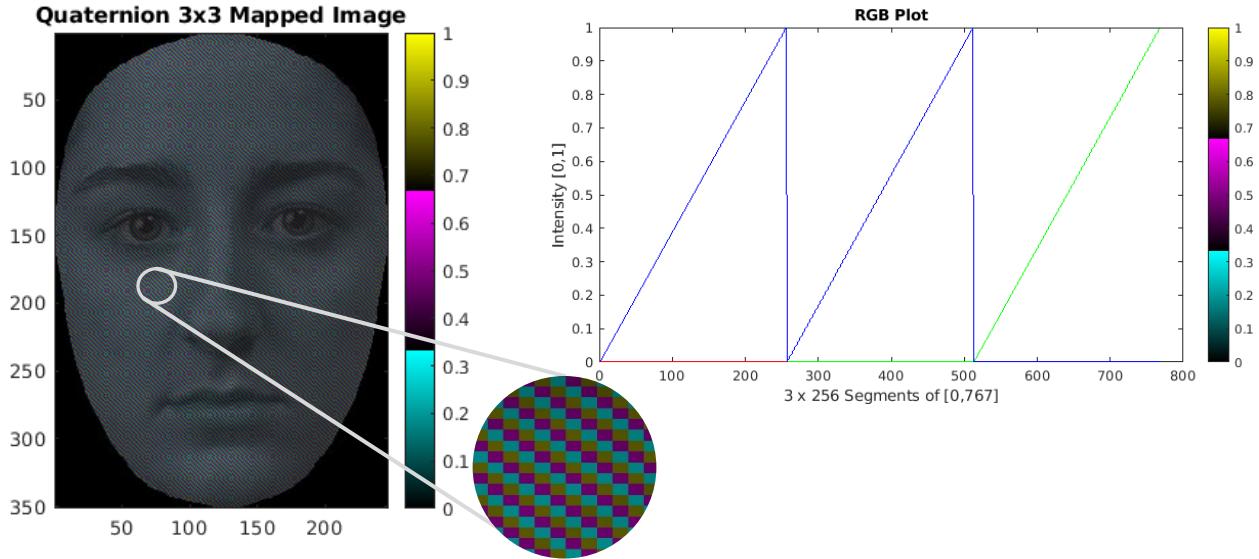


Figure 3.5: 2x2 mapped Quaternion image using colors cyan, magenta, yellow. (left) Quaternion image (FERET 314-B), (right) RGB plot of colors used for the image.

The quaternion image in the Fourier domain are show in figure 3.6. Since MATLAB uses RGB color spaces for displaying color images, we are able to see a unique Fourier quaternion image pattern.

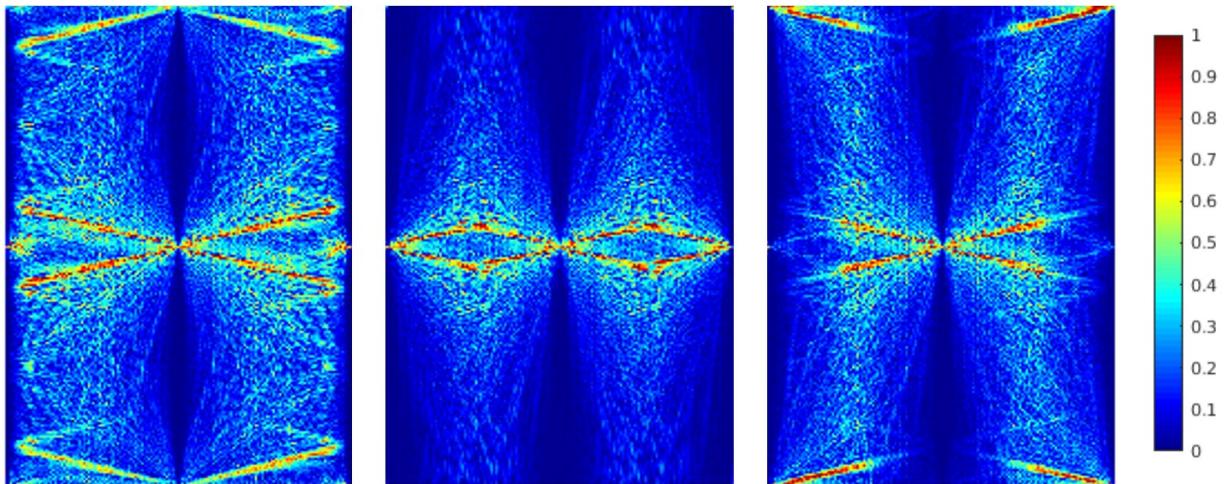


Figure 3.6: Fourier Transform of GrRGB Quaternion Images (left) Red Color Space, (middle) Green Color Space, (right) Blue Color Space.

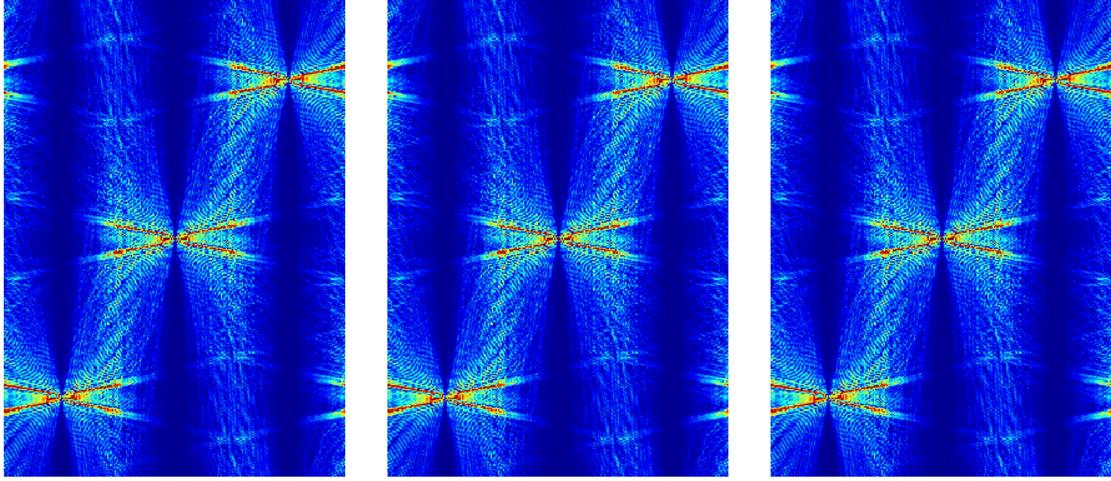


Figure 3.7: Fourier Transform of RGB Quaternion Images (left) Red Color Space, (middle) Green Color Space, (right) Blue Color Space.

Figure 3.6 and 3.7 are both quaternion images GrRGB and RGB respectively, the top figure shows a distinctive shape for the 2x2 mapping vs the nearly identical image of a 3x3 mapping of quaternion images.

3.2 Quantum Image

Currently quantum images representation is an emerging field where the use of quantum computing is being used [15, 16, 17, 18]. This section covers the use of a specific representation of a quantum image known as Multi-Channel Representation based on a previous model Flexible Representation [19] and Normal Arbitrary Superposition State. Using an open-source software for quantum computing called QISKIT [20], we can create a quantum circuit utilizing qubits which are the counterpart to classical bits. There are currently multiple representations of quantum images, to name the most common:

- Qubit Lattice
- Entangled Image
- Real Ket

- Flexible Representation of Quantum Image (FRQI)
 - Multi-Channel Representation of Quantum Images (MCRQI)
- Normal Arbitrary Quantum Superposition State (NASS)
 - NASS with Three Components (NASSTC)
 - NASS Relative Phases (NASSRP)
- Quantum Representation for Log-Polar images (QUALPI)
- Novel Enhanced Quantum Representation (NEQR) [21] [22]
- Color Quantum Image based on Phase Transform (CQIPT)
- Quantum Representation model of color digital images (CRQI) [23]

Quantum images are certainly more mathematically intricate since now we must consider the qubits super positional state $|\psi\rangle$. The idea of this representation to demonstrate the transformation process [24] as well as display usable date for potential use in pattern recognition. Understanding the fundamental properties of quantum computing is necessary to have a conceptual understanding of the Qubit itself and its operational implications. The following equation represents the qubits position and quantum state,

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle$$

Equation 3.1: Quantum State Equation for a Qubit

Where θ represents the spherical rotation of the qubit, Bloch Sphere show in Figure 3.1, and $|0\rangle, |1\rangle$ represents the spherical vector of the qubits quantum polar state. $|\psi\rangle$ Can be measured to a classical bit to provide its probable quantum state, $|0\rangle$ or $|1\rangle$, determined by the coefficients $\cos\frac{\theta}{2}$ and $e^{i\varphi}\sin\frac{\theta}{2}$. [25]

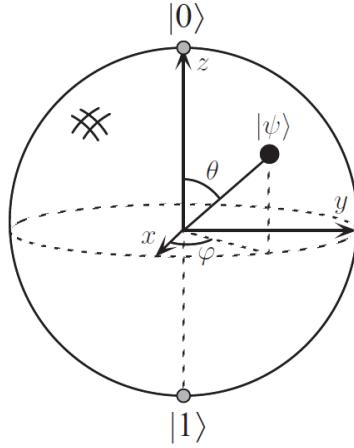


Figure 3.8: Bloch Sphere

3.2.1 MCRQI

As described above the MCRQI is a derivation from FRQI, which is defined by the following equation. [26] [27] [28] [29]

$$|I(\theta)\rangle = \frac{1}{2^{n+1}} \sum_{i=0}^{2^{2n}-1} |C_{RGB\alpha}^i\rangle \otimes |i\rangle$$

Equation 3.2: MCRQI Equation

Where n represents $2n \times 2n$ pixels to be used and $|C_{RGB\alpha}^i\rangle$ is the encoding of the color data,

$$\begin{aligned} |C_{RGB\alpha}^i\rangle = & \cos\theta_{Ri}|000\rangle + \cos\theta_{Gi}|001\rangle + \cos\theta_{Bi}|010\rangle + \cos\theta_{ai}|011\rangle + \sin\theta_{Ri}|100\rangle \\ & + \sin\theta_{Gi}|101\rangle + \sin\theta_{Bi}|110\rangle + \sin\theta_{ai}|111\rangle \end{aligned}$$

Equation 3.3: Quantum Encoding of Color Data

We may achieve $|I(\theta)\rangle$ by creating a quantum circuit using Hadamard, NOT, and Controlled Y-Rotations gates, from the initial quantum state, $|0\rangle^{2n+3}$. [30]

$$|I(\theta)\rangle = \mathcal{P}(|0\rangle^{2n+3})$$

Equation 3.4: Polynomial Preparation Theorem

And the operation \mathcal{P} functions,

$$|I(\theta)\rangle = |H\rangle \left(\prod_{i=0}^{2^{2n}-1} R_i \right)$$

Equation 3.5: Hadamard Gate and Rotation Function of the PPT

Using equation 3.4, an algorithm may be used to create a quantum circuit

$$R_i = C^{2n+2} \left(R_y \left(\frac{2\theta}{2^{2n}-1} \right) \right)$$

Equation 3.6: Controlled Rotation Gates

Where C^{2n+2} represents the $2n + 2$ controlled qubits where the most significant qubit from $C_{RGB\alpha}^i$ is the target qubit.

The Polynomial Preparation Theorem (PPT) which transforms the qubits initial quantum state over to the quantum image representation states that given four vectors $\theta_{Xi} \in \{\theta_{Ri}, \theta_{Gi}, \theta_{Bi}, \theta_{ai}\}$ of angles we may perform a unitary transform \mathcal{P} that transforms the initial quantum state $|0\rangle^{2n+3}$ to the MCRQI state, $|I(\theta)\rangle$, using gate operations. [31]

Initially an MCRQI quantum circuit of a 2x2 image shown in Figure 3.2 and 3.3, was implemented using controlled y-rotation gates, $R_y(2\theta_{Xi})$, representing the four $R_y \left(\frac{2\theta_{Xi}}{2^{2n}-1} \right)$ operation for each color as well as X (NOT) gates.

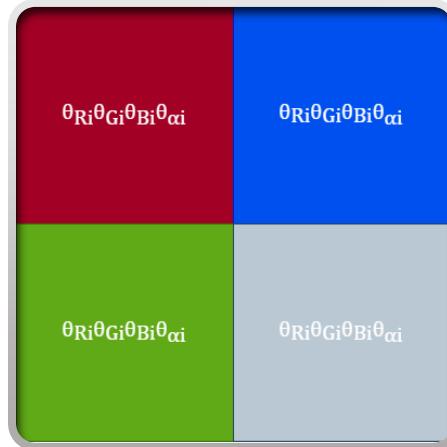


Figure 3.9: 2x2 image, n = 1

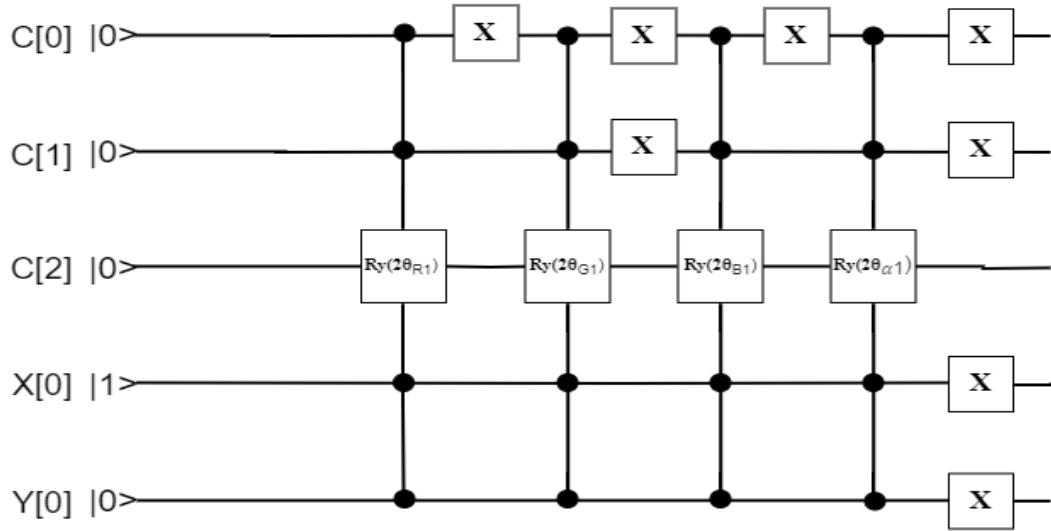
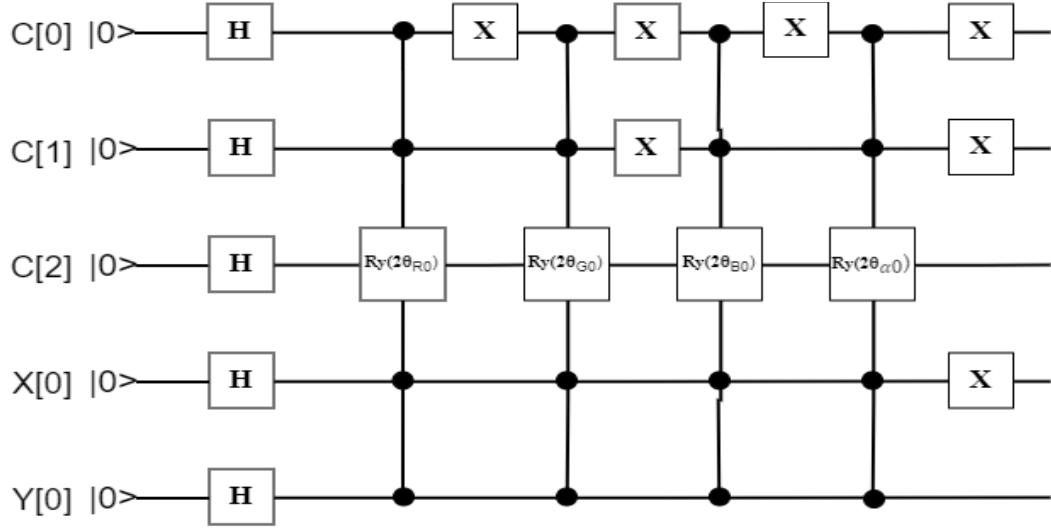


Figure 3.10: Quantum Circuit divided by pixel, (top) $i = 0$, (bottom) $i = 1$.

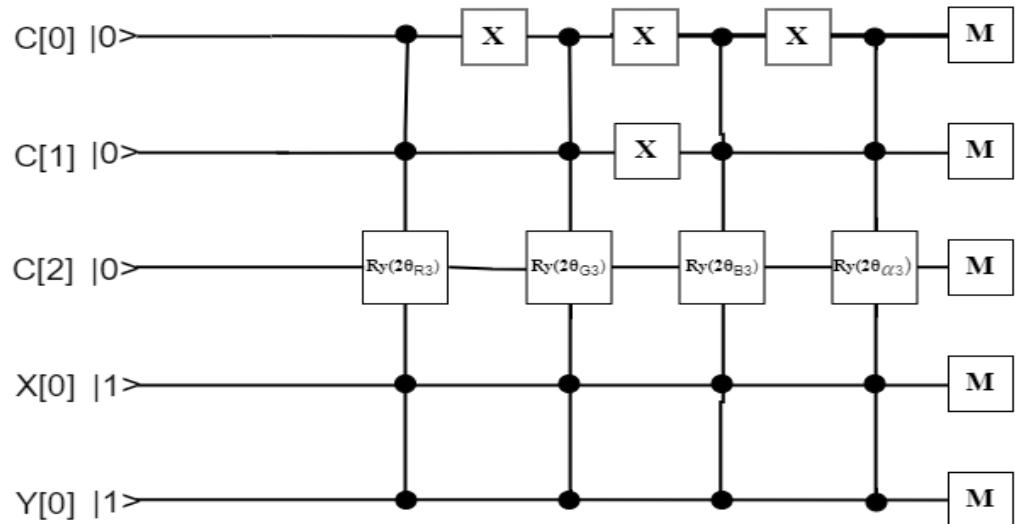
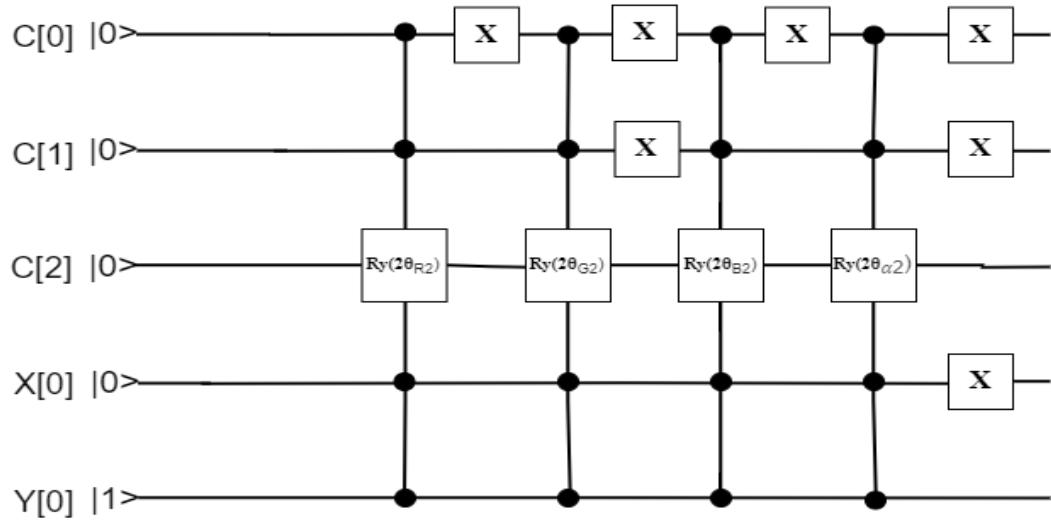


Figure 3.11: Quantum Circuit divided by pixel, (top) $i = 2$, (bottom) $i = 3$.

For each pixel four R_y rotation gates are performed as well as X (NOT) gates, to perform the appropriate color qubit iteration. Notice how each pixel value i , displays a change in the X and Y axes.

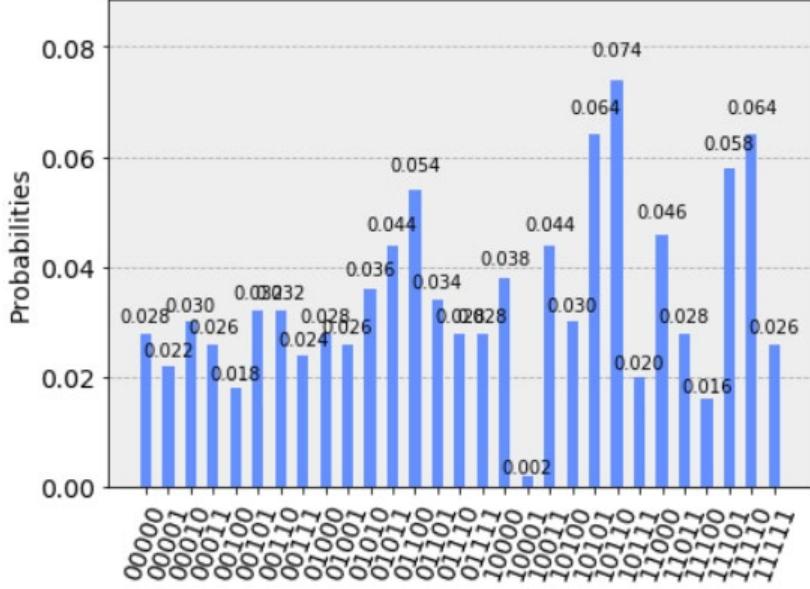


Figure 3.12: Histogram of count executions with probability measurements.

Interpreting the data using the Qiskit measurement function showing the classical bit representation of the Qubit. The histogram is divided into bins demonstrating the quantum state probability with the possible combination quantum state. Due to the amount of pixels of database images, computation time for processing was not feasible. However, code titled MCRQI.py is attached in the Appendix for testing.

3.2.1 NASS

Normal Arbitrary Superposition State is an alternative method for representing a quantum image. The NASS color representation of an image is as follows [32] [33]:

$$\phi_i = \frac{\pi}{2(M-1)} i,$$

Equation 3.7: NASS Color Angle

where ϕ_i , represents an angle of $\psi = \{\phi_0, \dots, \phi_{M-1}\}$

In this case $M = 2^{24} = 16,777,216$ of (255,255,255) RGB Color Scheme, and $i = (R \times 256 \times 256) + (G \times 256) + B$ or in Hex 0xRRGGBB. The NASS state representation of a 2^n image is as follows:

$$|\psi\rangle_k = \sum_{i=0}^{2^n-1} \theta_i |i\rangle = \sum_{i=0}^{2^n-1} \theta_i |i_n \dots i_1\rangle,$$

Equation 3.8: NASS Representation

where $\theta_i = \frac{a_i}{\sqrt{\sum_{j=0}^{2^n-1} a_j^2}}$ and $a_i \in \mathcal{Y}$ represent the color,

and $|i\rangle = |i_1 \dots i_n\rangle$ representing the coordinate of the i^{th} pixel.

$$|\psi\rangle_2 = \sum_{i=0}^{2^n-1} \theta_i |i_4 i_3\rangle |i_2 i_1\rangle$$

Equation 3.9: 4x4 NASS Representation

i_1 being the LSQB and i_n being the MSQB and are separated in their respective axes ($x - axis$, $y - axis$). Using a 4x4 image shown below the overall NASS calculation process is as follows:

$$|\psi\rangle_2 = \theta_0 |00\rangle |00\rangle + \theta_1 |00\rangle |01\rangle + \dots \theta_{14} |11\rangle |10\rangle + \theta_{15} |11\rangle |11\rangle$$

Equation 3.10: Full 4x4 NASS Representation

The quantum circuit is then constructed using controlled-Y gates with $O(N \log_2 N)$ operations.

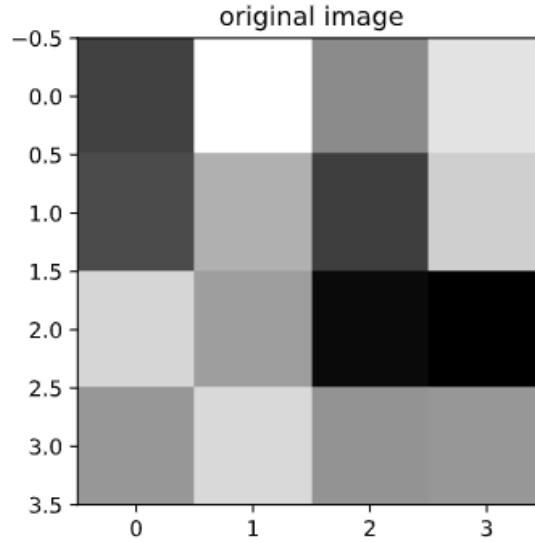


Figure 3.13: 4x4 Images used for NASS representation

To visualize the quantum image does require an additional transform (Li, 2019). The data observed is the quantum likelihood of the resulting bits, in this case procures 7 likely probabilities.

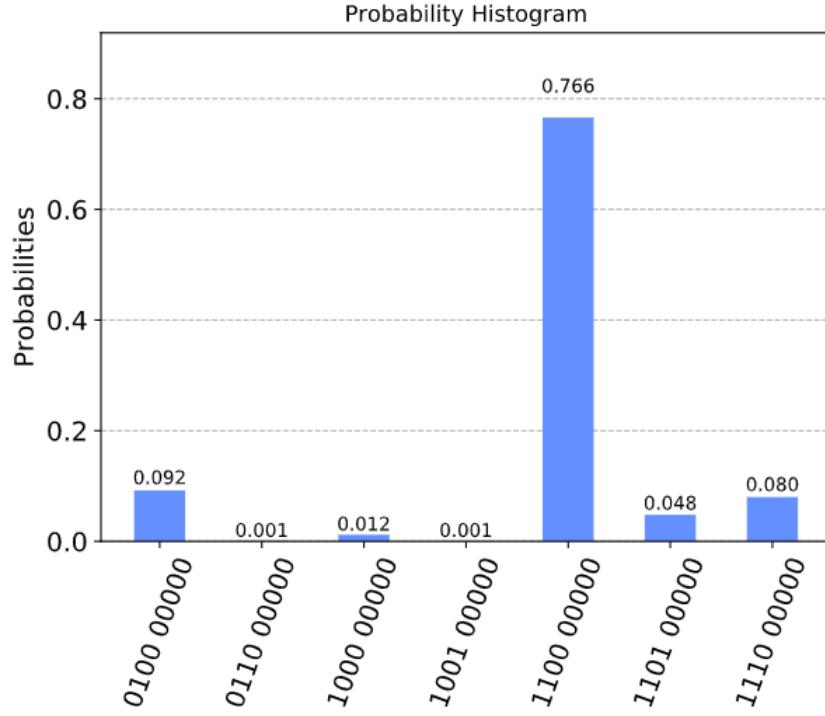


Figure 3.14: NASS Histogram of 4x4 image

The NASS Quantum Circuit was not include due to the similarity of the MCRQI.

CHAPTER 4: FEATURE EXTRACTION

Feature extraction is the process of gathering characteristics from the image which will later be used in a pattern recognition system. For our model we will be using two classes of features Local Binary Patterns (LBP), and Color Concentration also considered the images histogram.

4.1 Local Binary Pattern

The Local Binary Pattern process has many variants, we are using the generalized version for this application [34]. This process is to collect the neighboring pixels to determine the center pixel which will be a culmination of binary summation:

$$s(f_{x-1}^{y+1})2^0 + s(f_x^{y+1})2^1 + \dots + s(f_{x-1}^y)2^7$$

Equation 4.1: Binarization method for LBP

Producing a binary image shown on the next slide. Uniformity is the removal of binary translation less than or equal to 2.

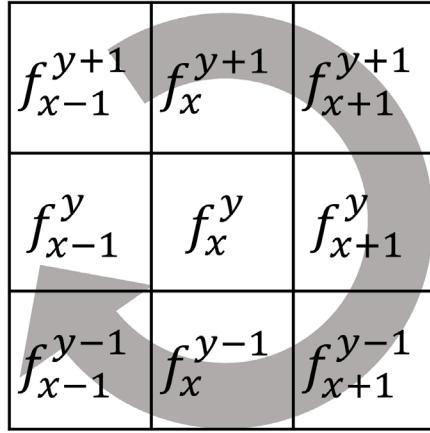


Figure 4.1: Binary Rotation of LBP process for Neighboring pixels f_x^y

The following equation represent the sequential process of deriving a local binary pattern. The function g_p represents the neighboring pixels with p representing the pixels position. g_p is subtracted by its center pixel in turn converted to 1 or 0 given its negative or positive value. Each of these values are used to determine the binary summation of operator 2^p providing the LBP image.

$$g_p = I(x_p, y_p), \quad p = 0, \dots, P - 1$$

Equation 4.2: The value of the Binary Rotation of neighboring pixels.

Neighboring Pixel Position x_p and y_p at radius R and in direction $\cos\left(\frac{2\pi p}{P}\right)$,

$$x_p = x + \text{round}\left(R \cos\left(\frac{2\pi p}{P}\right)\right) \quad \text{and} \quad y_p = y - \text{round}\left(R \sin\left(\frac{2\pi p}{P}\right)\right)$$

Equation 4.3: The value of the center pixel of the Binary Rotation.

The binarization process of the pixel difference as well as the functions piecewise operation.

$$S \in \{s(g_0 - g_c), s(g_1 - g_c), \dots, s(g_{p-1} - g_c)\}$$

Equation 4.4: Binarization process of pixel difference.

$$s(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$$

Equation 4.5: Piecewise operation of resulting difference.

The local binary pattern of the image,

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p$$

Equation 4.6: LBP representation of the image.

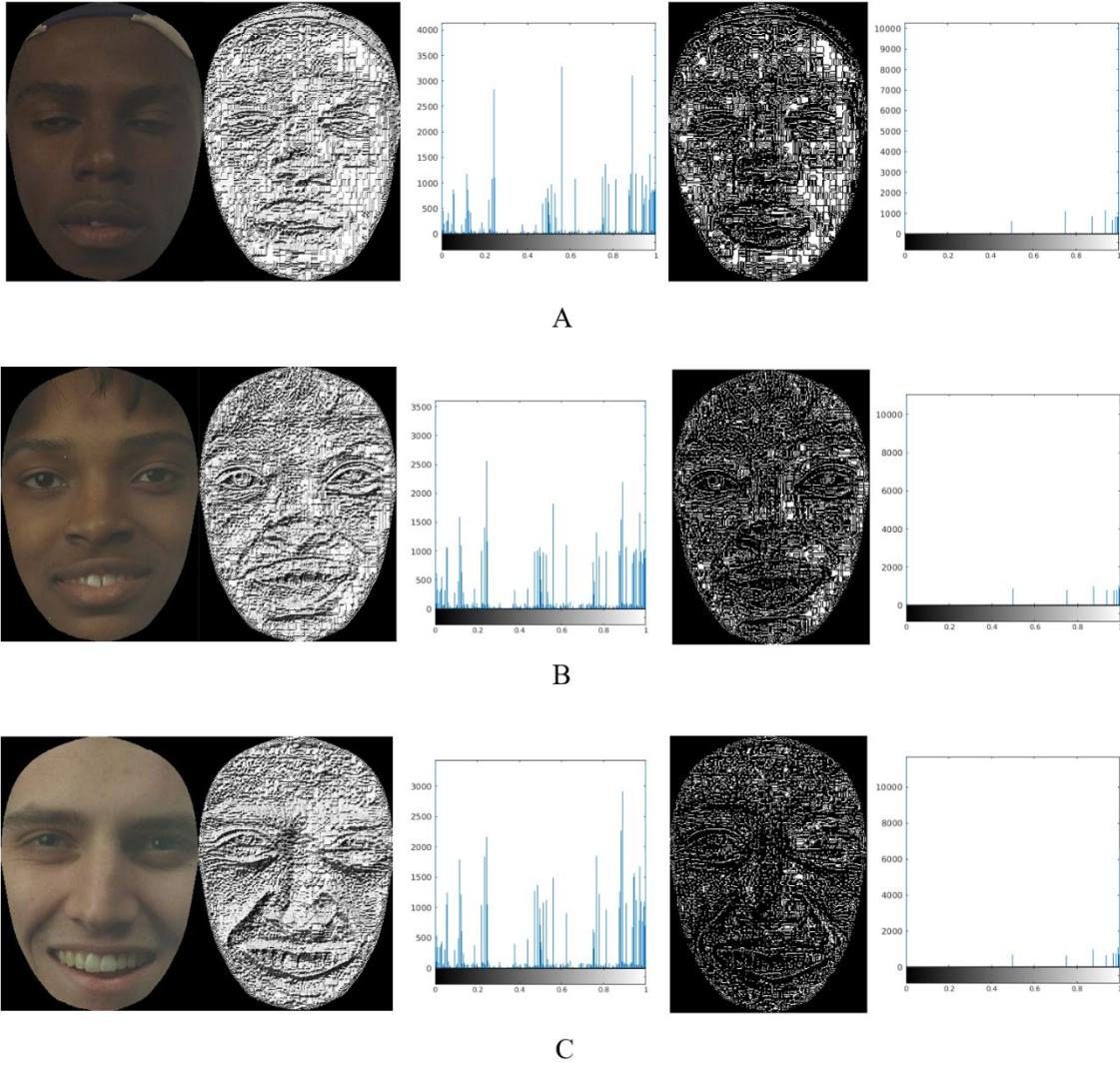


Figure 4.2: Local Binary Pattern and Uniform LBP of images 300-B,305-B,310-B (left) Original image, (second to left) Local Binary Pattern, (middle) Histogram of Local Binary Pattern, (second to right) Uniform Local Binary Pattern, (right) Histogram of Uniform Local Binary Pattern.

Figure 4.2 shows the LBP and uniform LBP transform of the image and their respective histograms. The nonuniform LBP histogram shows larger variation than the uniform LBP, this is caused by the removal of g_p with binary transitions other than 0 or 2 as explained in [5].

4.2 Color Concentration

Color concentration is considered the histogram of the facial image. Where the image color is considered from 0 to 1. This is a feature we use for our pattern recognition algorithm. Since our premise is to improve quality of the overall image through color image enhancement process, we do not consider physical facial measurements for our system.

CHAPTER 5: PATTERN RECOGNITION

For the pattern recognition process two primary methods were used, Correlation Coefficient and Minimum Distance of Kernel Distribution. The following sections describe the functions for the proposed pattern recognition system.

5.1 Correlation Coefficient

A common method used to describe the similarities between two images is Correlation Coefficient, this is known as Prototype Matching, [2] described by the following equation,

$$\gamma = \frac{\sum_s \sum_t [w(s, t) - \mu_w][f(x + s, y + t) - \mu_{xy}]}{\sqrt{\sum_s \sum_t [w(s, t) - \mu_w]^2 \sum_s \sum_t [f(x + s, y + t) - \mu_{xy}]^2}}$$

Equation 5.1: Correlation Coefficient

Where w , represents the compared image with the target image $f(x, y)$, providing a scalar value which defines the correlation between the two images between 0 and 1.

5.2 Minimum Distance of Kernel Distribution

The Density Distribution is another method for measuring patterns among a database. A kernel distribution is applied to describe the similarities between enhanced and transformed images.

$$\hat{f}_h(x) = \frac{1}{Nh} \sum_{i=1}^N \zeta\left(\frac{x - x_i}{h}\right)$$

Equation 5.2: Kernel distribution

This pdf uses a Smoothing Kernel, ζ as well as a bandwidth value, h .

$$\zeta = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

Equation 5.3: Smoothing Kernel

and

$$h = \left(\frac{4}{3} \frac{\sigma^5}{i} \right)^{\frac{1}{5}}$$

Equation 5.4: Bandwidth Value

Using the Formula:

$$\epsilon = \frac{1}{N} \sum |X - Y| \times 100$$

Equation 5.5: Error-Difference Estimation

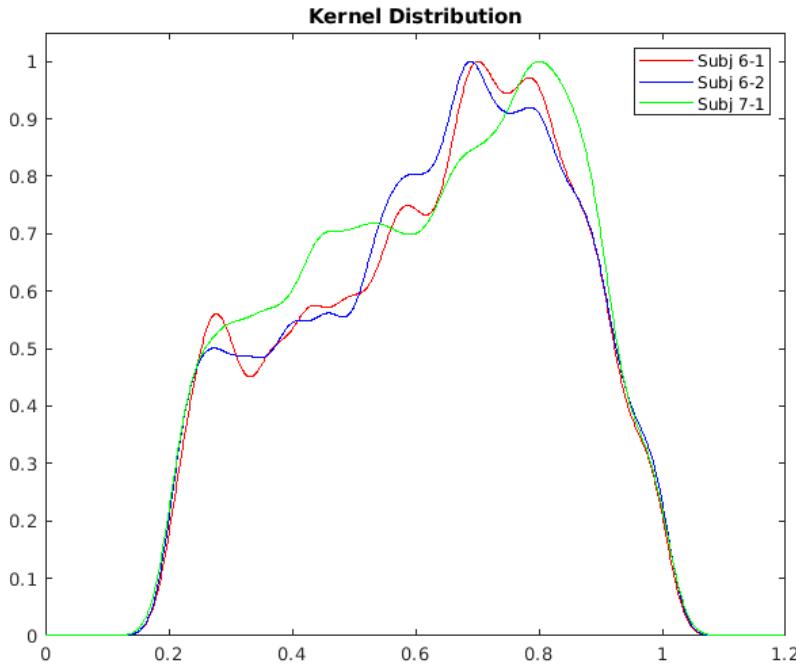


Figure 5.1: Kernel distribution of 3 images, 305-A/B, 306-A

Seen in Figure 5.1, shows a direct representation of kernel distribution. It may be observed that third image is not comparable to the first two images. Kernel distribution is a fitted density distribution function which estimates the overall image color concentration.

There are alternate methods to improve quality of the minimal distance measurement, one being centering the mean of the target with the matching distribution, another method used is to align maximum peak of the target with the matching distribution.

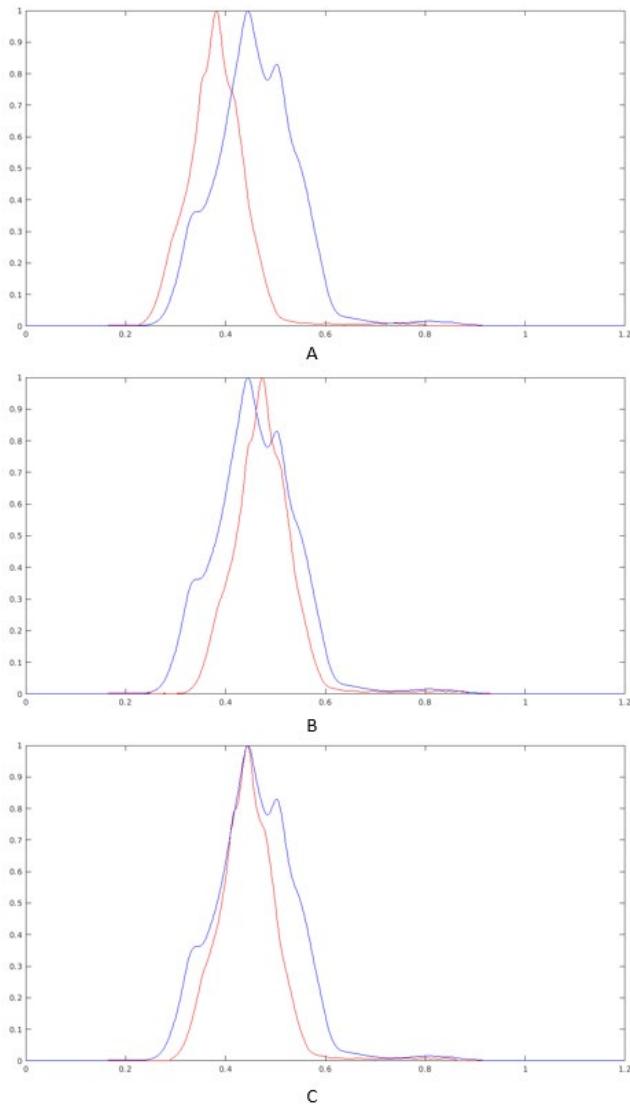


Figure 5.2: Kernel Distribution Adjustment methods, (Top A) Two image kernel distribution, (Middle B) Center Mean Target Adjustment, (Bottom C) Maximum Peak Alignment of two images.

CHAPTER 6: CONCLUSION

To conclude the enhancement of images for the proposed facial recognition system the following results were recorded and developed using MATLAB. Since we are not using any stochastic methods and are strictly using deterministic approach are results are considerably linear.

6.1 Results

Results for Local Binary Pattern as well as its alternative Uniform feature were measured by applying kernel distribution and correlation comparison measurements.

%	LBP		LBP Uniform		Kernel Dist.		LBP Kernel Dist.		RAW	
Method	Corr	Min Dist.	Corr	P-Min Dist.	P-Min Dist.	M-Min Dist.	P-Min Dist.	M-Min Dist.	Corr	Min Dist.
Red	53.13	53.13	65.63	46.88	25.00	28.13	34.38	34.38	84.38	31.25
Green	56.25	56.25	56.25	25.00	40.63	18.75	18.75	25.00	84.38	34.38
Blue	56.25	53.13	62.50	40.63	25.00	18.75	15.63	15.63	87.50	28.13
Gray	43.75	50.00	62.50	34.38	12.50	15.63	28.13	28.15	84.38	37.50

Table 6.1: Correlation Coefficient & Kernel Distribution Results.

The following table demonstrate the observation taken from applying all the enhancements to the database as well as the target image, this brings a linear relationship with the correlation coefficient were as the kernel distribution shows an increased performance on the quaternion images.

Method	Kernel Distribution	Correlation
GBHE 1	18.75	84.38
GBHE 2	6.25	84.38
GBHE 3	15.63	84.38
GBHE 4	3.13	84.38
Translational Block Filter $a = b = 0.5$	31.25	85.42
GrRGB	53.13	84.38
RGBY	84.38	84.38

RGB	84.38	84.38
CMY	84.38	84.38

Table 6.2: Enhanced Images using Correlation and Kernel Distribution

6.2 Improvements

Improvements can be made either through using alternative methods such as stochastic machine learning algorithms or by steadily improving the algorithm of the current processes. In conclusion a better approach for recognizing data could have been taken however, since color enhancement was the focus determining that Quaternion and Translational Spatial Filter may be further research.

CHAPTER 7: OTHER WORKS

7.1 Simulated Three-Dimensional Facial Construction

Using Machine learning algorithms to create a 3D reconstruction of the subject's face [35] [36] [37] [38] [39] [40], this area is being researched to produce viable practical application. Having the capability to produce a 3D face model based off a 2D image will be the leading research for facial analysis and biometrics. Morphable models shown in Figure 7.1 can replicate the original face by using a 3D mask as the prototype and modification are made to create a virtual image.

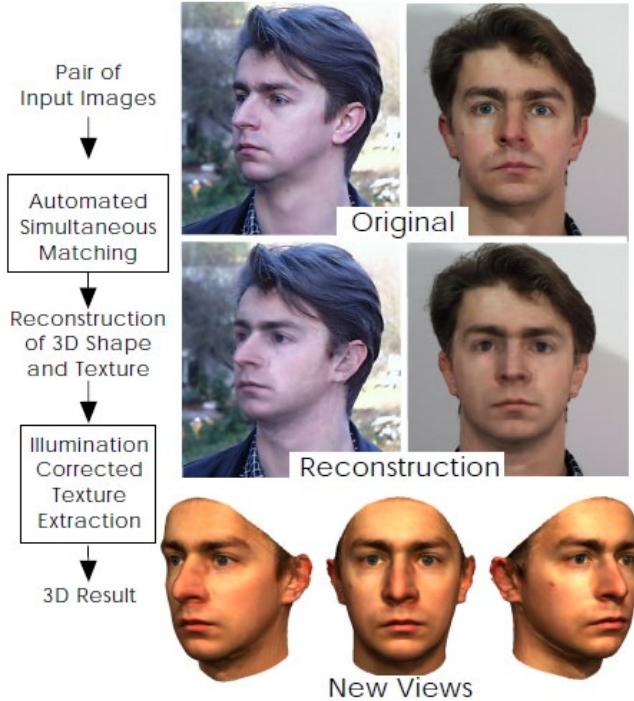


Figure 7.1: 3-D Morphable Model [7]

7.2 Range Imaging for Facial Analysis

Range imaging is acquired by using infrared and thermal imaging in conjunction with visible images for facial measurements in pattern recognition [41]. The ability to measure the face with range imaging allows for a three-dimensional scan giving the FR system data to process more efficiently over 2D. This application may be useful during weather condition to compensate for the visible quality. Figure 7.2 shows the process of range imaging which is eventually acquired and analyzed for further processing.

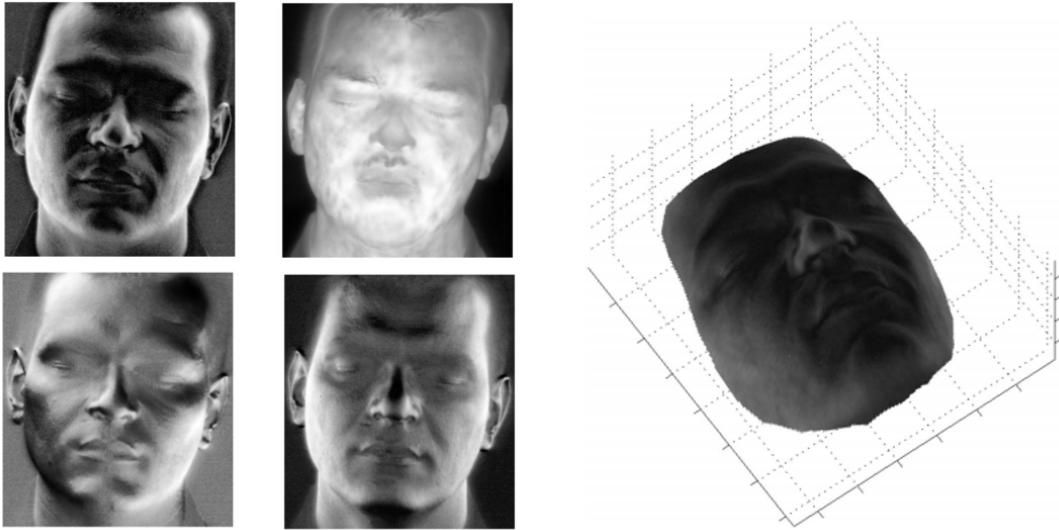


Figure 7.2: Example of facial measurement using range imaging.

CHAPTER 8: FURTHER RESEARCH

8.1 Normal Arbitrary Superposition State Three Component

Another Quantum Image representation that could potentially be utilized in such a system would be called a Normal Arbitrary Superposition State Three Component. Since the quantum computers are in the early stages of development it may be soon that we see quantum image representations being utilized for pattern recognition [42]. To discuss briefly on the how to produce a Normal Arbitrary Superposition State Three Component (NASSTC) image we may demonstrate a method which utilized three color components.

The following equations are used to derive a NASSTC quantum image representation.

$$r_j = y_1 \frac{\pi}{2}, g_j = y_2 \frac{\pi}{2}, b_j = y_3 \frac{\pi}{2}$$

Equation 8.1: RGB Components of NASSTC

Where y_1, y_2, y_3 represent the intensity values of the color,

$$\theta_{rj} = \frac{r_j}{G_{rgb}}, \theta_{gj} = \frac{g_j}{G_{rgb}}, \theta_{bj} = \frac{b_j}{G_{rgb}}$$

Equation 8.2: Calculate the average intensity

Having G_{rgb} be the sum of all color angles of all pixels.

$$G_{rgb} = \sqrt{\sum_{i=0}^{2^n-1} (r_i^2 + g_i^2 + b_i^2)}$$

Equation 8.3: Magnitude of each color component

Providing the three-color components $|\psi_{Ar}\rangle, |\psi_{Ag}\rangle, |\psi_{Ab}\rangle$

$$|\psi_{Ar}\rangle = \sum_{j=0}^{2^n-1} \sqrt{3} \theta_{rj} |j\rangle, \quad |\psi_{Ag}\rangle = \sum_{j=0}^{2^n-1} \sqrt{3} \theta_{gj} |j\rangle, \quad |\psi_{Ab}\rangle = \sum_{j=0}^{2^n-1} \sqrt{3} \theta_{bj} |j\rangle$$

Equation 8.4: Color Components

The compression of the above equation represents the NASSTC.

$$|\psi_c\rangle = \sum_{j=0}^{2^n-1} \theta_{jr} |j\rangle |01\rangle + \theta_{jg} |j\rangle |10\rangle + \theta_{jb} |j\rangle |11\rangle$$

Equation 8.5: NASSTC equation

It is possible to add an additional information $|X_j\rangle$ by multiplying an additional qubit alongside $|j\rangle$, where $|X_j\rangle = \cos \gamma |0\rangle + e^{i\lambda_j} \sin \gamma_j |1\rangle$, this is known as the Relative Phase. This representation is beneficial over MCRQI due to this feature. Where this may contain important data in the facial recognition process [43].

8.2 Potential of FR

With regards to the translation filter, the reduction of sparse pixels concentrates a dense area near the center leaving behind the primary data. Though this may be considered a filter algorithm, this virtually may be used to conserve data for alternative applications. Fourier

Transformed image using the spatial filter with parameters $\alpha=0.9$ and $\beta=0.1$ shows the empty space surrounding the center data.



Figure 8.1: Original Image, (middle) filtered image $\alpha=0.9, \beta=0.3$, (right) filtered image $\alpha=0.9, \beta=0.1$.

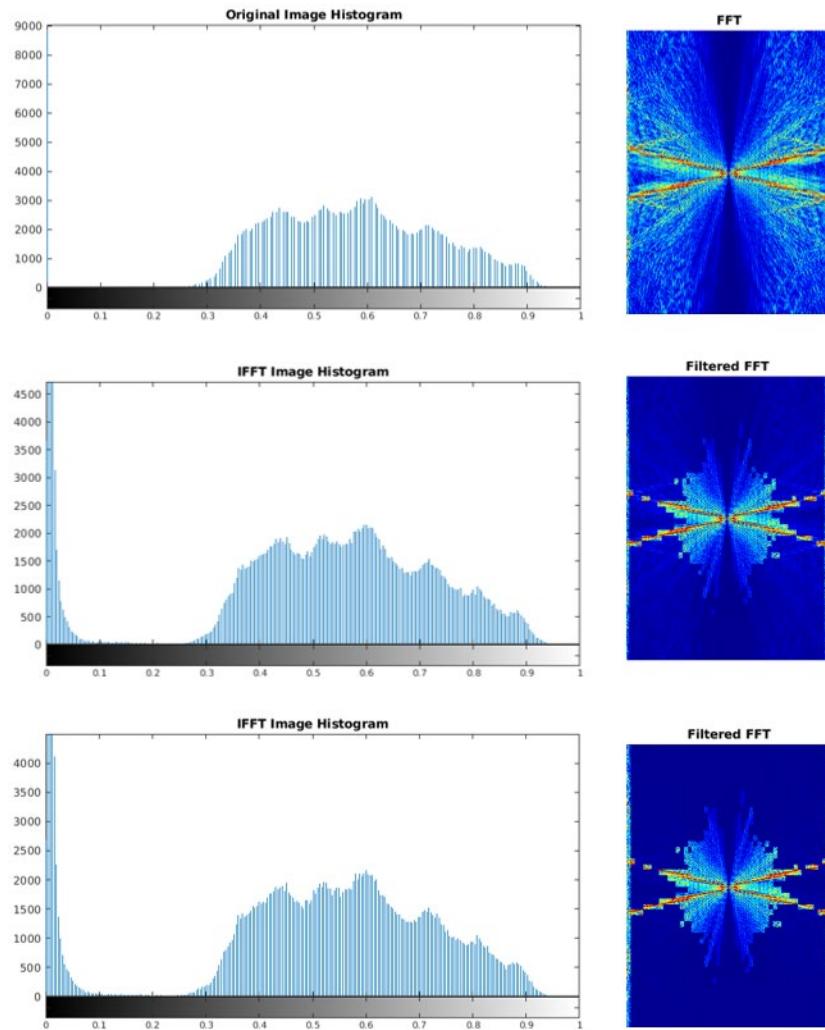


Figure 8.2: Histogram and FFT of Images (top) Original Image, (middle) Filtered Image $\alpha=0.9, \beta=0.3$, (right) Filtered Image $\alpha=0.9, \beta=0.1$.

APPENDICES

Appendix A

FERET DATABASE Subjects

The research in this paper use the FERET database of facial images collected under the FERET program, sponsored by the DOD Counterdrug Technology Development Program Office.

Images used where from the following subjects:

300 FA, 300 FB, 301 FA, 301 FB, 302 FA, 302 FB, 303 FA, 303 FB, 304 FA, 304 FB,
305 FA, 305 FB, 306 FA, 306 FB, 307 FA, 307 FB, 308 FA, 308 FB, 309 FA, 309 FB,
310 FA, 310 FB, 311 FA, 311 FB, 312 FA, 312 FB, 313 FA, 313 FB, 314 FA, 314 FB,
315 FA, 315 FB, 316 FA, 316 FB

Appendix B

List of Equations

Equation 2.1: Grigoryan-Agaian Enhancement Measurement	5
Equation 2.2: Estimated Enhancement Measurement	5
Equation 2.3: Estimation Enhancement Entropy Measurement	5
Equation 2.4: Agaian-Michelson Enhancement Measure.....	5
Equation 2.5: Agaian-Michelson Entropy Enhancement Measure.....	6
Equation 2.6: Parametrized Agaian-Michelson Entropy Enhancement Measure.....	6
Equation 2.7: Panetta-Agaian Enhancement Measure.....	6
Equation 2.8: Mean-Square-Root Measure of Enhancement Measure.....	6
Equation 2.9: Translation Block Filter.....	7
Equation 2.10: Translation Block Filter Kernel.....	7
Equation 2.11: Gradient Based Histogram Equalization	11

Equation 2.12: Convolution of the Original Image	12
Equation 2.13: Kernels used for gradient convolution	12
Equation 3.1: Quantum State Equation for a Qubit	24
Equation 3.2: MCRQI Equation	25
Equation 3.3: Quantum Encoding of Color Data.....	25
Equation 3.4: Polynomial Preparation Theorem.....	25
Equation 3.5: Hadamard Gate and Rotation Function of the PPT	26
Equation 3.6: Controlled Rotation Gates	26
Equation 3.7: NASS Color Angle.....	29
Equation 3.8: NASS Representation.....	30
Equation 3.9: 4x4 NASS Representation.....	30
Equation 3.10: Full 4x4 NASS Representation	30
Equation 4.1: Binarization method for LBP	32
Equation 4.2: The value of the Binary Rotation of neighboring pixels.....	32
Equation 4.3: The value of the center pixel of the Binary Rotation.	32
Equation 4.4: Binarization process of pixel difference.....	33
Equation 4.5: Piecewise operation of resulting difference.	33
Equation 4.6: LBP representation of the image.....	33
Equation 5.1: Correlation Coefficient.....	35
Equation 5.2: Kernel distribution.....	36
Equation 5.3: Smoothing Kernel.....	36
Equation 5.4: Bandwidth Value	36
Equation 5.5: Error-Difference Estimation.....	36

Equation 8.1: RGB Components of NASSTC.....	42
Equation 8.2: Calculate the average intensity.....	43
Equation 8.3: Magnitude of each color component.....	43
Equation 8.4: Color Components.....	43
Equation 8.5: NASSTC equation.....	43

Appendix C

The following section demonstrates the code used to develop the facial recognition process.

Spatial_Filter.m

```
im = cImage(22);
%% Regular image
im1 = mat2gray(imresize(im,[350 250]));
[row col rgb] = size(im);

im_r = im1(:,:,1); im_g = im1(:,:,2); im_b = im1(:,:,3);

%% Fourier Transforms
% Fast Fourier Transform 2-D
f.a = fft2(im_r); fc.a = fftshift(f.a); fd.a = abs(fc.a); fp.a = angle(fc.a); fe.a =
normalize(fd.a,'range',[0 1]);
f.b = fft2(im_g); fc.b = fftshift(f.b); fd.b = abs(fc.b); fp.b = angle(fc.b); fe.b =
normalize(fd.b,'range',[0 1]);
f.c = fft2(im_b); fc.c = fftshift(f.c); fd.c = abs(fc.c); fp.c = angle(fc.c); fe.c =
normalize(fd.c,'range',[0 1]);

%% Spatial Filtering (red)
k1 = 50;
k2 = 50;
alpha = 0.5;
beta = 0.8;

sf.a = SFilter(fd.a,alpha,beta,k1,k2);
sf.b = SFilter(fd.b,alpha,beta,k1,k2);
sf.c = SFilter(fd.c,alpha,beta,k1,k2);

%% Perform Operation leading to IFT
fm.a = zeros(row,col); fm.b = zeros(row,col); fm.c = zeros(row,col);
for i = 1:row
    for j = 1:col
        fm.a(i,j) = sf.a(i,j) * exp(fp.a(i,j)*1i);
        fm.b(i,j) = sf.b(i,j) * exp(fp.b(i,j)*1i);
        fm.c(i,j) = sf.c(i,j) * exp(fp.c(i,j)*1i);
    end
end

fsh.a = ifft2(ifftshift(fm.a)); fsh.b = ifft2(ifftshift(fm.b)); fsh.c = ifft2(ifftshift(fm.c));

f_img(:,:,1) = abs(fsh.a); f_img(:,:,2) = abs(fsh.b); f_img(:,:,3) = abs(fsh.c);
```

SFilter2.m

```
function qdft = SFilter(img,a,b,k1,k2)

%img2 = imresize(img,[350 250]); %img2 = normalize(img2,'range',[0 1]);
img2 = img;
[M N] = size(img2);
mean_img = mean(mean(img2));

L1 = M/k1; % width 10
L2 = N/k2; % height 10

%% Blocks
m = 1;
e = exp(1)-1;
%% Operation in Transform (Fourier) Domain using kernel
for k = 1:k1 % 35
    for l = 1:k2 % 25
        r = (1:L1)+(L1*(k-1));
        c = (1:L2)+(L2*(l-1));
        mean_val = mean(mean(img2(r,c)));
        if mean_val < (mean_img*a) % less than 0.2~
            img3(r,c) = img2(r,c) * ((exp(b)-1)/e); % Kernal operation
        else
            img3(r,c) = img2(r,c);
        end
    end
end

qdft = img3;
```

GB_HE.m

Gradient Based Histeq

```
function Y = GB_HE(im,a,option)
```

```
X = im;
```

```
G(:,:,1) = (1/4)*...
[1 0 1;
 0 -4 0;
```

```

1  0  1];
G(:,:,2) = (1/4)*...
[0  1  0;
 1 -4  1;
 0  1  0];
G(:,:,3) = (1/8)*...
[1  1  1;
 1 -8  1;
 1  1  1];
G(:,:,4) = (1/8)*...
[-1 -1 -1;
 -1  8 -1;
 -1 -1 -1];

Xs = conv2(X,G(:,:,option)); [r,c] = size(Xs);
thresh = Xs > 0;
% Xs_ = immultiply(Xs,thresh)+.3; Xs_ = recrop3(Xs_);
Xr = double(imresize(X,[r,c]));
Xg = Xr-Xs;
Xg_ = histeq(Xg); Xg_ = recrop3(Xg_);
Y = (a*Xg_) + Xs; Y = recrop3(Y);

```

cImage.m

Acquire image from database

```

function img = cImage(int)

fp = '/home/MATLAB/FERET/';
I = CF_extract2(fp); clear fp;
img = imresize(I(:,:,:,int),[350 250]);

end

```

pAMEE.m

Parameterized Agaian-Michelson Entropy Enhancement Measure

```

Function pamee = pAMEE(img,a,k1,k2)
%img = cImage(10);
img = imresize(img,[350 250]); img2 = normalize(img,'range',[0 1]);

[N1 N2] = size(img2);
%k1 = 35; % y-segments
%k2 = 25; % x-segments

L1 = N1/k1; % width 10
L2 = N2/k2; % height 10

%%
m = 1;

for k = 1:k1 % 35
    for l = 1:k2 % 25
        max_val(m) = max(max(img2((1:L1)+(L1*(k-1)),(1:L2)+(L2*(l-1)))));
        min_val(m) = min(min(img2((1:L1)+(L1*(k-1)),(1:L2)+(L2*(l-1)))));
        m = m + 1;
    end
end
m = m-1;
max_val = double(max_val);
min_val = double(min_val);

count = 1;
for t = 1:m
    if (max_val(t) ~= 0 && min_val(t) ~= 0)
        val(1,count) = max_val(t);
        val(2,count) = min_val(t);
        count = count + 1;
    end
end
v_len = length(val);
for h = 1:v_len
    summ(h) = a * (val(1,h) - val(2,h))/(val(1,h) + val(2,h))^a...
        * log((val(1,h) - val(2,h))/(val(1,h) + val(2,h)));
end

pamee = sum(summ)/(-k1*k2);

% 2nd Derivative, Panetta-Agaian Measure of Enhancement
function pame = PAME(img,k1,k2)

```

```
% error func introduced for invalid k1,k2
if mod(350,k1) ~= 0 || mod(250,k2) ~= 0
    error('Please choose a value k1 or k2 that is divisible by integer (x,y)[350,250]');
end
```

PAME.m

Parameterized Agaian-Michelson Enhancement Measure

```
%img = cImage(10);
img = imresize(img,[350 250]); img2 = normalize(img,'range',[0 1]);
%img2 = rgb2gray(img); img2 = mat2gray(img2);
```

```
[N1 N2] = size(img2);
%k1 = 35; % y-segments
%k2 = 25; % x-segments
```

```
L1 = N1/k1; % width 10
L2 = N2/k2; % height 10
```

```
%%
m = 1;
cen(1) = round(L1/2); cen(2) = round(L2/2);

for k = 1:k1 % 35
    for l = 1:k2 % 25
        max_val(m) = max(max(img2((1:L1)+(L1*(k-1)),(1:L2)+(L2*(l-1)))));
        min_val(m) = min(min(img2((1:L1)+(L1*(k-1)),(1:L2)+(L2*(l-1)))));
        M = img2((1:L1)+(L1*(k-1)),(1:L2)+(L2*(l-1)));
        cen_val(m) = M(cen(1),cen(2));
    end
    m = m + 1;
end
```

```
m = m-1;
max_val = double(max_val);
min_val = double(min_val);
cen_val = double(cen_val);
```

```
count = 1;
for t = 1:m
    if (max_val(t) ~= 0 && min_val(t) ~= 0 && cen_val(t) ~= 0)
        val(1,count) = max_val(t);
        val(2,count) = min_val(t);
    end
end
```

```

    val(3,count) = cen_val(t);
    count = count + 1;
end
v_len = length(val);
for h = 1:v_len
    summ(h) = 20 * log(abs(val(1,h) + val(3,h) - val(2,h))/(val(1,h) + val(3,h) + val(2,h)));
end

pame = sum(summ)/(-k1*k2);

```

AMEE.m

Agaian-Michelson Entropy Enhancement Measure

```

function amee = AMEE(img,k1,k2)
%img = cImage(10);
img = imresize(img,[350 250]); img2 = normalize(img,'range',[0 1]);
% img2 = rgb2gray(img); img2 = mat2gray(img2);

```

```

[N1 N2] = size(img2);
%k1 = 35; % y-segments
%k2 = 25; % x-segments

```

```

L1 = N1/k1; % width 10
L2 = N2/k2; % height 10

```

```

%%%
m = 1;

```

```

for k = 1:k1 % 35
    for l = 1:k2 % 25
        max_val(m) = max(max(img2((1:L1)+(L1*(k-1)),(1:L2)+(L2*(l-1)))));
        min_val(m) = min(min(img2((1:L1)+(L1*(k-1)),(1:L2)+(L2*(l-1)))));
        m = m + 1;
    end
end
m = m-1;
max_val = double(max_val);
min_val = double(min_val);

count = 1;
for t = 1:m
    if (max_val(t) ~= 0 && min_val(t) ~= 0)

```

```

    val(1,count) = max_val(t);
    val(2,count) = min_val(t);
    count = count + 1;
end

end
v_len = length(val);
for h = 1:v_len
    summ(h) = (val(1,h) - val(2,h))/(val(1,h) + val(2,h))...
        * log((val(1,h) - val(2,h))/(val(1,h) + val(2,h)));
end

amee = sum(summ)/(-k1*k2);

```

AME.m

Agaian-Michelson Enhancement Measure

```

function ame = AME(img,k1,k2)
%img = cImage(10);
img = imresize(img,[350 250]); img2 = normalize(img,'range',[0 1]);
% img2 = rgb2gray(img); img2 = mat2gray(img2);

```

```

[N1 N2] = size(img2);
%k1 = 35; % y-segments
%k2 = 25; % x-segments

```

```

L1 = N1/k1; % width 10
L2 = N2/k2; % height 10

```

```

%%%
m = 1;

```

```

for k = 1:k1 % 35
    for l = 1:k2 % 25
        max_val(m) = max(max(img2((1:L1)+(L1*(k-1)),(1:L2)+(L2*(l-1))))));
        min_val(m) = min(min(img2((1:L1)+(L1*(k-1)),(1:L2)+(L2*(l-1))))));
        m = m + 1;
    end
end
m = m-1;
max_val = double(max_val);
min_val = double(min_val);

```

```

count = 1;
for t = 1:m
%    if (min_val(t) == 0)
%        min_val(t) = .01;
%    end
    if (max_val(t) ~= 0 && min_val(t) ~= 0)
        val(1,count) = max_val(t);
        val(2,count) = min_val(t);
        count = count + 1;
    end
end
v_len = length(val);
for h = 1:v_len
    summ(h) = 20 * log((val(1,h) - val(2,h))/(val(1,h) + val(2,h)));
end

ame = sum(summ)/(-k1*k2);

```

EMEE.m

Enhancement Measurement Estimation Entropy

```

function emee = EMEE(img,k1,k2)
%img = cImage(10);
img = imresize(img,[350 250]); img2 = normalize(img,'range',[0 1]);
% img2 = rgb2gray(img); img2 = mat2gray(img2);

[N1 N2] = size(img2);
%k1 = 35; % y-segments
%k2 = 25; % x-segments

L1 = N1/k1; % width 10
L2 = N2/k2; % height 10

%%
m = 1;

for k = 1:k1 % 35
    for l = 1:k2 % 25
        max_val(m) = max(max(img2((1:L1)+(L1*(k-1)),(1:L2)+(L2*(l-1))))));
        min_val(m) = min(min(img2((1:L1)+(L1*(k-1)),(1:L2)+(L2*(l-1))))));
        m = m + 1;
    end
end
m = m-1;

```

```

max_val = double(max_val);
min_val = double(min_val);

count = 1;
for t = 1:m
    if (max_val(t) ~= 0 && min_val(t) ~= 0)
        val(1,count) = max_val(t);
        val(2,count) = min_val(t);
        count = count + 1;
    end

end
v_len = length(val);
for h = 1:v_len
    summ(h) = (val(1,h)/val(2,h)) * log(val(1,h)/val(2,h));
end

eme = sum(summ)/(k1*k2);

```

EME.m

Enhancement Measurement Estimation

```

function eme = EME(img,k1,k2)
%img = cImage(10);
img2 = imresize(img,[350 250]); img2 = normalize(img2,'range',[0 1]);
%img2 = rgb2gray(img); img2 = mat2gray(img2);

[N1 N2] = size(img2);
%k1 = 35; % y-segments
%k2 = 25; % x-segments

L1 = N1/k1; % width 10
L2 = N2/k2; % height 10

%%
m = 1;

for k = 1:k1 % 35
    for l = 1:k2 % 25
        max_val(m) = max(max(img2((1:L1)+(L1*(k-1)),(1:L2)+(L2*(l-1)))));
        min_val(m) = min(min(img2((1:L1)+(L1*(k-1)),(1:L2)+(L2*(l-1)))));
        m = m + 1;
    end
end

```

```

m = m-1;
max_val = double(max_val);
min_val = double(min_val);

count = 1;
for t = 1:m
    if (max_val(t) ~= 0 && min_val(t) ~= 0)
        val(1,count) = max_val(t);
        val(2,count) = min_val(t);
        count = count + 1;
    end
end

v_len = length(val);
for h = 1:v_len
    summ(h) = 20 * log(val(1,h)/val(2,h));
end

eme = sum(summ)/(k1*k2);

```

MSRE.m

Mean-Square-Root measure of enhancement

```

function msrme = MSRME(img,k1,k2)
% error func introduced for invalid k1,k2
if mod(350,k1) ~= 0 || mod(250,k2) ~= 0
    error('Please choose a value k1 or k2 that is divisible by integer (x,y)[350,250]');
end

%img = cImage(10);
img = imresize(img,[350 250]); img2 = normalize(img,'range',[0 1]);
%img2 = rgb2gray(img); img2 = mat2gray(img2);

[N1 N2] = size(img2);
%k1 = 35; % y-segments
%k2 = 25; % x-segments

L1 = N1/k1; % width 10
L2 = N2/k2; % height 10

%%
m = 1;

```

```

cen(1) = round(L1/2); cen(2) = round(L2/2);

for k = 1:k1 % 35
    for l = 1:k2 % 25
        mean_val(m) = mean(mean(img2((1:L1)+(L1*(k-1)),(1:L2)+(L2*(l-1))))));
        M = img2((1:L1)+(L1*(k-1)),(1:L2)+(L2*(l-1)));
        cen_val(m) = M(cen(1),cen(2));

        m = m + 1;
    end
end
m = m-1;
mean_val = double(mean_val);
cen_val = double(cen_val);

count = 1;

for t = 1:m
    if (mean_val(t) ~= 0 && cen_val(t) ~= 0)
        val(1,count) = mean_val(t);
        val(2,count) = cen_val(t);
        count = count + 1;
    end
end

v_len = length(val);

for h = 1:v_len
    summ(h) = sqrt(abs(log(abs(val(2,h) - val(1,h))/(val(2,h) + val(1,h))))) ;
end

msrme = sum(summ)/(k1*k2);

```

quat_mo_1.m

Quaternion Image 2x2 GrRGB

```

function [imgrgb map_norm] = quat_mo_1(im)

img = im; %
[N, M, L] = size(img);
N = 4*floor(N/4); M=4*floor(M/4); % trim
img = imresize(img,[N,M]);

%% average grayscale image calculation

```

```

Gr = rgb2gray(img);
X(:,:,2:4) = img;
X(:,:,1) = Gr;
X = double(X);

% At this point there is a conversion of integers (0-1024)
% for color map encoding.
%% Custom Colormap
map = zeros(256*4,3);
V = [0:255]';
BLK = zeros(256,1)+255;
Ze = zeros(256,1);

map(1:256,1:3) = [V V V]; % Gr
map(257:512,:) = [V Ze Ze]; % R
map(513:768,:) = [Ze V Ze]; % G
map(769:1024,:) = [Ze Ze V]; % B
% Normalize 0:1
map_norm = normalize(map,'range',[0 1]);
% Adjust X MAT with new colormap
Y(:,:,1) = X(:,:,1)+1;
Y(:,:,2) = X(:,:,2)+256+1;
Y(:,:,3) = X(:,:,3)+256*2+1;
Y(:,:,4) = X(:,:,4)+256*3+1;

%% Creating and assigning Q
Q = zeros(N,M);

for i = 1:N
    for j = 1:M
        if mod(i+1,2) == 0      % odd 1,3,5...
            Q(i,j) = Y(i,j,mod(j+1,2)+1); % Gray or Red
        elseif mod(i+1,2) == 1   % even 2,4,6...
            Q(i,j) = Y(i,j,mod(j+1,2)+3); % Green or Blue
        end
    end
end
imgrgb = ind2rgb(Q,map_norm);

```

quat_mo_1Y.m

Quaternion Image 2x2 RGBY

```
%clc;clear;
```

```

function [imgrgb map_norm] = quat_mo_1Y(im)

img = im; % imread(im);
[N, M, L] = size(img);
N = 4*floor(N/4); M=4*floor(M/4); % trim
img = imresize(img,[N,M]);

%% average grayscale image calculation
Gr = rgb2gray(img);
X(:,:,2:4) = img;
X(:,:,1) = Gr;
X = double(X);

% At this point there is a conversion of integers (0-1024)
% for color map encoding.
%% Custom Colormap
map = zeros(256*4,3);
V = [0:255]';
BLK = zeros(256,1)+255;
Ze = zeros(256,1);

map(1:256,1:3) = [V Ze Ze];
map(257:512,:) = [Ze V Ze];
map(513:768,:) = [Ze Ze V];
map(769:1024,:) = [V V Ze];
% Normalize 0:1
map_norm = normalize(map,'range',[0 1]);
% Adjust X MAT with new colormap
Y(:,:,:,1) = X(:,:,:,1)+1;
Y(:,:,:,2) = X(:,:,:,2)+256+1;
Y(:,:,:,3) = X(:,:,:,3)+256*2+1;
Y(:,:,:,4) = X(:,:,:,4)+256*3+1;

%% Creating and assigning Q
Q = zeros(N,M);

for i = 1:N
    for j = 1:M
        if mod(i+1,2) == 0      % odd 1,3,5...
            Q(i,j) = Y(i,j,mod(j+1,2)+1); % Gray or Red
        elseif mod(i+1,2) == 1   % even 2,4,6...
            Q(i,j) = Y(i,j,mod(j+1,2)+3); % Green or Blue
        end
    end
end

```

```
imgrgb = ind2rgb(Q,map_norm);
```

quat_mo_2CMY.m

Quaternion Image 3x3 CMY

```
function [imgrgb map_norm] = quat_mo_2CMY(im)
```

```
X = im;
[N M L] = size(X);
N = 3 * floor(N/3); M = 3 * floor(M/3);
X = imresize(X,[N M]);
X = double(X);
%% Create Colormap
map = zeros(256*3,3);
V = [0:255]';
Ze = zeros(256,1);

map(1:256,:) = [V Ze Ze];
map(257:512,:) = [Ze V Ze];
map(513:768,:) = [Ze Ze V];
% Normalize 0:1
map_norm = normalize(map,'range',[0 1]);
% Adjust X MAT with new colormap
Y(:,:,1) = X(:,:,1)+1;
Y(:,:,2) = X(:,:,2)+256+1;
Y(:,:,3) = X(:,:,3)+(256*2)+1;
```

```
%% Calculate the model of the image in the quaternion space
```

```
%% RGB
for i = 1:3:N
    for j = 1:3:M
        Q(i,j) = Y(i,j,1);
        Q(i,j+1) = Y(i,j+1,2);
        Q(i,j+2) = Y(i,j+2,3);
    end
end
```

```
%% BRG
for i = 2:3:N
    for j = 1:3:M
        Q(i,j) = Y(i,j,3);
        Q(i,j+1) = Y(i,j+1,1);
        Q(i,j+2) = Y(i,j+2,2);
```

```

    end
end

%% GBR
for i = 3:3:N
    for j = 1:3:M
        Q(i,j) = Y(i,j,2);
        Q(i,j+1) = Y(i,j+1,3);
        Q(i,j+2) = Y(i,j+2,1);
    end
end

imgrgb = ind2rgb(Q,map_norm);

```

quat_mo_2.m

Quaternion Image 3x3 RGB

```

X = im; %
[N M L] = size(X);
N = 3 * floor(N/3); M = 3 * floor(M/3);
X = imresize(X,[N M]);
X = double(X);
%% Create Colormap
map = zeros(256*3,3);
V = [0:255];
Ze = zeros(256,1);

map(1:256,:) = [Ze V V];
map(257:512,:) = [V Ze V];
map(513:768,:) = [V V Ze];
% Normalize 0:1
map_norm = normalize(map,'range',[0 1]);
% Adjust X MAT with new colormap
Y(:,:,1) = X(:,:,1)+1;
Y(:,:,2) = X(:,:,2)+256+1;
Y(:,:,3) = X(:,:,3)+(256*2)+1;

```

```

%% RGB
for i = 1:3:N
    for j = 1:3:M
        Q(i,j) = Y(i,j,1);
        Q(i,j+1) = Y(i,j+1,2);
        Q(i,j+2) = Y(i,j+2,3);
    end
end

```

```

    end
end

%% BRG
for i = 2:3:N
    for j = 1:3:M
        Q(i,j) = Y(i,j,3);
        Q(i,j+1) = Y(i,j+1,1);
        Q(i,j+2) = Y(i,j+2,2);
    end
end

%% GBR
for i = 3:3:N
    for j = 1:3:M
        Q(i,j) = Y(i,j,2);
        Q(i,j+1) = Y(i,j+1,3);
        Q(i,j+2) = Y(i,j+2,1);
    end
end

%% Display img
% colormap(map_norm);

imgrgb = ind2rgb(Q,map_norm);

```

MCRQI.py

Multi-Channel Representation Quantum Image

```

import qiskit
from qiskit import(QuantumCircuit,
QuantumRegister,ClassicalRegister,execute,Aer)
from qiskit.visualization import plot_histogram
import matplotlib
import numpy as np
from numpy import (zeros,ones)
import math as mp
import cv2
import csv
from csv import writer
import timeit

```

```
# import image for MCRQI transform
```

```

path = r'/home-new/bqc058/Anaconda/Jupyter/alaska_ice_cavern.jpg'
img = cv2.imread(path)/255

# image size is 256
n = 3
N = np.power(2,(2*n))
K = int(np.sqrt(N))
Q = (2*n)+3

theta = np.ndarray([K,K,4])
theta[0:K,0:K,3] = np.ones([K,K],'float64')
theta[0:K,0:K,3] = theta[0:K,0:K,3] * (mp.pi/2)

for k in range(3):
    for i in range(K):
        for j in range(K):
            if k == 0:
                tr = (mp.pi/2) * img[i][j][k]
                theta[i][j][k] = tr
            elif k == 1:
                tg = (mp.pi/2) * img[i][j][k]
                theta[i][j][k] = tg
            elif k == 2:
                tb = (mp.pi/2) * img[i][j][k]
                theta[i][j][k] = tb

#####
# Use Aer's qasm_simulator
simulator = Aer.get_backend('qasm_simulator')
# Create a Quantum Circuit for Color Quantum States (c) and Positional Quantum States (i)
qr_c = QuantumRegister(3,'c')
qr_x = QuantumRegister(n,'x')
qr_y = QuantumRegister(n,'y')
qr_a = QuantumRegister(16,'a')
# perhaps to fit instructions on circuit*** Then why allocate actual Qubits?
cr_m = ClassicalRegister(Q,'m')

qc = QuantumCircuit(qr_c,qr_x,qr_y,qr_a)

#####
l = np.zeros(K)
lm = np.zeros(K)
T = np.zeros(K)
d = [0]

```

```

start = timeit.timeit()

for j in range(1,K):
    l[j] = mp.log2(j)
    lm = (l%2)%1

## Y-Axis Rows
for j in range(K):
    if j == 0:
        a = 0
    else:
        Ixor = j ^ j-1
        Ib = format(Ixor,'b')
        Ibin = [int(j) for j in Ib]
        for index, item in enumerate(Ibin):
            if item == 1:
                b.append(index)
        for o in range(len(b)):
            qc.x(qr_y[b[o]])

## X-Axis columns
for i in range(K):
    if i == 0:
        a = 0
    else:
        Txor = i ^ (i-1)
        Tb = format(Txor,'b')
        Tbin = [int(i) for i in Tb]
        for index, item in enumerate(Tbin):
            if item == 1:
                d.append(index)
        for p in range(len(d)):
            qc.x(qr_x[d[p]])

# perform rotation
qc.mcry((2*theta[j,i,0])/1,qr_c[0:2] + qr_x[0:8] + qr_y[0:8],qr_c[2],qr_a)
qc.mcry((2*theta[j,i,1])/1,qr_c[0:2] + qr_x[0:8] + qr_y[0:8],qr_c[2],qr_a)
qc.mcry((2*theta[j,i,2])/1,qr_c[0:2] + qr_x[0:8] + qr_y[0:8],qr_c[2],qr_a)
qc.mcry((2*theta[j,i,3])/1,qr_c[0:2] + qr_x[0:8] + qr_y[0:8],qr_c[2],qr_a)

Tbin = 0
d = [0]

qc.measure_all()
# zero out the x qubits

```

```

for r in range(n):
    qc.x(qr_x[r])

# perform rotation
Ibin = 0
b = [0]

qc.measure(range(0,Q),range(0,Q))
end = timeit.timeit()
print('time Ry/NOT:',end-start)
# Map the quantum measurement to the classical bits

start = timeit.timeit()
# Execute the circuit on the qasm simulator
sh = 100
job = execute(qc, simulator, shots=sh)

# Grab results from the job
result = job.result()

# Returns counts
counts = result.get_counts(qc)
print("For ", sh, " shots the total count is:\n",counts)
end = timeit.timeit()
print('time counts:',end-start)
# Draw the circuit
qc.draw()
w = csv.writer(open('counts2.csv', 'w'))
for key, val in counts.items():
    w.writerow([key,val])

debug = 0 % debug here
# Plot a histogram
plot_histogram(counts)

```

NASS.py

```

%matplotlib inline
import numpy as np
from qiskit import QuantumCircuit, QuantumRegister, ClassicalRegister, execute, Aer
from qiskit.visualization import *
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from skimage.color import rgb2gray
import timeit

```

```

path = r'/home/pathfinder/python/images/color-box.png';
img = mpimg.imread(path);
img = rgb2gray(img);

n = int(np.log2(np.size(img))); # n = 18-4
print('n: ', n)
print('size: ', np.size(img), ', width: ', len(img))
plt.imshow(img, cmap='gray');
plt.title('original image');

# we ignore (M-1) since the image is normalized from 0 to 1
a = np.divide(np.multiply(img, np.pi), 2);
aj = np.sqrt(np.sum(np.power(a, 2)));
theta = np.divide(a, aj);

# set up quantum registers
e = int(n/2);
qr = QuantumRegister(n, 'q');
#anc = QuantumRegister(1, 'ancilla');
cr = ClassicalRegister(n+1, 'c');
# set up quantum circuit
qc = QuantumCircuit(qr, cr);

def Qbit_target(control, target):
    arr = []
    if target != 0:
        arr.extend(qr[:target])
        arr.extend(qr[target+1:])
    return arr;

start = timeit.timeit()

## Y-Axis Rows
for j in range(n):
    if j != 0:
        Jxor = j ^ j-1
        Jb = format(Jxor, 'b')
        Jbin = [int(j) for j in Jb]
        for index, item in enumerate(Jbin):
            if item == 1:
                qc.x(qr[index+e])

## X-Axis columns

```

```

for i in range(n):
    if i != 0:
        Ixor = i ^ (i-1)
        Ib = format(Ixor,'b')
        Ibin = [int(i) for i in Ib]
        for index, item in enumerate(Ibin):
            if item == 1:
                qc.x(qr[index])

    # perform rotation
    for q in range(n):
        qc.mcrx(theta[j][i], Qbit_target(qr[:,q]), qr[q]);

    # zero out the x qubits
    if (j == n and i == n):
        break;
    else:
        for r in range(e):
            qc.x(qr[r])

end = timeit.timeit()
print('time:',end-start)

qc.measure_all()
draw_path = r'/home/pathfinder/python/images/q_circuit1.png';

# Use Aer's qasm_simulator
simulator = Aer.get_backend('qasm_simulator')
# Execute the circuit on the qasm simulator
sh = 1000
job = execute(qc, simulator, shots=sh)

# Grab results from the job
result = job.result()

# Returns counts
counts = result.get_counts(qc)

print("For ", sh, " shots the total count for 0000 and 1111 are:\n",counts)

# Draw Quantum Circuit
qc.draw(scale=0.5,filename=draw_path);

# Plot a histogram
plot_histogram(counts, title='Probability Histogram')

```

Appendix D

Enhancement Measurements starting from image 300A-316B of the Ferret Database

eme	emee	ame	amee	pamee $\alpha=0.8$	pamee $\alpha=0.6$	pamee $\alpha=0.4$	pamee $\alpha=0.2$	pame	msrme	std2
6.43	1.01	25.62	0.21	0.17	0.13	0.09	0.05	11.01	1.56	0.15866688
5.68	0.68	27.7	0.2	0.16	0.13	0.09	0.05	11.5	1.58	0.17419939
4.93	0.67	30.65	0.18	0.15	0.12	0.09	0.05	12.08	1.64	0.23812638
5.05	0.86	30.46	0.18	0.15	0.12	0.09	0.05	12.07	1.62	0.23817944
4.17	18.05	33.83	0.16	0.14	0.11	0.08	0.05	12.72	1.69	0.26467211
4.18	0.73	32.66	0.17	0.15	0.12	0.09	0.05	12.53	1.65	0.26511992
4.65	7.9	31.87	0.17	0.15	0.12	0.09	0.05	12.36	1.63	0.25723553
4.69	0.4	30.78	0.18	0.16	0.13	0.09	0.05	12.15	1.65	0.22077434
3.95	0.44	32.83	0.17	0.15	0.12	0.09	0.05	12.64	1.67	0.27082462
4.21	0.86	32.32	0.17	0.15	0.12	0.09	0.05	12.52	1.66	0.27594245
4.83	4.23	30.6	0.18	0.16	0.13	0.09	0.05	12.28	1.63	0.22496148
5.71	1.45	28.1	0.19	0.16	0.13	0.09	0.05	11.55	1.61	0.16548848
3.77	5.01	34.68	0.16	0.14	0.11	0.08	0.04	12.9	1.68	0.29748958
3.98	0.54	32.82	0.17	0.15	0.12	0.09	0.05	12.61	1.65	0.29349716
4.48	0.59	31.09	0.18	0.15	0.12	0.09	0.05	12.29	1.62	0.24647846
4.1	1.1	32.89	0.17	0.15	0.12	0.09	0.05	12.72	1.64	0.24374404
4.41	0.48	31.56	0.18	0.15	0.12	0.09	0.05	12.35	1.67	0.22273755
3.65	0.38	34.79	0.16	0.14	0.11	0.08	0.04	12.99	1.7	0.28424596
4.65	0.56	31.36	0.18	0.15	0.12	0.09	0.05	12.18	1.66	0.23258349
3.97	0.71	33.47	0.17	0.14	0.12	0.08	0.05	12.74	1.68	0.28249302
3.66	0.36	34.33	0.16	0.14	0.11	0.08	0.04	12.88	1.7	0.26918719
4.47	0.63	30.61	0.18	0.16	0.12	0.09	0.05	12.28	1.63	0.26906875
4.28	0.51	32.74	0.17	0.15	0.12	0.09	0.05	12.58	1.65	0.2265224
3.99	0.4	32.91	0.17	0.15	0.12	0.09	0.05	12.7	1.67	0.24340223
4.39	0.67	31.77	0.17	0.15	0.12	0.09	0.05	12.38	1.66	0.25989142
5.61	0.56	27.61	0.2	0.17	0.14	0.1	0.05	11.54	1.62	0.22734479
4.13	0.8	33.31	0.17	0.14	0.12	0.08	0.05	12.74	1.68	0.25182124
4.72	2.11	30.74	0.18	0.15	0.12	0.09	0.05	12.23	1.63	0.24922144
4.27	0.36	31.83	0.18	0.15	0.12	0.09	0.05	12.48	1.69	0.2472138
4.91	0.62	30.43	0.18	0.16	0.13	0.09	0.05	12.09	1.64	0.25689347
3.71	0.33	33.41	0.17	0.15	0.12	0.09	0.05	12.81	1.65	0.25300359
4.25	0.56	31.92	0.17	0.15	0.12	0.09	0.05	12.46	1.67	0.23907763

Table A.1: Original Image Enhancement Measurements

eme	emee	ame	amee	pamee $\alpha=0.8$	pamee $\alpha=0.6$	pamee $\alpha=0.4$	pamee $\alpha=0.2$	pame	msrme	std2
7.3	1.51	24.83	0.21	0.17	0.14	0.1	0.05	10.73	1.55	0.29716191
6.9	0.95	25.86	0.2	0.17	0.13	0.09	0.05	10.93	1.58	0.2996729
6.05	1.01	29.04	0.19	0.16	0.13	0.09	0.05	11.51	1.61	0.30572963
6.04	1.64	29.42	0.18	0.15	0.12	0.09	0.05	11.58	1.61	0.3070082
5.7	1.13	29.27	0.19	0.16	0.13	0.09	0.05	11.7	1.64	0.30432446
6.04	0.88	27.98	0.19	0.16	0.13	0.09	0.05	11.38	1.61	0.30388989
5.99	76.02	29.25	0.19	0.16	0.12	0.09	0.05	11.64	1.61	0.31172105
5.84	0.6	29.05	0.19	0.16	0.13	0.09	0.05	11.54	1.62	0.31750658
5.72	0.75	28.24	0.19	0.16	0.13	0.09	0.05	11.58	1.61	0.31201584
5.68	2	28.78	0.19	0.16	0.13	0.09	0.05	11.66	1.62	0.31544376
5.92	1.18	28.38	0.19	0.16	0.13	0.09	0.05	11.61	1.62	0.30540103
6.34	1.88	27.28	0.2	0.17	0.13	0.09	0.05	11.27	1.6	0.30540699
5.87	0.7	28	0.19	0.16	0.13	0.09	0.05	11.47	1.6	0.31572591
6.15	2.02	27.21	0.2	0.17	0.13	0.09	0.05	11.35	1.59	0.31772859
6.13	0.84	27.74	0.19	0.16	0.13	0.09	0.05	11.34	1.59	0.30635486
5.61	0.83	29.17	0.19	0.16	0.13	0.09	0.05	11.73	1.62	0.30409197
6.76	0.94	26.52	0.2	0.17	0.13	0.09	0.05	11.06	1.61	0.3132983
6.06	0.84	28.19	0.19	0.16	0.13	0.09	0.05	11.52	1.63	0.3113167
5.93	1.38	29.04	0.18	0.16	0.12	0.09	0.05	11.59	1.62	0.31226863
5.2	0.53	30.31	0.18	0.15	0.12	0.09	0.05	11.96	1.65	0.31483666
5.34	0.61	29.66	0.18	0.16	0.12	0.09	0.05	11.85	1.64	0.31001929
6.04	0.76	27.5	0.2	0.16	0.13	0.09	0.05	11.33	1.6	0.3185698
6.14	1.14	29.21	0.19	0.16	0.12	0.09	0.05	11.54	1.62	0.3130253
6.38	0.84	27.97	0.19	0.16	0.13	0.09	0.05	11.28	1.61	0.31350649
5.86	1.79	28.88	0.19	0.16	0.13	0.09	0.05	11.62	1.62	0.30729767
6.41	0.71	26.58	0.2	0.17	0.14	0.1	0.05	11.15	1.6	0.30937945
5.39	0.82	30.3	0.18	0.15	0.12	0.09	0.05	11.95	1.64	0.29992295
6.24	21.46	28.21	0.19	0.16	0.13	0.09	0.05	11.45	1.61	0.30444656
5.18	0.48	30.33	0.19	0.16	0.12	0.09	0.05	11.93	1.66	0.31747407
6.15	0.8	28.21	0.19	0.16	0.13	0.09	0.05	11.41	1.62	0.31578252
5.46	0.56	28.86	0.19	0.16	0.13	0.09	0.05	11.71	1.6	0.30268543
5.97	0.83	28.33	0.19	0.16	0.13	0.09	0.05	11.55	1.63	0.30209483

Table A.2: Histeq Image Enhancement Measurement

eme	emee	ame	amee	pamee $\alpha=0.8$	pamee $\alpha=0.6$	pamee $\alpha=0.4$	pamee $\alpha=0.2$	pame	msrme	std2
23.99	2989.8	26.07	0.21	0.17	0.13	0.09	0.05	11.19	1.78	0.15776588
23.5	1536.66	28.24	0.2	0.16	0.13	0.09	0.05	11.68	1.79	0.17319297
23.38	4737.41	30.93	0.18	0.16	0.12	0.09	0.05	12.19	1.85	0.23728358
23.28	7137.4	30.69	0.18	0.15	0.12	0.09	0.05	12.16	1.83	0.23739938
21.95	1215.62	34.2	0.17	0.14	0.11	0.08	0.05	12.83	1.89	0.26348057
21.51	2068.19	32.89	0.17	0.15	0.12	0.09	0.05	12.63	1.86	0.26389734
21.55	1957.77	32.18	0.18	0.15	0.12	0.09	0.05	12.47	1.85	0.25632213
22.81	4541.79	31.03	0.19	0.16	0.13	0.09	0.05	12.25	1.87	0.21991337
20.33	849.15	33.19	0.17	0.15	0.12	0.09	0.05	12.76	1.87	0.26961246
21.53	2487.29	32.87	0.17	0.15	0.12	0.08	0.05	12.65	1.86	0.27475028
22.1	3846.89	30.87	0.18	0.16	0.13	0.09	0.05	12.38	1.83	0.22408096
23.98	6856.78	28.43	0.2	0.16	0.13	0.09	0.05	11.68	1.84	0.16480153
21.91	1978.7	34.89	0.16	0.14	0.11	0.08	0.04	12.98	1.89	0.29614943
21.9	2148.03	33.05	0.17	0.15	0.12	0.09	0.05	12.69	1.86	0.29210995
22.16	1387.97	31.68	0.18	0.15	0.12	0.09	0.05	12.47	1.84	0.24527536
21.98	1581.07	33.72	0.17	0.14	0.12	0.08	0.05	12.92	1.86	0.24260164
23.18	438236.4	31.94	0.18	0.15	0.12	0.09	0.05	12.46	1.88	0.22174233
22.06	2023.1	35.07	0.16	0.14	0.11	0.08	0.04	13.08	1.91	0.28295889
22.85	1092.51	31.73	0.18	0.15	0.12	0.09	0.05	12.29	1.87	0.2317099
21.98	8177.52	33.98	0.17	0.14	0.12	0.08	0.04	12.86	1.9	0.2814159
20.83	1490.26	34.86	0.16	0.14	0.11	0.08	0.04	13.03	1.91	0.26794636
22.73	2437.02	31.85	0.18	0.15	0.12	0.09	0.05	12.55	1.86	0.26771844
21.9	1492.43	33.05	0.17	0.15	0.12	0.09	0.05	12.69	1.87	0.22572873
21.32	6282.39	33.2	0.17	0.15	0.12	0.09	0.05	12.78	1.86	0.24248314
21.54	1601.43	32.01	0.18	0.15	0.12	0.09	0.05	12.48	1.86	0.25902389
23.11	1810.87	27.94	0.2	0.17	0.14	0.1	0.05	11.67	1.83	0.22666803
22.46	1999.18	33.59	0.17	0.14	0.12	0.08	0.05	12.84	1.89	0.25074423
21.66	2028.57	31.01	0.18	0.16	0.12	0.09	0.05	12.34	1.85	0.24804607
21.2	1329.11	32.04	0.18	0.15	0.12	0.09	0.05	12.58	1.9	0.24645013
22.53	1660.17	30.72	0.19	0.16	0.13	0.09	0.05	12.2	1.85	0.25600404
21.56	1497.83	34.09	0.17	0.14	0.12	0.09	0.05	12.97	1.87	0.25186949
21.74	2266.4	32.91	0.17	0.15	0.12	0.09	0.05	12.68	1.88	0.23799519

Table A.3: Spatial Filter Enhancement Measurement

eme	emee	ame	amee	pamee $\alpha=0.8$	pamee $\alpha=0.6$	pamee $\alpha=0.4$	pamee $\alpha=0.2$	pame	msrme	std2
19.7	7.3368E+16	19.4	0.28	0.21	0.16	0.11	0.05	9.83	1.7	0.18534204
19.47	1.0522E+17	21.29	0.27	0.21	0.15	0.1	0.05	10.45	1.74	0.18455563
14.91	6.5073E+15	26.95	0.26	0.19	0.15	0.1	0.05	12.01	1.83	0.18821685
17.87	4.1422E+16	27.19	0.25	0.19	0.14	0.1	0.05	11.96	1.84	0.1882602
15.54	5.4798E+16	27.24	0.25	0.19	0.14	0.1	0.05	12.01	1.86	0.18670004
18.37	1.1281E+17	26.47	0.25	0.19	0.14	0.1	0.05	11.81	1.83	0.1866394
15.51	5.9076E+16	28.85	0.25	0.19	0.14	0.09	0.05	12.46	1.85	0.18855798
16.86	1.0635E+17	29.64	0.24	0.18	0.14	0.09	0.05	12.58	1.9	0.18904637
16.36	2.1974E+16	26.76	0.26	0.2	0.15	0.1	0.05	12.07	1.84	0.18857375
18.3	1.1131E+17	27.88	0.25	0.19	0.14	0.1	0.05	12.36	1.86	0.18907178
14.94	35.75	26.31	0.25	0.19	0.14	0.1	0.05	11.8	1.84	0.18699357
18.9	6.4431E+16	24.64	0.26	0.2	0.15	0.1	0.05	11.27	1.82	0.18742462
15.7	2.2117E+16	27.2	0.25	0.19	0.14	0.1	0.05	12.08	1.85	0.18867089
16.52	7.9499E+16	27.09	0.26	0.19	0.14	0.1	0.05	12.22	1.85	0.18847601
18.04	1.7624E+16	26.25	0.25	0.19	0.14	0.1	0.05	11.75	1.85	0.18770257
16.45	1.5236E+16	26.95	0.25	0.19	0.14	0.1	0.05	12	1.84	0.18785478
17.33	2.6486E+16	25.79	0.26	0.2	0.15	0.1	0.05	11.72	1.85	0.18790686
16.27	1.1953E+17	27.12	0.26	0.19	0.14	0.1	0.05	12.13	1.87	0.1881954
16.2	1.745E+16	27.93	0.25	0.19	0.14	0.1	0.05	12.33	1.88	0.18886134
14.28	5.2766E+16	29.36	0.24	0.19	0.14	0.1	0.05	12.68	1.88	0.18902022
16.6	1.2222E+17	28.05	0.25	0.19	0.14	0.09	0.05	12.38	1.88	0.18816569
15.94	6.1872E+16	28.15	0.25	0.19	0.14	0.09	0.05	12.35	1.87	0.18840073
18.44	1.3953E+17	27.95	0.25	0.19	0.14	0.09	0.05	12.05	1.86	0.18901254
18.47	1.4452E+17	27.14	0.25	0.19	0.14	0.1	0.05	11.85	1.85	0.18851955
16.45	5.5457E+16	26.86	0.25	0.19	0.15	0.1	0.05	11.95	1.86	0.18958187
17.56	3.2064E+16	25.23	0.27	0.2	0.15	0.11	0.06	11.63	1.83	0.18960334
16.36	2.6717E+16	27.4	0.25	0.19	0.14	0.1	0.05	12.14	1.85	0.1875095
16.44	1.626E+16	27.07	0.25	0.19	0.14	0.1	0.05	11.98	1.85	0.18753229
14.95	1.8503E+16	29.78	0.25	0.19	0.14	0.09	0.05	12.71	1.9	0.19007977
15.02	5.2501E+16	28.02	0.25	0.19	0.14	0.1	0.05	12.25	1.89	0.18932957
14.3	75.45	26.14	0.26	0.2	0.15	0.1	0.05	11.9	1.83	0.18825979
15.84	1.3207E+16	25.74	0.26	0.2	0.15	0.1	0.05	11.69	1.85	0.18826708

Table A.4: GBHE (1) Enhancement Measurement

eme	emee	ame	amee	pamee $\alpha=0.8$	pamee $\alpha=0.6$	pamee $\alpha=0.4$	pamee $\alpha=0.2$	pame	msrme	std2
22.3	1.0623E+17	19.66	0.27	0.21	0.15	0.1	0.05	9.86	1.69	0.1869497
19.1	1.7171E+17	21.56	0.28	0.21	0.15	0.1	0.05	10.56	1.75	0.18638855
18.62	1.5832E+17	26.76	0.25	0.19	0.14	0.1	0.05	11.86	1.82	0.1894253
18.54	5.4977E+16	26.98	0.25	0.19	0.14	0.09	0.05	11.85	1.83	0.18949724
16.9	2.0944E+16	27.25	0.25	0.19	0.14	0.09	0.05	11.97	1.84	0.18798781
18.24	8.6776E+16	26.47	0.25	0.19	0.14	0.1	0.05	11.76	1.82	0.188034
15.8	1.2177E+16	28.65	0.25	0.19	0.14	0.09	0.05	12.35	1.84	0.18975225
17.15	5.1399E+16	29.45	0.24	0.18	0.14	0.09	0.05	12.47	1.88	0.19018582
16.12	1.7914E+16	26.88	0.26	0.19	0.14	0.1	0.05	12.1	1.82	0.18985139
16.81	7.2142E+16	27.78	0.25	0.19	0.14	0.1	0.05	12.31	1.84	0.19012976
15.3	28.24	26.47	0.25	0.19	0.14	0.1	0.05	11.79	1.81	0.18859889
18.92	1.0715E+17	24.87	0.26	0.19	0.14	0.1	0.05	11.3	1.82	0.18884858
17.33	4.5778E+16	27.63	0.25	0.19	0.14	0.1	0.05	12.19	1.83	0.18993904
17.09	5.7844E+16	27.25	0.25	0.19	0.14	0.1	0.05	12.24	1.84	0.18951471
17.07	4.5365E+16	26.09	0.25	0.19	0.14	0.1	0.05	11.67	1.84	0.18902581
21.32	2.0661E+17	26.97	0.25	0.19	0.14	0.1	0.05	11.99	1.82	0.18890884
17.87	7.3568E+16	26.02	0.26	0.19	0.14	0.1	0.05	11.76	1.84	0.18923302
15.65	9.1059E+16	27.25	0.25	0.19	0.14	0.1	0.05	12.13	1.86	0.18918467
17.01	7.4604E+16	27.81	0.25	0.19	0.14	0.1	0.05	12.21	1.86	0.19012092
17.41	1.5222E+17	29.28	0.24	0.18	0.14	0.09	0.05	12.62	1.86	0.19008121
18.67	1.2721E+17	28.03	0.25	0.19	0.14	0.09	0.05	12.33	1.86	0.18938759
16.2	4.7827E+16	27.84	0.25	0.19	0.14	0.09	0.05	12.24	1.83	0.1897667
17.45	2.7953E+16	28.16	0.24	0.18	0.14	0.09	0.05	12.06	1.85	0.18983209
18.55	7.5066E+16	27.21	0.25	0.19	0.14	0.1	0.05	11.83	1.86	0.18964864
19.5	1.2294E+17	26.82	0.25	0.19	0.14	0.1	0.05	11.89	1.83	0.19031369
18.52	3.201E+16	25.23	0.26	0.2	0.15	0.1	0.05	11.54	1.82	0.19050146
19.65	1.7938E+17	27.56	0.25	0.19	0.14	0.1	0.05	12.17	1.83	0.18882359
18.2	1.2713E+17	26.95	0.25	0.19	0.14	0.1	0.05	11.94	1.84	0.1889634
14.13	1.0273E+17	29.57	0.25	0.19	0.14	0.09	0.05	12.61	1.89	0.19053087
16.09	1.2341E+17	27.98	0.25	0.19	0.14	0.1	0.05	12.19	1.87	0.19040241
16.66	1.0934E+17	26.26	0.26	0.2	0.15	0.1	0.05	11.89	1.8	0.18956043
19.08	7.2017E+16	25.76	0.25	0.19	0.15	0.1	0.05	11.65	1.83	0.18937513

Table A.5: GBHE (2) Enhancement Measurement

eme	emee	ame	amee	pamee $\alpha=0.8$	pamee $\alpha=0.6$	pamee $\alpha=0.4$	pamee $\alpha=0.2$	pame	msrme	std2
19.74	5.1429E+16	19.48	0.28	0.21	0.16	0.11	0.05	9.84	1.7	0.18610959
19.28	3.4141E+16	21.51	0.28	0.21	0.15	0.1	0.05	10.54	1.76	0.18551462
15.1	5.8613E+15	26.9	0.26	0.19	0.14	0.1	0.05	11.95	1.82	0.18888879
19.99	1.2521E+17	27.1	0.25	0.19	0.14	0.1	0.05	11.91	1.84	0.18894779
20.7	2.6442E+17	27.28	0.25	0.19	0.14	0.1	0.05	12.02	1.85	0.18734322
15.81	6.0776E+15	26.5	0.25	0.19	0.14	0.1	0.05	11.8	1.82	0.18732366
15.15	1.1794E+16	28.83	0.25	0.19	0.14	0.09	0.05	12.45	1.84	0.18917215
19.02	1.5392E+17	29.63	0.24	0.18	0.14	0.09	0.05	12.55	1.9	0.18967107
17.67	6.1992E+16	26.89	0.26	0.19	0.14	0.1	0.05	12.1	1.83	0.1889453
13.76	44.24	27.88	0.25	0.19	0.14	0.1	0.05	12.36	1.85	0.18957425
16.1	1.0811E+16	26.4	0.25	0.19	0.14	0.1	0.05	11.8	1.82	0.18772529
18.89	1.3553E+17	24.76	0.26	0.19	0.15	0.1	0.05	11.28	1.82	0.18829872
15.83	1.9169E+16	27.45	0.25	0.19	0.14	0.1	0.05	12.13	1.83	0.18924124
17.76	7.257E+16	27.25	0.26	0.19	0.14	0.1	0.05	12.27	1.84	0.18892668
16.8	2.7749E+17	26.32	0.25	0.19	0.14	0.1	0.05	11.75	1.84	0.18829829
17.85	1.4308E+17	27.25	0.25	0.19	0.14	0.1	0.05	12.08	1.83	0.18828202
19.73	1.5368E+17	25.94	0.26	0.2	0.15	0.1	0.05	11.76	1.84	0.18855036
20.81	1.949E+17	27.31	0.25	0.19	0.14	0.1	0.05	12.18	1.87	0.18853978
20.09	9.0018E+16	27.86	0.25	0.19	0.14	0.1	0.05	12.28	1.86	0.18939778
15.15	5.2553E+16	29.39	0.24	0.19	0.14	0.1	0.05	12.67	1.86	0.18943528
18.28	1.5266E+17	28.15	0.25	0.19	0.14	0.09	0.05	12.39	1.87	0.18877949
16.75	1.0567E+17	28.07	0.25	0.19	0.14	0.09	0.05	12.32	1.84	0.18906069
17.52	6.1634E+16	28.15	0.24	0.18	0.14	0.09	0.05	12.08	1.86	0.1894544
16.6	1.4834E+15	27.22	0.25	0.19	0.14	0.1	0.05	11.85	1.85	0.18909973
17.35	1.4716E+16	26.87	0.25	0.19	0.15	0.1	0.05	11.94	1.85	0.18997187
18.26	1.8116E+16	25.25	0.27	0.2	0.15	0.11	0.05	11.6	1.82	0.19019125
17.05	3.438E+16	27.54	0.25	0.19	0.14	0.1	0.05	12.18	1.84	0.18802209
18.11	1.289E+17	27.08	0.25	0.19	0.14	0.1	0.05	11.99	1.84	0.1880878
14.89	8.2638E+16	29.7	0.25	0.19	0.14	0.09	0.05	12.68	1.9	0.19027791
15.75	1.2141E+16	28.03	0.25	0.19	0.14	0.1	0.05	12.22	1.88	0.18992736
17.39	5.972E+16	26.24	0.26	0.2	0.15	0.1	0.05	11.92	1.83	0.18885342
17.95	3.9952E+16	25.85	0.25	0.19	0.15	0.1	0.05	11.71	1.84	0.18880544

Table A.6: GBHE (3) Enhancement Measurement

eme	emee	ame	amee	pamee $\alpha=0.8$	pamee $\alpha=0.6$	pamee $\alpha=0.4$	pamee $\alpha=0.2$	pame	msrme	std2
20.79	7.0576E+16	21.31	0.26	0.2	0.15	0.1	0.05	10.45	1.72	0.18856136
20.68	4.6442E+17	21.83	0.26	0.2	0.15	0.1	0.05	10.59	1.74	0.18745919
18.69	6.9129E+16	26.68	0.25	0.19	0.14	0.09	0.05	11.77	1.8	0.19077483
17.06	1.2372E+16	26.3	0.24	0.18	0.14	0.09	0.05	11.56	1.8	0.19107347
17.7	8.636E+16	26.91	0.25	0.18	0.14	0.09	0.05	11.82	1.84	0.18885667
20.61	1.4044E+17	26.28	0.25	0.19	0.14	0.09	0.05	11.71	1.81	0.18896016
21.85	8.7569E+17	27.7	0.24	0.18	0.14	0.09	0.05	12.04	1.81	0.19098623
16.22	1.1287E+17	28.28	0.24	0.18	0.14	0.09	0.05	12.1	1.86	0.19138923
17.73	2.3995E+17	26.64	0.25	0.19	0.14	0.1	0.05	11.99	1.8	0.19076011
16.03	1.7789E+16	27.57	0.25	0.19	0.14	0.09	0.05	12.26	1.8	0.19135994
20.01	1.4357E+17	26.13	0.25	0.19	0.14	0.09	0.05	11.73	1.81	0.18987493
22.78	1.6397E+17	25.12	0.25	0.19	0.14	0.1	0.05	11.41	1.76	0.18955295
18.58	1.9362E+17	26.57	0.24	0.18	0.14	0.09	0.05	11.74	1.78	0.1909962
21.23	1.954E+17	26.1	0.25	0.19	0.14	0.09	0.05	11.75	1.8	0.19088182
24.14	2.5001E+17	26.41	0.25	0.19	0.14	0.09	0.05	11.71	1.84	0.19021849
15.27	4.0855E+16	27.77	0.24	0.18	0.14	0.09	0.05	12.15	1.84	0.1898782
17.67	5.6654E+16	25.72	0.25	0.19	0.14	0.09	0.05	11.54	1.86	0.18984876
15.4	72.78	26.72	0.25	0.19	0.14	0.09	0.05	11.87	1.85	0.19021599
17.92	7.695E+16	27.98	0.24	0.18	0.14	0.09	0.05	12.19	1.83	0.19106359
15.38	1.939E+16	28.69	0.24	0.18	0.14	0.09	0.05	12.4	1.82	0.19101086
18.47	1.4642E+17	27.83	0.24	0.18	0.13	0.09	0.05	12.19	1.83	0.190305
17.3	8.9246E+16	27.4	0.25	0.19	0.14	0.09	0.05	12.13	1.82	0.19091468
17.21	1.2995E+17	27.73	0.24	0.18	0.13	0.09	0.05	11.96	1.81	0.19081818
18.3	9.2517E+16	26.85	0.25	0.19	0.14	0.09	0.05	11.72	1.82	0.19052062
16.31	1.3243E+16	26.35	0.25	0.19	0.14	0.1	0.05	11.81	1.79	0.19175131
17.64	7.0706E+15	24.38	0.26	0.2	0.15	0.1	0.05	11.21	1.8	0.19193614
20.35	9.2643E+16	27.69	0.25	0.18	0.14	0.09	0.05	12.18	1.82	0.18969258
19.49	1.5455E+17	26.56	0.25	0.19	0.14	0.09	0.05	11.86	1.81	0.19003424
16.25	7.1797E+16	28.96	0.24	0.18	0.14	0.09	0.05	12.37	1.86	0.19170393
16.71	2.4047E+16	27.28	0.25	0.19	0.14	0.09	0.05	11.98	1.85	0.19159607
16.64	5.504E+16	26.98	0.25	0.19	0.14	0.1	0.05	12.07	1.83	0.19043918
19.54	2.5206E+17	26.63	0.24	0.18	0.14	0.09	0.05	11.7	1.82	0.19030083

Table A.7: GBHE (4) Enhancement Measurement

eme	eme	ame	amee	pamee1	pamee2	pamee3	pamee4	pame	msrme	std2
25.34	4.7283E+15	13.96	0.26	0.2	0.15	0.1	0.05	7.95	1.4	0.24754484
26.45	5.2573E+16	14.43	0.27	0.21	0.15	0.1	0.05	8.2	1.4	0.24772882
25.46	7.7667E+16	14.97	0.28	0.22	0.16	0.11	0.05	8.51	1.43	0.25001804
23.28	5.3512E+15	14.96	0.28	0.21	0.16	0.11	0.05	8.45	1.43	0.25098865
28.16	6.4157E+16	14.94	0.28	0.22	0.16	0.11	0.05	8.45	1.42	0.25560516
25.97	6.4915E+16	14.42	0.28	0.21	0.16	0.1	0.05	8.24	1.4	0.25430535
23.84	1.7621E+16	15.71	0.28	0.22	0.16	0.11	0.05	8.76	1.43	0.25442074
25.05	3.3714E+16	15.43	0.28	0.22	0.16	0.11	0.05	8.7	1.44	0.25586433
22.91	2.9268E+15	14.98	0.28	0.21	0.16	0.11	0.05	8.46	1.41	0.25450829
25.24	2.7025E+16	15.29	0.28	0.22	0.16	0.11	0.05	8.63	1.4	0.25552081
26.8	9.4334E+16	14.87	0.27	0.21	0.16	0.11	0.05	8.37	1.4	0.25444204
30.38	8.3953E+16	14.49	0.27	0.21	0.16	0.11	0.05	8.25	1.41	0.25442987
27.89	4.7782E+16	15.34	0.28	0.22	0.16	0.11	0.05	8.59	1.44	0.25629898
23.77	2.6225E+16	14.97	0.28	0.21	0.16	0.1	0.05	8.48	1.42	0.25807866
25.17	1.0093E+16	14.75	0.28	0.21	0.16	0.11	0.05	8.34	1.4	0.25207825
27.7	3.0127E+16	15.08	0.27	0.21	0.16	0.1	0.05	8.43	1.41	0.25276039
29.85	1.2902E+17	14.65	0.28	0.21	0.16	0.1	0.05	8.32	1.42	0.25597318
24.12	2.2776E+16	15.23	0.28	0.21	0.16	0.1	0.05	8.51	1.43	0.2555102
24.18	7.6031E+16	15.84	0.28	0.21	0.16	0.11	0.05	8.77	1.43	0.25336385
25.05	4.842E+16	16.02	0.28	0.22	0.16	0.11	0.06	8.92	1.44	0.25536412
25.38	3.9798E+16	15.56	0.28	0.21	0.16	0.1	0.05	8.61	1.41	0.25439329
23.19	7.6663E+15	15.04	0.28	0.22	0.16	0.1	0.05	8.55	1.43	0.25592697
28.46	1.3569E+17	15.32	0.28	0.21	0.16	0.1	0.05	8.58	1.4	0.25496833
25.42	4.4689E+16	14.95	0.28	0.21	0.16	0.1	0.05	8.4	1.42	0.25524901
23.99	2.0582E+16	15.72	0.28	0.22	0.16	0.11	0.05	8.81	1.4	0.25044326
23.51	2.5593E+16	15.14	0.28	0.22	0.16	0.11	0.05	8.56	1.41	0.25141822
21.92	3.1372E+15	15.44	0.28	0.21	0.16	0.11	0.05	8.54	1.39	0.25068963
26.49	4.2879E+16	15.03	0.28	0.22	0.16	0.1	0.05	8.49	1.42	0.25199824
24.33	5.7491E+16	16.27	0.28	0.22	0.16	0.11	0.06	9.04	1.43	0.25483681
24.55	1.6366E+16	15.38	0.28	0.21	0.16	0.11	0.05	8.6	1.41	0.25506408
24.93	2.6006E+16	14.82	0.28	0.21	0.16	0.11	0.05	8.33	1.39	0.25059415
24.49	2.2719E+16	14.74	0.28	0.21	0.16	0.1	0.05	8.27	1.39	0.25122052

Table A.8: 2x2 Quaternion Image GrRGB Enhancement Measurement

eme	eme	ame	amee	pamee1	pamee2	pamee3	pamee4	pame	msrme	std2
22.54	1.87E+16	15.78	0.28	0.22	0.16	0.11	0.06	8.49	1.59	0.19386
23.15	1.46E+16	16.73	0.28	0.22	0.16	0.11	0.06	8.78	1.62	0.197755
22.64	3.29E+16	18.13	0.29	0.23	0.17	0.12	0.06	9.32	1.6	0.195454
22.91	6.24E+16	18.42	0.29	0.23	0.17	0.12	0.06	9.45	1.62	0.195694
21.12	1.03E+16	17.58	0.29	0.23	0.17	0.12	0.06	9.22	1.58	0.184488
22.39	1.15E+17	17.04	0.29	0.23	0.17	0.12	0.06	8.97	1.59	0.186264
22.09	1.59E+16	18.96	0.29	0.23	0.17	0.12	0.06	9.66	1.63	0.193631
24.07	1.05E+17	19.28	0.29	0.23	0.17	0.11	0.06	9.69	1.67	0.198848
21.59	2.74E+16	18.52	0.29	0.23	0.17	0.12	0.06	9.42	1.62	0.193169
21.15	1.51E+17	18.85	0.29	0.23	0.17	0.12	0.06	9.5	1.62	0.195349
22.73	2.22E+16	17.75	0.29	0.23	0.17	0.12	0.06	9.31	1.59	0.187738
21.73	2.4E+16	17.29	0.29	0.23	0.17	0.12	0.06	9.17	1.59	0.188414
19.33	2.15E+16	18.75	0.29	0.23	0.17	0.12	0.06	9.55	1.63	0.194117
22.98	6.85E+16	18.65	0.29	0.23	0.17	0.11	0.06	9.55	1.64	0.193037
20.39	4.68E+15	17.3	0.29	0.23	0.17	0.12	0.06	8.98	1.59	0.192722
24.81	7.76E+16	17.74	0.29	0.23	0.17	0.12	0.06	9.18	1.59	0.18936
22.28	4.89E+16	18.13	0.29	0.23	0.17	0.11	0.06	9.33	1.62	0.192283
22.01	4.4E+16	18.41	0.29	0.23	0.17	0.12	0.06	9.46	1.61	0.19085
20.58	1.52E+16	19.43	0.29	0.22	0.17	0.11	0.06	9.73	1.63	0.195771
21.98	7.95E+16	19.62	0.29	0.23	0.17	0.12	0.06	9.8	1.63	0.195093
21.79	5.4E+16	18.76	0.29	0.23	0.17	0.12	0.06	9.51	1.61	0.191523
20.22	1.44E+16	18.99	0.29	0.22	0.17	0.11	0.06	9.6	1.66	0.199671
20.79	2.71E+16	18.64	0.29	0.22	0.17	0.11	0.06	9.54	1.65	0.195042
20.21	3.51E+16	18.27	0.29	0.23	0.17	0.12	0.06	9.39	1.65	0.195246
24.09	1.53E+17	18.78	0.29	0.23	0.17	0.12	0.06	9.54	1.61	0.196804
18.25	1.32E+16	18.09	0.3	0.23	0.17	0.12	0.06	9.4	1.65	0.197145
22.3	4.08E+16	17.95	0.29	0.23	0.17	0.12	0.06	9.24	1.58	0.189449
18.45	2.82E+15	17.89	0.3	0.23	0.17	0.12	0.06	9.24	1.59	0.191713
20.91	2.92E+16	20.16	0.29	0.23	0.17	0.12	0.06	9.96	1.66	0.200025
18	2.96E+15	18.89	0.29	0.23	0.17	0.11	0.06	9.56	1.63	0.197654
17.79	46.42	17.69	0.29	0.23	0.17	0.12	0.06	9.19	1.58	0.191348
22.12	2.6E+16	17.38	0.29	0.23	0.17	0.12	0.06	9.06	1.57	0.190395

Table A.9: 2x2 Quaternion RGBY Enhancement Measure

eme	eme	ame	amee	pamee1	pamee2	pamee3	pamee4	pame	msrme	std2
35.09	1.34E+17	8.85	0.25	0.19	0.14	0.09	0.04	5.54	1.36	0.15371
30.98	6.37E+16	8.98	0.25	0.19	0.14	0.09	0.04	5.6	1.34	0.153618
29.17	1.5E+16	9.72	0.26	0.2	0.14	0.09	0.05	5.96	1.38	0.154753
30.01	4.43E+16	9.8	0.26	0.2	0.14	0.09	0.05	5.97	1.36	0.155561
30.18	1.01E+16	9.26	0.26	0.19	0.14	0.09	0.04	5.77	1.34	0.161853
30.56	1.61E+16	9.08	0.26	0.19	0.14	0.09	0.04	5.67	1.34	0.160443
29.18	1.05E+16	10.07	0.26	0.2	0.14	0.09	0.05	6.1	1.35	0.158813
28.28	5.85E+14	9.69	0.26	0.2	0.14	0.09	0.05	5.97	1.36	0.159541
31.61	1.02E+17	9.7	0.26	0.2	0.14	0.09	0.05	5.97	1.36	0.158855
31.37	4.05E+16	9.67	0.27	0.2	0.14	0.09	0.05	5.96	1.38	0.159324
29.89	1.87E+16	9.32	0.26	0.19	0.14	0.09	0.04	5.76	1.35	0.160171
30.08	1.53E+16	9.28	0.25	0.19	0.14	0.09	0.04	5.71	1.37	0.160036
29.37	2.64E+16	9.45	0.26	0.19	0.14	0.09	0.04	5.89	1.36	0.160407
35.54	1.71E+17	9.35	0.26	0.19	0.14	0.09	0.04	5.79	1.36	0.162078
32.41	1.97E+16	9.37	0.26	0.19	0.14	0.09	0.05	5.82	1.35	0.157066
30.97	2.45E+16	9.7	0.26	0.2	0.14	0.09	0.05	6.03	1.34	0.158299
30.46	3.24E+16	9.08	0.26	0.19	0.14	0.09	0.04	5.65	1.35	0.16046
28.86	4.06E+15	9.34	0.26	0.19	0.14	0.09	0.04	5.81	1.35	0.1602
28.35	1.32E+15	10.04	0.26	0.2	0.14	0.09	0.05	6.1	1.35	0.157494
30.16	7.08E+16	10.15	0.26	0.2	0.14	0.09	0.05	6.15	1.36	0.159334
28.76	9.83E+15	9.8	0.27	0.2	0.14	0.09	0.05	6.03	1.35	0.15906
30.71	2.38E+16	9.59	0.26	0.2	0.14	0.09	0.04	5.95	1.35	0.159423
31.32	8.47E+16	9.94	0.26	0.19	0.14	0.09	0.05	6.03	1.37	0.159311
28.68	6.88E+15	9.29	0.26	0.19	0.14	0.09	0.04	5.8	1.35	0.159586
30.04	8.48E+16	10.21	0.27	0.2	0.14	0.09	0.05	6.2	1.38	0.154899
30.82	2.26E+16	9.77	0.26	0.2	0.14	0.09	0.05	5.96	1.38	0.155667
30.83	8.3E+16	9.86	0.26	0.2	0.14	0.09	0.05	5.99	1.36	0.15662
33.66	1.09E+17	9.7	0.26	0.2	0.14	0.09	0.04	5.95	1.36	0.157201
28.89	7.92E+15	9.78	0.27	0.2	0.14	0.09	0.05	6.01	1.36	0.158346
29.66	1.47E+16	9.33	0.26	0.19	0.14	0.09	0.05	5.8	1.36	0.158783
30.23	4.87E+16	9.76	0.26	0.2	0.14	0.09	0.05	5.98	1.38	0.156036
31.65	1.05E+17	9.76	0.27	0.2	0.14	0.09	0.05	6.02	1.36	0.156724

Table A.10: 3x3 Quaternion Image RGB Enhancement Measurement

eme	emee	ame	amee	pamee1	pamee2	pamee3	pamee4	pame	msrme	std2
23.45	2.6943E+16	13.28	0.29	0.22	0.16	0.11	0.05	7.57	1.49	0.22047471
26.05	1.2577E+17	14.11	0.29	0.22	0.16	0.11	0.05	7.88	1.53	0.22308592
24.55	6.277E+16	16.26	0.3	0.23	0.17	0.12	0.06	8.94	1.54	0.22103355
21.34	4.6966E+16	16.41	0.3	0.23	0.17	0.12	0.06	9	1.54	0.22098888
19.62	2.4479E+16	16.58	0.3	0.23	0.17	0.12	0.06	9.04	1.56	0.21153346
23.98	9.6446E+16	15.74	0.3	0.23	0.17	0.12	0.06	8.71	1.54	0.21344961
21.97	4.5244E+16	17.31	0.3	0.23	0.17	0.12	0.06	9.35	1.56	0.21848802
22.94	5.2371E+16	17.66	0.3	0.23	0.17	0.12	0.06	9.42	1.57	0.22144329
21.65	5.9461E+16	17.03	0.3	0.23	0.17	0.12	0.06	9.35	1.57	0.21801892
19.44	1.9616E+16	17.57	0.31	0.23	0.17	0.12	0.06	9.53	1.58	0.21927847
22.04	4.12E+16	15.89	0.3	0.23	0.17	0.12	0.06	8.84	1.54	0.21435345
20.24	1.3243E+16	15.75	0.3	0.23	0.17	0.11	0.06	8.72	1.53	0.21507384
23.42	6.8547E+16	17.47	0.31	0.23	0.17	0.12	0.06	9.46	1.55	0.21784578
20.24	4.5255E+16	17.65	0.31	0.23	0.17	0.12	0.06	9.52	1.58	0.21672999
22.62	2.5841E+16	15.75	0.3	0.23	0.17	0.12	0.06	8.63	1.55	0.21858687
22.9	5.6014E+16	16.14	0.3	0.23	0.17	0.12	0.06	8.82	1.52	0.21603826
20.8	5.3413E+15	16.38	0.3	0.23	0.17	0.11	0.06	9.01	1.55	0.21695655
21.33	6.5477E+16	16.78	0.3	0.23	0.17	0.12	0.06	9.2	1.57	0.21608217
18.77	39.43	17.27	0.3	0.23	0.17	0.12	0.06	9.34	1.57	0.22005758
18.51	4.4418E+16	17.99	0.3	0.24	0.18	0.12	0.06	9.63	1.57	0.21899234
20.4	1.6539E+16	17.28	0.3	0.23	0.17	0.12	0.06	9.28	1.56	0.21703904
17.32	48	17.77	0.3	0.23	0.17	0.11	0.06	9.5	1.59	0.22179829
19.43	1.7614E+16	17.5	0.3	0.23	0.17	0.12	0.06	9.44	1.59	0.21916882
21.87	5.5498E+16	16.83	0.3	0.23	0.17	0.11	0.06	9.25	1.56	0.21913329
20.08	2.1953E+16	17.37	0.3	0.23	0.17	0.12	0.06	9.4	1.57	0.22180518
24.3	3.8158E+16	16.92	0.31	0.23	0.17	0.12	0.06	9.24	1.57	0.22172846
18.67	106.74	16.33	0.3	0.23	0.17	0.12	0.06	8.97	1.53	0.2170012
20.72	3.255E+16	16.52	0.3	0.23	0.17	0.12	0.06	9.05	1.57	0.21808677
19.35	3.0362E+16	18.37	0.31	0.24	0.18	0.12	0.06	9.87	1.59	0.22251263
19.92	4.8725E+16	17.25	0.31	0.23	0.17	0.12	0.06	9.43	1.6	0.22093824
22.68	3.7268E+16	16.51	0.31	0.23	0.18	0.12	0.06	9.09	1.55	0.21808853
23.11	4.8632E+16	15.98	0.3	0.23	0.17	0.12	0.06	8.86	1.54	0.21741172

Table A.11: 3x3 Quaternion Image CMY Enhancement Measurement

REFERENCES

- [1] R. Naik, D. P. Singh and J. Chaudhary, "A survey on comparative analysis of different ICA based face recognition technologies," in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2018.
- [2] P. J. Phillips, H. Moon, S. A. Rizvi and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 10, pp. 1090-1104, 2000.
- [3] P. J. Phillips, H. Wechsler, J. Huang and P. J. Rauss, "The FERET database and evaluation procedure for face-recognition algorithms," *Image and vision computing*, vol. 16, no. 5, pp. 295-306, 1998.
- [4] O. Marques, Practical image and video processing using MATLAB, John Wiley & Sons, 2011.
- [5] S. H. Abdalla and S. E. O. Fattoh, "Digital Image Processing Technology based on MATLAB," *International Journal of Advanced Research in Computer Science*, vol. 7, no. 3, 2016.
- [6] A. M. Grigoryan, A. John and S. S. Agaian, "A Novel color image enhancement method by the transformation of color images to 2-D grayscale images," *International Journal of Signal Processing and Analysis*, vol. 2, no. 1, p. 18, 2017.
- [7] A. M. Grigoryan and S. S. Agaian, "Color image enhancement via combine homomorphic ratio and histogram equalization approaches: Using underwater images as illustrative

- examples," *International Journal on Future Revolution in Computer Science & Communication Engineering*, vol. 4, no. 5, pp. 36-47, 2018.
- [8] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, 4th ed., Pearson, 2018, p. 1192.
- [9] A. M. Grigoryan and S. S. Agaian, "Image Processing Contrast Enhancement," in *Wiley Encyclopedia of Electrical and Electronics Engineering*, American Cancer Society, 2017, pp. 1-22.
- [10] A. Grigoryan and S. Agaian, "Gradient based histogram equalization in grayscale image enhancement," in *Proc. of SPIE, Defense + Commercial Sensing, Mobile Multimedia/Image Processing Security, and Applications 2019*, vol. 10993, p. 11, Baltimore, Maryland, 4-8 April 2019.
- [11] A. Grigoryan and S. Agaian, "Color Visibility Images and Measures of Image Enhancement," *Electronic Imaging*, vol. 2018, no. 13, pp. 211-219, 2018.
- [12] A. Grigoryan and S. Agaian, "Color facial image representation with new quaternion gradients," *Electronic Imaging*, vol. 2018, no. 13, pp. 381-383, 2018.
- [13] A. M. Grigoryan and S. S. Agaian, "Quaternion and Octonion Color Image Processing with MATLAB," *SPIE*, vol. PM279, p. 404, April 5, 2018.
- [14] A. M. Grigoryan and S. S. Agaian, "Quaternion alpha-rooting image enhancement of grayscale images," in *Proc. of SPIE, Defense + Commercial Sensing, Mobile Multimedia/Image Processing, Security, and Applications 2019*, vol. 10993, p. 8, Baltimore, Maryland, April 4-8 2019.

- [15] H.-S. Li, S. Song, P. Fan, H. Peng, H.-Y. Xia and Y. Liang, "Quantum vision representations and multi-dimensional quantum transforms," *Information Sciences*, vol. 502, pp. 42-58, 2019.
- [16] G. Beach, C. Lomont and C. Cohen, "Quantum image processing (quip)," in *32nd Applied Imagery Pattern Recognition Workshop, 2003. Proceedings.*, 2003.
- [17] M. A. Nielsen and I. Chuang, *Quantum computation and quantum information*, American Association of Physics Teachers, 2002.
- [18] A. M. Grigoryan and S. S. Agaian, "New look on quantum representation of images: Fourier transform representation," *Quantum Information Processing*, vol. 19, no. 5, pp. 1-26, 2020.
- [19] P. Q. Le, F. Dong and K. Hirota, "A flexible representation of quantum images for polynomial preparation, image compression, and processing operations," *Quantum Information Processing*, vol. 10, no. 1, pp. 63-84, 2011.
- [20] N. H. E. J. Mercedes Gimeno-Segovia, *Programming Quantum Computers: Essential Algorithms and Code Samples*, O'Reilly Media, 2019, p. 336.
- [21] J. Sang, S. Wang and Q. Li, "A novel quantum representation of color digital images," *Quantum Information Processing*, vol. 16, no. 2, p. 42, 2017.
- [22] Y. Zhang, K. Lu, Y. Gao, M. Wang, Y. Zhang, ·. K. Lu, ·. M. Wang, K. Lu, M. Wang and Y. Gao, "NEQR: A novel enhanced quantum representation of digital images," vol. 12, pp. 2833-2860, 2013.
- [23] L. Wang, Q. Ran, J. Ma, S. Yu and L. Tan, "QRCI: A new quantum representation model of color digital images," *Optics Communications*, vol. 438, pp. 147-158, 1 5 2019.

- [24] R.-G. Zhou and Y.-J. Sun, "Quantum multidimensional color images similarity comparison," *Quantum Information Processing*, vol. 14, no. 5, pp. 1605-1624, 2015.
- [25] A. M. Grigoryan and S. S. Agaian, "Paired quantum Fourier transform with $\log 2 N$ Hadamard gates," *Quantum Information Processing*, vol. 18, no. 7, p. 217, 2019.
- [26] H.-S. Li, P. Fan, H.-Y. Xia, H. Peng and S. Song, "Quantum implementation circuits of quantum signal representation and type conversion," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 341-354, 2018.
- [27] X.-H. Song, S. Wang and X.-M. Niu, "Multi-channel quantum image representation based on phase transform and elementary transformations," *J. Inf. Hiding Multimed. Signal Process*, vol. 5, no. 4, pp. 574-585, 2014.
- [28] B. Sun, P. Q. Le, A. M. Iliyasu, F. Yan, J. A. Garcia, F. Dong and K. Hirota, "A Multi-Channel Representation for images on quantum computers using the RGB α color space," in *2011 IEEE 7th International Symposium on Intelligent Signal Processing*, 2011.
- [29] R.-G. Zhou, C. Tan and H. Ian, "Global and local translation designs of quantum image based on FRQI," *International Journal of Theoretical Physics*, vol. 56, no. 4, pp. 1382-1398, 2017.
- [30] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin and H. Weinfurter, "Elementary gates for quantum computation," *Physical review A*, vol. 52, no. 5, p. 3457, 1995.
- [31] C. Bernhardt, Quantum Computing for Everyone, Mit Press, 2019.
- [32] P. Fan, R.-G. Zhou, N. Jing and H.-S. Li, "Geometric transformations of multidimensional color images based on NASS," *Information Sciences*, vol. 340, pp. 191-208, 2016.

- [33] H.-S. Li, C. Li, X. Chen and H.-y. Xia, "Quantum image encryption algorithm based on NASS," *International Journal of Theoretical Physics*, vol. 57, no. 12, pp. 3745-3760, 2018.
- [34] M. Pietikäinen, A. Hadid, G. Zhao and T. Ahonen, Computer vision using local binary patterns, vol. 40, Springer Science & Business Media, 2011.
- [35] P. Paysan, R. Knothe, B. Amberg, S. Romdhani and T. Vetter, "A 3D face model for pose and illumination invariant face recognition," in *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, 2009.
- [36] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999.
- [37] V. Blanz and T. Vetter, "Face recognition based on fitting a 3d morphable model," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 25, no. 9, pp. 1063-1074, 2003.
- [38] B. Egger, S. Schönborn, A. Schneider, A. Kortylewski, A. Morel-Forster, C. Blumer and T. Vetter, "Occlusion-aware 3d morphable models and an illumination prior for face image analysis," *International Journal of Computer Vision*, vol. 126, no. 12, pp. 1269-1287, 2018.
- [39] M. Lüthi, T. Gerig, C. Jud and T. Vetter, "Gaussian process morphable models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 8, pp. 1860-1873, 2017.
- [40] S. Schönborn, B. Egger, A. Morel-Forster and T. Vetter, "Markov chain monte carlo for automated face image analysis," *International Journal of Computer Vision*, vol. 123, no. 2, pp. 160-183, 2017.

- [41] A. J. Yuffa, K. P. Gurton and G. Videen, "Three-dimensional facial recognition using passive long-wavelength infrared polarimetric imaging," *Applied optics*, vol. 53, no. 36, pp. 8514-8521, 2014.
- [42] H.-S. Li, Q. Zhu, M.-C. Li and H. Ian, "Multidimensional color image storage, retrieval, and compression based on quantum amplitudes and phases," *Information Sciences*, vol. 273, pp. 212-232, 2014.
- [43] S. E. Venegas-Andraca and S. Bose, "Quantum computation and image processing: new trends in artificial intelligence," in *IJCAI*, 2003.

VITA

Isaac Trevino is from Edinburg, TX of the Rio Grande Valley. He studied his undergraduate degree at the University of Texas of the Rio Grande Valley in Electrical Engineering and graduated in December 2015. He later went to the University of Texas San Antonio to pursue his master's degree in electrical engineering and plans to graduate in May 2020. Isaac plans to start his career as an Electrical Engineer and potentially pursue a future degree.