12-1-2014

Spectral Decomposition of the Scattered Light due to Deposits on the Solar Panel Surface, and Cross Correlated to Power Loss

Suzanna Ho
University of Nevada, Las Vegas, sh20017@gmail.com

SPECTRAL DECOMPOSITION OF THE SCATTERED LIGHT DUE TO DEPOSITS ON THE SOLAR PANEL SURFACE, AND CROSS CORRELATED TO POWER LOSS

by

Suzanna Ho

Bachelor of Science (B.Sc.)
University of Nevada, Las Vegas
2010

A thesis submitted in partial fulfillment of
the requirements for the

Master of Science – Computer Science

Department of Computer Science
Howard R. Hughes College of Engineering
The Graduate College

University of Nevada, Las Vegas
December 2014

The Graduate College

We recommend the dissertation prepared under our supervision by

Suzanna Ho

entitled

Spectral Decomposition of the Scattered Light due to Deposits on the Solar Panel Surface, and Cross Correlated to Power Loss

be accepted in partial fulfillment of the requirements for the degree of

Master of Science – Computer Science

Department of Computer Science

Evangelos A. Yfantis, Ph.D., Committee Chair

John T. Minor, Ph.D., Committee Member

Jan B. Pedersen, Ph.D., Committee Member

Robert F. Boehm, Ph.D., Graduate College Representative

Kathryn Hausbeck Korgan, Ph.D., Interim Graduate College Dean

December 2014

Abstract

The electric energy generated by solar panels declines due to dust particulates, bird deposits, water spots, and other contaminants that inhibit sunlight absorption and promote light scattering. As part of our research, we use cameras to capture images of solar panels, and analyze the images to detect the amount of scattered light. The more scattered light there is, the less light there is to penetrate the solar panel glass and reach the part of the panel that converts incident light to electric energy; therefore, less energy is generated. In this paper, we discuss the classification algorithm we developed to classify panels as clean or dirty. Dirty panels suffer from loss of electric energy generation and they need cleaning in order to restore their performance.

Acknowledgements

First and foremost, I would like to thank my committee members, Dr. Robert F. Boehm, Dr. John T. Minor, Dr. Jan B. Pedersen, and Dr. Evangelos A. Yfantis, for their time and effort in reviewing my thesis and attending the defense.

Most importantly, I would like to thank my family: my parents for their support, and their encouragement to educate myself as much as possible and to do my best; and my brother and his fiance, who helped me get started with college in the first place.

Lastly, I would like to acknowledge my appreciation for the NSF, who supported this project. You and many more people have all been an integral part of my education.

Suzanna Ho

University of Nevada, Las Vegas

November 2014

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The amount of energy generated by solar panels depends on many factors, including the location, the month of the year, day of the month, time of day, weather conditions, and the overall cleanliness of the solar panel. It is important to know when a panel is dirty, so that it can be cleaned promptly to minimize the loss of energy. We devised an algorithm that classifies panels as clean or dirty. Our algorithm utilizes various multivariate statistics and signal processing methods including the Principal Component Analysis, which is based on the eigenvalues and eigenvectors of the variance-covariance matrix of the multivariate probability function of the RGB components. Additionally, the frequency domain analysis, which is based on the frequency spectrum of the scattered light, as well as phase spectrum analysis. Time-frequency domain analysis, which is based on wavelet decomposition. Furthermore, the Mahalanobis distance, which is based on the trivariate probability density function (PDF) obtained from sampling the clean panels, and the trivariate PDF from sampling the dirty panels. Following this section is background information on the dynamics of light. The next chapters describe the methodology utilized in the classification algorithm, and the experimental results, respectively.

1.1 Background

1.1.1 The Color of the Sky

Sunlight is comprised of ultraviolet, visible, and infrared light. For now we will only focus on the visible spectrum. White light from the sun is comprised of all colors in the visible spectrum. Each color is distinguished by its different wavelength, where violet and blue are short, yellow and red are long, and green is in the middle. Different molecules and particles in the atmosphere affect different components of white light, and these phenomena are explained by Rayleigh scattering and the Mie solution to Maxwell's equation. Rayleigh scattering applies to particles that are much

smaller than the wavelength, where shorter wavelengths scatter more than longer ones. The Mie solution to Maxwell's equation applies to particles similar to or larger than the wavelength, where all wavelengths of white light scatter equally. The color of the sky is indicative of how clean or dirty the air is.

Clean air is composed of approximately 78.04% Nitrogen, 20.94% Oxygen, 0.934% Argon, and 0.0350% other gases [BF85]. Due to the abundance of gases, particularly Nitrogen and Oxygen, light is scattered most frequently by these gases while traveling through the atmosphere. Because the gas molecules are small, the color components most affected are the ones with shorter wavelengths, namely violet and blue. This partially explains why the sky is blue on a clear day. Although violet has a smaller wavelength than blue and scatters more, it has lower intensity compared to other colors and the human eye has low sensitivity to it; therefore, we see the sky as blue rather than violet [HP08, Ram11, TSG91].

Dirty air is comprised of not only gas, but also any combination of water vapor, sulfur, aerosols, soot, and pollutants. Due to the addition of larger molecules, color components of larger wavelengths scatter, such as green and red. This mixture of short and long wavelengths in the atmosphere makes the sky look gray on a cloudy day. The same can be said for clouds. Considering the primary component of clouds is water vapor, the condensation makes clouds appear white or gray.

1.1.2 From Sunlight to Electric Energy

In the previous section, we discussed the dynamics of white light in the atmosphere. In this section, we will discuss the dynamics of the light that actually makes it through the atmosphere to the solar panels. Recall that red light has the longest wavelength in the visible spectrum. Therefore, it penetrates the glass more readily and produces the most amount of electricity. Meanwhile, blue and violet light suffer the most reflection, and contribute very little to the production of electric energy.

Solar output depends on the time of day, month, or year, and climatological conditions. In terms of climate, dust storms and sand storms are common in arid climates, and as the dust settles on the glass of a solar panel, the energy output decreases, because part of the light is refracted by the dust and less light penetrates the glass. The loss of light energy depends on the amount, size, and chemical composition of the dust [RVM89, MBY$^+$06]. In terms of time, trees are sparse in arid climates, and as birds migrate in the fall and spring, solar farms are used by them as rest areas; therefore, the solar panels become dirty by bird excrement. It is safe to say that the dirtier a panel is, the less energy it produces.

Chapter 2

Classification Algorithm

The classification algorithm involves sampling, determining the classification vector, and developing, training, and testing a classifier. Our goal is to use videos of solar panel surfaces captured by cameras from a solar panel site and to classify the solar panels as clean or dirty. All of the samples we collected are images of solar panels captured by a digital camera. First, we took pictures of a panel when it was dirty, then we took pictures of it after we cleaned it. Figures 2.1 and 2.2 show an example of a bad and good image, respectively. Some of these images included both clean and dirty sections, so we cropped them into smaller samples of 200×200 dimensions to ensure separation between clean and dirty data. The goal was to use crops from the same panel, and crops from a combination of panels with similar characteristics. We settled on three groups of training data, where a group contains one clean sample set and one dirty sample set.

1. The first group contains data from the same panel. Each set has 12 samples.
2. The second group builds upon the first group by incorporating data from another panel of similar characteristics. Each set has 20 samples.
3. The third group does not build upon the first and second group. Instead, it incorporates data from two panels of similar characteristics, but with a lighter shade of blue. Each set has 19 samples.

The process of building and testing a classifier starts with defining the classification vector, then defining, training, and testing the classifier. If the classifier is proven to work properly, then we can use it to classify new incoming data. The power of a classifier is a function of its ability to correctly classify objects. The higher the probability of the classifer to correctly classify objects, the more powerful it is. Another way of expressing the power of a classifier is by computing its misclassification error, which is the probability of a classifier to incorrectly classify objects. The lower the misclassification error, the more powerful the classifier is. We train the classifier as

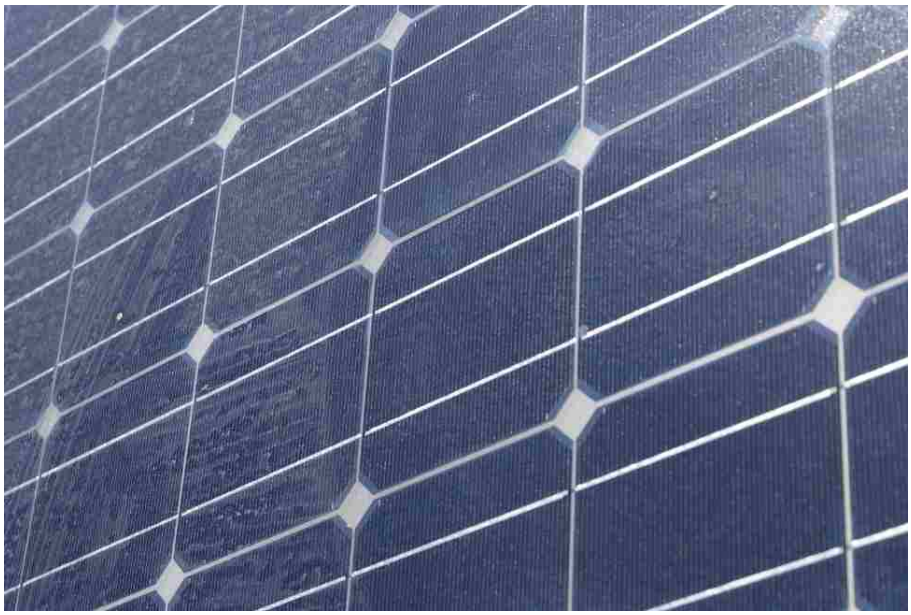


Figure 2.1: This is an example of an image we did not use. Notice the band angle and glare at the top-right corner.

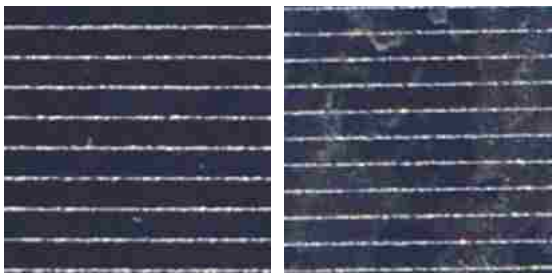
Figure 2.2: This is an example of an image we used. Although it has both clean and dirty areas, it has a good angle, which makes it easy to crop.

follows: define the classification vectors for each class, where each vector represents a sample of data, or an image; build the classifier, which is a metric, or a distance, that will classify a vector as belonging to the correct class; and classify vectors already used to define a class c as belonging to class c. For the last step, we use the Jackknifing approach, where we take a vector v that already belongs to a class c, recompute c without v, and then we use our classification algorithm to classify v. We repeat this process for each vector from each class, and we estimate the probability of correct classification as the ratio of the number of vectors classified correctly over the total number of classification vectors, also known as the accuracy. The vectors and parameters obtained by the classifier will be used to classify new vectors corretly.

An important component associated with the panel classification process is the noise. The noise depends on the atmospheric pressure, the amount of particulates floating in the atmosphere, the orientation of the sun, temperature fluctuations, the orientation of the camera, fluctuations of the power source of the chip obtaining the image, the type and quality of the imager chip and other factors. We made no separate considerations for noise in this study, thus the noise is embedded in the signal and the idea here is to use algorithms that are robust with respect to the noise. We explored various methods and found that the Mahalanobis distance was the strongest classifier. In subsequent sections of this chapter, we will discuss the Mahalanobis distance and how we used it during training and testing, as well as other notable classifiers. In the next chapter, we will discuss the experimental results of each classifier.

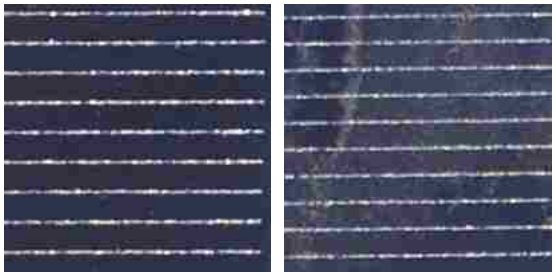
2.1 Samples

These are the sample sets we used for the Discrete Fourier Transform, the Discrete Wavelet Transform, and the Principal Component Analysis. We chose three pairs from each group. We avoided samples that were either false negatives or false positives, and we made the pairs as homogeneous as possible.



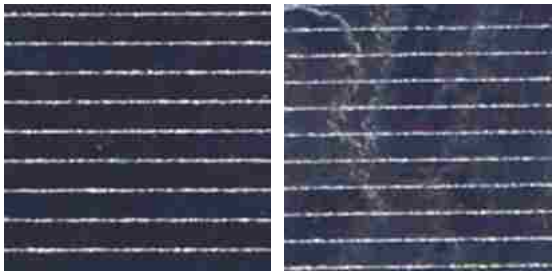
(a) Clean. (b) Dirty.

Figure 2.3: Group 1 sample set 1.



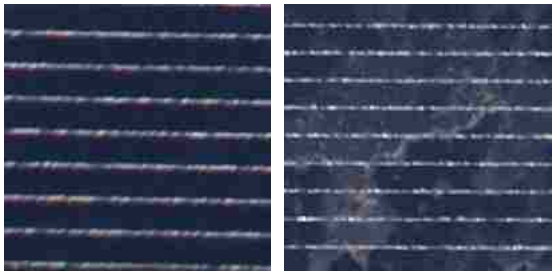
(a) Clean. (b) Dirty.

Figure 2.4: Group 1 sample set 2.



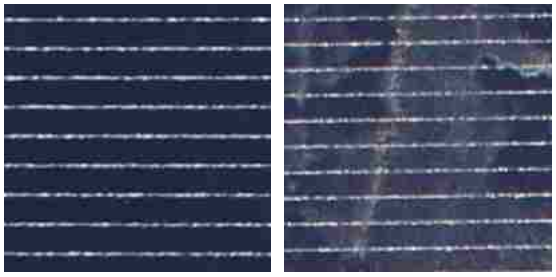
(a) Clean. (b) Dirty.

Figure 2.5: Group 1 sample set 3.



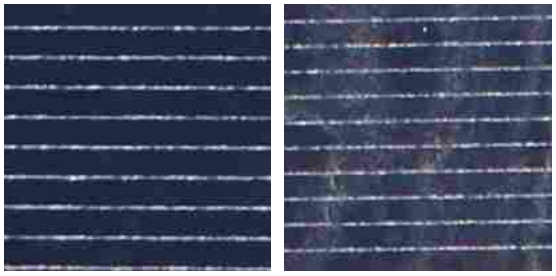
(a) Clean. (b) Dirty.

Figure 2.6: Group 2 sample set 1.



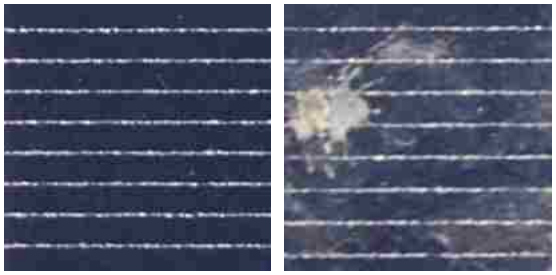
(a) Clean. (b) Dirty.

Figure 2.7: Group 2 sample set 2.



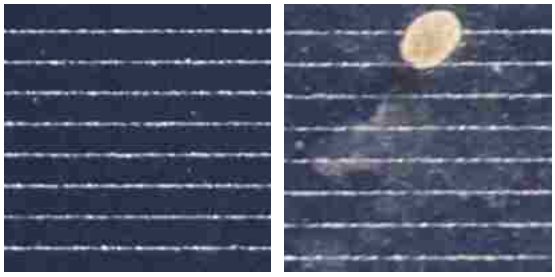
(a) Clean. (b) Dirty.

Figure 2.8: Group 2 sample set 3.



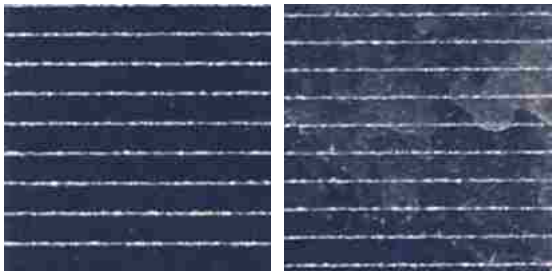
(a) Clean. (b) Dirty.

Figure 2.9: Group 3 sample set 1.



(a) Clean. (b) Dirty.

Figure 2.10: Group 3 sample set 2.



(a) Clean. (b) Dirty.

Figure 2.11: Group 3 sample set 3.

2.2 Mahalanobis Distance

The Mahalanobis Distance measures the distance between a data point to a common point. Consider a sample set of K clean panels, another sample set of D dirty panels, and an unknown sample x from an arbitrary panel. For this panel, we compute the averages of the RGB values, and the variance-covariance matrix of x. The averages constitute a classification vector. If the panel is clean, then its vector belongs to the clean class; otherwise, it belongs to the dirty class. All the clean classification vectors have a grand average, which is another vector, and they have their own variance-covariance matrix. The grand average of the clean classification vectors defines the centroid, or center of gravity, of the clean class. The variance-covariance matrix of the vectors of the clean class defines how close the vectors are to the centroid. Together, the variance-covariance matrix and the centroid make up the parameters of the classifier.

Theorem 2.2.1 *Let $\overline{x_{cm}}$ be a classification vector from the clean class of images $m = 1, 2, \cdots, K$ with centroid $\overline{\overline{x_c}}$ and variance-covariance matrix Σ_c, then the mean of the vector $\overline{x_{cm}} - \overline{\overline{x_c}}$ is zero, the variance-covariance matrix of $\overline{x_{cm}} - \overline{\overline{x_c}}$ is $\frac{K-1}{K}\Sigma_c$, where Σ_c is the variance-covariance matrix of $\overline{x_{cm}}$, $m = 1, 2, \cdots, K$, and K is the number of classification vectors in the clean class. The Mahalanobis distance of $\overline{x_{cm}}$ from $\overline{\overline{x_c}}$ is*

$$d_{cm}^2 = \left(\overline{x_{cm}} - \overline{\overline{x_c}}\right)^T \left(\frac{K}{K-1}\right) \Sigma_c^{-1} \left(\overline{x_{cm}} - \overline{\overline{x_c}}\right) \tag{2.1}$$

Alternatively, the Mahalanobis distance of $\overline{x_{dm}}$ from $\overline{\overline{x_d}}$, where d is the dirty class, is

$$d_{dm}^2 = \left(\overline{x}_{dm} - \overline{\overline{x_d}}\right)^T \left(\frac{D}{D-1}\right) \Sigma_d^{-1} \left(\overline{x}_{dm} - \overline{\overline{x_d}}\right) \tag{2.2}$$

Proof: Let $\mu = E\left(\overline{x_{cm}}\right)$, then

$$\begin{aligned} E\left(\overline{\overline{x}}\right) &= E\left[\frac{\sum_{m=1}^{K} \overline{x_{cm}}}{K}\right] \\ &= \frac{1}{K}\sum_{m=1}^{K} E\left[\overline{x_{cm}}\right] \\ &= \frac{1}{K} K\mu \\ &= \mu \end{aligned}$$

Thus $E\left(\overline{x_{cm}} - \overline{\overline{x_c}}\right) = E\left[\overline{x_{cm}}\right] - E\left[\overline{\overline{x_c}}\right] = \mu - \mu = 0$, where μ is a column vector with three components, namely the means of the red, green, and blue components. The variance-covariance matrix Σ of the classification vectors is a 3×3 positive definite symmetric matrix and is denoted by

$$\Sigma = E\left(\overline{x_{cm}} - \mu\right)' \left(\overline{x_{cm}} - \mu\right)$$

The variance-covariance matrix of the centroid $\overline{\overline{x_c}}$ is

$$\begin{aligned}
E\left(\overline{\overline{x_c}}-\mu\right)'\left(\overline{\overline{x_c}}-\mu\right) &= E\left[\frac{\sum_{i=1}^{K} x_{ci}}{K}-\mu\right]'\left[\frac{\sum_{j=1}^{K} x_{cj}}{K}-\mu\right] \\
&= \frac{1}{K^2}E\left[\left(\sum_{i=1}^{K} x_{ci}-\mu\right)'\left(\sum_{j=1}^{K} x_{cj}-\mu\right)\right] \\
&= \frac{1}{K^2}\left\{\sum_{i=1}^{K} E\left(x_{ci}-\mu\right)'\left(x_{ci}-\mu\right)+\sum_{\substack{i=1\\ i\neq j}}^{K}\sum_{\substack{j=1\\ i\neq j}}^{K} E\left(x_{ci}-\mu\right)\left(x_{cj}-\mu\right)\right\} \\
&= \frac{1}{K^2}K\Sigma+\sum_{\substack{i=1\\ i\neq j}}^{K}\sum_{\substack{j=1\\ i\neq j}}^{K} 0 \\
&= \frac{\Sigma}{K}
\end{aligned}$$

The variance covariance matrix of $\overline{x_{cm}}-\overline{\overline{x_c}}$ is

$$\begin{aligned}
E\left(\overline{x_{cm}}-\overline{\overline{x_c}}\right)'\left(\overline{x_{cm}}-\overline{\overline{x_c}}\right) &= E\left[\left(\overline{x_{cm}}-\mu\right)\left(\overline{x_c}-\mu\right)\right]'\left[\left(\overline{x_{cm}}-\mu\right)\left(\overline{x_c}-\mu\right)\right] \\
&= E\left(\overline{x_{cm}}-\mu\right)'\left(\overline{x_{cm}}-\mu\right)-E\left(\overline{x_{cm}}-\mu\right)'\left(\overline{\overline{x_c}}-\mu\right) \\
&\quad -E\left(\overline{\overline{x_c}}-\mu\right)'\left(\overline{x_{cm}}-\mu\right)+E\left(\overline{x_c}-\mu\right)'\left(\overline{x_c}-\mu\right) \\
&= \Sigma-\frac{1}{K}\Sigma-\frac{1}{K}\Sigma+\frac{1}{K}\Sigma \\
&= \frac{K-1}{K}\Sigma
\end{aligned}$$

From the above, we infer that the Mahalanobis distance of $\overline{x_{cm}}$ from the centroid $\overline{\overline{x}}_c$ is

$$d_{cm}^2=\left(\overline{x_{cm}}-\overline{\overline{x_c}}\right)^T\left(\frac{K}{K-1}\right)\Sigma_c^{-1}\left(\overline{x_{cm}}-\overline{\overline{x_c}}\right)$$

■

Theorem 2.2.1 describes how we can compute the Mahalanobis distance of the mth vector of a class to the centroid of that class, and it is illustrated in Figure 2.12. The clean classification vectors form a space with the centroid $\overline{\overline{x_c}}$, which is a subspace of the 3-D space defined by the *RGB* values, and its shape is similar to an ellipsoid, where each axis has a different size. Every classification vector $\overline{x_{cm}}$ of the clean class that does not belong to the intersection of the clean class with the dirty class, has a smaller distance from the centroid of the clean space than from the centroid of the dirty space. Similarly, the dirty classification vectors form a space with the centroid $\overline{\overline{x_d}}$, and its subspace has the same characteristics as the clean subspace. Every classification vector $\overline{x_{dm}}$ of the dirty class that does not belong to the intersection of the clean class with the dirty class, has a smaller distance from the centroid of the dirty space than from the centroid of the clean space. Classification vectors belonging to the intersection of the clean space and the dirty

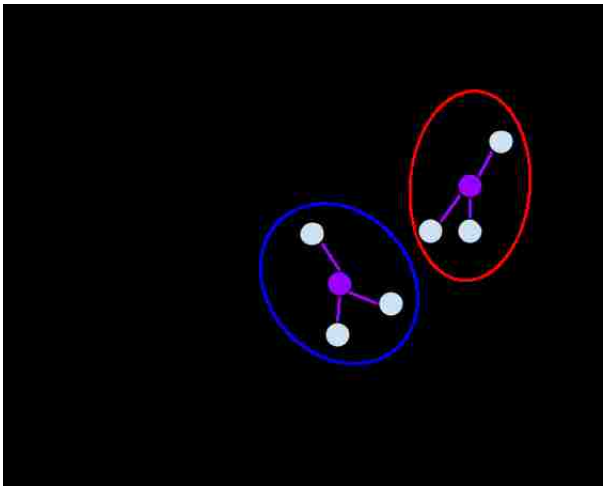


Figure 2.12: Illustration for Theorem 2.2.1. The clean space is blue, and any points within that space are clean classification vectors. The dirty space is red, and any points within that space are dirty classification vectors. The centroids are purple.

space could have a smaller distance from the centroid of the dirty space's centroid, although they actually belong to the clean space, and vice versa. The issue of intersections is addressed in Theorem 2.2.3, and is illustrated in Figure 2.14a and Figure 2.14b. Although Figure 2.14b shows that not all vectors in the intersection are misclassified, the worst case for misclassification of the entire intersection is that all of them are misclassified. The figure also shows a point sitting on the misclassification plane. The probability of a point to be equidistant from both centroids is close to zero, but when it happens, we arbitrarily choose a class.

Theorem 2.2.2 *Let* $\overline{x_{cm}}$, $m = 1, 2, \cdots, K$ *be a classification of the clean space with mean vector* μ_c, *variance-covariance matrix* Σ_c, *and centroid* $\overline{\overline{x_c}}$. *Let* $\overline{x}$ *be a new classification vector. If* $\overline{x}$ *belongs to the clean space, then* $E\left(\overline{x} - \overline{\overline{x_c}}\right) = 0$, *the variance-variance matrix of* $\overline{x} - \overline{\overline{x_c}}$ *is* $\frac{K+1}{K}\Sigma_c$, *and the Mahalanobis distance is*

$$d_1^2 = \left(\overline{x} - \overline{\overline{x_c}}\right)' \frac{K}{K+1} \Sigma_c^{-1} \left(\overline{x} - \overline{\overline{x_c}}\right)$$

Proof: Let $E\left(\overline{x}\right) = \mu_c$, $E\left(\overline{\overline{x_c}}\right) = \mu_c$ from Theorem 2.2.1, and $E\left(\overline{x} - \overline{\overline{x_c}}\right) = \mu_c - \mu_c = 0$.

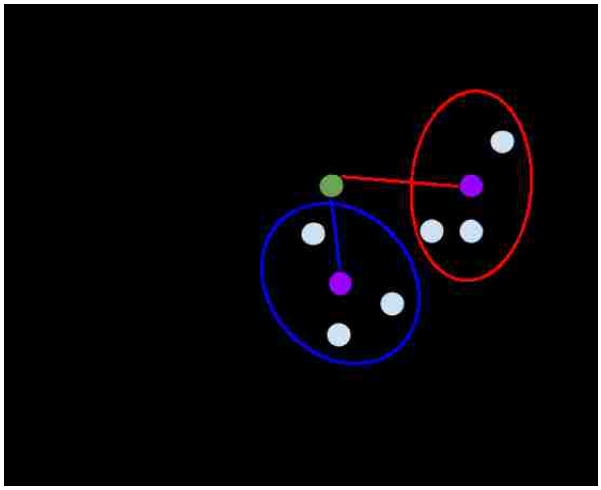


Figure 2.13: Illustration for Theorem 2.2.2. The blue space is clean class, and the red space is dirty class. The purple points are the centroids of each class. The green point is the unknown vector. It belongs to the clean class since it is closer to the clean space's centroid.

The variance-covariance matrix of $\overline{x} - \overline{\overline{x_c}}$ is

$$\begin{aligned} E\left(\overline{x} - \overline{\overline{x_c}}\right)'\left(\overline{x} - \overline{\overline{x_c}}\right) &= E\left[\left(\overline{x} - \mu_c\right)\left(\overline{x_c} - \mu_c\right)\right]'\left[\left(\overline{x} - \mu_c\right)\left(\overline{x_c} - \mu_c\right)\right] \\ &= E\left(\overline{x} - \mu_c\right)'\left(\overline{x} - \mu_c\right) - E\left(\overline{x} - \mu_c\right)'\left(\overline{\overline{x_c}} - \mu_c\right) \\ &\quad - E\left(\overline{\overline{x_c}} - \mu_c\right)'\left(\overline{x} - \mu_c\right) + E\left(\overline{\overline{x_c}} - \mu_c\right)'\left(\overline{\overline{x_c}} - \mu_c\right) \\ &= \Sigma_c - 0 - 0 + \frac{1}{K}\Sigma_c \\ &= \frac{K+1}{K}\Sigma \end{aligned}$$

Therefore, the Mahalanobis distance of $\overline{x}$ from $\overline{\overline{x_c}}$ is

$$d_1^2 = \left(\overline{x} - \overline{\overline{x_c}}\right)' \frac{K}{K+1}\Sigma_c^{-1}\left(\overline{x} - \overline{\overline{x_c}}\right)$$

■

Theorem 2.2.2 describes how we can classify a new, arbitrary vector. If $\overline{x}$ is classified correctly to class c, then c's space is recalculated, which means a new centroid and variance-covariance matrix is computed. Recall that we used the Jackknife method to calculate the misclassification error during the design of the classifier, where we start by removing a vector r from the clean space, recompute the clean space (centroid and variance covariance matrix), and then classify r against the new clean space and the original dirty space. If it is correctly classified as clean, then we consider it a true positive; otherwise, we count it as being misclassified, or a false negative. We repeat this process until we go through all the clean and dirty classification vectors. Afterwards, we divide the number of misclassified vectors over the total number of vectors in the two classes, which is an estimate of the misclassification probability. As the process continues and more vectors

are classified correctly, the process comes into a steady state, and the estimated misclassification probability converges to the true probability of misclassification. This is due to the Law of Large Numbers, which states that as the number of trials of a random process increases, the results should converge to the expected value.

Theorem 2.2.3 *Let $\overline{x_{cm}}$, where $m = 1, 2, \cdots, K$, be a classification vector of the clean space with mean vector μ_c, variance-covariance matrix Σ_c, and centroid $\overline{\overline{x_c}}$. Let $\overline{x_{dm}}$, where $m = 1, 2, \cdots, D$, be a classification vector of the dirty space with mean vector μ_d, variance-covariance matrix Σ_d, which is not significantly different from Σ_c, and centroid $\overline{\overline{x_d}}$. A better estimate of the variance for both classes is $\Sigma = \frac{(K-1)\Sigma_c + (D-1)\Sigma_d}{K+D-2}$. Let $\overline{x}$ be a new classification vector. The Mahalanobis distance of $\overline{x}$ from $\overline{\overline{x_c}}$ is*

$$d_1^2 = \left(\overline{x} - \overline{\overline{x_c}}\right)' \frac{K}{K+1} \Sigma^{-1} \left(\overline{x} - \overline{\overline{x_c}}\right)$$

and from $\overline{\overline{x_d}}$ is

$$d_2^2 = \left(\overline{x} - \overline{\overline{x_d}}\right)' \frac{D}{D+1} \Sigma^{-1} \left(\overline{x} - \overline{\overline{x_d}}\right)$$

If $d_1^2 < d_2^2$, then $\overline{x}$ is more likely to be in the clean space, and if $d_1^2 > d_2^2$, then $\overline{x}$ is more likely to be in the dirty space.

Proof: Consider the following functions expressing the probability of $\overline{x}$ belonging to the clean space and to the dirty space:

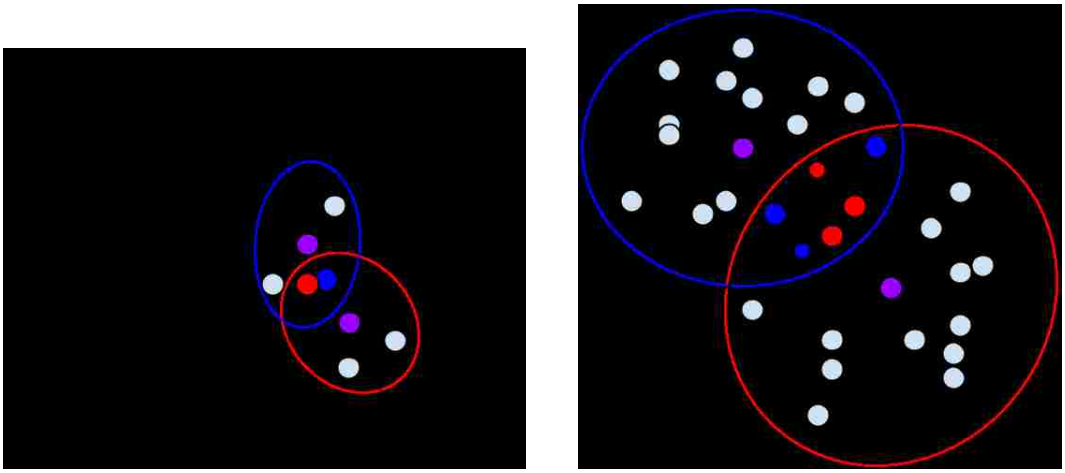
$$\frac{1}{(2\pi)^{\frac{3}{2}} \left(\frac{K+1}{K}\right)^{\frac{1}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}\left(\overline{x} - \overline{\overline{x_c}}\right)' \frac{K}{K+1} \Sigma^{-1} \left(\overline{x} - \overline{\overline{x_c}}\right)}$$

$$>$$

$$\frac{1}{(2\pi)^{\frac{3}{2}} \left(\frac{D+1}{D}\right)^{\frac{1}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}\left(\overline{x} - \overline{\overline{x_d}}\right)' \frac{D}{D+1} \Sigma^{-1} \left(\overline{x} - \overline{\overline{x_d}}\right)}$$

The greater relation is due to $d_1^2 < d_2^2$, and since $\sqrt{\frac{K+1}{K}} \approx \sqrt{\frac{D+1}{D}} \approx 1$, which implies that if $d_1^2 < d_2^2$, $\overline{x}$ is more likely to belong to the clean class. ■

So far we have explored the Mahalanobis distance of the trivariate probability distribution function derived from the *RGB* data of clean and dirty samples. The red, green, and blue histograms of a panel can be used to estimate the marginal probability functions of the panel section, as well as the mean vector, and the variance-covariance matrix of the panel. This means if we were to compute the marginals of two samples, the bigger the difference, the more apparent it will be when we plot the Mahalanobis distances. The estimated mean vector and the variance-covariance matrix can be used to estimate the trivariate probability function of the average *RGB* vector, which is a trivariate normal according to the Central Limit Theorem.



(a) (b)

Figure 2.14: Illustrations for Theorem 2.2.3. The clean space is blue, and the dirty space is red. The centroids of each class are the purple points. (a) A three-dimensional example. The blue point is a sample classified as clean, and the red point is a sample classified as dirty. (b) The black line represents a plane. Anything to the left of the plane is clean, and anything to the right of the plane is dirty. The blue point (thickened black border) now belongs to the dirty class and is closer to the dirty class's centroid, even though it was part of the clean class. The same can be said for the red point (thickened black border) that was a dirty classification vector.

2.3 Histogram Analysis

The histogram allows us to view the data distribution through a mathematical model. The histogram of a digital image is a discrete function $h(x_i) = n_i$ where x_i is the ith intensity value in the range $[0, X-1]$, and n_i is the number of elements in the data with intensity x_i [GW08]. We refer to these counts as bins. It is common practice to normalize a histogram, and the method we chose was to divide each bin by the total number of elements times its width. As a result, the area, or integral, under the histogram is equal to one. The function obtained through normalization is the probability density function.

The gamma distribution provides enough flexibility to model these probability density functions. The gamma density function [WMMY12] is denoted by

$$f(x) = \frac{x^{\alpha-1}e^{-\frac{x}{\beta}}}{\Gamma(\alpha)\beta^{\alpha}}$$

where $x, \alpha, \beta > 0$ and the gamma function $\Gamma(\alpha) = \int_0^\infty x^{\alpha-1}e^{-x}\,\mathrm{d}x$. In our application, we computed $\Gamma(\alpha)$ numerically by using the Lanczos approximation [PTVF92]

$$\Gamma(\alpha+1) = (\alpha+\gamma+0.5)^{\alpha+0.5}e^{-(\alpha+\gamma+0.5)}\sqrt{2\pi}\left[p_0 + \sum_{n=1}^{N}\frac{p_n}{\alpha+n}\right]$$

where $\alpha > 0$, $\gamma = 5$, $N = 6$, and

$$p = \left\{\begin{array}{l} 76.18009172947146, \\ -86.50532032941677, \\ 24.01409824083091, \\ -1.231739572450155, \\ 1.208650973866179 \times 10^{-3}, \\ -5.395239384953 \times 10^{-6} \end{array}\right\}$$

Another way to compute it numerically is to solve for the integral using integration by parts, which shows that

$$\begin{aligned} \Gamma(\alpha) &= (\alpha-1)\Gamma(\alpha-1) \\ &= (\alpha-1)(\alpha-2)\Gamma(\alpha-2) \\ &= (\alpha-1)(\alpha-2)\cdots(2)(1)\Gamma(1) \\ &= (\alpha-1)! \end{aligned}$$

for positive integers α. For floating points, we can still use the factorial pattern until we reach an α that is between 0 and α, in which case the integral is small enough to be computed numerically.

The global mode of the gamma density function, or the point x that maximizes $f(x)$ is $x_m = (\alpha-1)/\beta$. If $\hat{x}$ is the global mode based on the empirical density function computed from the data

and $\overline{x}$ is the average, then an estimate of the parameters α and β from the data is: $\hat{\alpha} = \frac{\overline{x}}{\overline{x}-\hat{x}}$ and $\hat{\beta} = \overline{x} - \hat{x}$. The alpha parameter affects the shape of the curve, while the beta parameter affects the scale of the curve. The idea here is that samples with similar parameters belong to the same class.

2.4 Discrete Fourier Transform

The Discrete Fourier Transform (DFT) is a mathematical procedure that transforms data from the spatial domain to the frequency domain. Our input data is a two dimensional image, so we need to compute a two dimensional transform. The 2-D DFT [GW08] is denoted by

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-i2\pi(ux/M+vy/N)} : i = \sqrt{-1} \tag{2.3}$$

where the input $f(x,y)$ may be either real or complex, and the output $F(u,v)$ is always complex. Due to the separability property, the 2-D DFT can be computed with the 1-D DFT along the rows and columns [GW08]. The 1-D DFT [GW08] is denoted by

$$F(u) = \sum_{x=0}^{M-1} f(x) W_M^{ux} : W_M = e^{-i2\pi/M} \tag{2.4}$$

Although Equation 2.4 improves upon Equation 2.3, it can be optimized further. We can compute the 1-D DFT using the Fast Fourier Transform (FFT), which is an algorithm that efficiently computes the 1-D DFT in $O(NlogN)$ time as opposed to $O(N^2)$ time, because it exploits the periodicity property

$$W_N^{k+N} = W_N^k$$

and symmetry property

$$W_N^{k+\frac{N}{2}} = -W_N^k$$

of the complex exponential [GW08]. The main advantage of the FFT is that it does not perform the unnecessary duplicate computations that is prevalent in Equation 2.3. There are several well-known FFT algorithms, such as the Cooley-Tukey algorithm, but we used our own, which we describe next.

Given an RGB image of $N \times M$ dimensions, pad the image with extra (R,G,B) pixels that are zeroes $(0,0,0)$ row-wise if N is not a power of two, and column-wise if M is not a power of two. It is unnecessary for $N = M$. Assuming we had to pad the image with zeroes, the new dimensions are N' and M'. The FFT-butterfly consists of stages, and within each stage are blocks. There are S stages where $N' = 2^S$; each stage s_i will use a certain subset of multipliers; and the number of blocks equals the number of multipliers used for that particular stage. Each block consists of one or more butterflies, and each butterfly corresponds to a particular multiplier. Figure 2.15 illustrates

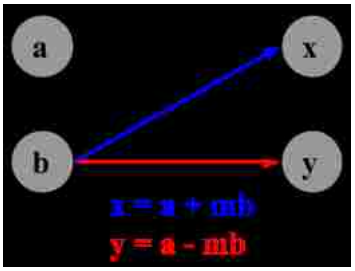


Figure 2.15: Butterfly.

the form of the butterfly. The total number of multipliers is $\frac{N'}{2}$, or 2^{S-1}, and they are of the form $e^{-j2k\pi/8}$, where $0 \leq k < \frac{N'}{2}$ and the set of k's are in reverse bit order. The complex exponential can be rewritten using Euler's equation

$$e^{i\theta} = \cos\theta + i\sin\theta$$

as

$$\begin{aligned} e^{i(-\theta)} &= \cos(-\theta) + i\sin(-\theta) \\ e^{-i\theta} &= \cos\theta - i\sin\theta \end{aligned} \tag{2.5}$$

where

$$\theta = \frac{2k\pi}{8}$$

The algorithm to compute in reverse bit order is as follows: start with stage 0 when $i = 0$, and for each subsequent stage s_i, where $i = 1, 2, \cdots, \frac{N'}{2} - 1$, multiply the previous values by two, then add one to these new values. Table 2.1 displays the results when $S = 4$. Each bit value is substituted into Equation 2.5 to compute the corresponding multiplier.

stage s_i	0	1	2	3
k	0	0	0	0
		1	2	4
			1	2
			3	6
				1
				5
				3
				7

Table 2.1: Reverse bit order when $S = 4$.

Because the FFT-butterfly must be computed row-wise and column-wise in order for it to be two-dimensional, we must compute separate mutipliers for the rows and columns. For optimization purposes, it is better to precompute the multipliers, and the benefit of $N = M$ is that you will only have to compute them once. Each row will have the same multipliers, and the same goes for

columns. Figure 2.16 illustrates an example of transforming a sequence of $N = 8$ data elements. There are $\frac{N}{2} = 4$ multipliers and $S = 3$ stages, so if we refer to Table 2.1, we should use the column of k's under s_2. The numbers above each stage signify the group of multipliers to use. Therefore, from left to right (for stages) and top to bottom (for blocks), the first stage uses the multipler when $k = 0$ on the one and only block; the second stage uses the multiplier when $k = 0$ on the first block, and $k = 2$ on the second block; and the third stage uses the multiplier when $k = 0$ on the first block, $k = 2$ on the second block, $k = 1$ on the third block, and $k = 3$ on the fourth block. Each stage reuses the computations of previous stages. The results are in reverse bit order, so the

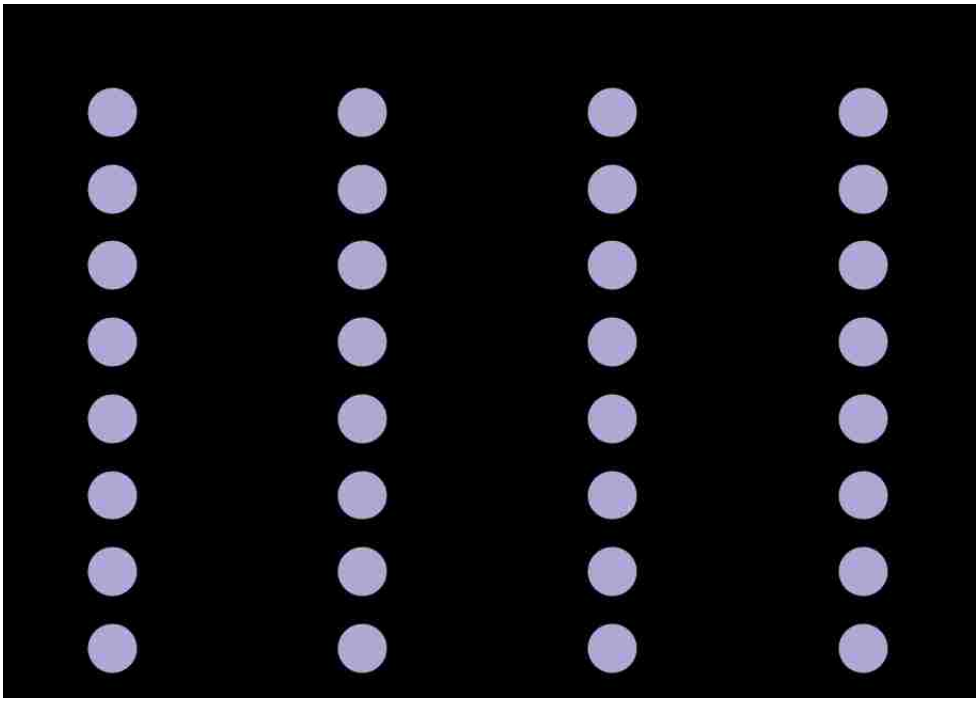


Figure 2.16: Forward FFT on a sequence of data where $N = 8$. Notice the subscripts after the transform are out of order.

last step is to unscramble them in ascending order.

The Fourier Transform produces a complex number valued output image which can be displayed with two images, either with the real and imaginary part, or with the magnitude

$$|F(u,v)| = \left[R^2(u,v) + I^2(u,v)\right]^{1/2}$$

and phase

$$\phi(u,v) = \arctan\left[\frac{I(u,v)}{R(u,v)}\right]$$

It is common to display the magnitude only [FPWW00].

2.5 Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) converts an input signal into low-pass and high-pass wavelet coefficients. This transformation can be applied recursively on the low-pass coefficients until the desired number of transforms is reached. We used the Daubechies wavelet, a biorthogonal wavelet, which consists of the following low-pass filter

$$\begin{bmatrix} -\frac{1}{8} & \frac{1}{4} & \frac{3}{4} & \frac{1}{4} & -\frac{1}{8} \end{bmatrix}$$

and high-pass filter

$$\begin{bmatrix} -\frac{1}{2} & 1 & -\frac{1}{2} \end{bmatrix}$$

Figure 2.17 illustrates how one would apply these filters to an image, which is similar to how one would apply edge detection filters.

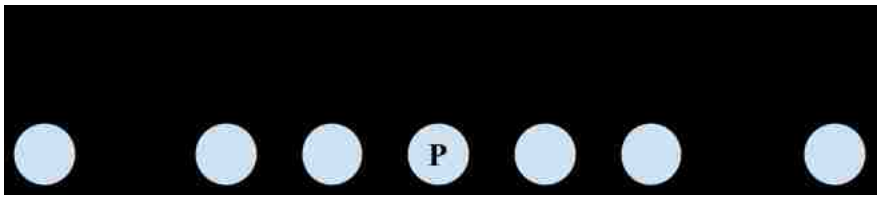


Figure 2.17: Applying the low-pass filter to an image. P is the current pixel.

To show that the low-pass filter passes low frequencies, and the high-pass filter passes high frequencies, we must derive their equations and substitute a set of values that range from 0 to N. We begin the derivation with Equation 2.6

$$A_m = \sum_{m=-N}^{N-1} x_n e^{-j2\pi mn/2N} = \sum_{m=-N}^{N-1} x_n e^{-j\pi mn/N} \tag{2.6}$$

where the summation for the low-pass filter is

$$\begin{aligned}
A_m &= x_{-2}e^{-j\pi m(-2)/N} + x_{-1}e^{-j\pi m(-1)/N} + x_0 e^{-j\pi m(0)/N} + x_1 e^{-j\pi m(1)/N} + x_2 e^{-j\pi m(2)/N} \\
&= -\frac{1}{8}e^{j2\pi m/N} + \frac{1}{4}e^{j\pi m/N} + \frac{3}{4} + \frac{1}{4}e^{-j\pi m/N} - \frac{1}{8}e^{-2j\pi m/N} \\
&= -\frac{1}{8}\left[e^{j2\pi m/N} + e^{-2j\pi m/N}\right] + \frac{1}{4}\left[e^{j\pi m/N} + \frac{1}{4}e^{-j\pi m/N}\right] + \frac{3}{4} \\
&= \frac{3}{4} - \frac{1}{8}\left[\cos\left(\frac{2\pi m}{N}\right) + j\sin\left(\frac{2\pi m}{N}\right) + \cos\left(\frac{2\pi m}{N}\right) - j\sin\left(\frac{2\pi m}{N}\right)\right] \\
&+ \frac{1}{4}\left[\cos\left(\frac{\pi m}{N}\right) + j\sin\left(\frac{\pi m}{N}\right) + \cos\left(\frac{\pi m}{N}\right) - j\sin\left(\frac{\pi m}{N}\right)\right] \\
&= \frac{3}{4} - \frac{1}{8}\left[2\cos\left(\frac{2\pi m}{N}\right)\right] + \frac{1}{4}\left[2\cos\left(\frac{\pi m}{N}\right)\right] \\
&= \frac{3}{4} + \frac{1}{2}\cos\left(\frac{\pi m}{N}\right) - \frac{1}{4}\cos\left(\frac{2\pi m}{N}\right)
\end{aligned}$$

and the summation for the high-pass filter is

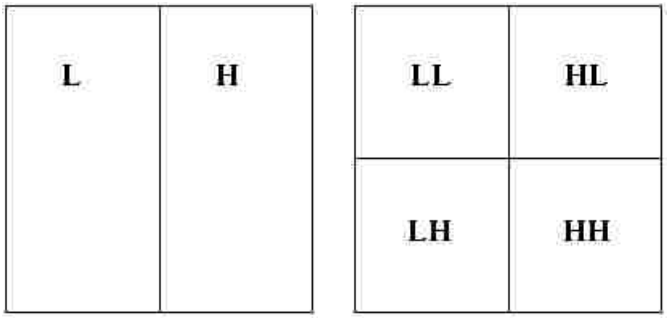
$$\begin{aligned}
A_m &= x_{-1}e^{-j\pi m(-1)/N} + x_0 e^{-j\pi m(0)/N} + x_1 e^{-j\pi m(1)/N} \\
&= -\frac{1}{2}e^{j\pi m/N} + e^{j\pi m(0)/N} + \frac{1}{2}e^{-j\pi m/N} \\
&= 1 - \frac{1}{2}\left[e^{j\pi m/N} + e^{-j\pi m/N}\right] \\
&= 1 - \frac{1}{2}\left[\cos\left(\frac{\pi m}{N}\right) + j\sin\left(\frac{\pi m}{N}\right) + \cos\left(\frac{\pi m}{N}\right) - j\sin\left(\frac{\pi m}{N}\right)\right] \\
&= 1 - \frac{1}{2}\left[2\cos\left(\frac{\pi m}{N}\right)\right] \\
&= 1 - \cos\left(\frac{\pi m}{N}\right)
\end{aligned}$$

Notice that we substituted the complex exponential with Euler's equation. The next step is to substitute a set of values from 0 to N. Table 2.2 lists the results of the low-pass and high-pass filter; the values converge from 1 to 0 on the low-pass filter, and the values converge from 0 to 2 on the high-pass filter.

$m =$	Low pass	High pass
0	1	0
$N/8$	1.035	0.076
$N/4$	1.104	0.293
$3N/8$	0.765	0.617
$N/2$	1	1
$5N/8$	0.735	1.383
$3N/4$	0.396	1.707
$7N/8$	0.111	1.924
N	0	2

Table 2.2: Results of substituting m for specific values from 0 to N. Notice that the low-pass values converge to zero, and the high-pass values diverge away from zero.

For our application, we are given a low-pass filter, a high-pass filter, and an RGB image where each row and column of data are pixels. For each channel, apply the low-pass filter then the high-pass filter to each row of data. Place the new low-pass coefficients (L) and high-pass coefficients (H) side-by-side as shown in Figure 2.18a. The output size from each pass is half the original input size. If the original input size was odd, then the half would round up one to be even. Apply the low-pass filter then the high-pass filter to each column of data in L (low) and H (high) separately. Place the new low-pass coefficients (LL) and high-pass coefficients (LH) of L on top of one another. Do the same for the new sets of coefficients of H. Figure 2.18b illustrates the resulting output, which consists of four quadrants, or bands, where the LL (low-low) is the approximation, the LH (low-high) is the vertical detail, the HL (high-low) is the horizontal detail, and the HH (high-high) is the diagonal detail [GW08]. It is optional to repeat these steps on the LL quadrant of the current iteration to produce more subbands.



(a) Intermediate. (b) Final.

Figure 2.18: One level transform. There are two passes per level: (a) one row-wise, and (b) one column-wise.

Implementing the DWT using the filters in their current form has a disadvantage in that they are not reversible. Fortunately, they can be rewritten. The filters reproduce the equivalent forward formulas below. While reversing is unnecessary for our application, a feature of the rewritten formulas is that they are faster to compute because they can be computed with integer arithmetic and bit shifting instead of floating point arithmetic. We will not show the inverse formulas because we did not use them in our application. Below are the forward formulas for when $n = 10$, but they can be expanded for any even n:

$$
\begin{aligned}
z_0 &= x_1 - \frac{x_0 + x_2}{2} & y_0 &= x_0 + \frac{z_0}{2} \\
z_1 &= x_3 - \frac{x_2 + x_4}{2} & y_1 &= x_2 + \frac{z_0 + z_1}{4} \\
z_2 &= x_5 - \frac{x_4 + x_6}{2} & y_2 &= x_4 + \frac{z_1 + z_2}{4} \\
z_3 &= x_7 - \frac{x_6 + x_8}{2} & y_3 &= x_6 + \frac{z_2 + z_3}{4} \\
z_4 &= x_9 - \frac{x_8 + x_8}{2} = x_9 - x_8 & y_4 &= x_8 + \frac{z_3 + z_4}{4}
\end{aligned}
$$

Below are the forward formulas for when $n = 9$, but they can be expanded for any odd n:

$$
\begin{aligned}
z_0 &= x_1 - \frac{x_0 + x_2}{2} & y_0 &= x_0 + \frac{z_0}{2} \\
z_1 &= x_3 - \frac{x_2 + x_4}{2} & y_1 &= x_2 + \frac{z_0 + z_1}{4} \\
z_2 &= x_5 - \frac{x_4 + x_6}{2} & y_2 &= x_4 + \frac{z_1 + z_2}{4} \\
z_3 &= x_7 - \frac{x_6 + x_8}{2} & y_3 &= x_6 + \frac{z_2 + z_3}{4} \\
z_4 &= -\frac{1}{2}x_8 + x_7 - \frac{1}{2}x_6 = z_3 & y_4 &= x_8 + \frac{z_3 + z_3}{4} = x_8 + \frac{z_3}{2}
\end{aligned}
$$

2.6 Principal Component Analysis

The Principal Component Analysis (PCA) is a statistical procedure that uses eigen decomposition to compute the most meaningful basis that best re-expresses a data set to reveal hidden patterns and dynamics [Shl03]. Eigen decomposition is a matrix decomposition of a square matrix into eigenvectors and eigenvalues [Wei]. The square matrix we use is the variance-covariance matrix, which is a positive definite symmetric matrix. Note that for our case it is a positive definite symmetric matrix, but in general it is a positive *semi-definite* symmetric matrix. An important property of a symmetric matrix is that it is diagonalizable by a matrix of its orthonormal vectors [Shl03]. These vectors are the eigenvectors. Note that a diagonal matrix is one where all but the diagonal elements are zero. The vectors that make up the new basis are the eigenvectors, or principal components [Shl03].

The goal is to compute a set of eigenvectors and eigenvalues that satisfy the following equation

$$\Sigma x = \lambda x \tag{2.7}$$

where Σ is the covariance matrix, λ is the vector $[\lambda_0\lambda_1\lambda_2]$, and x is a matrix where each row is an eigenvector. It can be rewritten as

$$(\Sigma - \lambda I)x = 0 \tag{2.8}$$

But x cannot be zero, so $(\Sigma - \lambda I)$ must be

$$(\Sigma - \lambda I) = 0 \tag{2.9}$$

where λ is the vector $[\lambda_0\lambda_1\lambda_2]^T$, and I is the identity matrix. We start by computing the covariance matrix in Equation 2.9

$$\Sigma = \begin{bmatrix} \sigma_R^2 & \sigma_{RG} & \sigma_{RB} \\ \sigma_{RG} & \sigma_G^2 & \sigma_{GB} \\ \sigma_{RB} & \sigma_{GB} & \sigma_B^2 \end{bmatrix} \tag{2.10}$$

which is a matrix where the diagonal terms are the variances

$$\sigma_a^2 = \frac{1}{(R \times C) - 1} \sum_{j=0}^{R} \sum_{i=0}^{C} (a_{ij} - \overline{a})^2 \tag{2.11}$$

and the off-diagonal terms are the covariances

$$\sigma_{ab} = \frac{1}{(R \times C) - 1} \sum_{j=0}^{R} \sum_{i=0}^{C} (a_{ij} - \overline{a})(b_{ij} - \overline{b}) \tag{2.12}$$

where R is the number of rows and C is the number of columns. To diagonalize the covariance matrix, we solve for the determinant of the left-hand side of Equation 2.9:

$$|\Sigma - \lambda I| = 0 \tag{2.13}$$

Expanding Equation 2.13 produces the cubic polynomial in Equation 2.15:

$$\left| \begin{bmatrix} \sigma_{00}^2 & \sigma_{01} & \sigma_{02} \\ \sigma_{10} & \sigma_{11}^2 & \sigma_{12} \\ \sigma_{20} & \sigma_{21} & \sigma_{22}^2 \end{bmatrix} - \begin{bmatrix} \lambda_0 \\ \lambda_1 \\ \lambda_2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right| = 0$$

$$\left| \begin{bmatrix} \sigma_{00}^2 & \sigma_{01} & \sigma_{02} \\ \sigma_{10} & \sigma_{11}^2 & \sigma_{12} \\ \sigma_{20} & \sigma_{21} & \sigma_{22}^2 \end{bmatrix} - \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \right| = 0$$

$$\left| \begin{bmatrix} \sigma_{00}^2 - \lambda_0 & \sigma_{01} & \sigma_{02} \\ \sigma_{10} & \sigma_{11}^2 - \lambda_1 & \sigma_{12} \\ \sigma_{20} & \sigma_{21} & \sigma_{22}^2 - \lambda_2 \end{bmatrix} \right| = 0$$

$$\left| \begin{bmatrix} \sigma_{00}^2 - \lambda & \sigma_{01} & \sigma_{02} \\ \sigma_{10} & \sigma_{11}^2 - \lambda & \sigma_{12} \\ \sigma_{20} & \sigma_{21} & \sigma_{22}^2 - \lambda \end{bmatrix} \right| = 0 \tag{2.14}$$

$$-\lambda^3 + A\lambda^2 - B\lambda + C = 0 \tag{2.15}$$

where

$$A = \sigma_R^2 + \sigma_G^2 + \sigma_B^2$$
$$B = \sigma_{RB}^2 + \sigma_{RG}^2 + \sigma_{GB}^2 - \sigma_R^2\sigma_B^2 - \sigma_R^2\sigma_G^2 - \sigma_G^2\sigma_B^2$$
$$C = \sigma_R^2\sigma_B^2\sigma_G^2 + 2\sigma_{RG}\sigma_{RB}\sigma_{GB} - \sigma_{RB}^2\sigma_G^2 - \sigma_{RG}^2\sigma_B^2 - \sigma_{GB}^2\sigma_R^2$$

The roots of the polynomial are the eigenvalues, and they can be solved for the roots numerically. First, estimate a range, then find a turning point at which it goes from positive to negative or negative to positive, and lastly, interpolate between those two values until a certain threshold, or margin of error, is reached. The threshold we chose was 0.01. Each eigenvalue represents a dimension, or variable, in the data; therefore, there is an eigenvalue for red, green, and blue. One way to check if the eigenvalues are correct is to add them together, because the sum of the eigenvalues should equal the sum of the variances, or the trace, of the covariance matrix. We verified that they were equal for all samples.

Next, substitute the eigenvalues back into Equation 2.7 one at a time to get the eigenvectors.

This turns into a problem of solving a system of equations, with three unknown variables:

$$\Sigma x_i = \lambda_i x_i$$

$$\begin{bmatrix} \sigma_{00}^2 & \sigma_{01} & \sigma_{02} \\ \sigma_{10} & \sigma_{11}^2 & \sigma_{12} \\ \sigma_{20} & \sigma_{21} & \sigma_{22}^2 \end{bmatrix} \begin{bmatrix} x_{i0} \\ x_{i1} \\ x_{i2} \end{bmatrix} = \lambda_i \begin{bmatrix} x_{i0} \\ x_{i1} \\ x_{i2} \end{bmatrix}$$

so

$$\sigma_{00}^2 x_{i0} + \sigma_{01} x_{i1} + \sigma_{02} x_{i2} = \lambda_i x_{i0}$$
$$\sigma_{10} x_{i0} + \sigma_{11}^2 x_{i1} + \sigma_{12} x_{i2} = \lambda_i x_{i1}$$
$$\sigma_{20} x_{i0} + \sigma_{21} x_{i1} + \sigma_{22}^2 x_{i2} = \lambda_i x_{i2}$$

or

$$(\sigma_{00}^2 - \lambda_i) x_{i0} + \sigma_{01} x_{i1} + \sigma_{02} x_{i2} = 0$$
$$\sigma_{10} x_{i0} + (\sigma_{11}^2 - \lambda_i) x_{i1} + \sigma_{12} x_{i2} = 0$$
$$\sigma_{20} x_{i0} + \sigma_{21} x_{i1} + (\sigma_{22}^2 - \lambda_i) x_{i2} = 0$$

where $i = 0, 1, 2$.

Assume that x_{i0} is r, x_{i1} is g, x_{i2} is b, $(\sigma_{00}^2 - \lambda_{i0})$ is σ_{00}, and $(\sigma_{11}^2 - \lambda_{i1})$ is σ_{11}. We set $r = 1$ and solved for the rest. The formulas below are what we used to solve for each variable. The last step is to normalize the eigenvectors and sort them by decreasing eigenvalue.

$$r = 1 \tag{2.16}$$
$$g = \frac{\sigma_{02}\sigma_{10} - \sigma_{12}\sigma_{00}}{\sigma_{12}\sigma_{01} - \sigma_{02}\sigma_{11}} \tag{2.17}$$
$$b = \frac{-\sigma_{00} - \sigma_{01} y}{\sigma_{02}} \tag{2.18}$$

Recall that the principal components make up a new basis to represent the data. The method we used to transform the data from the old R, G, B basis to the new PC_1, PC_2, PC_3 basis is

$$D_n = E D_o \tag{2.19}$$

where D_o is the original $3 \times N$ data matrix such that each column represents a channel, D_n is the new data matrix, and E is 3×3 eigenvector matrix such that each row represents a principal component.

Chapter 3

Experimental Results

In this chapter, we will go over the results of each image processing technique described in Chapter 2. Only the DFT, DWT, and PCA were applied to the samples in Section 2.1. The Gamma Distribution was applied to a single sample set, and the Mahalanobis distance was applied to the groups of training data.

3.1 Mahalanobis Distance

To develop a classification rule, we needed a training sample of individuals from each population [Mai12]. For us, that meant we needed training samples of clean and dirty panels. As we mentioned in Chapter 2, we settled on three groups of training samples. The procedure we relied on was the Jackknife approach, whose results are described next. It allowed us to find outliers; for example, clean samples with a lighter blue color might be considered dirty because their data behaves similarly to a dirty panel, which means we should not include it with the other clean samples. The more overlap there is between the set of characteristics that distinguish samples from clean and dirty, the larger the probability of a classification error [Mai12]. This was another important concept that was taken care of by the testing method, because it allowed us to filter out samples that barely made the cut; for example, a clean sample could have been correctly classified, but it could have easily been misclassified if the dirty and clean samples were slightly different, because sometimes taking out, adding in, or substiting an image would upset the balance. However, we also needed to pay attention to samples that were far from the threshold, known as outliers, because they were too good to be a true negative or positive.

3.1.1 Jackknife

We tested Theorems 2.2.2 and 2.2.3 by using the Jackknife approach on the clean sample set and the dirty sample set of each group. We calculated the accuracy by using Equation 3.1:

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \tag{3.1}$$

as well as the misclassification error, which is the number of false negatives and false positives divided by the total number of classification vectors. Tables 3.1 and 3.2 list the results of applying the jackknife method on all three groups of data. Figures 3.1 to 3.4 illustrate the distribution of samples in group 1; Figures 3.5 to 3.8 illustrate the distribution of samples in group 2; and Figures 3.9 to 3.12 illustrate the distribution of samples in group 3. The data points in these figures are the classification vectors. Both theorems resulted in zero misclassification.

Group	TN	FP	TP	FN	Accuracy (%)	Misclass. Error
1	12	0	12	0	100	0
2	20	0	20	0	100	0
3	19	0	19	0	100	0

Table 3.1: Jackknife results of all three groups based on theorem 2.2.2.

Group	TN	FP	TP	FN	Accuracy (%)	Misclass. Error
1	12	0	12	0	100	0
2	20	0	20	0	100	0
3	19	0	19	0	100	0

Table 3.2: Jackknife results of all three groups based on theorem 2.2.3.

We performed another test on the theorems, but each classification vector was based on the mode RGB values rather than the average RGB values. Tables 3.3 and 3.4 list the results of applying the jackknife method on all three groups of data. Figures 3.13 to 3.16 illustrate the distribution of samples in group 1; Figures 3.17 to 3.20 illustrate the distribution of samples in group 2; and Figures 3.21 to 3.24 illustrate the distribution of samples in group 3. Unlike the results based on the average, group 2 has one misclassified sample for Theorem 2.2.2, and group 3 has three misclassified samples for Theorems 2.2.2 and 2.2.3.

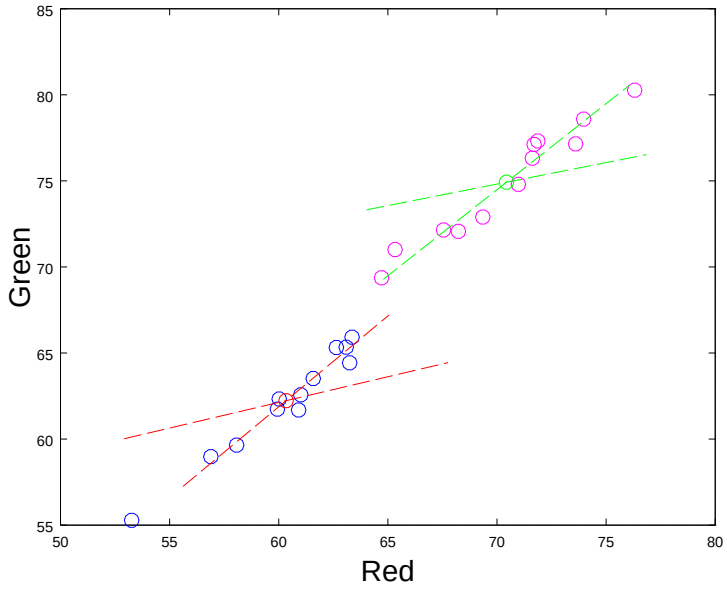


Figure 3.1: Two dimensional view of red and green averages for group 1. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The vectors that cross the centroids are the first and second principal components.

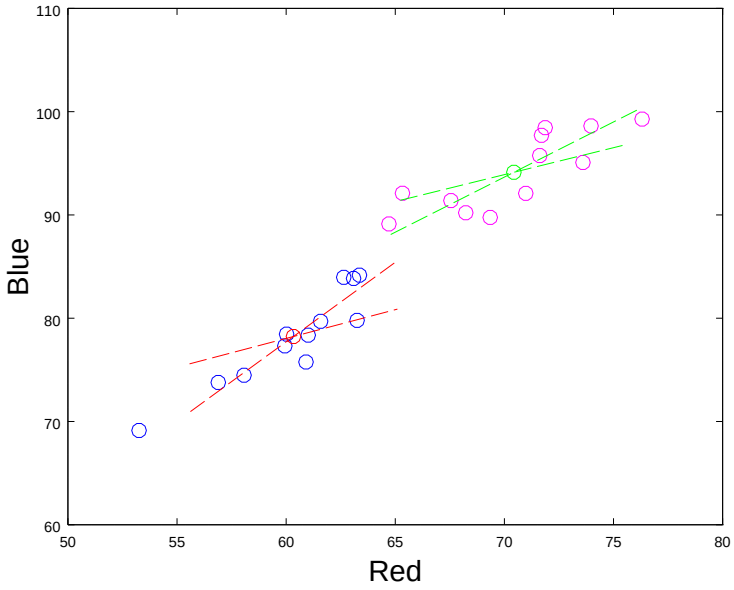


Figure 3.2: Two dimensional view of red and blue averages for group 1. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The vectors that cross the centroids are the first and third principal components.

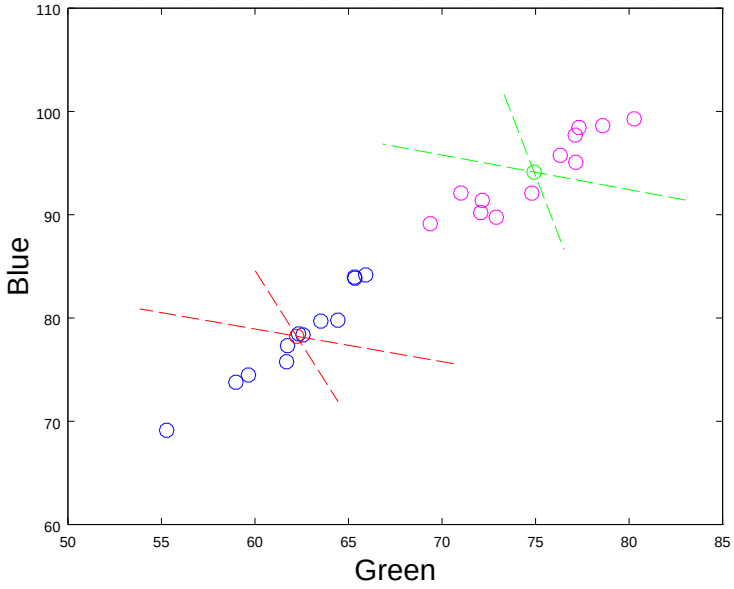


Figure 3.3: Two dimensional view of green and blue averages for group 1. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The vectors that cross the centroids are the second and third principal components.

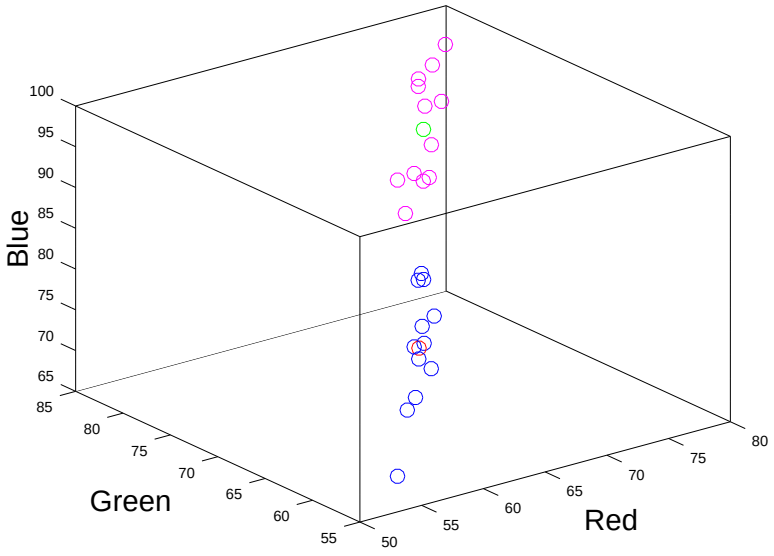


Figure 3.4: Three dimensional view of red, green, and blue averages for group 1. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively.

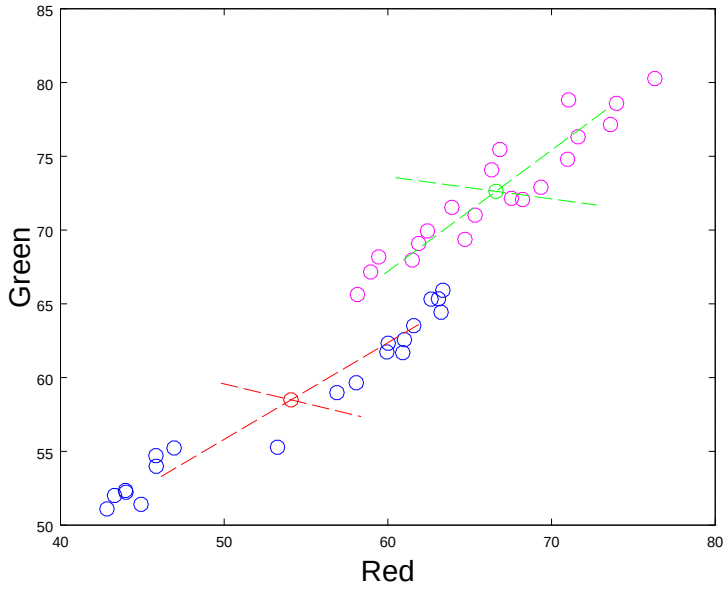


Figure 3.5: Two dimensional view of red and green averages for group 2. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The vectors that cross the centroids are the first and second principal components.

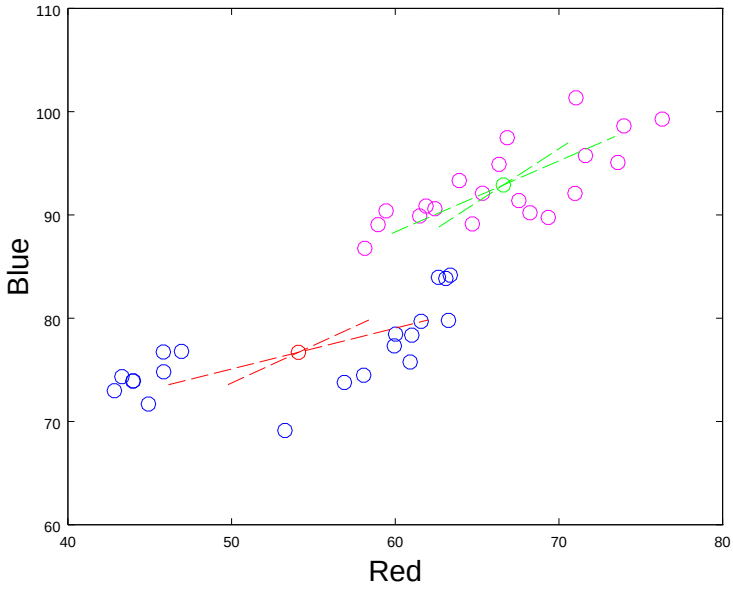


Figure 3.6: Two dimensional view of red and blue averages for group 2. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The vectors that cross the centroids are the first and third principal components.

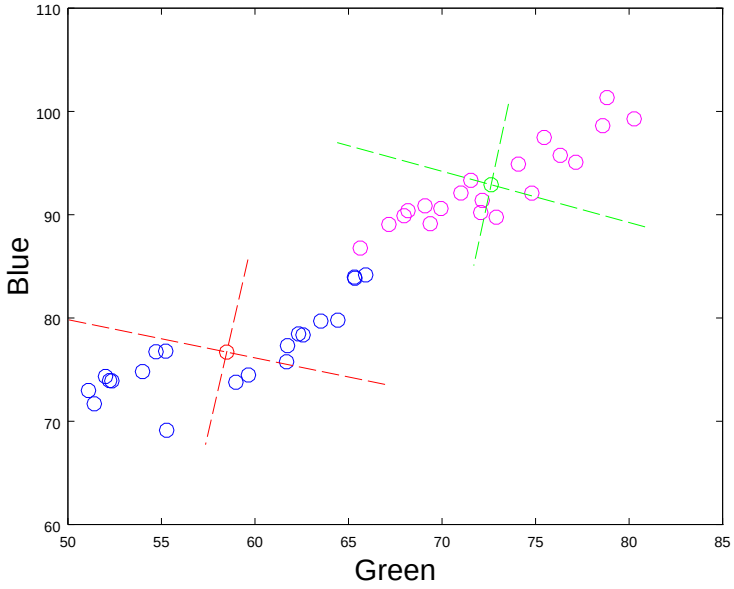


Figure 3.7: Two dimensional view of green and blue averages for group 2. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The vectors that cross the centroids are the second and third principal components.

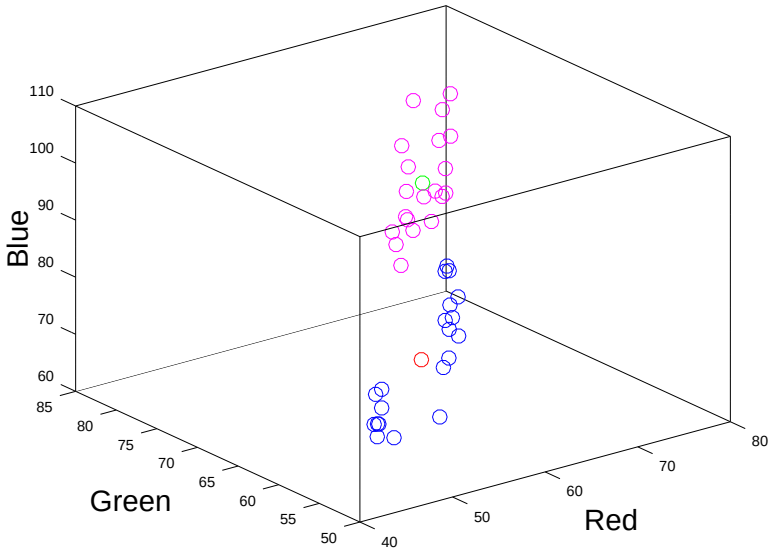


Figure 3.8: Three dimensional view of red, green, and blue averages for group 2. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively.

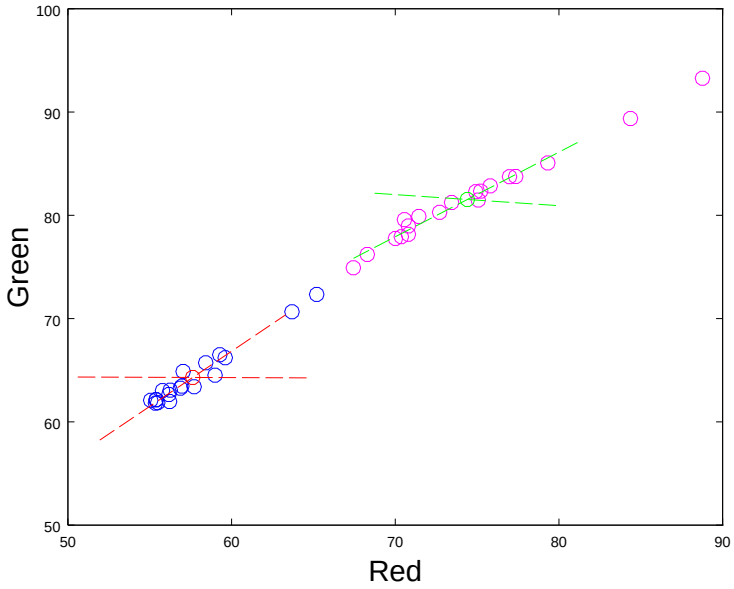


Figure 3.9: Two dimensional view of red and green averages for group 3. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The vectors that cross the centroids are the first and second principal components.

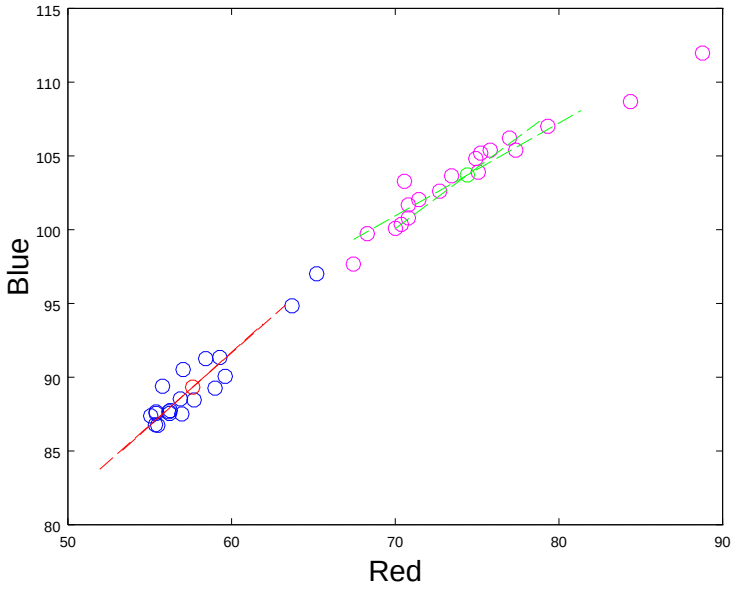


Figure 3.10: Two dimensional view of red and blue averages for group 3. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The vectors that cross the centroids are the first and third principal components.

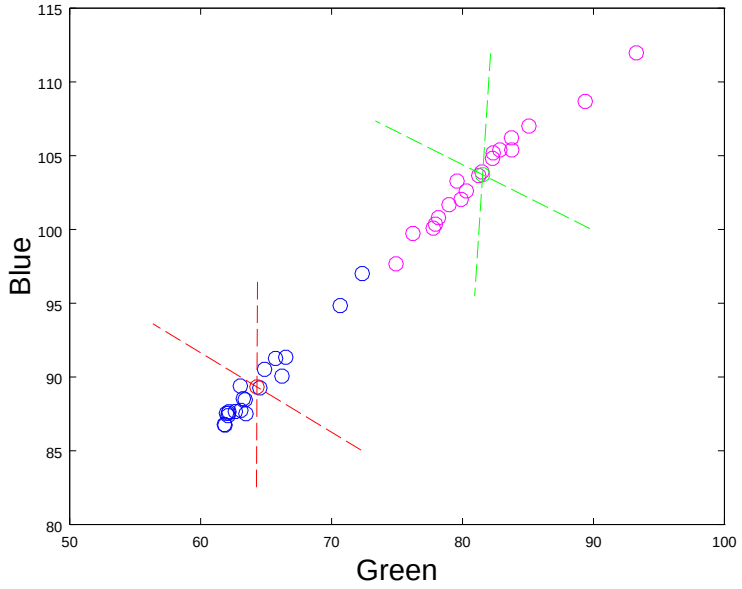


Figure 3.11: Two dimensional view of green and blue averages for group 3. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The vectors that cross the centroids are the second and third principal components.

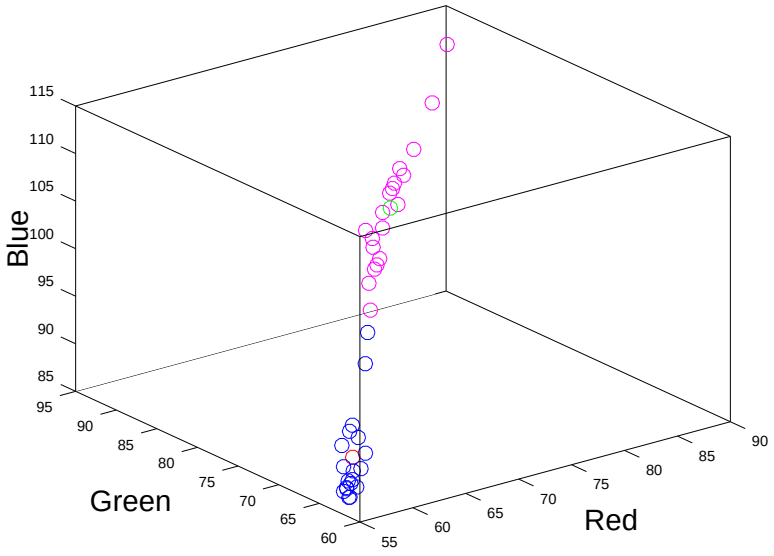


Figure 3.12: Three dimensional view of red, green, and blue averages for group 3. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively.

Group	TN	FP	TP	FN	Accuracy (%)	Misclass. Error
1	12	0	12	0	100	0
2	20	0	20	0	100	0
3	17	2	18	1	92.1	0.0789

Table 3.3: Jackknife results of all three groups based on theorem 2.2.2. However, each classification vector is based on the mode RGB values instead of the average RGB values.

Group	TN	FP	TP	FN	Accuracy (%)	Misclass. Error
1	12	0	12	0	100	0
2	20	0	19	1	97.5	0.025
3	17	2	18	1	92.1	0.0789

Table 3.4: Jackknife results of all three groups based on theorem 2.2.3. However, each classification vector is based on the mode RGB values instead of the average RGB values.

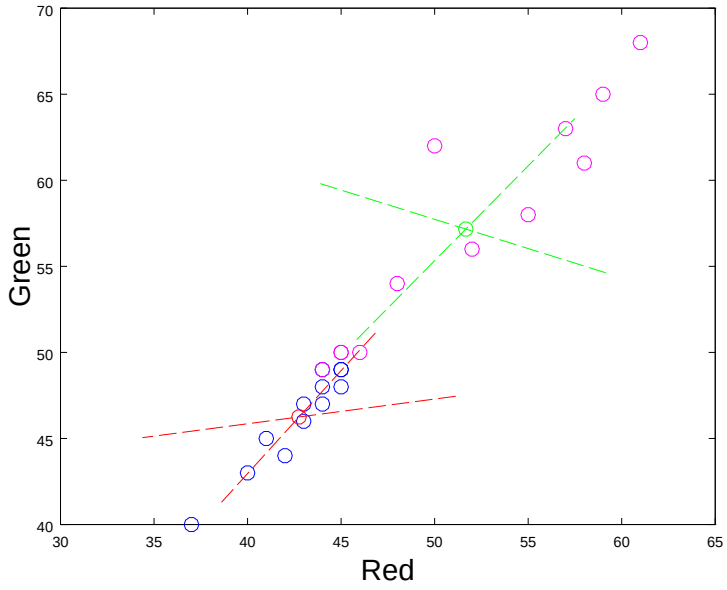


Figure 3.13: Two dimensional view of red and green modes for group 1. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The vectors that cross the centroids are the first and second principal components.

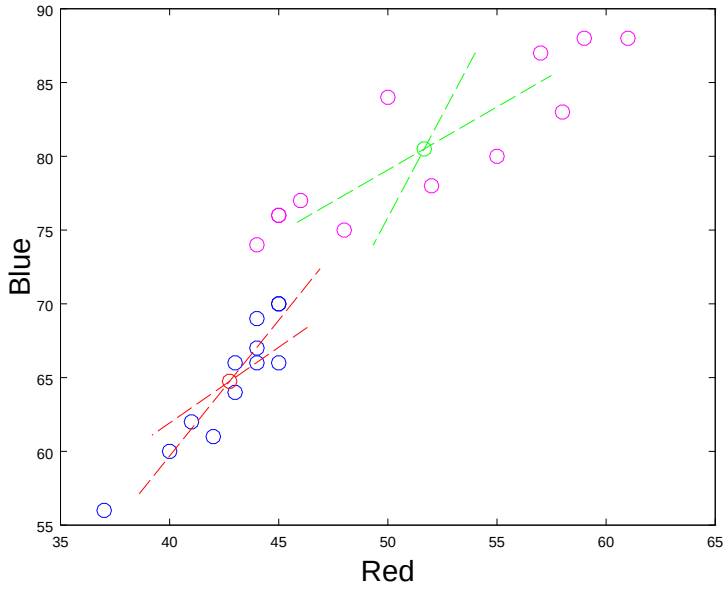


Figure 3.14: Two dimensional view of red and blue modes for group 1. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The vectors that cross the centroids are the first and third principal components.

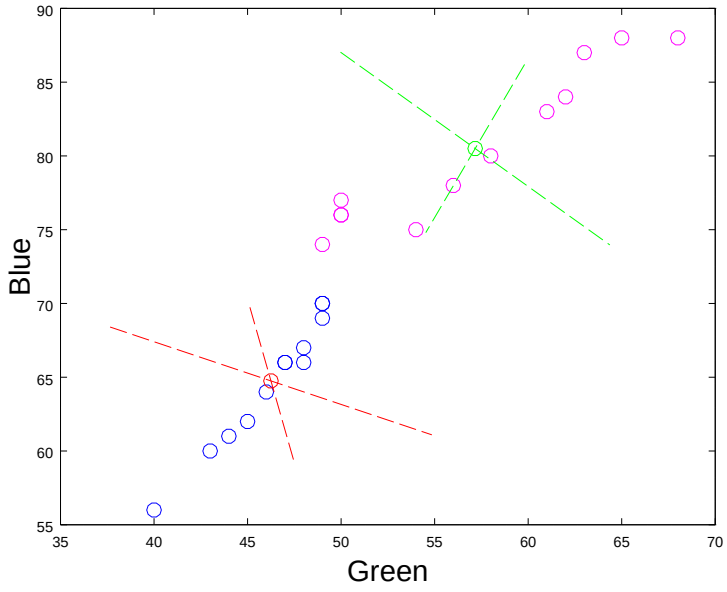


Figure 3.15: Two dimensional view of green and blue modes for group 1. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The vectors that cross the centroids are the second and third principal components.

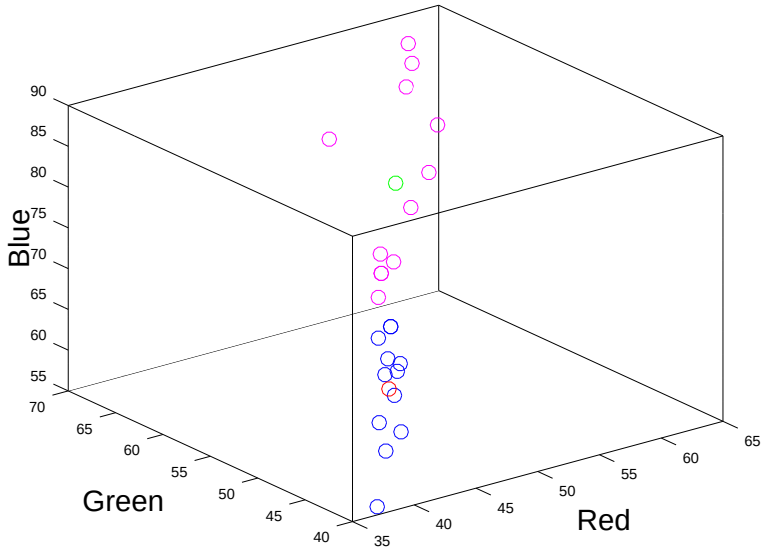


Figure 3.16: Three dimensional view of red, green, and blue modes for group 1. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively.

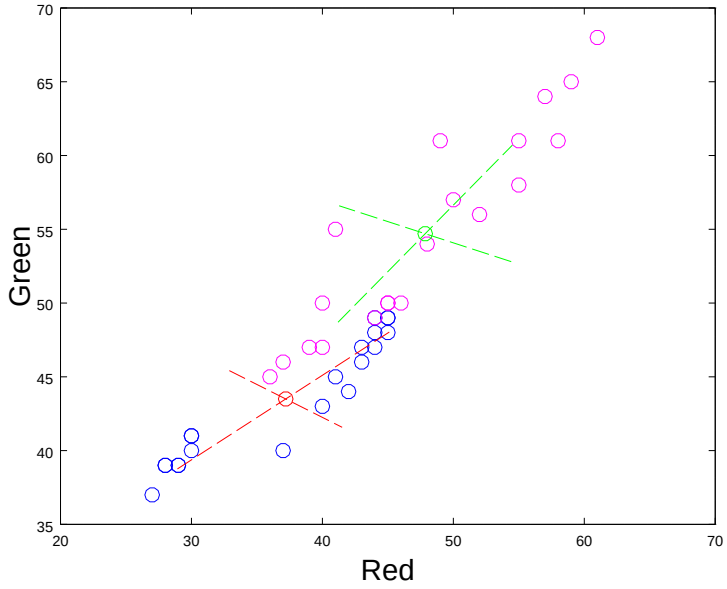


Figure 3.17: Two dimensional view of red and green modes for group 2. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The vectors that cross the centroids are the first and second principal components.

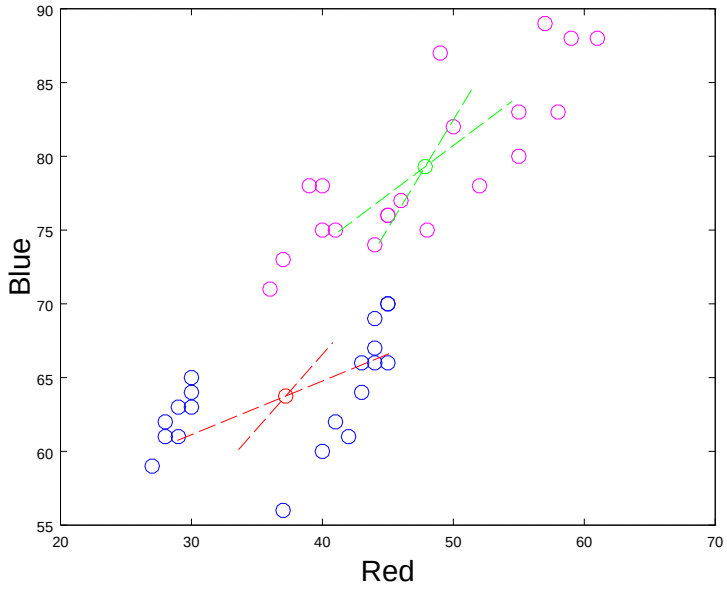


Figure 3.18: Two dimensional view of red and blue modes for group 2. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The vectors that cross the centroids are the first and third principal components.

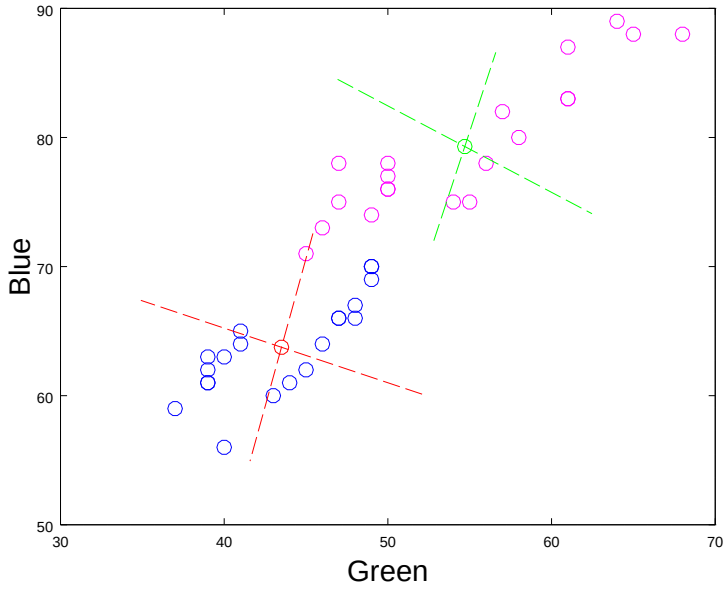


Figure 3.19: Two dimensional view of green and blue modes for group 2. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The vectors that cross the centroids are the second and third principal components.

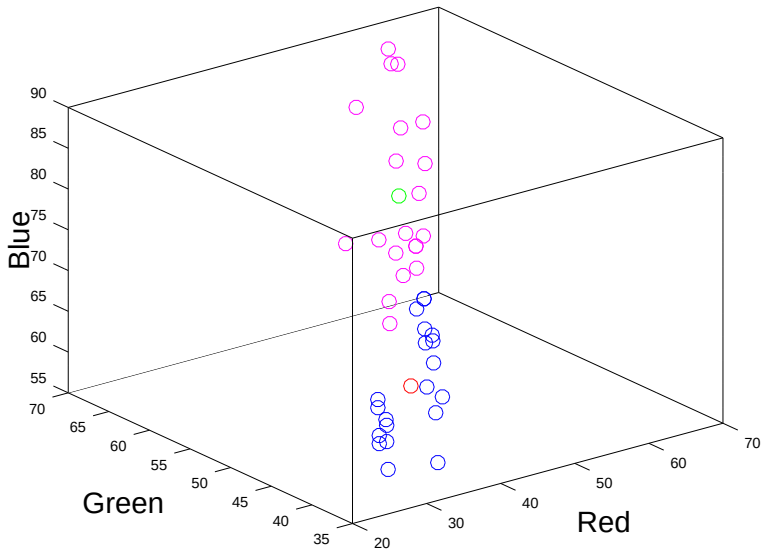


Figure 3.20: Three dimensional view of red, green, and blue modes for group 2. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The intersection here accounts for the misclassified clean sample.

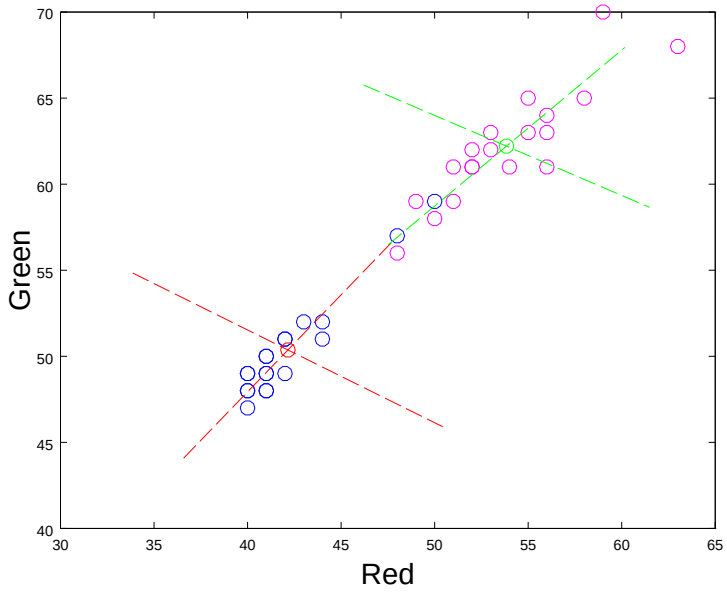


Figure 3.21: Two dimensional view of red and green modes for group 3. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The vectors that cross the centroids are the first and second principal components.

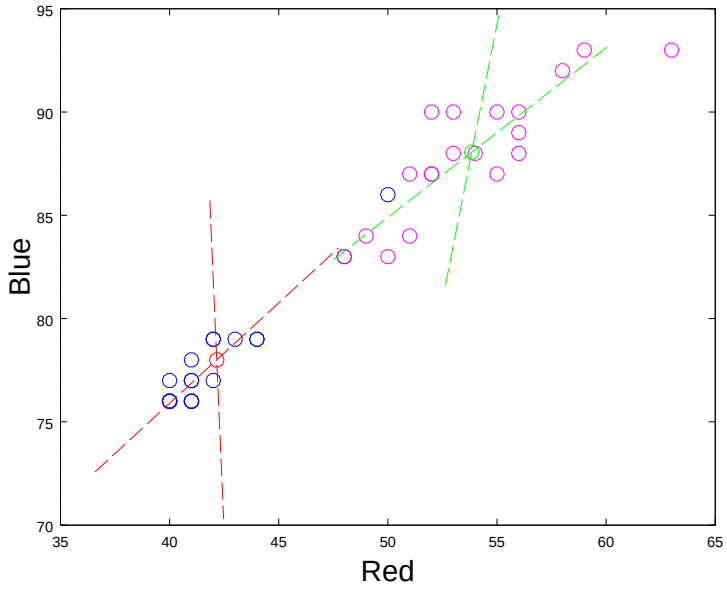


Figure 3.22: Two dimensional view of red and blue modes for group 3. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The vectors that cross the centroids are the first and third principal components.

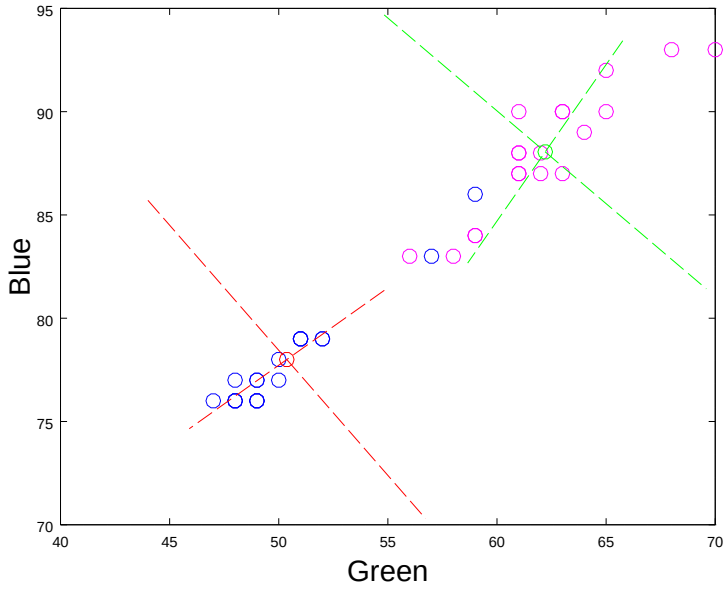


Figure 3.23: Two dimensional view of green and blue modes for group 3. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The vectors that cross the centroids are the second and third principal components.

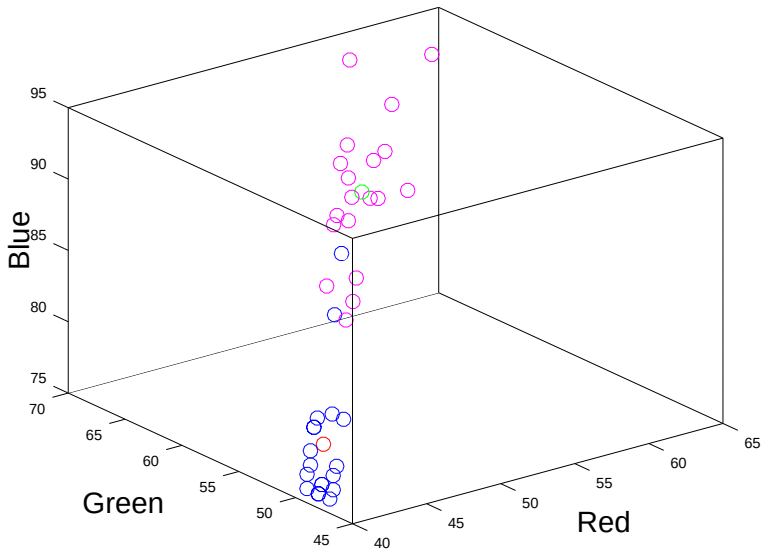


Figure 3.24: Three dimensional view of red, green, and blue modes for group 3. The blue and magenta circles are clean and dirty samples, respectively. The red and green circle are centroids of the clean and dirty samples, respectively. The overlap is clear, and there is no question on which two clean samples are misclassified.

Recall that we used the jackknife method to gather our training data, and we noted some of the advantages of doing so, for instance, being aware of outliers and points that are susceptible to misclassification. The following tables list the classification vector and the Mahalanobis distances of each sample in group 3. We specifically chose to list the results of group 3 only, because it is the only group with consistent misclassification, while the rest have nothing to show because they have perfect accuracy. Tables 3.5 to 3.8 list the results of Theorems 2.2.2 and 2.2.3 where the classification vectors are based on the average RGB values. Tables 3.9 to 3.12 list the results of both theorems where the classification vectors are based on the mode RGB values. When the classification vectors are based on the mode, the difference in distances are similar for both theorems, but when the classification vectors are the mean, the difference in distances of theorem 2 is bigger, and the difference in distances of theorem 3 is smaller to the point where the samples are at the cusp of misclassification.

i	$x_m = RGB$ average vector	v_1 = clean		v_2 = dirty	Classification
20	56.21 61.97 87.54	2.35706526151629	$<$	10.1848483003662	CLEAN
21	63.69 70.66 94.84	2.44002257389201	$<$	4.8587556664668	CLEAN
22	65.2 72.34 97.01	3.54869806284974	$<$	4.41271760153356	CLEAN
23	55.78 63.03 89.39	2.68233330006873	$<$	7.96795065739642	CLEAN
24	58.42 65.72 91.26	1.34651299605151	$<$	6.6040652004053	CLEAN
25	57.04 64.88 90.52	2.41782880238462	$<$	6.15767743248276	CLEAN
26	59.28 66.5 91.34	1.10405394434877	$<$	6.16037253004963	CLEAN
27	57.71 63.4 88.46	1.99057702124235	$<$	9.60751531077478	CLEAN
28	58.99 64.52 89.26	2.4382975634377	$<$	9.36498647893965	CLEAN
29	56.87 63.26 88.53	0.727713407293908	$<$	8.65856974460388	CLEAN
30	56.17 62.65 87.66	0.80165177265945	$<$	8.67208783897508	CLEAN
31	55.35 61.81 86.78	1.19576607047464	$<$	8.98653464580726	CLEAN
32	55.39 62.12 87.55	1.01484789375189	$<$	8.58895779090618	CLEAN
33	55.49 61.86 86.74	1.26030256399803	$<$	9.06830990035708	CLEAN
34	55.39 62.16 87.65	1.05620996651935	$<$	8.53854769454094	CLEAN
35	56.25 63.07 87.74	1.49141971665988	$<$	7.97233421511727	CLEAN
36	55.06 62.09 87.38	1.45335801772928	$<$	8.12760549552636	CLEAN
37	59.61 66.21 90.06	2.33355269490492	$<$	6.99351892962558	CLEAN
38	56.96 63.46 87.51	2.49239813395392	$<$	8.21528438385842	CLEAN

Table 3.5: Jackknife on clean sample set of group 3. These results are based on Theorem 2.2.2, where the classification vectors are the average RGB values.

i	$x_m = RGB$ average vector	v_1 = dirty		v_2 = clean	Classification
0	76.98 83.74 106.21	1.3613499811059	$<$	6.59730723463006	DIRTY
1	75.8 82.85 105.39	0.959187029413874	$<$	6.18709194493406	DIRTY
2	75.22 82.34 105.19	1.34634596101952	$<$	5.92237533861357	DIRTY
3	75.07 81.48 103.9	1.74031966653875	$<$	6.09036260008453	DIRTY
4	77.36 83.76 105.39	0.710726145075818	$<$	7.11090810698612	DIRTY
5	88.77 93.27 111.97	4.07444994655178	$<$	12.4931585417989	DIRTY
6	84.37 89.37 108.68	2.96792308128091	$<$	10.8191453691049	DIRTY
7	79.32 85.07 107.01	2.26394265608027	$<$	7.81002737288318	DIRTY
8	72.71 80.29 102.61	0.664949701968289	$<$	5.68156817943284	DIRTY
9	68.29 76.21 99.73	1.61723153332615	$<$	4.08535627935491	DIRTY
10	70.8 78.16 100.8	1.34334199707529	$<$	4.91443448620639	DIRTY
11	71.44 79.89 102.04	2.64522475342059	$<$	6.3273109985509	DIRTY
12	70.01 77.76 100.09	1.70395002891043	$<$	5.36379536518132	DIRTY
13	70.8 78.97 101.68	1.17885658888482	$<$	5.39819792644326	DIRTY
14	70.37 77.93 100.36	1.47491652806493	$<$	5.15970287838447	DIRTY
15	67.44 74.92 97.67	3.15516904568329	$<$	4.36995154262149	DIRTY
16	70.56 79.58 103.28	2.878891784708	$<$	5.51232695583294	DIRTY
17	73.44 81.24 103.65	0.909319855165994	$<$	5.85916152583036	DIRTY
18	74.92 82.29 104.82	0.689526745436323	$<$	5.98021503664103	DIRTY

Table 3.6: Jackknife on dirty sample set of group 3. These results are based on Theorem 2.2.2, where the classification vectors are the average RGB values.

i	$x_m = RGB$ average vector	v_1 = clean		v_2 = dirty	Classification
20	56.21 61.97 87.54	1.22614288716497	$<$	1.73019844591296	CLEAN
21	63.69 70.66 94.84	0.392600326408559	$<$	0.623190678684538	CLEAN
22	65.2 72.34 97.01	1.55359671596084	$<$	1.58421870815981	CLEAN
23	55.78 63.03 89.39	1.70100628961979	$<$	1.98673996972909	CLEAN
24	58.42 65.72 91.26	1.05092403617618	$<$	1.33993912884979	CLEAN
25	57.04 64.88 90.52	1.31296991110775	$<$	1.52295194631611	CLEAN
26	59.28 66.5 91.34	0.574438048621323	$<$	0.938174314717332	CLEAN
27	57.71 63.4 88.46	1.09643932366743	$<$	1.58372165523264	CLEAN
28	58.99 64.52 89.26	1.27277405327129	$<$	1.68589616205805	CLEAN
29	56.87 63.26 88.53	0.328214983301313	$<$	1.10589806216841	CLEAN
30	56.17 62.65 87.66	0.592205236424604	$<$	1.20068375947045	CLEAN
31	55.35 61.81 86.78	0.927360089696872	$<$	1.4182600403219	CLEAN
32	55.39 62.12 87.55	0.247716925781373	$<$	1.08362513059613	CLEAN
33	55.49 61.86 86.74	1.04861696656753	$<$	1.50297854020997	CLEAN
34	55.39 62.16 87.65	0.226065360859211	$<$	1.07676013175849	CLEAN
35	56.25 63.07 87.74	1.2997398503741	$<$	1.60736535154831	CLEAN
36	55.06 62.09 87.38	0.771044564161142	$<$	1.26190113908403	CLEAN
37	59.61 66.21 90.06	1.66835752897789	$<$	1.83418543891725	CLEAN
38	56.96 63.46 87.51	2.18597778435758	$<$	2.37278598448897	CLEAN

Table 3.7: Jackknife on clean sample set of group 3. These results are based on Theorem 2.2.3, where the classification vectors are the average RGB values.

i	$x_m = RGB$ average vector	v_1 = dirty		v_2 = clean	Classification
0	76.98 83.74 106.21	1.44947265108666	$<$	1.71206777250281	DIRTY
1	75.8 82.85 105.39	1.0064304355654	$<$	1.36410225350444	DIRTY
2	75.22 82.34 105.19	1.37786023562976	$<$	1.63593857991641	DIRTY
3	75.07 81.48 103.9	1.06907010424383	$<$	1.30768988811768	DIRTY
4	77.36 83.76 105.39	0.834026498264436	$<$	1.24748658787489	DIRTY
5	88.77 93.27 111.97	3.98784229889776	$<$	4.1978633663645	DIRTY
6	84.37 89.37 108.68	3.27982521803679	$<$	3.47585459582169	DIRTY
7	79.32 85.07 107.01	2.26499223294416	$<$	2.41042115177014	DIRTY
8	72.71 80.29 102.61	0.786398267377804	$<$	1.22023851250614	DIRTY
9	68.29 76.21 99.73	1.03067153262497	$<$	1.24322992635778	DIRTY
10	70.8 78.16 100.8	0.642345049588495	$<$	1.00385980045734	DIRTY
11	71.44 79.89 102.04	2.32797402665849	$<$	2.57116857539001	DIRTY
12	70.01 77.76 100.09	1.72776788921921	$<$	1.92921670769973	DIRTY
13	70.8 78.97 101.68	1.40795322904159	$<$	1.69768795165935	DIRTY
14	70.37 77.93 100.36	1.26963332196892	$<$	1.51145227450516	DIRTY
15	67.44 74.92 97.67	1.58243697031374	$<$	1.69331246123673	DIRTY
16	70.56 79.58 103.28	2.38726418536442	$<$	2.60207117529912	DIRTY
17	73.44 81.24 103.65	0.777022032976213	$<$	1.26279038609913	DIRTY
18	74.92 82.29 104.82	0.626201176094157	$<$	1.14219144218767	DIRTY

Table 3.8: Jackknife on dirty sample set of group 3. These results are based on Theorem 2.2.3, where the classification vectors are the average RGB values.

i	$x_m = RGB$ mode vector	v_1 = clean		v_2 = dirty	Classification
20	40 47 76	2.08870112896722	$<$	4.60930955775586	CLEAN
21	48 57 83	2.92245422626679	$>$	1.76158599614127	DIRTY
22	50 59 86	4.13315054050489	$>$	1.2781671780659	DIRTY
23	41 50 78	1.65668474395726	$<$	3.71500504110027	CLEAN
24	42 51 79	1.63804752563811	$<$	3.43884509146719	CLEAN
25	42 51 79	1.63804752563811	$<$	3.43884509146719	CLEAN
26	44 52 79	1.24198560279261	$<$	3.13715412107146	CLEAN
27	41 48 77	2.15624230025714	$<$	4.33558288764729	CLEAN
28	44 51 79	1.82875646626716	$<$	3.46183674925202	CLEAN
29	40 49 77	1.81510006459549	$<$	3.99703908344801	CLEAN
30	40 48 76	0.963062908472878	$<$	4.26250720786527	CLEAN
31	40 48 76	0.963062908472878	$<$	4.26250720786527	CLEAN
32	41 50 77	1.62308927111597	$<$	3.74592052779846	CLEAN
33	40 49 76	2.05348373106815	$<$	4.05052587233422	CLEAN
34	42 49 77	1.56236115385263	$<$	4.03528499815245	CLEAN
35	41 48 76	1.64924538559342	$<$	4.32808373481987	CLEAN
36	41 49 76	1.87486582365815	$<$	4.05947121482065	CLEAN
37	43 52 79	1.07688438451357	$<$	3.1433419122968	CLEAN
38	41 49 76	1.87486582365815	$<$	4.05947121482065	CLEAN

Table 3.9: Jackknife on clean sample set of group 3. These results are based on Theorem 2.2.3, where the classification vectors are the mode RGB values.

i	$x_m = RGB$ mode vector	v_1 = dirty		v_2 = clean	Classification
0	55 65 90	1.2402117277138	$<$	4.5472102535774	DIRTY
1	53 63 90	1.77312979145326	$<$	4.43264343100751	DIRTY
2	58 65 92	1.58460182236632	$<$	6.15962913992373	DIRTY
3	52 62 87	1.14332725886376	$<$	3.82973273081517	DIRTY
4	52 61 87	0.639676193982939	$<$	3.3199167286041	DIRTY
5	63 68 93	3.56271120084	$<$	9.50537684924153	DIRTY
6	59 70 93	3.91777170830362	$<$	6.69421440547089	DIRTY
7	53 62 88	0.463553267680182	$<$	3.64771147260397	DIRTY
8	55 63 87	1.67113843583844	$<$	6.10828120515683	DIRTY
9	51 59 84	1.7820329465295	$<$	4.39603036162401	DIRTY
10	48 56 83	2.22931843878967	$>$	2.02620414602682	CLEAN
11	56 61 88	2.37095895168415	$<$	6.85959928315677	DIRTY
12	54 61 88	1.09471232426117	$<$	4.67562435558731	DIRTY
13	51 61 87	1.16733073918263	$<$	3.24124856435369	DIRTY
14	50 58 83	2.17232578042924	$<$	4.30133431504125	DIRTY
15	49 59 84	1.85529032089294	$<$	3.42411499207017	DIRTY
16	52 61 90	3.30921892798612	$<$	5.85007570486638	DIRTY
17	56 64 89	0.757409877765688	$<$	5.34381993616665	DIRTY
18	56 63 90	1.14933637400672	$<$	5.40550404856429	DIRTY

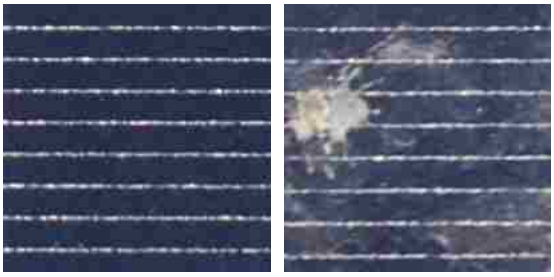
Table 3.10: Jackknife on dirty sample set of group 3. These results are based on Theorem 2.2.3, where the classification vectors are the mode RGB values.

i	$x_m = RGB$ mode vector	v_1 = clean		v_2 = dirty	Classification
20	40 47 76	1.57287741868849	$<$	4.80839907497174	CLEAN
21	48 57 83	2.19954872439612	$>$	1.99341356438361	DIRTY
22	50 59 86	3.03637415490451	$>$	1.56251183964729	DIRTY
23	41 50 78	0.853083855195661	$<$	3.97302012187084	CLEAN
24	42 51 79	0.870537954503574	$<$	3.69314324341758	CLEAN
25	42 51 79	0.870537954503574	$<$	3.69314324341758	CLEAN
26	44 52 79	0.703991794680023	$<$	3.22623837174443	CLEAN
27	41 48 77	1.43382475332139	$<$	4.49984201190484	CLEAN
28	44 51 79	0.966828299567452	$<$	3.50345128917664	CLEAN
29	40 49 77	0.977511151215543	$<$	4.27208244954977	CLEAN
30	40 48 76	0.901589482262144	$<$	4.42473119067707	CLEAN
31	40 48 76	0.901589482262144	$<$	4.42473119067707	CLEAN
32	41 50 77	0.939002694361858	$<$	3.96569652105976	CLEAN
33	40 49 76	1.2042579695218	$<$	4.304631694314	CLEAN
34	42 49 77	1.04721272786961	$<$	4.09458661207538	CLEAN
35	41 48 76	1.25477351107931	$<$	4.41930490163166	CLEAN
36	41 49 76	1.17572730386569	$<$	4.20020992642732	CLEAN
37	43 52 79	0.730915265820127	$<$	3.34178991198043	CLEAN
38	41 49 76	1.17572730386569	$<$	4.20020992642732	CLEAN

Table 3.11: Jackknife on clean sample set of group 3. These results are based on Theorem 2.2.3, where the classification vectors are the mode RGB values.

i	$x_m = RGB$ mode vector	v_1 = dirty		v_2 = clean	Classification
0	55 65 90	1.39770689873618	$<$	4.32348428956287	DIRTY
1	53 63 90	2.25338042484708	$<$	4.05558871710288	DIRTY
2	58 65 92	1.97701094956598	$<$	4.8027654509091	DIRTY
3	52 62 87	1.48532649512144	$<$	3.46809104939575	DIRTY
4	52 61 87	0.791668877622761	$<$	3.02602203687327	DIRTY
5	63 68 93	4.32385979414738	$<$	7.37185288566641	DIRTY
6	59 70 93	3.67077351839131	$<$	6.87130663697498	DIRTY
7	53 62 88	0.612083082810199	$<$	3.31877011954062	DIRTY
8	55 63 87	2.15933632488032	$<$	4.29833750182493	DIRTY
9	51 59 84	2.02158791622551	$<$	2.93712228395988	DIRTY
10	48 56 83	2.22581246519754	$>$	1.67728719278959	CLEAN
11	56 61 88	2.85837475746531	$<$	4.32437965823672	DIRTY
12	54 61 88	1.30774561006224	$<$	3.37139432244775	DIRTY
13	51 61 87	1.53155568945809	$<$	3.12978566311312	DIRTY
14	50 58 83	2.39331824188308	$<$	2.77128224609938	DIRTY
15	49 59 84	2.26772635734048	$<$	2.80485705555113	DIRTY
16	52 61 90	4.12107941977887	$<$	4.65210169249728	DIRTY
17	56 64 89	0.975509732890581	$<$	4.12025443804426	DIRTY
18	56 63 90	1.48319379041058	$<$	4.03899719393608	DIRTY

Table 3.12: Jackknife on dirty sample set of group 3. These results are based on Theorem 2.2.3, where the classification vectors are the mode RGB values.

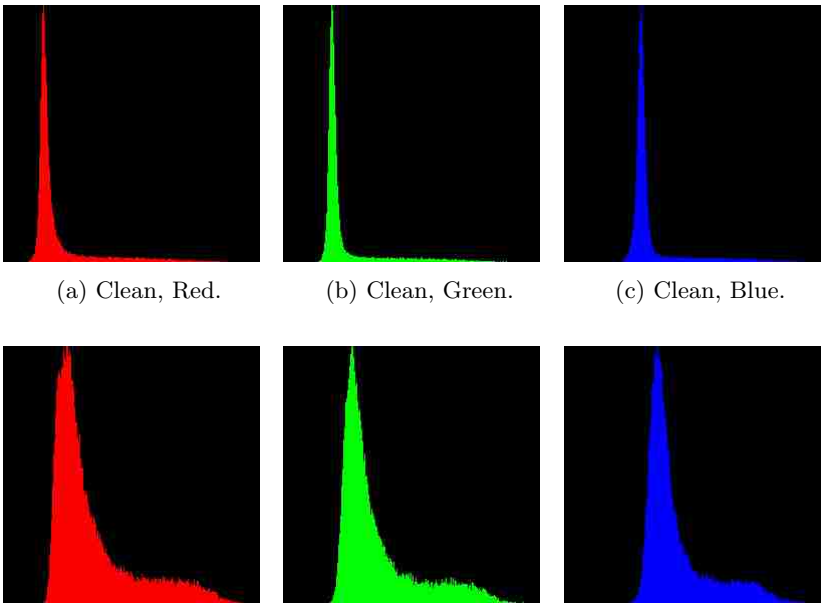


(a) Clean (b) Dirty

Figure 3.25: The sample pair on which the results are based.

3.2 Histogram Analysis

Figure 3.25 represents the sample pair we used to produce the results in Figure 3.26, which compares the probability density function of the channels of both samples. The empirical and global modes for the clean sample are 40 and 48 for the red, 48 and 45 for the green, and 76 and 72 for the blue. The modes for the dirty sample are 63 and 52 for the red, 68 and 75 for the green, and 93 and 95 for the blue. Notice how the modes increased in the dirty sample.



(a) Clean, Red. (b) Clean, Green. (c) Clean, Blue.

(d) Dirty, Red. (e) Dirty, Green. (f) Dirty, Blue.

Figure 3.26: Histograms of the sample pair.

For all groups of sample sets, we computed the histogram, probability density function, and gamma distribution. We compared the average mode from unnormalized histogram values, and the average alpha parameter, average beta parameter, and average global mode of the gamma distribution. Tables 3.13 to 3.15 list the results. Notice how the values are higher in the dirty

sample set of every group.

Channel	Red					
Group	1		2		3	
Category	C	D	C	D	C	D
α	4	4	3	4	4	4
β	18	19	17	19	15	20
x_m	44	51	39	49	44	53
Mode	43	52	37	48	42	54

Table 3.13: Probability density function and gamma distribution averages of the red channel.

Channel	Green					
Group	1		2		3	
Category	C	D	C	D	C	D
α	4	4	4	4	5	4
β	16	18	15	18	15	19
x_m	47	56	44	53	51	63
Mode	46	57	44	55	50	62

Table 3.14: Probability density function and gamma distribution averages of the green channel.

Channel	Blue					
Group	1		2		3	
Category	C	D	C	D	C	D
α	6	7	6	7	8	7
β	13	14	13	14	11	16
x_m	66	80	65	79	78	89
Mode	65	81	64	79	78	88

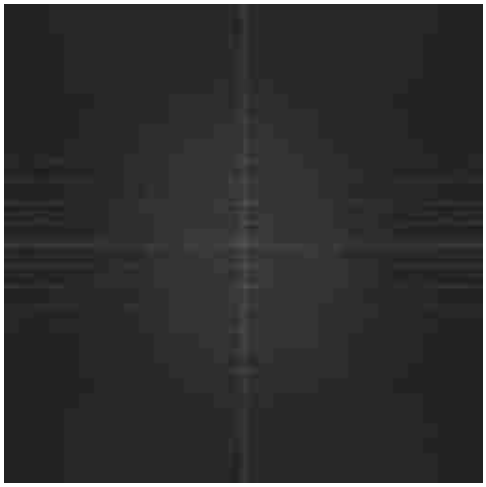
Table 3.15: Probability density function and gamma distribution averages of the blue channel.

In conclusion, we noticed a significant shift in red, green, and blue values to the right of the graphs. These observations, as well as the probability density function and the gamma density function values, are correlated to the amount of reflection. Clean panels reflect less and absorb more; therefore, they have low values. Dirty panels reflect more and absorb less; therefore, they have higher values.

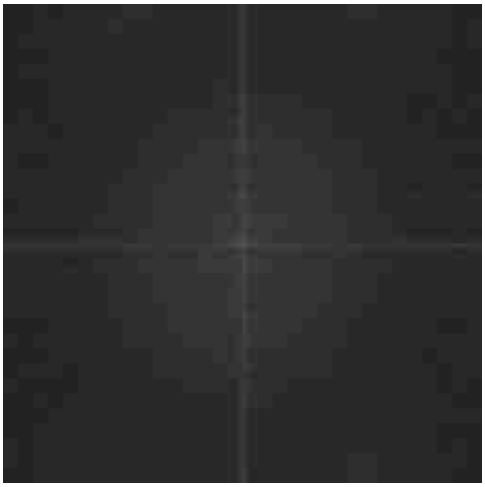
3.3 Discrete Fourier Transform

We applied the FFT to all the samples in Section 2.1 and compared their magnitudes. Recall from Section 2.4 that it is common to display only the magnitude of the results rather than also including the phase. However, the dynamic range of the Fourier coefficients, that is, the intensity values of the Fourier image, is too large to be displayed on the screen, and as a result, the coefficients are hard to visualize on an image. A common solution to help visualize the results better is to shift the values to where the frequency at position $(0, 0)$ is at the center before computing the FFT, and to apply a log transformation on the magnitude after computing the FFT. In addition, sometimes there is another step performed before the log transformation, and it is contrast stretching. Both contrast stretching and log transformation bring out more details and frequencies. Data that is closer to the origin have lower frequencies, and data that is further from the origin have higher frequencies [GW08].

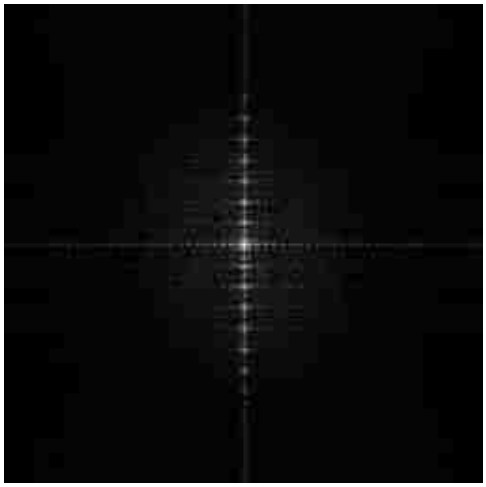
We graphed the results of all sample sets in two dimensions for both methods. Figure 3.27 displays the resulting images of group 1 sample set 1, which illustrate the issue present in the results for all sample sets. The figures show that it is difficult to determine if there is a significant difference between the clean and dirty sample when the results are plotted in two dimensions. Alternatively, we graphed the log transform results in three dimensions and saw a difference for all sample sets. For brevity, we only describe the results of one sample set. Figure 3.28 displays the results of group 1 sample set 1. For the clean sample, the frequencies range from very high to very low values, which are represented by red and blue, respectively. For the dirty sample, the frequency range is not as wide, but the values are more spread out in between, which is represented, from high to low, by orange, yellow, green, and light blue.



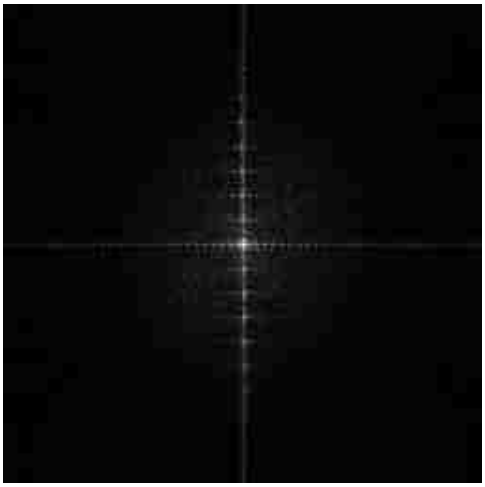
(a) Clean RGB.



(b) Dirty RGB.

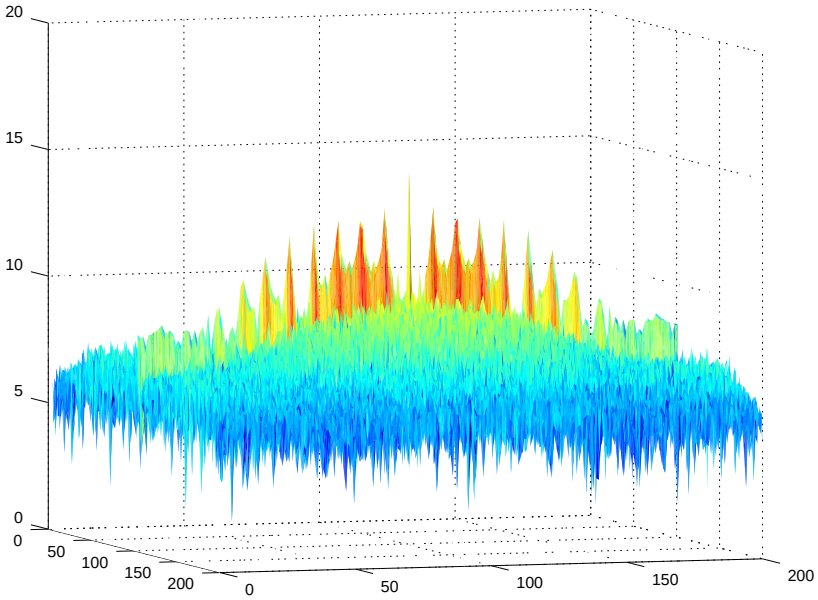


(c) Clean RGB.

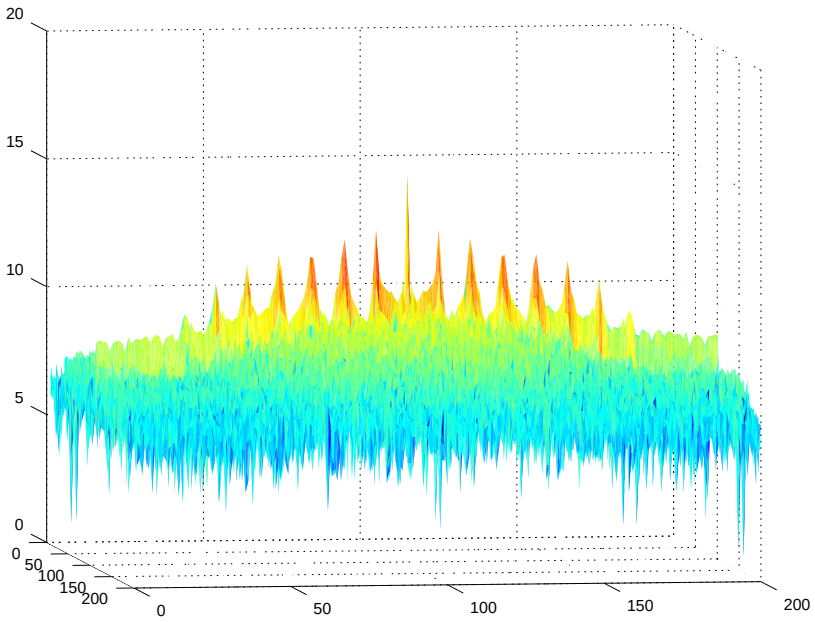


(d) Dirty RGB.

Figure 3.27: DFT results of group 1 sample set 1. (a)-(b) Log transform only. (c)-(d) Contrast stretch and log transform.



(a) Clean, Red.

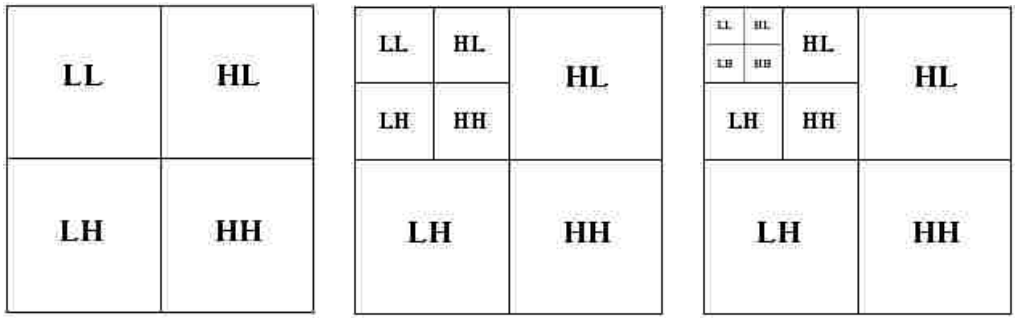


(b) Dirty, Red.

Figure 3.28: 3-D view of the log transform results of group 1 sample set 1. The bottom axes represent the dimensions of the sample, which is 200×200, and the vertical axis represents the results. The value range is from low (i.e., blue), to high (i.e., red). These figures only display the results of the red channel, but the blue and green channels are similar. Notice that the difference is more evident here than in Figure 3.27.

3.4 Discrete Wavelet Transform

In Chapter 2 Section 2.4, we mentioned that one could repeat the DWT on the LL quadrant of the current transform to produce more subbands. Our desired number of transforms was three, for a total of six passes, and an example of how the output looks is shown in Figure 3.29c.



(a) One level transform (b) Two level transform (c) Three level transform

Figure 3.29: Recursively apply the transform on the LL quadrant. (a) One level, (b) two level, and (c) three level DWT, for a total of six passes.

We performed a three-level DWT on all the samples in Section 2.1, and the results are shown below. The trick was to find the quadrant that was consistently different across all samples, excluding the LL quadrant of the third decomposition. However, before we exclude the LL quadrant, we noticed that the blue, green, and red LL quadrant of the third decomposition follow the same decrease in intensity similar to the histogram and gamma distribution results from Section 3.2. With that being said, only the blue channel appears brighter in the dirty panel than the clean panel.

The LH quadrant of the third decomposition was consistently different between clean and dirty samples. The clean samples have the same orientation as the original input, but the lines alternate between white and black; the exception is the first sample of the second group. The dirty samples also have the same orientation as the original input, but the white and black intensities alternate at a diagonal, usually mimicing the placement and pattern of the dirt. Moreover, the intensity of the quadrant may even be higher than the clean sample. The exceptions are the first and second samples of the third group, where instead of the diagonal, there is a distinct edge detection of the contaminants. The LH quadrant of the third decomposition varied among all dirty samples, but it was consistent among dirty samples that came from the sample panel.

The following images are the results obtained by applying the DWT to all the samples. Each sample set contains six images. The first column represents the clean sample and the second column represents the dirty sample. The first row represents the red channel, the second row represents the green channel, and the third row represents the blue channel.

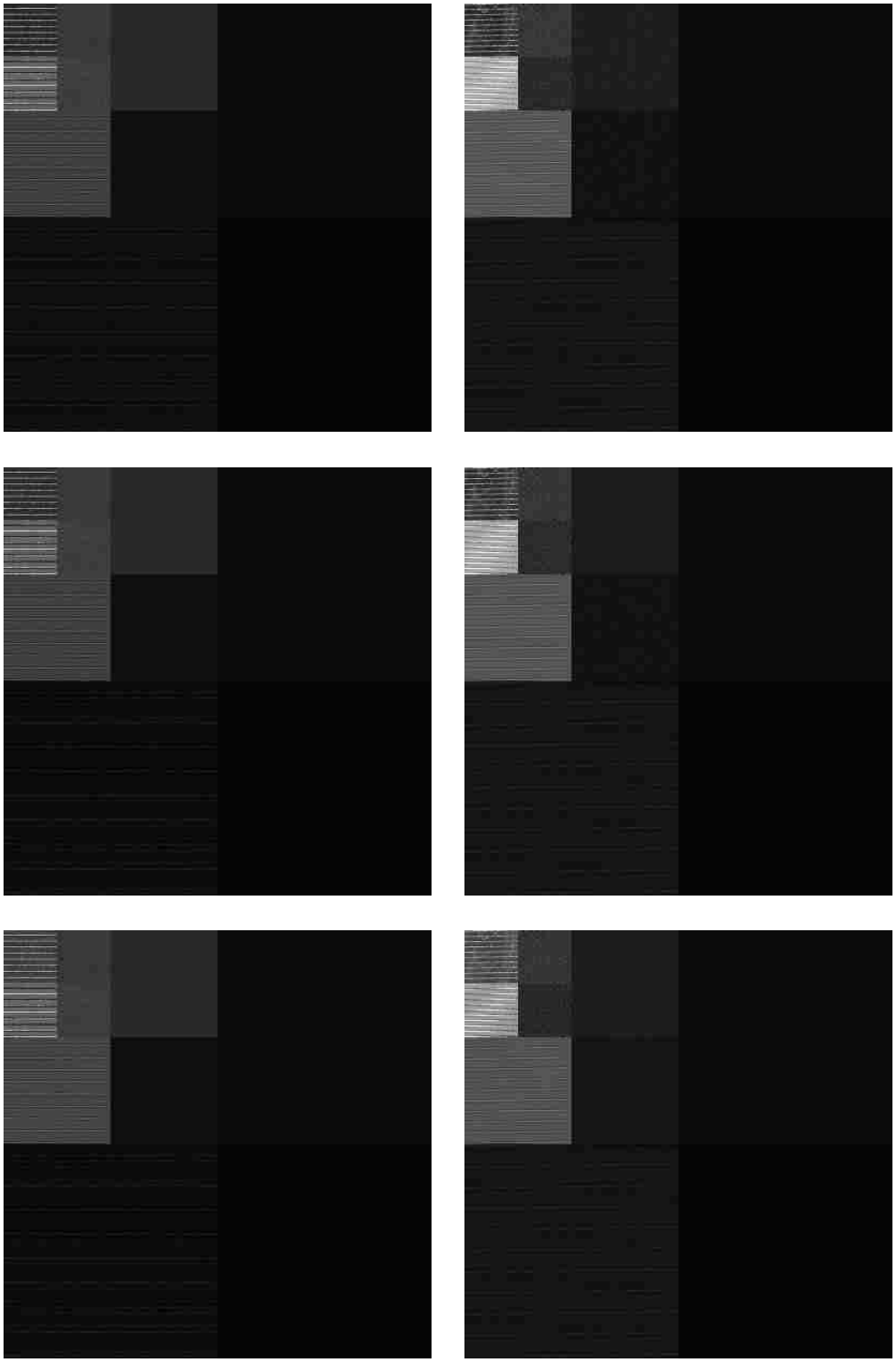


Figure 3.30: Three level DWT of group 1 sample 1.

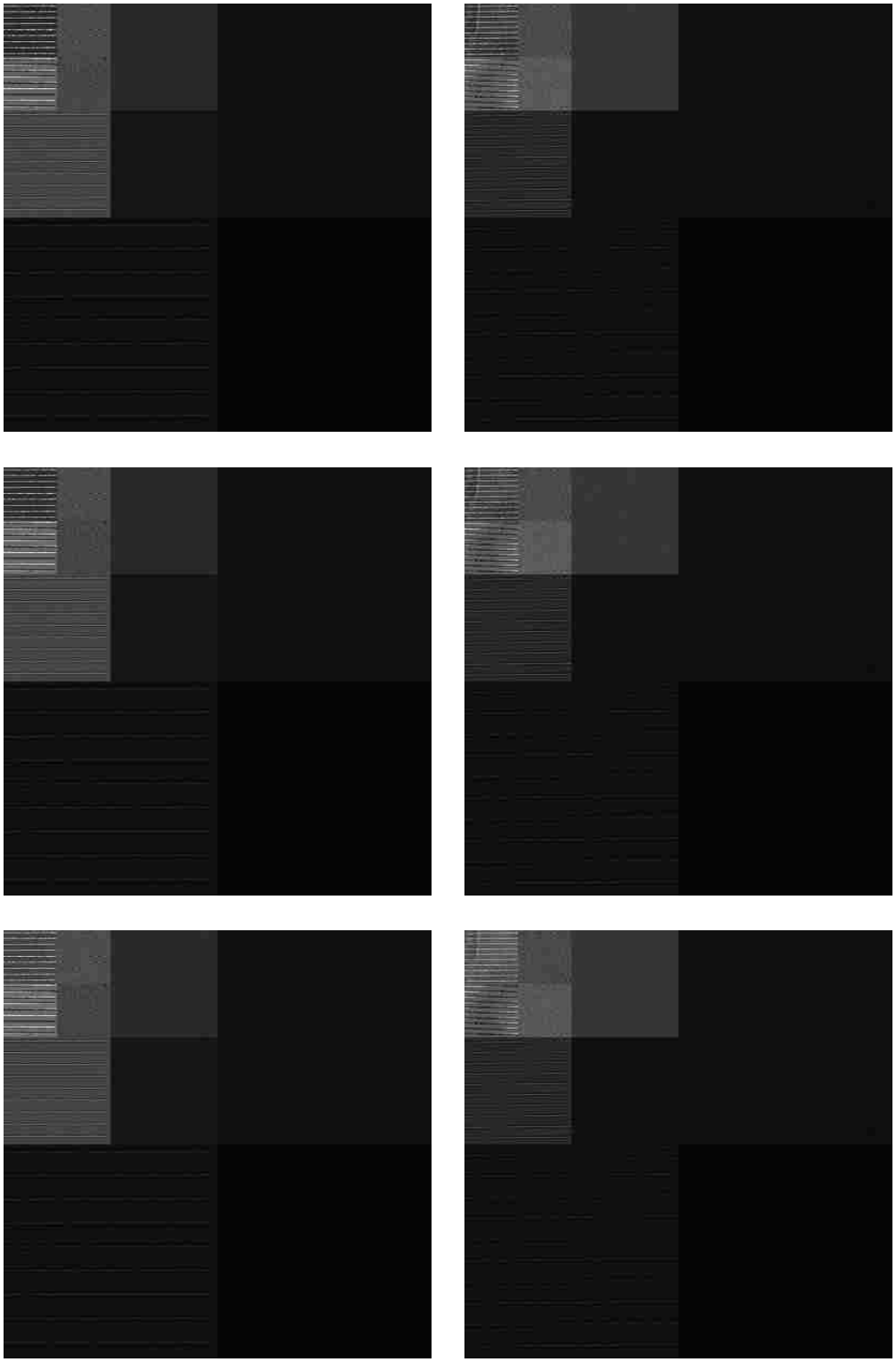


Figure 3.31: Three level DWT of group 1 sample 2.

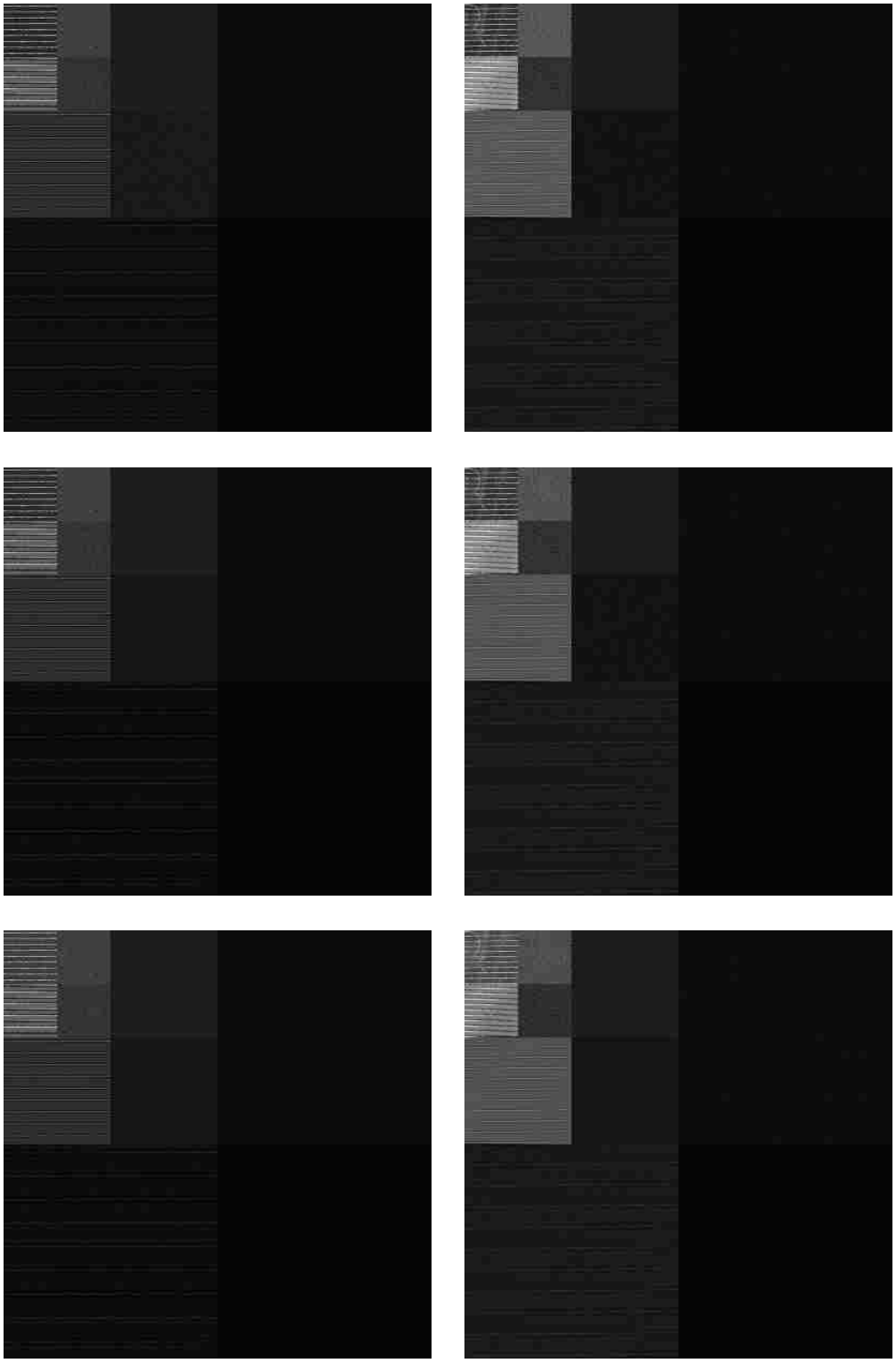


Figure 3.32: Three level DWT of group 1 sample 3.

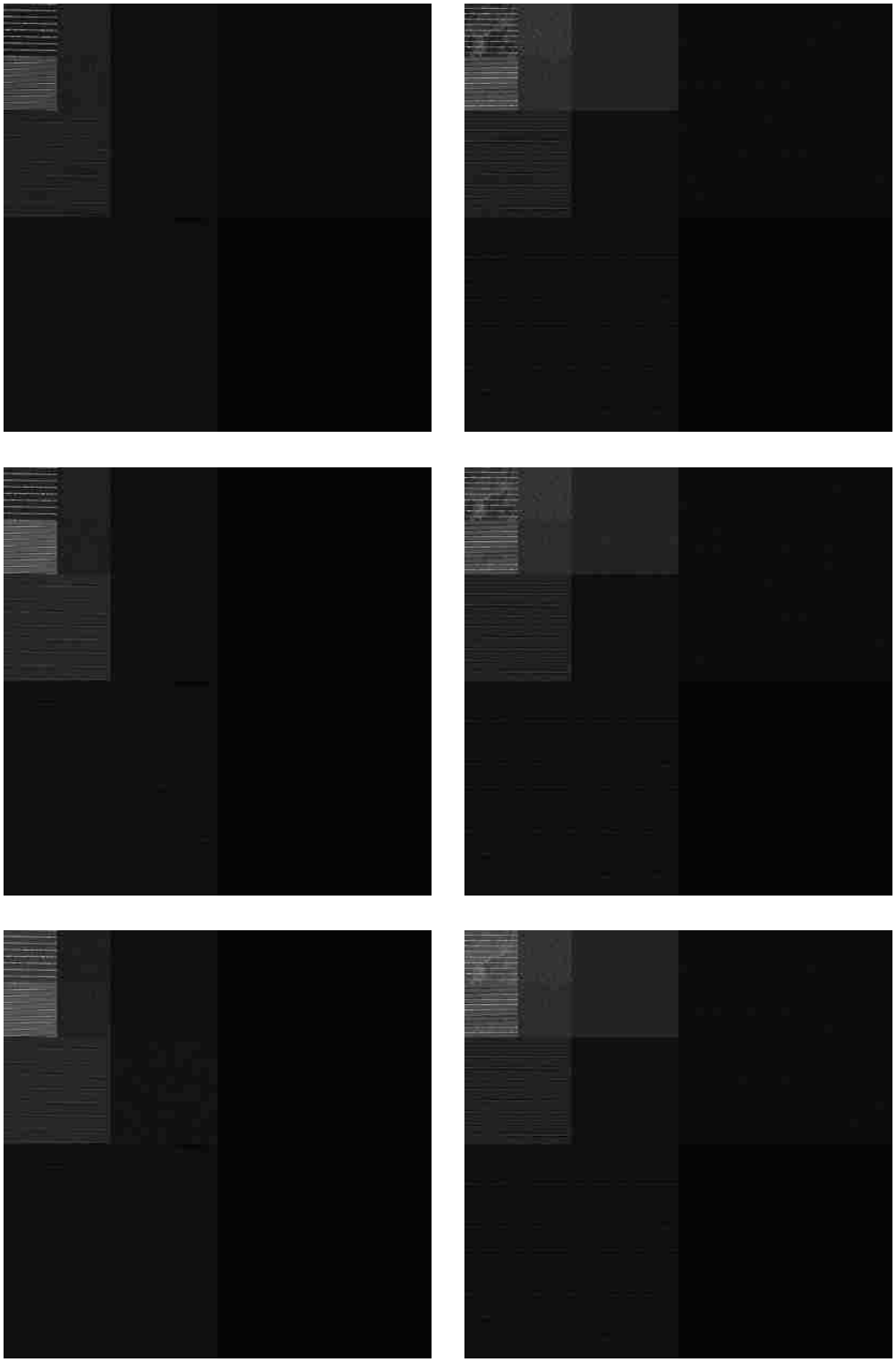


Figure 3.33: Three level DWT of group 2 sample 1.

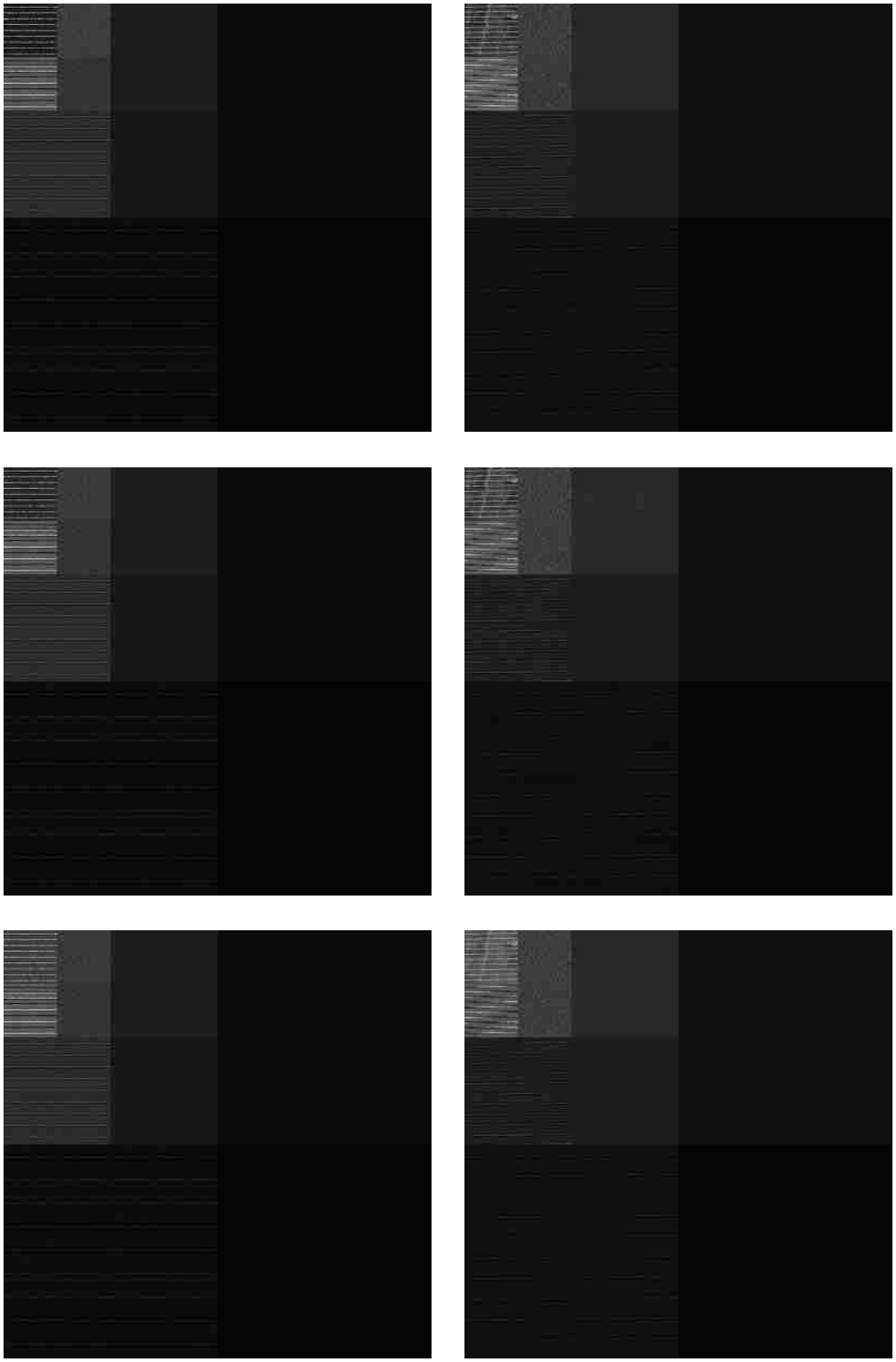


Figure 3.34: Three level DWT of group 2 sample 2.

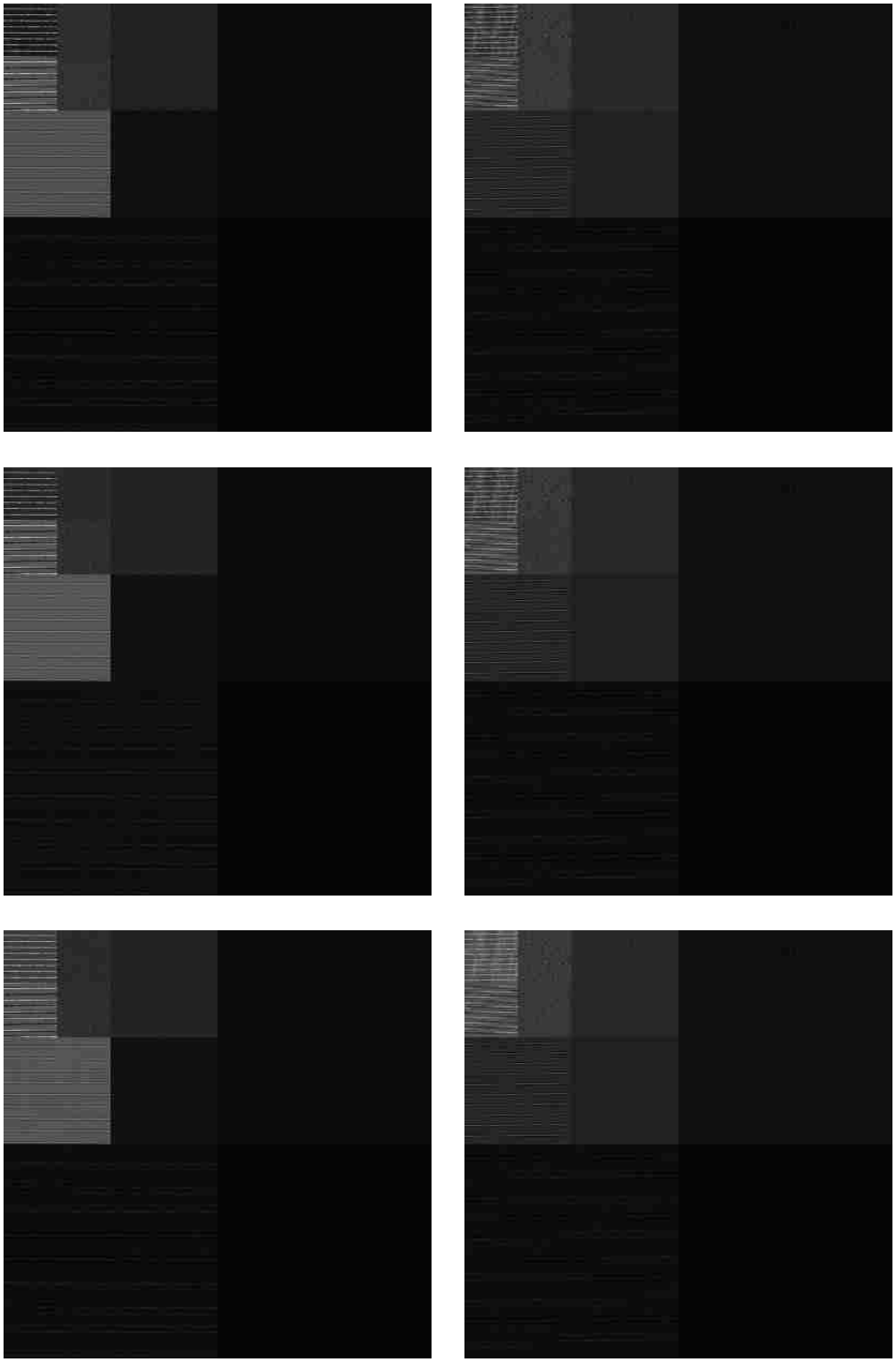


Figure 3.35: Three level DWT of group 2 sample 3.

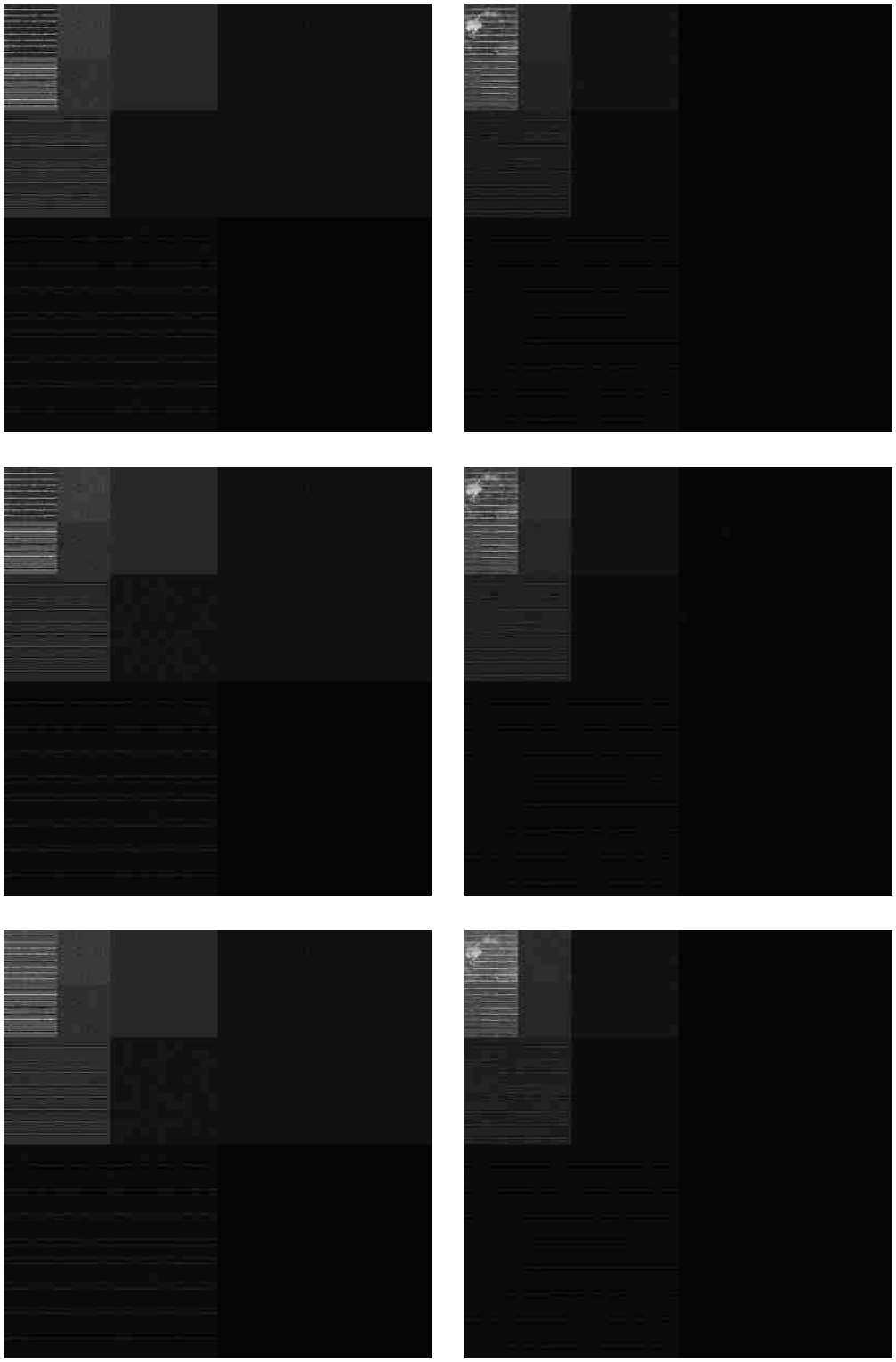


Figure 3.36: Three level DWT of group 3 sample 1.

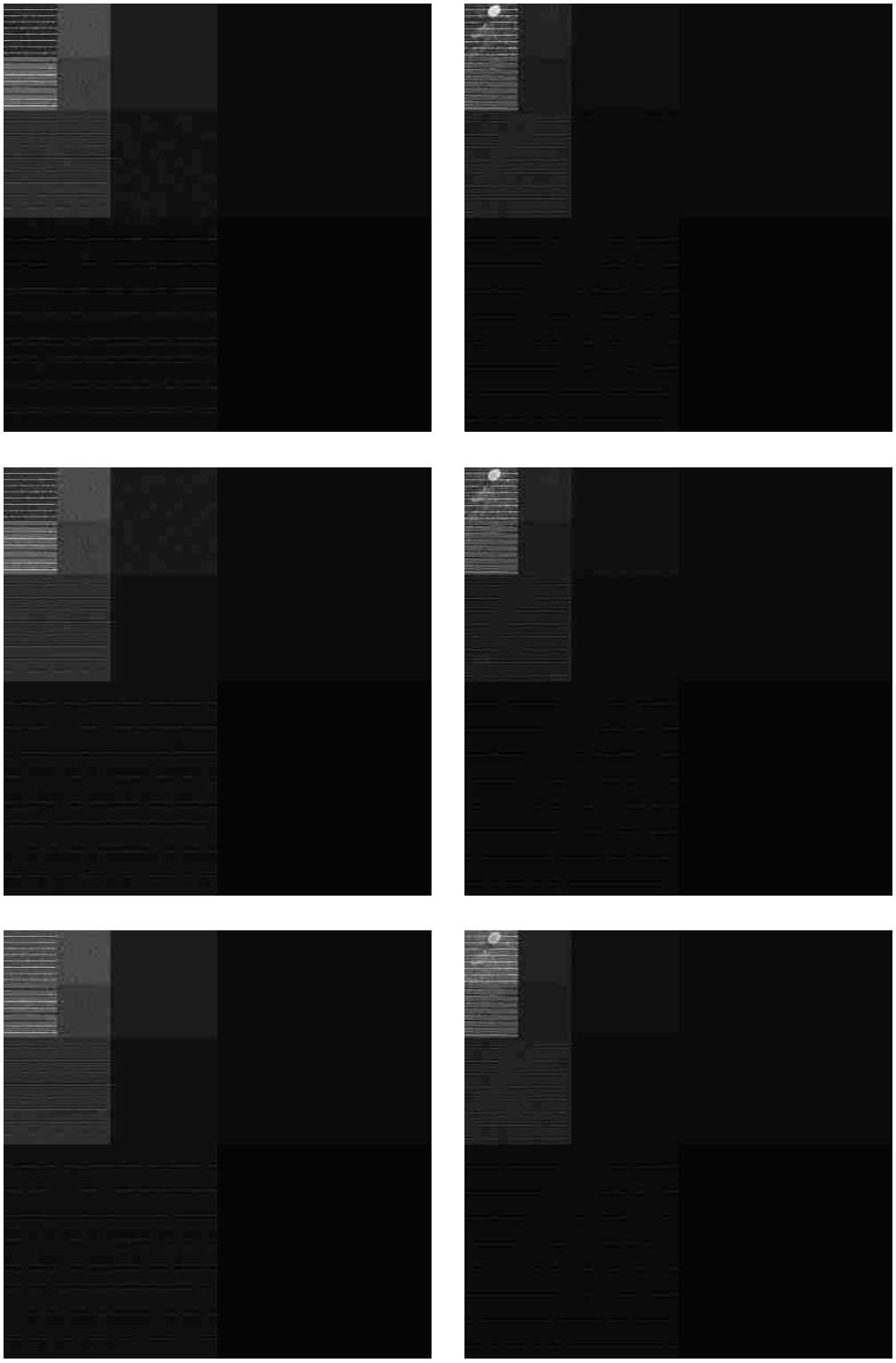


Figure 3.37: Three level DWT of group 3 sample 2.

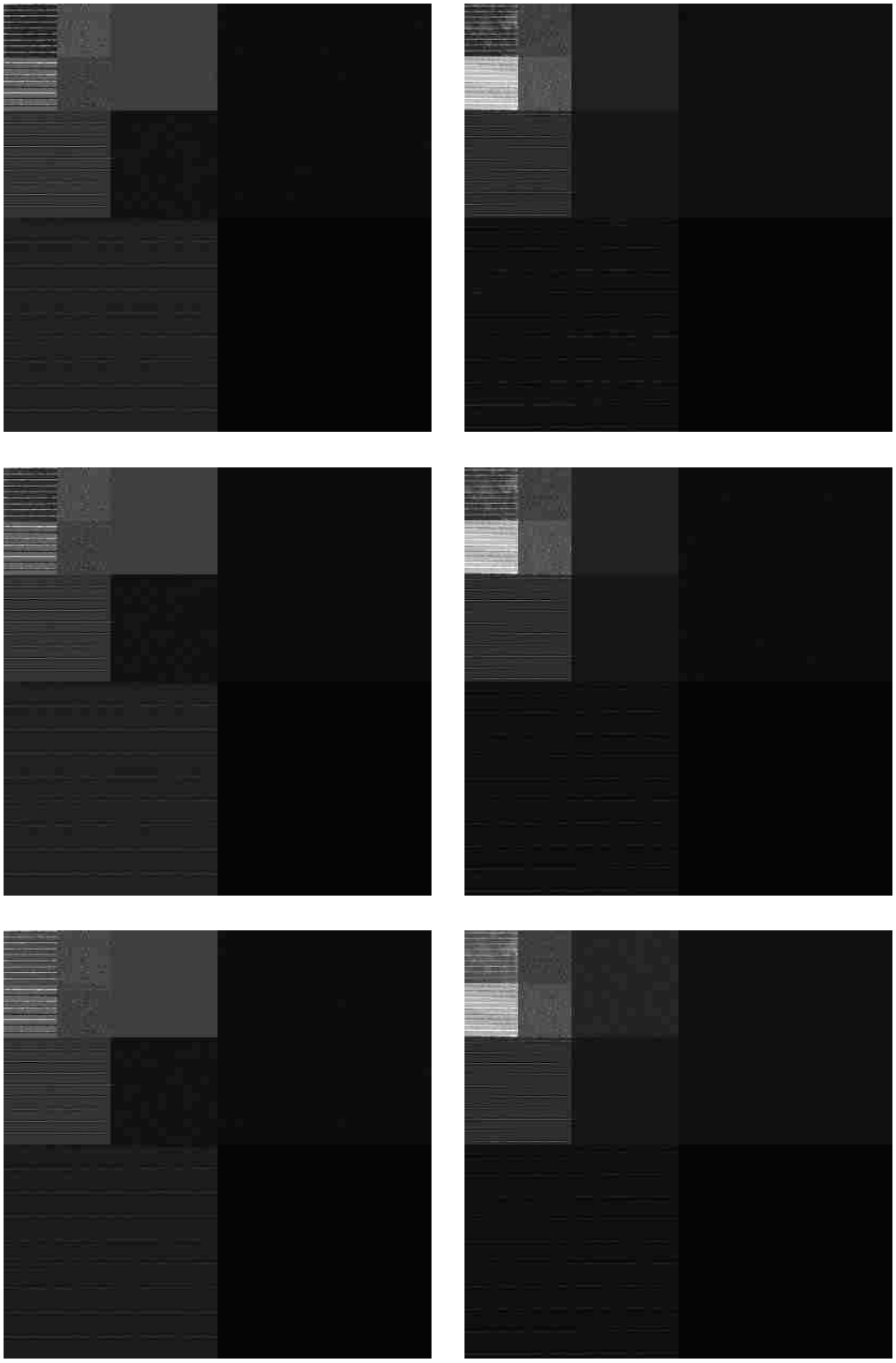


Figure 3.38: Three level DWT of group 3 sample 3.

3.5 Principal Component Analysis

We applied the PCA on the third sample set of group 1. While we can plot the results from Tables 3.16 and 3.17 in three dimensions, notice that the third variable holds little information about the data. What we have not mentioned so far is that PCA is used for dimensional reduction, where a data set of n dimensions is reduced to k dimensions. Often times large variances are associated only with the first $k < n$ principal components, the remaining $n-k$ dimensions are ignored [Shl03]. This process of removing less important variables simplifies the dynamics of the data [Shl03], and it is particularly useful for high dimensional data. Although our data is not high dimensional, we can still reduce the dimensions from three to two, or even to one, as shown in Table 3.18.

Eigenvalue	Eigenvector		
4150.392322082414	0.6037252397288499	0.5834112358604482	0.5432744838349465
11.0857346997606	0.6813986258022624	-0.02392949583528244	-0.731521217726361
1.856166624657606	0.4137774131935325	-0.8118243092571532	0.4119824550026552

Table 3.16: Clean results.

Eigenvalue	Eigenvector		
3451.817331414454	0.6027102281166483	0.5947692928539576	0.5319679212147862
39.7574926605084	-0.7211738637135189	0.1206679358334026	0.6821638421656638
3.914070853661152	-0.3415386350328364	0.7947884860387129	-0.5016598680781612

Table 3.17: Dirty results.

	Component 1	Component 2	Component 3
Clean	99.7%	0.26%	0.04%
Dirty	98.8%	1.1%	0.1%

Table 3.18: A comparison of how much data each principal component holds. Notice the first component accounts for nearly all of the data.

We graphed the eigenvectors against the RGB basis and noticed that the vectors are parallel as shown in Figures 3.39 to 3.41. Not being parallel would mean that there is a fundamental difference between clean and dirty panels. However, notice that the sum of the clean sample's eigenvalues are greater than the sum of the dirty sample's eigenvalues, and the eigenvalue of the first principal component of clean sample is greater than that of the dirty sample. The differences in the eigenvalues are significant, and it is due to the higher amount of variance and scattered light in the dirty panel. Perhaps the dirtier a panel is, the more variance is distributed from the first principal component to the other two components. If we were to graph the eigenvalues and classification vectors as points, it would show that although the vectors are parallel, the two classes have different shapes.

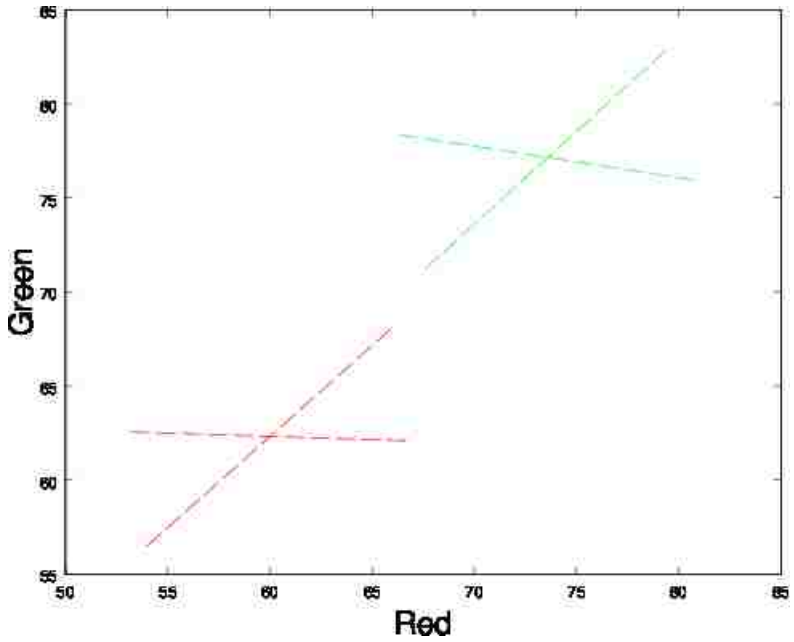


Figure 3.39: A red and green view. The first and second principal components of the clean (red lines) and dirty (green lines) sample plotted against red and green of the R, G, B basis.

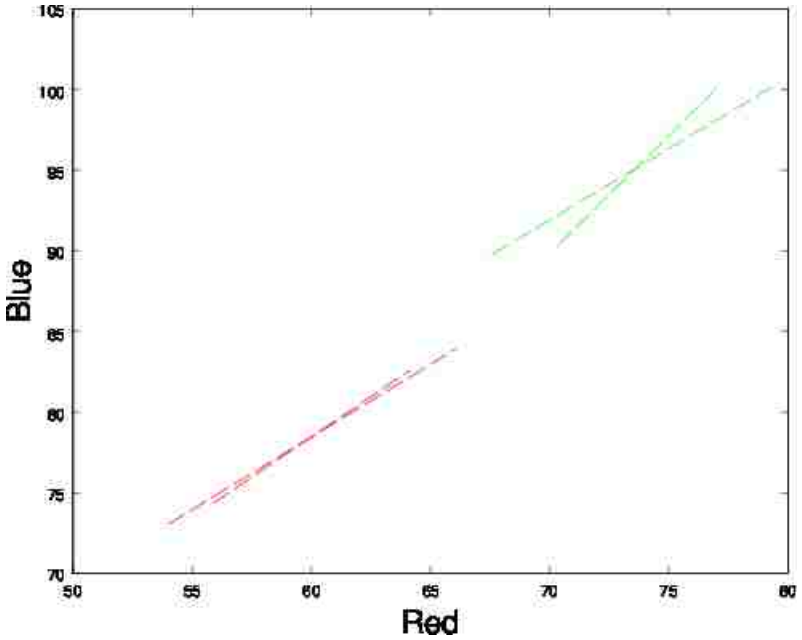


Figure 3.40: A red and blue view. The first and second principal components of the clean (red lines) and dirty (green lines) sample plotted against red and blue of the R, G, B basis.

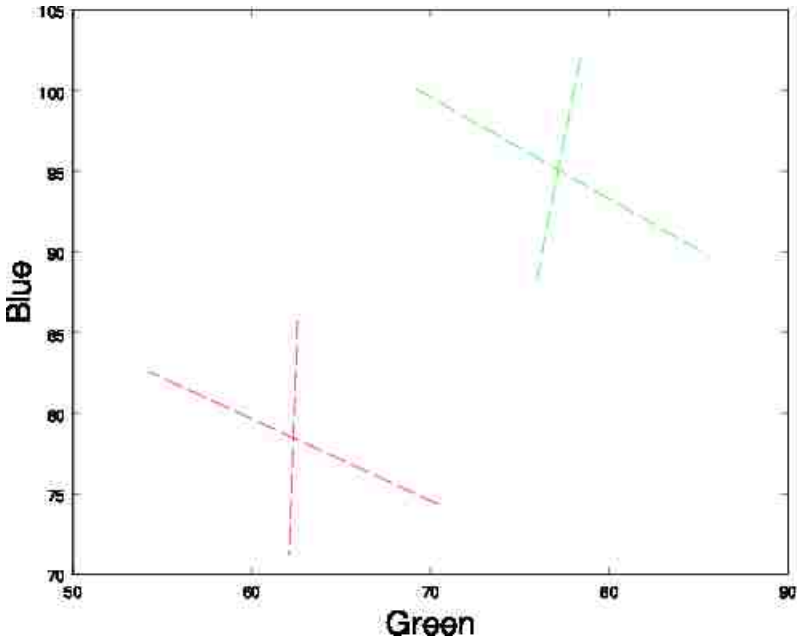


Figure 3.41: A green and blue view. The first and second principal components of the clean (red lines) and dirty (green lines) sample plotted against green and blue of the R, G, B basis.

The following results were obtained by applying the PCA on the sample sets from Section 2.1. We transformed the original data set by multiplying the pixels with the eigenvectors to produce a new data set. Each graph is a plot of the new data set, two out of three principal components at a time. The blue graphs are clean samples, and the red graphs are dirty samples. Instead of a three dimensional display, we plotted the eigenvectors in pairs, which comes out to a total of three graphs. When we compared the orientation of each graph for each pair, we noticed the following:

- All the dirty samples coming from the same panel image had the same results. That is not to say that *all* samples from that image will have principal components of the same orientation. We did not test that.

- Even though all the clean samples came from the same panel, the results were not like the dirty samples. Instead, clean samples that came from the same panel had similar results in subsets, but not as a whole. In other words, although all our clean samples came from the same panel, a few would have the same result, while the remaining few would have the same result that was different from the first subset. Perhaps the consistency of the dirty samples was a coincidence.

- Samples pairs from the same group rarely had the same results. Group 2 did the best with the first and second sample pairs.

- Sometimes sample pairs from different groups have the same results.

Furthermore, we provide the eigenvalues and eigenvectors of each sample pair per group, which appear before the graphs. There are three pairs of tables, each representing a sample pair. The first table lists the results of the clean sample, and the second table lists the results of the dirty sample. We compared the eigenvectors looking at the sign (+ or -) and order after sorting, and the eigenvalue sums and distribution.

- As expected, the clean samples have higher total variance than the dirty samples, and more of the variance is distributed to the lesser principal components in the dirty samples.
- Like the orientation, all dirty samples from the same panel had parallel eigenvectors.
- Like the orientation, clean samples had parallel eigenvectors in subsets.

Eigenvalue	Eigenvector		
3210.041013589082	0.6043881283468774	0.595047683636071	0.529748284105484
37.55978626634278	-0.7183827877592479	0.1195667104577083	0.685295536248102
3.830843115022727	-0.3444432617494644	0.7947465356638357	-0.4997367141662911

Table 3.22: Dirty results of group 1 sample 2.

Eigenvalue	Eigenvector		
4150.392322082414	0.6037252397288499	0.5834112358604482	0.5432744838349465
11.0857346997606	0.6813986258022624	-0.02392949583528244	-0.731521217726361
1.856166624657606	0.4137774131935325	-0.8118243092571532	0.4119824550026552

Table 3.23: Clean results of group 1 sample 3.

Eigenvalue	Eigenvector		
4207.794742341159	0.6039615569219726	0.5865161770311635	0.5396565684221176
16.0858075097764	-0.7165362524072486	0.1030755949321833	0.6898921805003678
2.39211322376632	-0.3490075024215987	0.803351850567437	-0.4825138002620364

Table 3.19: Clean results of group 1 sample 1.

Eigenvalue	Eigenvector		
3587.576656567072	0.6013096916187074	0.5951389439923165	0.5331381548052301
43.82456305199561	-0.7365112020378037	0.1541322984347225	0.6586307644287952
3.829137827368165	-0.3098030084396283	0.7887032850970891	-0.5310077438595557

Table 3.20: Dirty results of group 1 sample 1.

Eigenvalue	Eigenvector		
4800.071314795519	0.60.6057039673272	0.5855004292872928	0.5379597814241225
15.74842602228708	0.647139130368572	0.02963983931845276	-0.7617955275997522
2.241637741030261	0.4619766499199957	-0.8101300680937428	0.3609249890200535

Table 3.21: Clean results of group 1 sample 2.

Eigenvalue	Eigenvector		
3451.817331414454	0.6027102281166483	0.5947692928539576	0.5319679212147862
39.7574926605084	-0.7211738637135189	0.1206679358334026	0.6821638421656638
3.914070853661152	-0.3415386350328364	0.7947884860387129	-0.5016598680781612

Table 3.24: Dirty results of group 1 sample 3.

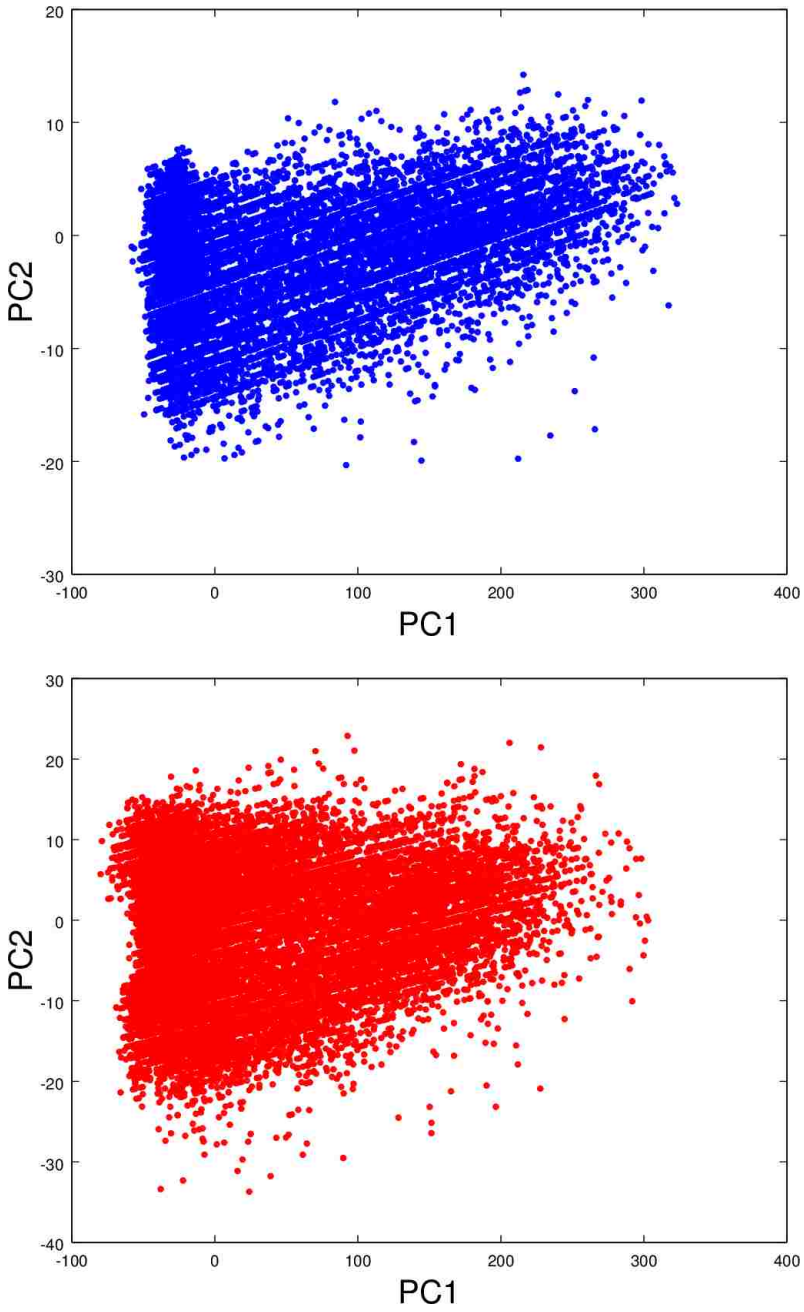


Figure 3.42: First and second principal components for group 1 sample set 1.

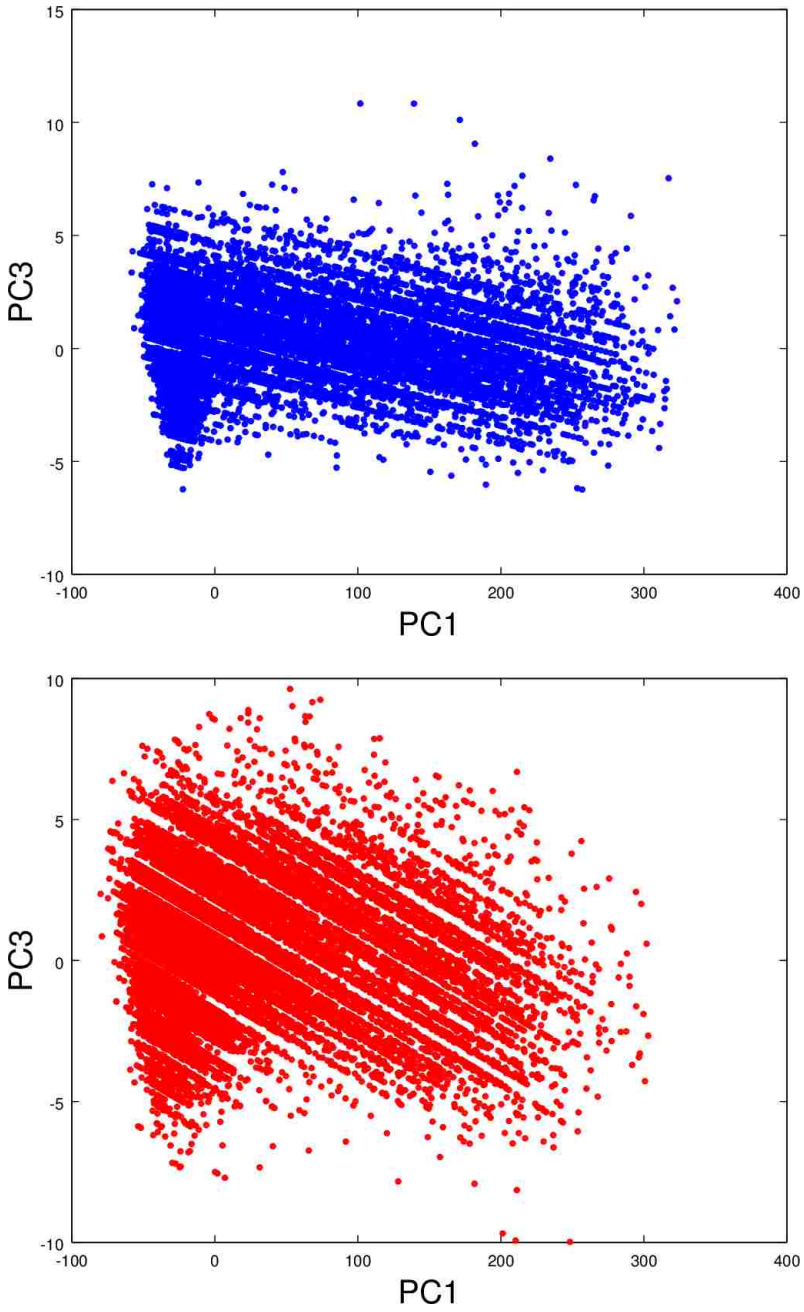


Figure 3.43: First and third principal components for group 1 sample set 1.

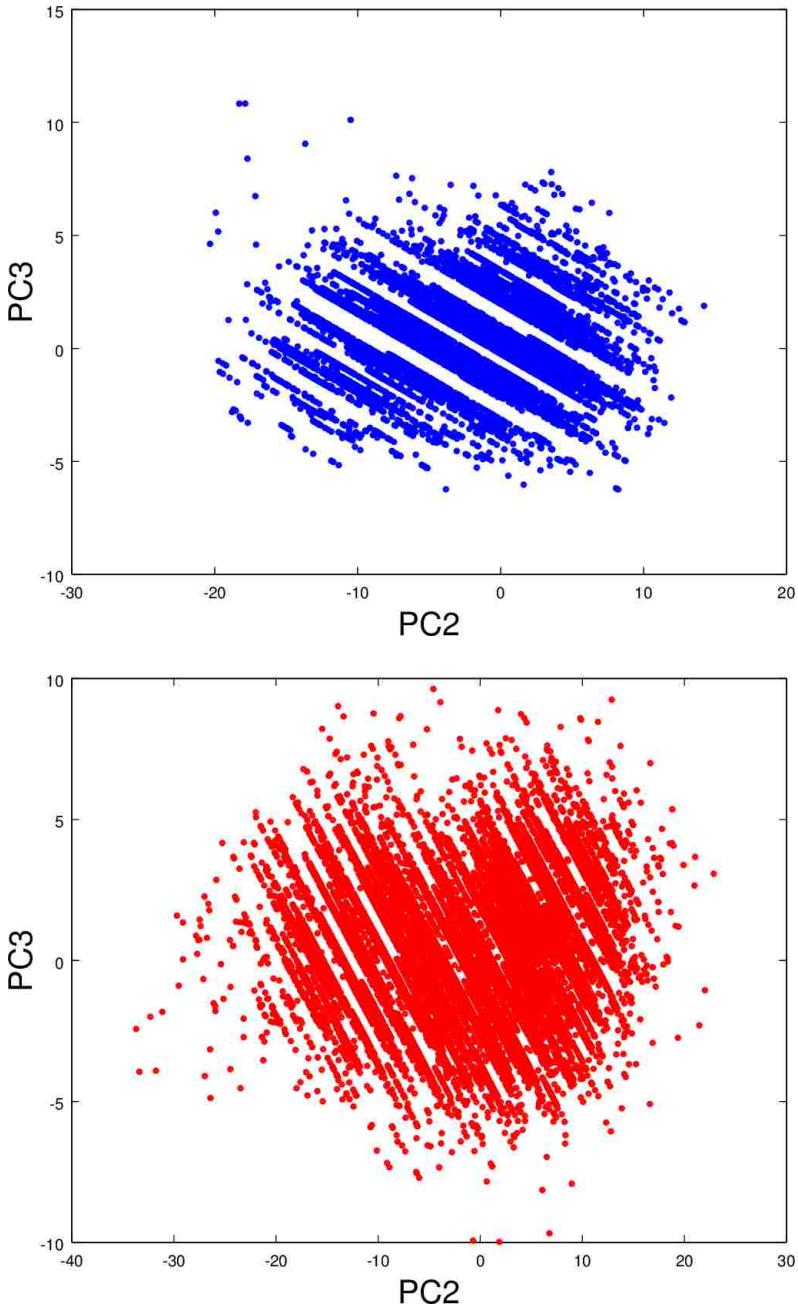


Figure 3.44: Second and third principal components for group 1 sample set 1.

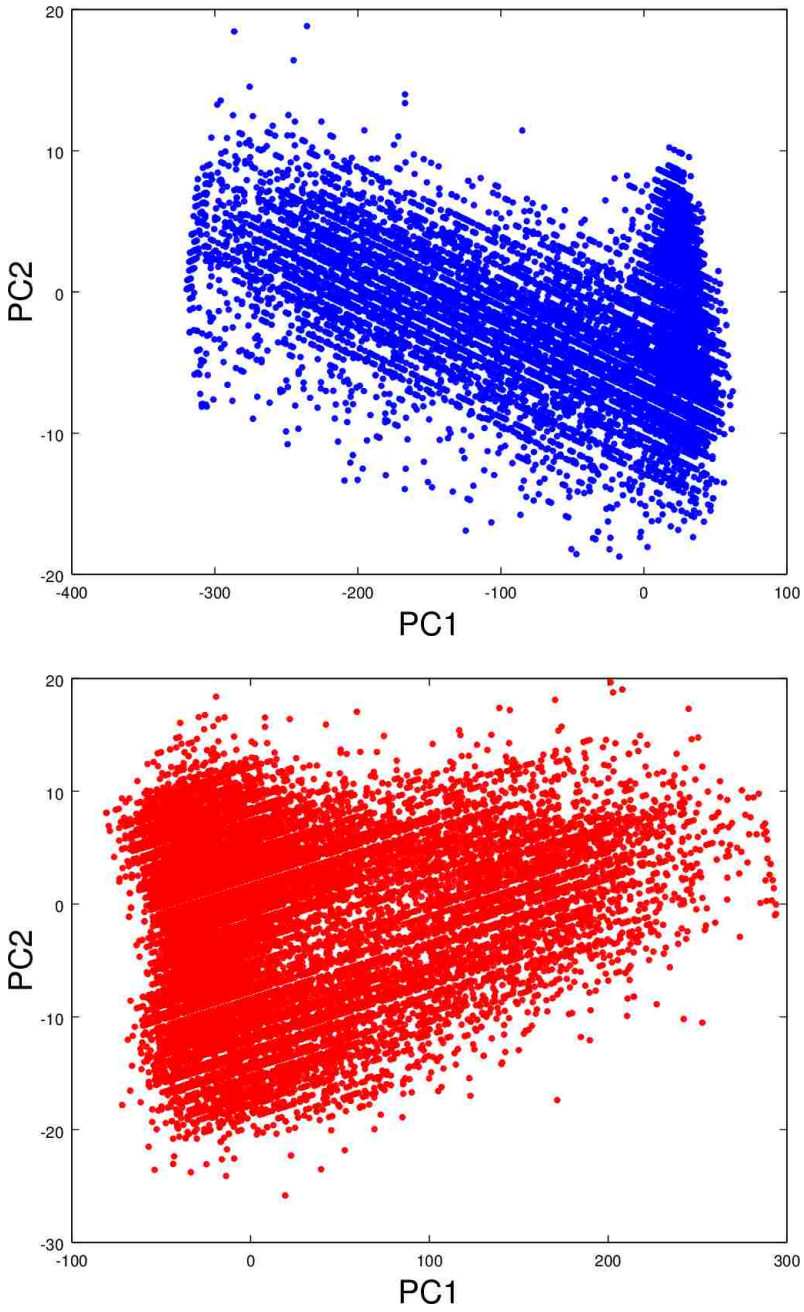


Figure 3.45: First and second principal components for group 1 sample set 2.

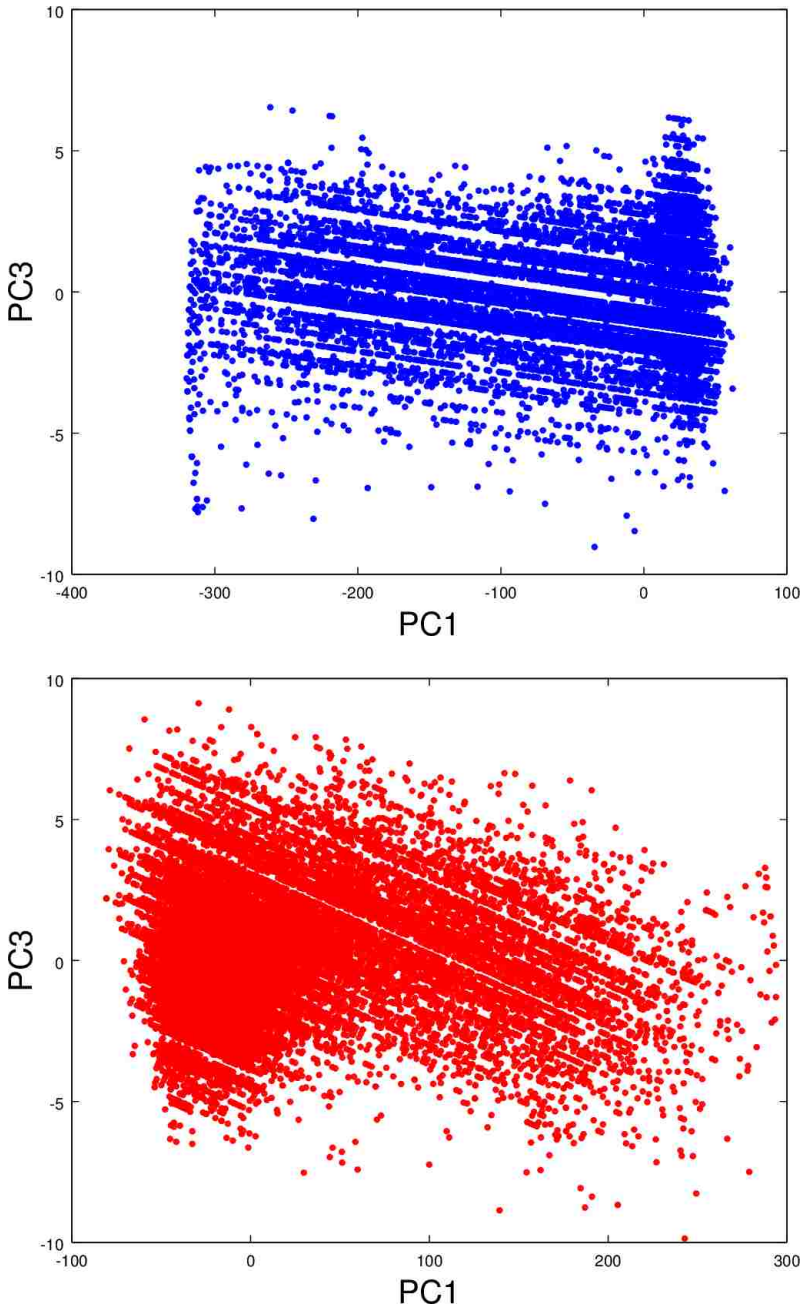


Figure 3.46: First and third principal components for group 1 sample set 2.

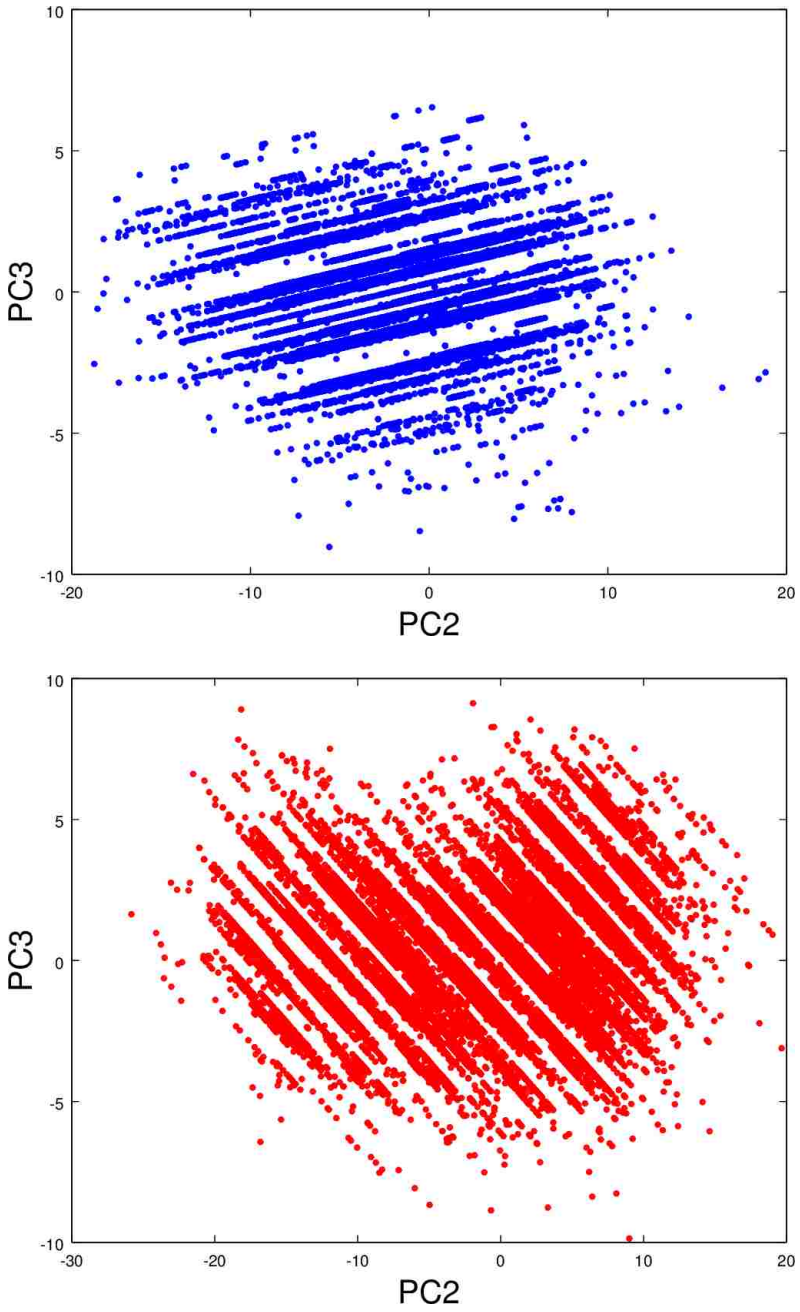


Figure 3.47: Second and third principal components for group 1 sample set 2.

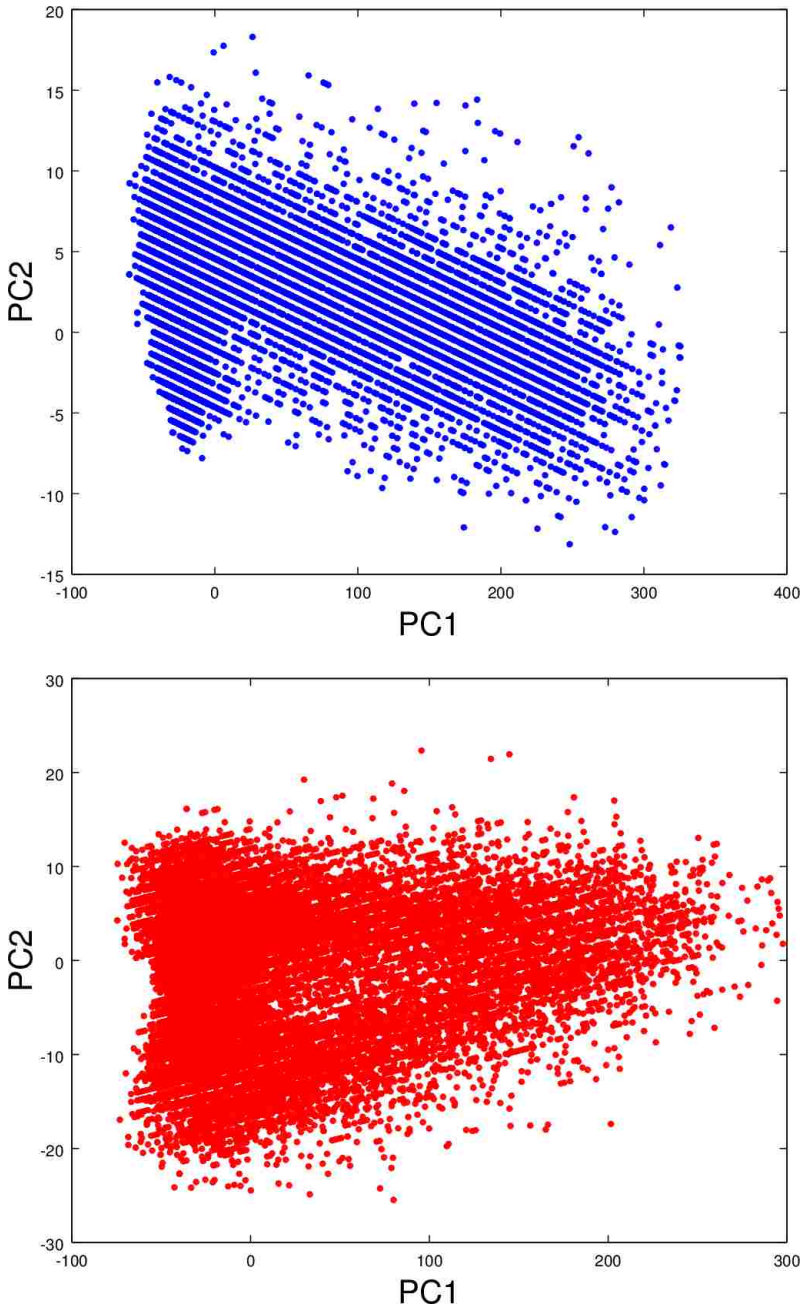


Figure 3.48: First and second principal components for group 1 sample set 3.

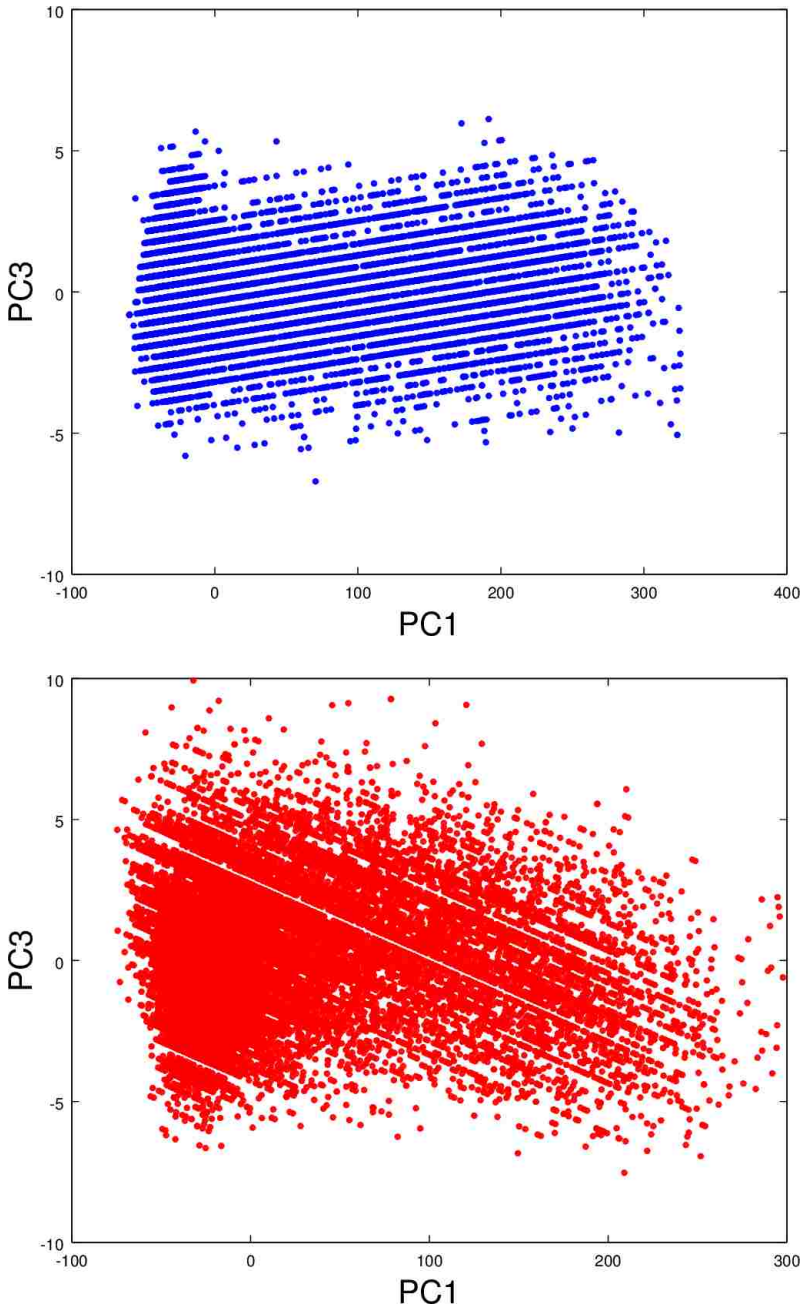


Figure 3.49: First and third principal components for group 1 sample set 3.

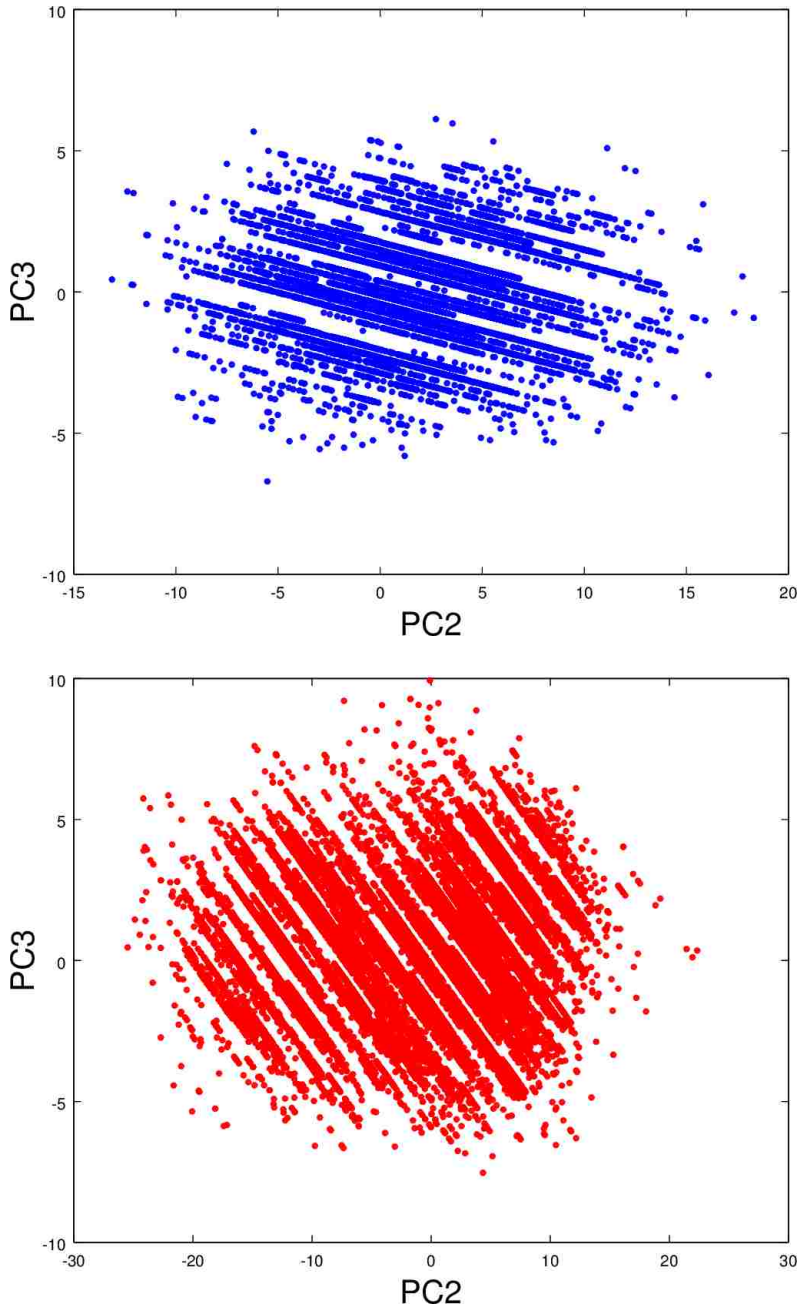


Figure 3.50: Second and third principal components for group 1 sample set 3.

Eigenvalue	Eigenvector		
2528.502935773575	0.6073500881525973	0.5926637356830375	0.5290326708505433
61.80688257472466	0.7749402910615207	-0.2953613374003466	-0.5587747539559783
3.027698298991973	-0.1749097358940263	0.7493406280025832	-0.6386667421387908

Table 3.25: Clean results of group 2 sample 1.

Eigenvalue	Eigenvector		
2445.597628644024	0.6185467140802302	0.5903324232449233	0.5185630073253654
14.58025236296168	0.6613344230102307	-0.0347504977484083 3	-0.7492857824942122
2.636109827159747	0.4243073690643088	-0.8064118259127993	0.411901958705843

Table 3.26: Dirty results of group 2 sample 1.

Eigenvalue	Eigenvector		
3577.0366038379	0.604677704321451	0.5787552087234572	0.5471812152038912
11.79288654302961	-0.7416003794198222	0.1585385756777157	0.6518392418890233
2.453240779224713	-0.2905060259975952	0.7999424531786949	-0.5250698243677079

Table 3.27: Clean results of group 2 sample 2.

Eigenvalue	Eigenvector		
2830.174317704229	0.6056115835740309	0.5976484284274326	0.5254055251319445
52.20028627806973	-0.7297306487404928	0.1537651887502839	0.6662202691432985
4.368754649167502	-0.3173764170976379	0.7868752269113634	-0.5292443548527055

Table 3.28: Dirty results of group 2 sample 2.

Eigenvalue	Eigenvector		
3626.439343360916	0.6071792702574716	0.5794146815449535	0.5437020880774369
16.28933072766827	-0.7550496615428487	0.2076428439893199	0.6219199771225045
2.486786992317051	-0.24745371763955	0.7881389954507941	-0.5635632896810954

Table 3.29: Clean results of group 2 sample 3.

Eigenvalue	Eigenvector		
2626.733251557247	0.6019728964053597	0.5992469548331535	0.5277610435760999
48.00530683955849	-0.7357942458584334	0.1595006497410417	0.658153759006058
4.044478647244546	-0.3102184065379757	0.7845142636404882	-0.5369375293175749

Table 3.30: Dirty results of group 2 sample 3.

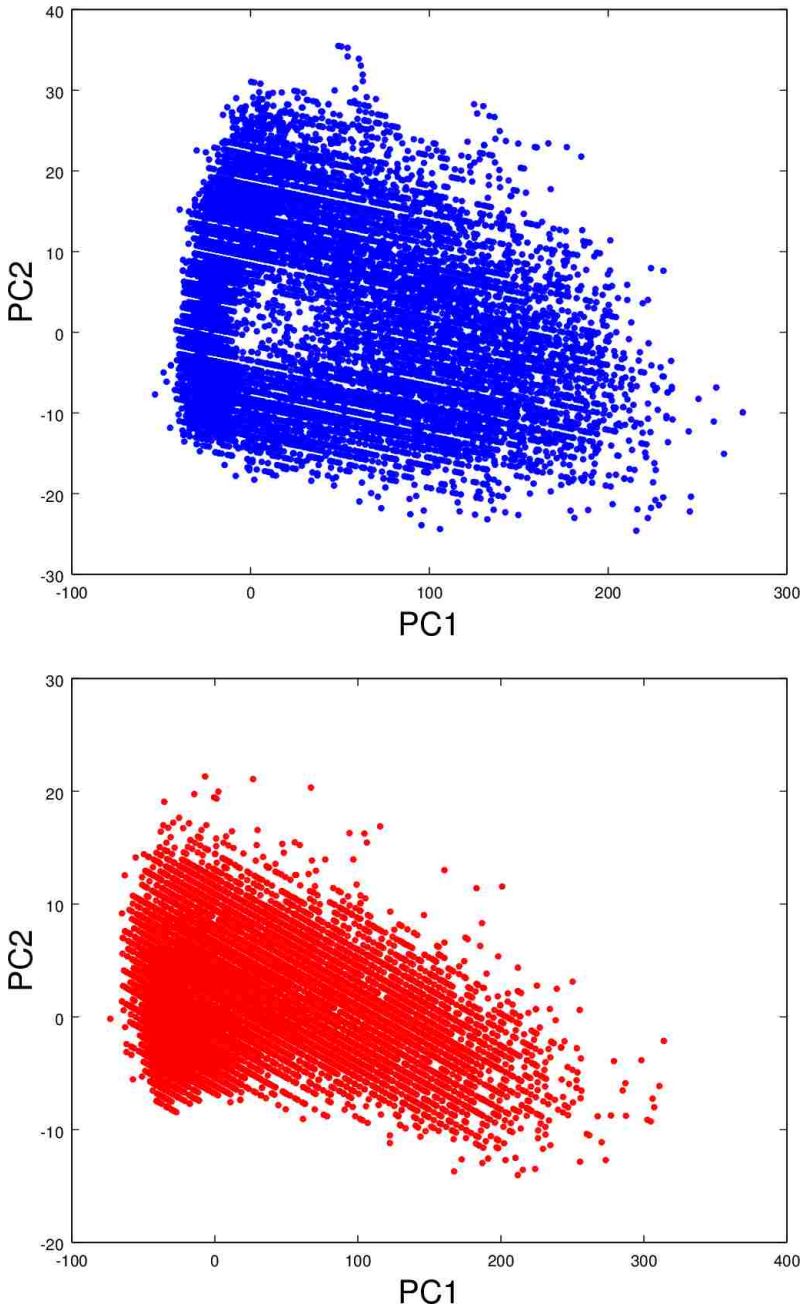


Figure 3.51: First and second principal components for group 2 sample set 1.

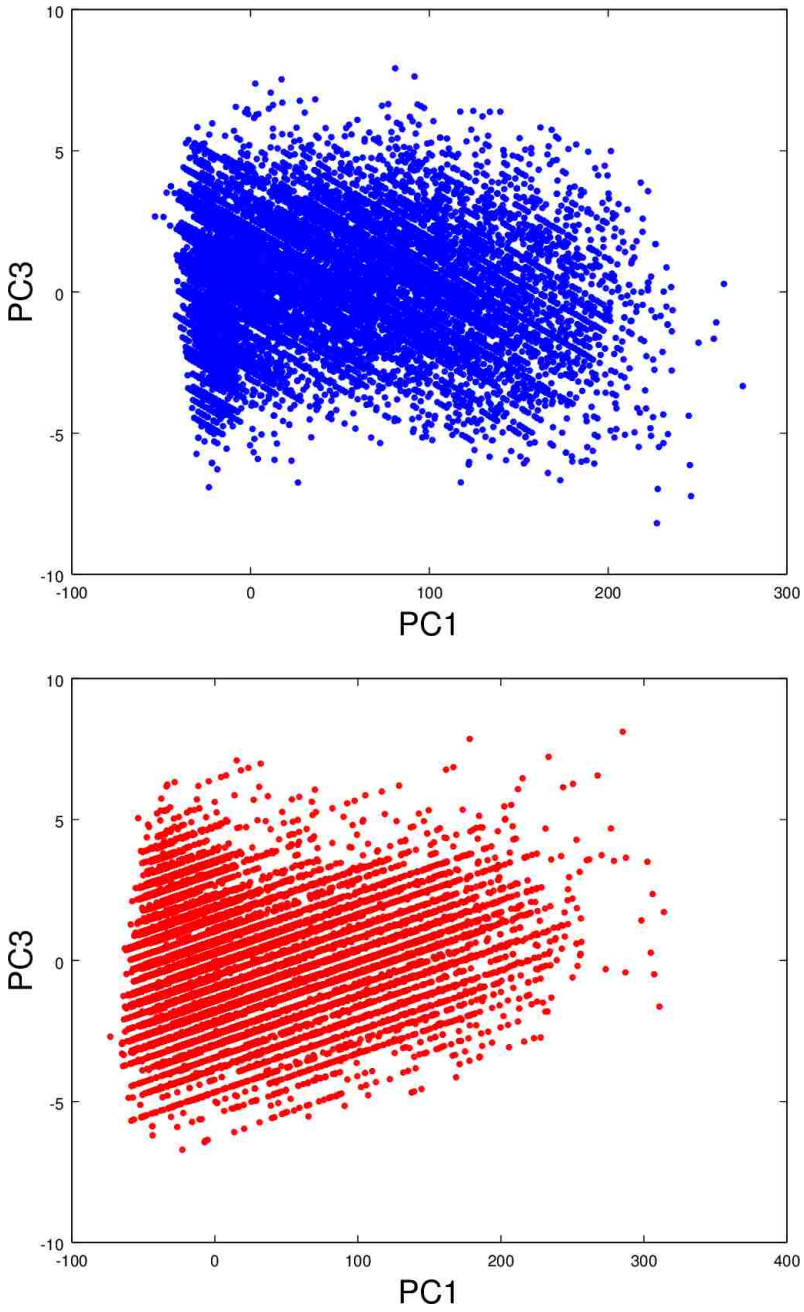


Figure 3.52: First and third principal components for group 2 sample set 1.

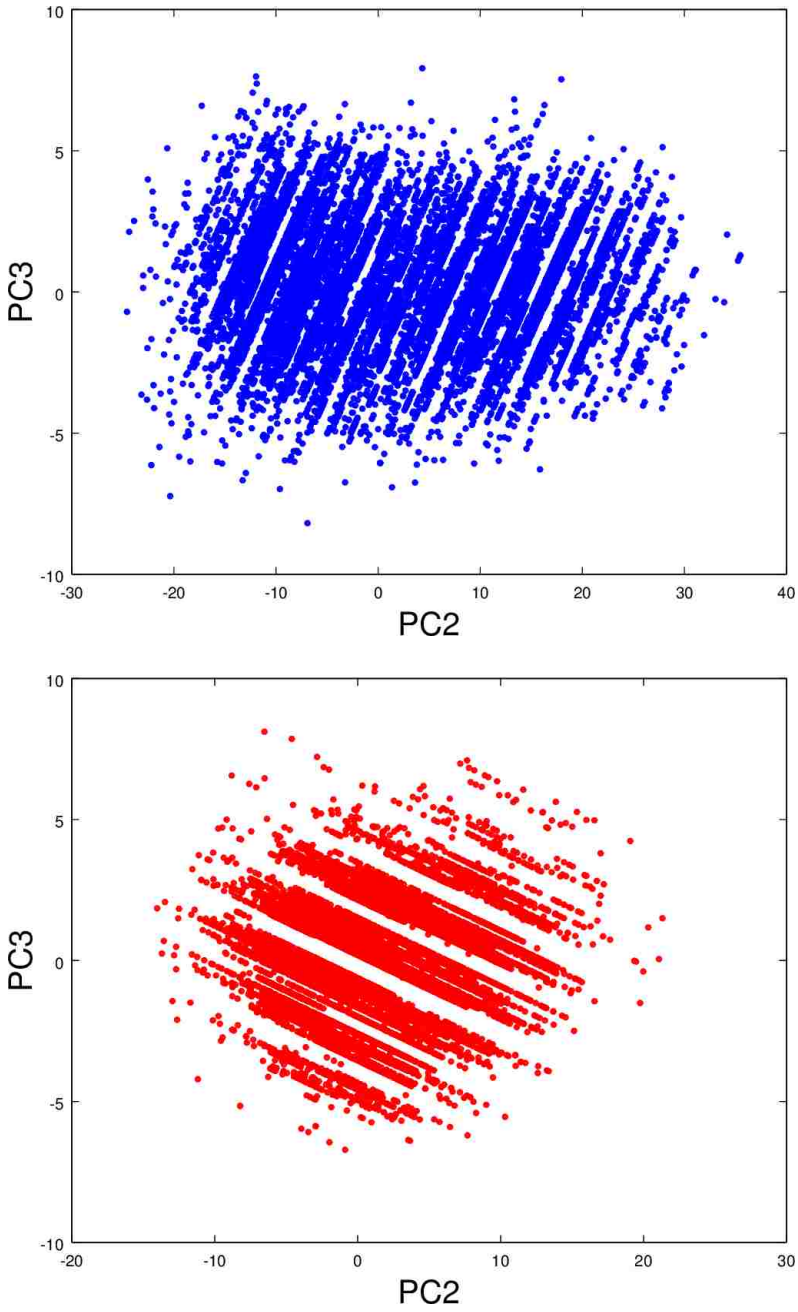


Figure 3.53: Second and third principal components for group 2 sample set 1.

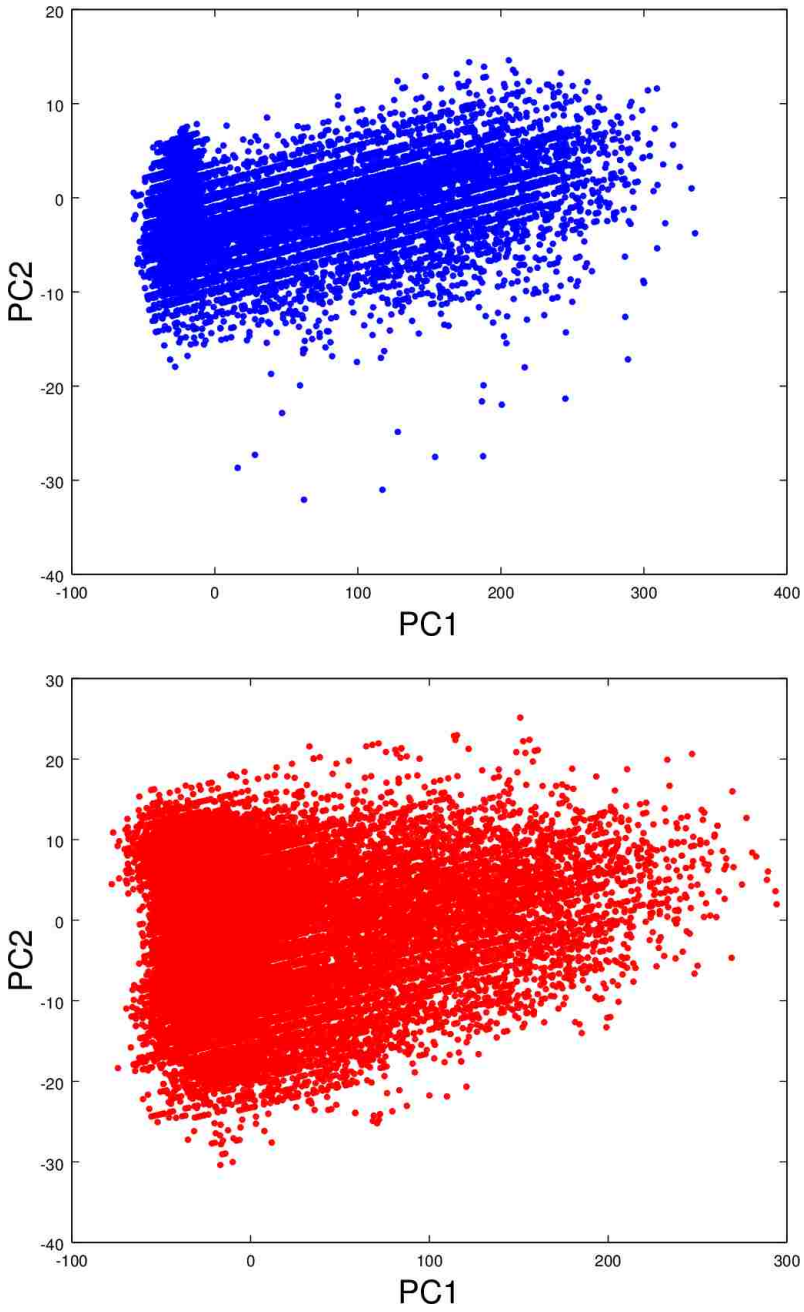


Figure 3.54: First and second principal components for group 2 sample set 2.

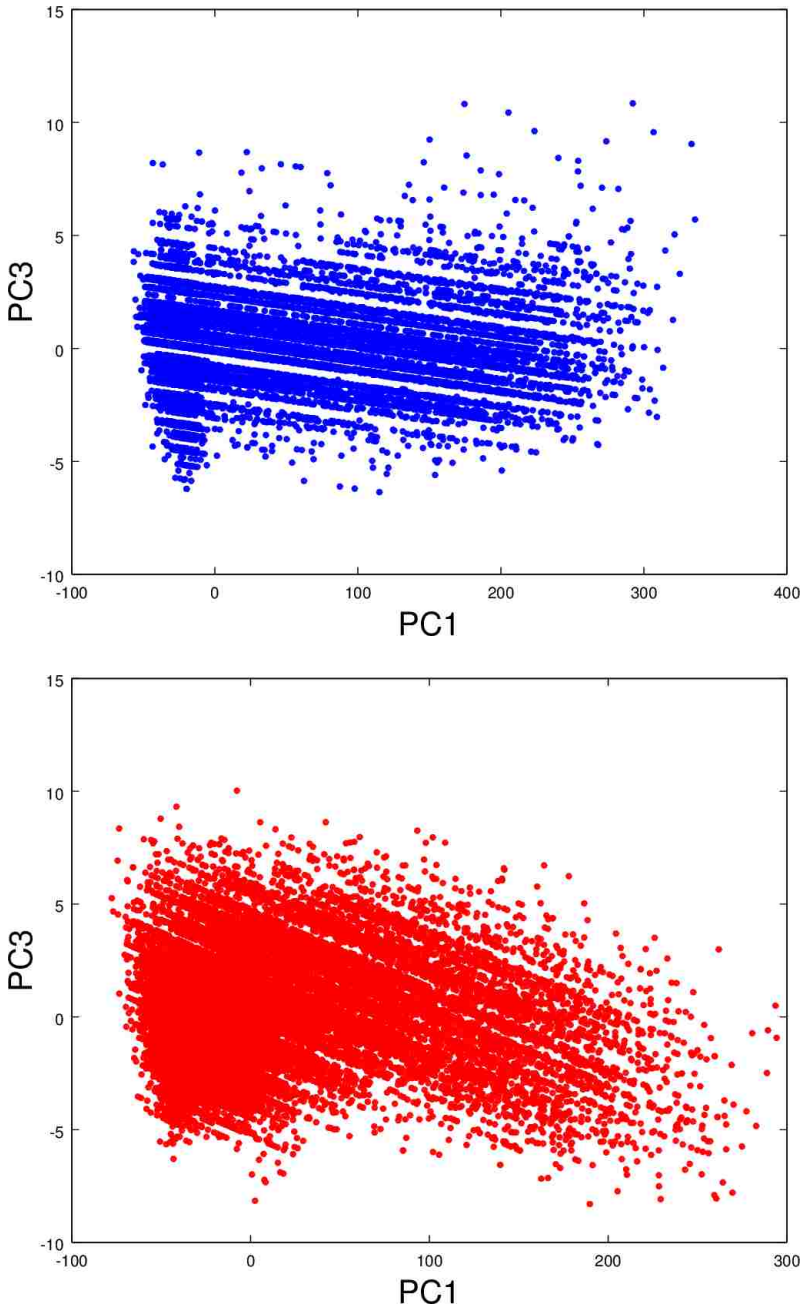


Figure 3.55: First and third principal components for group 2 sample set 2.

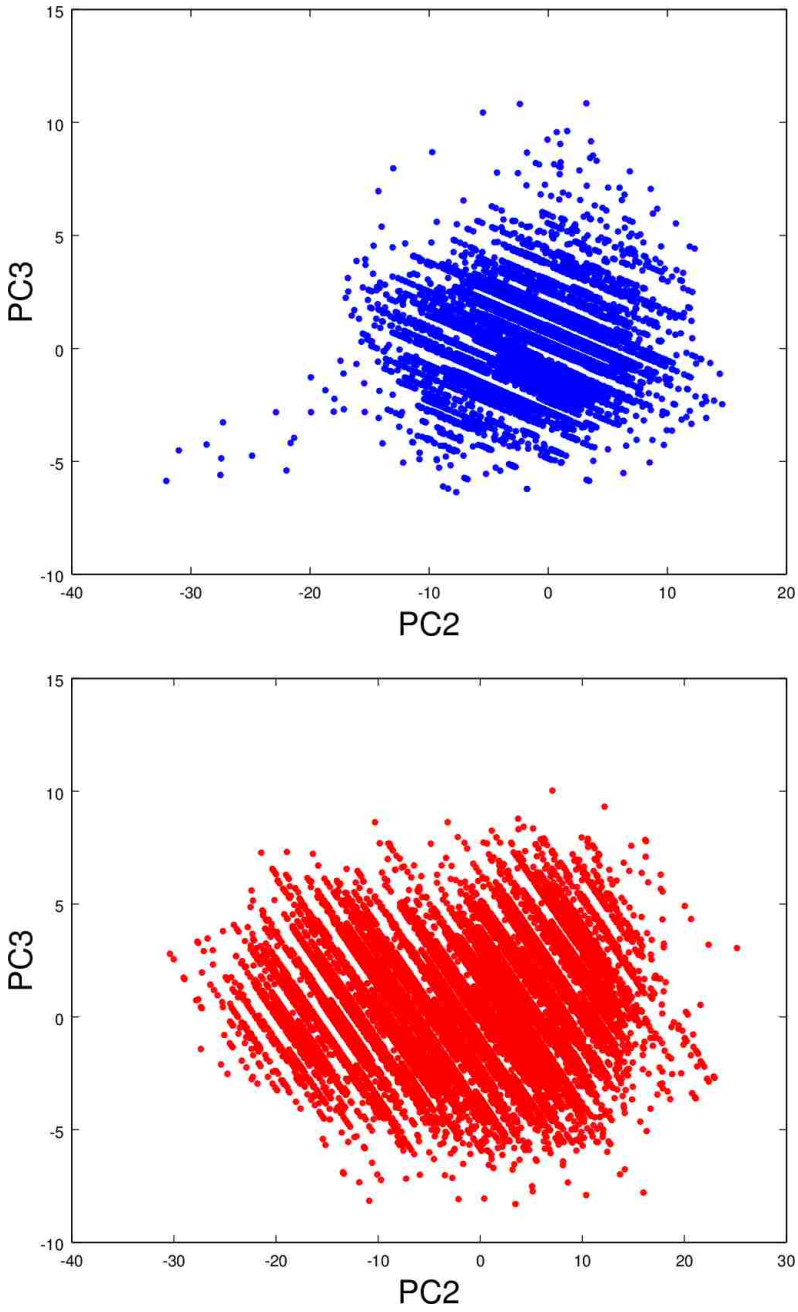


Figure 3.56: Second and third principal components for group 2 sample set 2.

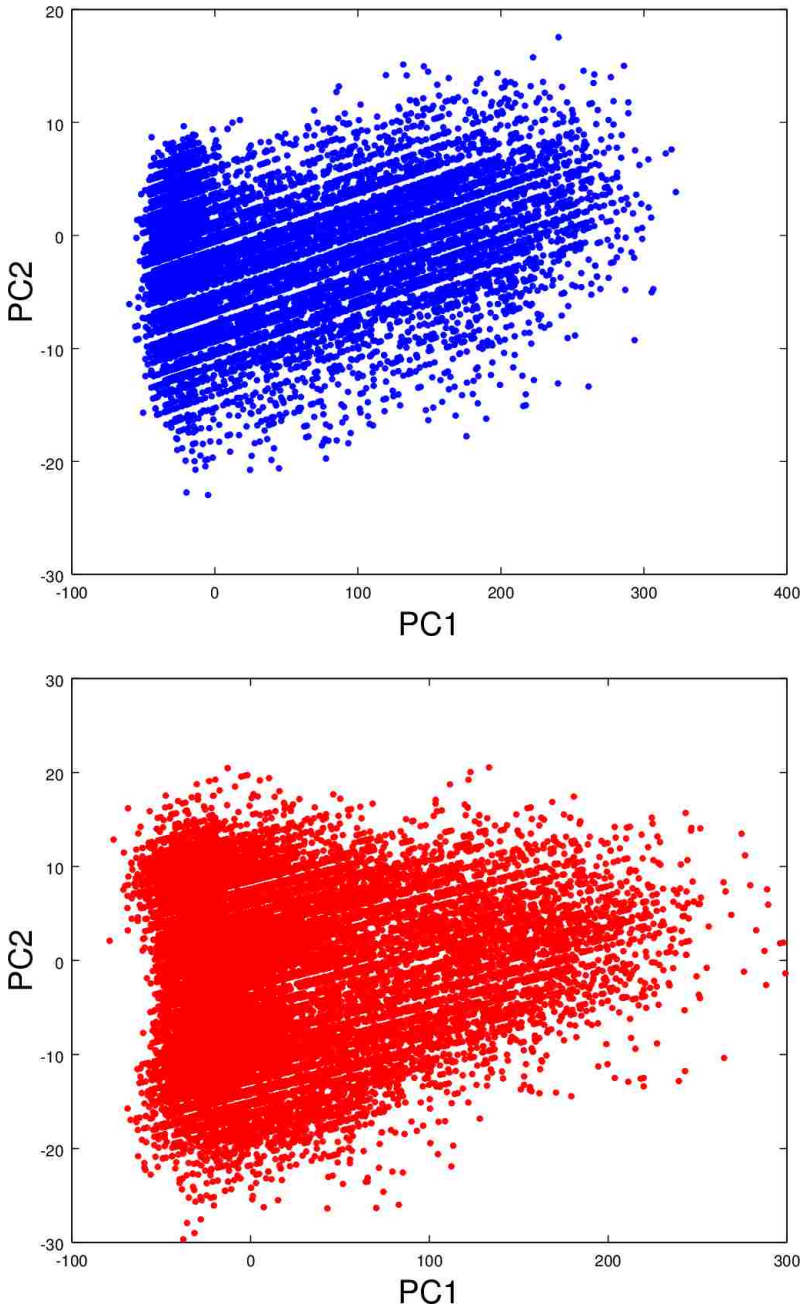


Figure 3.57: First and second principal components for group 2 sample set 3.

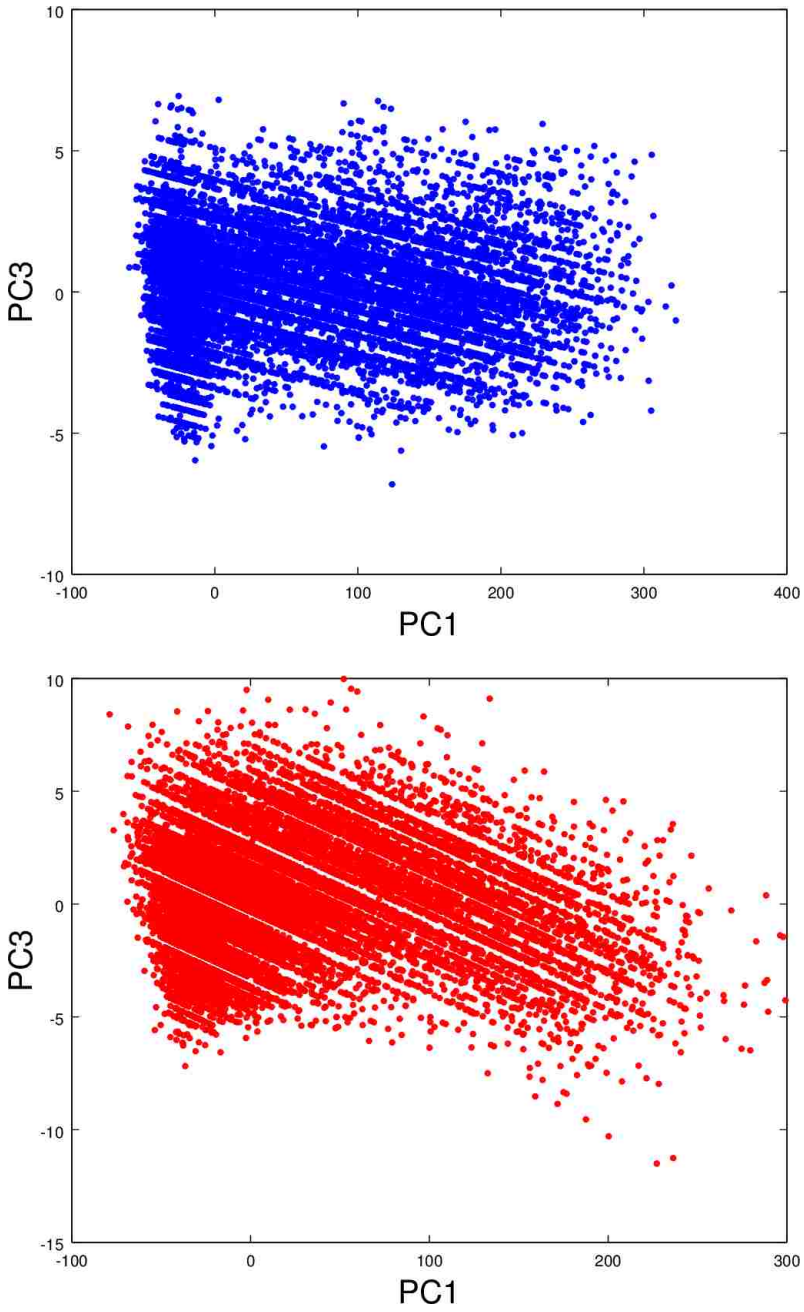


Figure 3.58: First and third principal components for group 2 sample set 3.

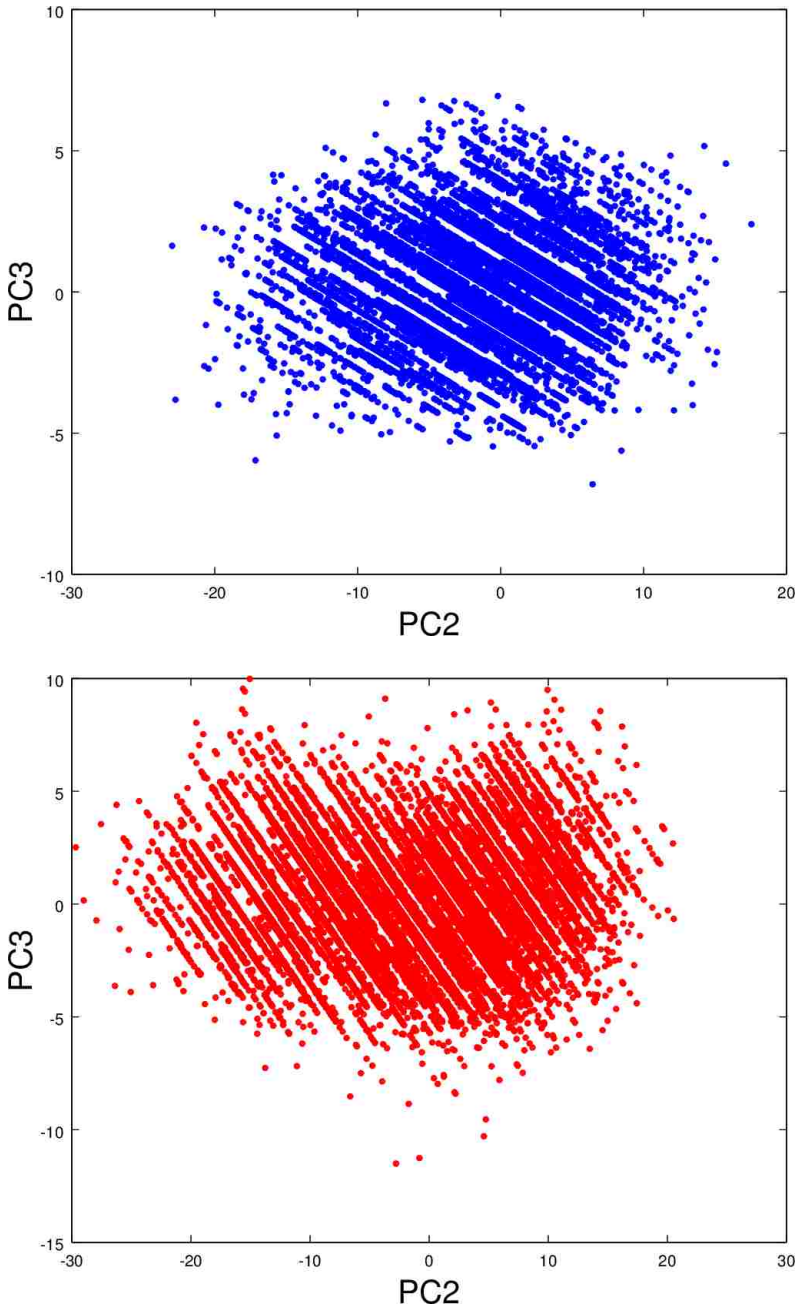


Figure 3.59: Second and third principal components for group 2 sample set 3.

Eigenvalue	Eigenvector		
3290.302818933097	0.609748539718542	0.5881676799082678	0.5312866444984584
19.14943538936752	-0.7087023684186949	0.1044499363460112	0.6977328742398817
2.499416637434105	-0.3548910696380483	0.8019657044466676	-0.4805240239389931

Table 3.31: Clean results of group 3 sample 1.

Eigenvalue	Eigenvector		
4072.643934293934	0.6344450841993349	0.5905163874983442	0.4987683141812483
27.25759324965203	0.6117844740267014	0.01076643656495638	-0.7909512255402134
2.88462536836114	0.4724396178086376	-0.8069538276379871	0.3544380447768418

Table 3.32: Dirty results of group 3 sample 1.

Eigenvalue	Eigenvector		
3121.925817692429	0.6066210266810175	0.5873238409765121	0.5357813320834021
16.4522091504832	-0.7164993127579606	0.1119148559890755	0.6885519586976092
2.523652507821268	-0.3444410904724315	0.8015770523346717	-0.488707033266988

Table 3.33: Clean results of group 3 sample 2.

Eigenvalue	Eigenvector		
4097.106236164673	0.6522168898208406	0.5933080326680173	0.4718036742163376
55.27947266622876	0.563476587731584	0.03686925878418695	-0.8253089075219501
3.333857702637625	0.5070574560252515	-0.8041307332320482	0.3102684324272917

Table 3.34: Dirty results of group 3 sample 2.

Eigenvalue	Eigenvector		
4095.566856141882	0.608436734994489	0.5849419541109724	0.5363279312417758
15.65633045017098	0.662070638547968	-0.001508758145177966	-0.7494399196877263
2.353923823831428	0.4375696619761613	-0.8110737537176852	0.3881908769514961

Table 3.35: Clean results of group 3 sample 3.

Eigenvalue	Eigenvector		
3371.667680659702	0.607483882937955	0.5881380259360441	0.5339072900969586
24.74639566039329	-0.7047380294586908	0.08895208343186903	0.7038691900401742
3.730120082314824	-0.3664800701338428	0.8038539601825585	-0.4685201904865047

Table 3.36: Dirty results of group 3 sample 3.

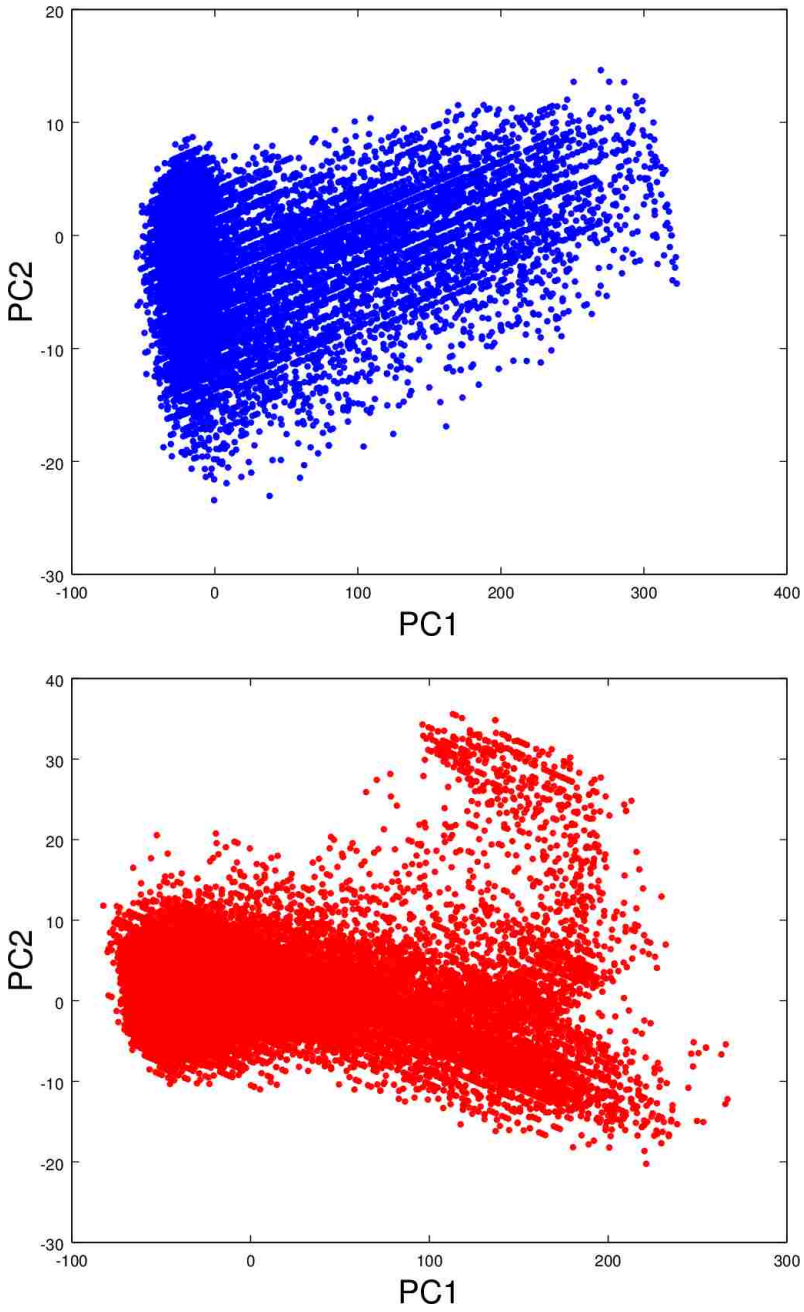


Figure 3.60: First and second principal components for group 3 sample set 1.

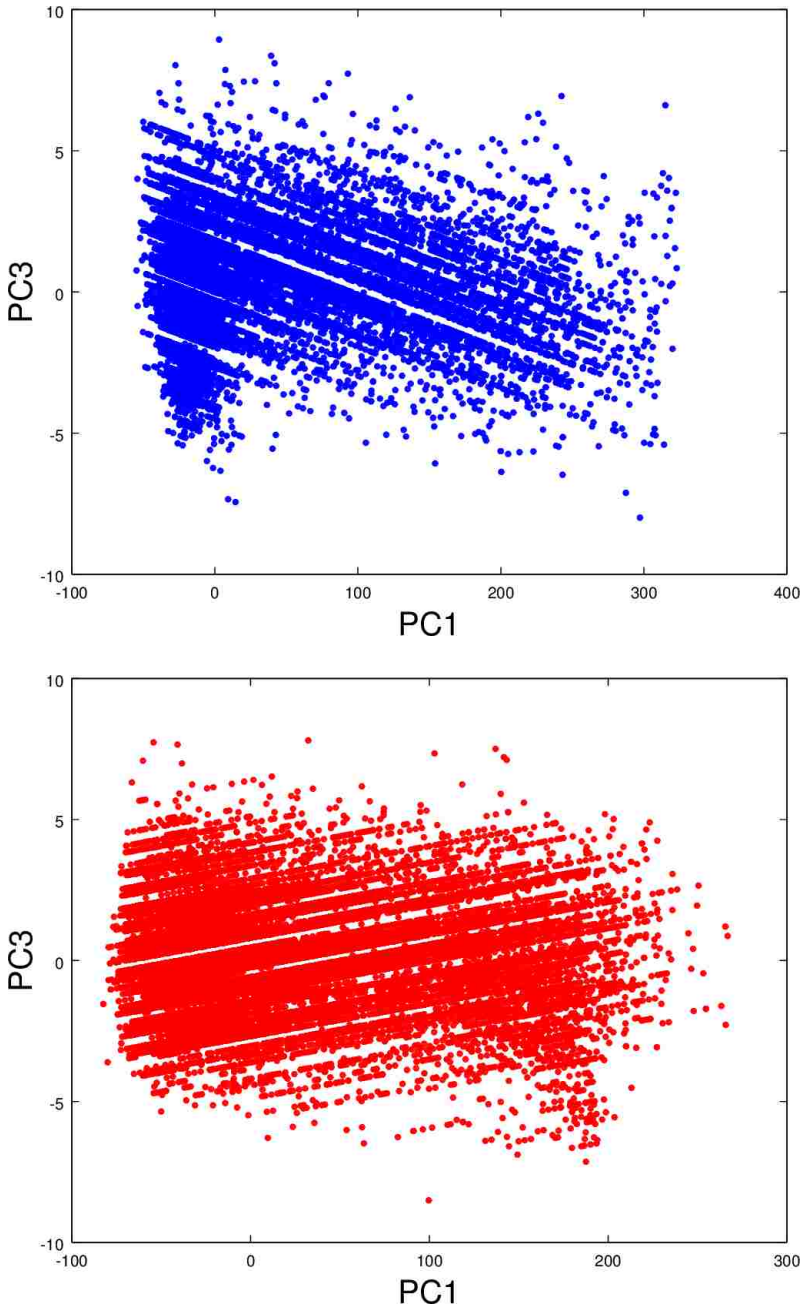


Figure 3.61: First and third principal components for group 3 sample set 1.

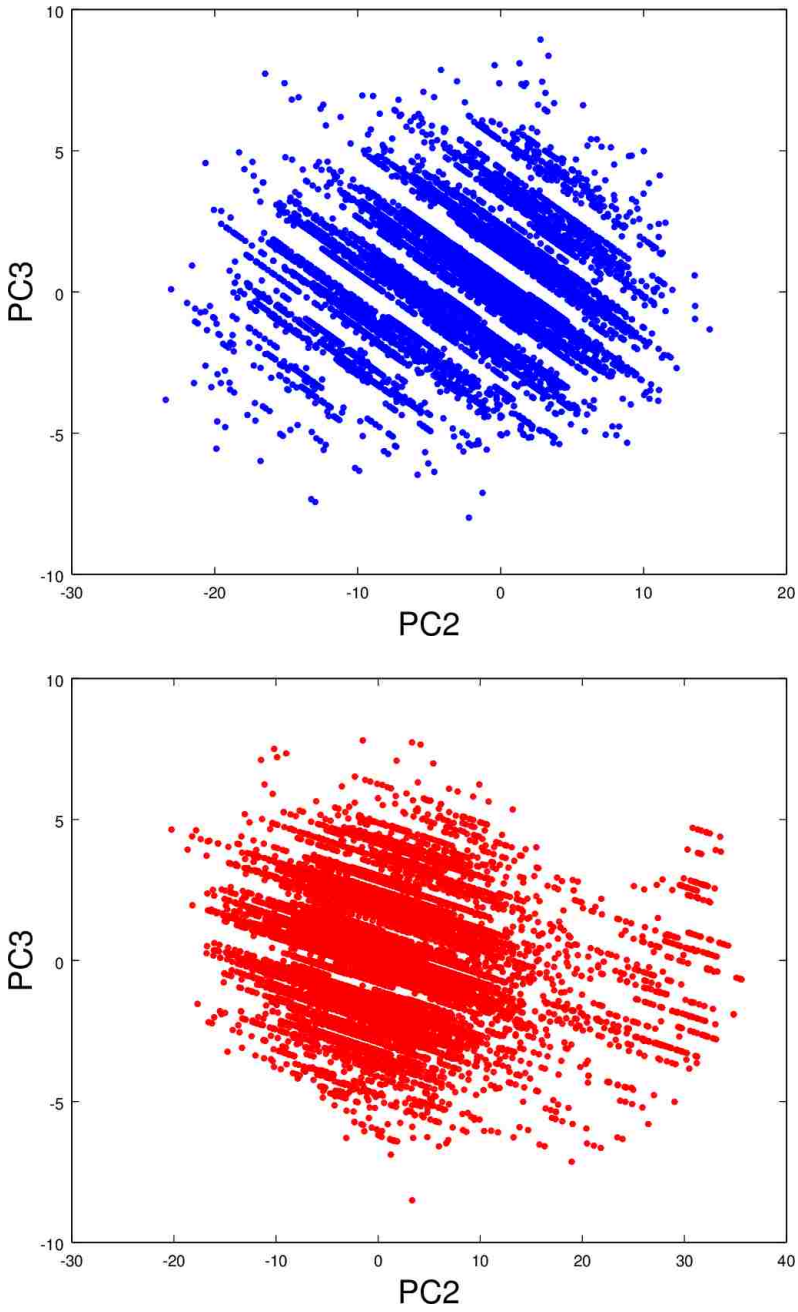


Figure 3.62: Second and third principal components for group 3 sample set 1.

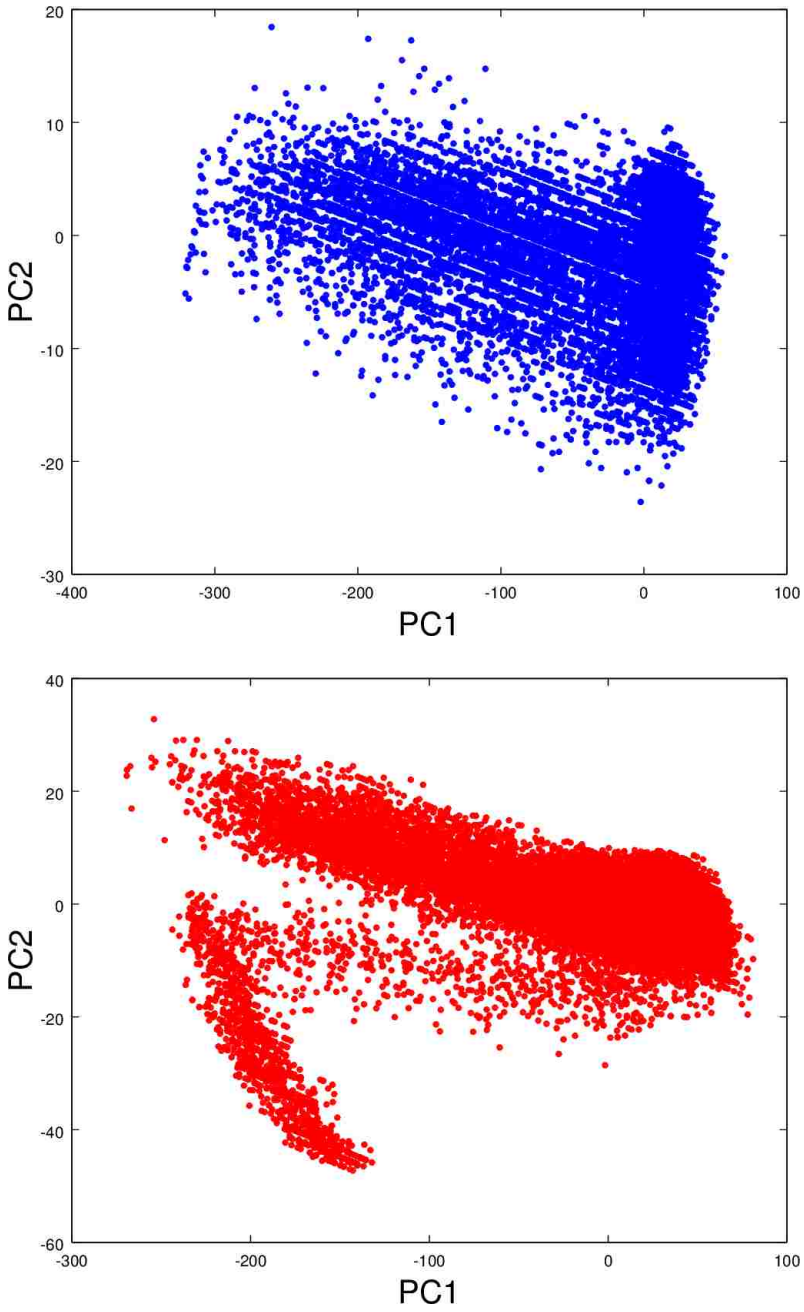


Figure 3.63: First and second principal components for group 3 sample set 2.

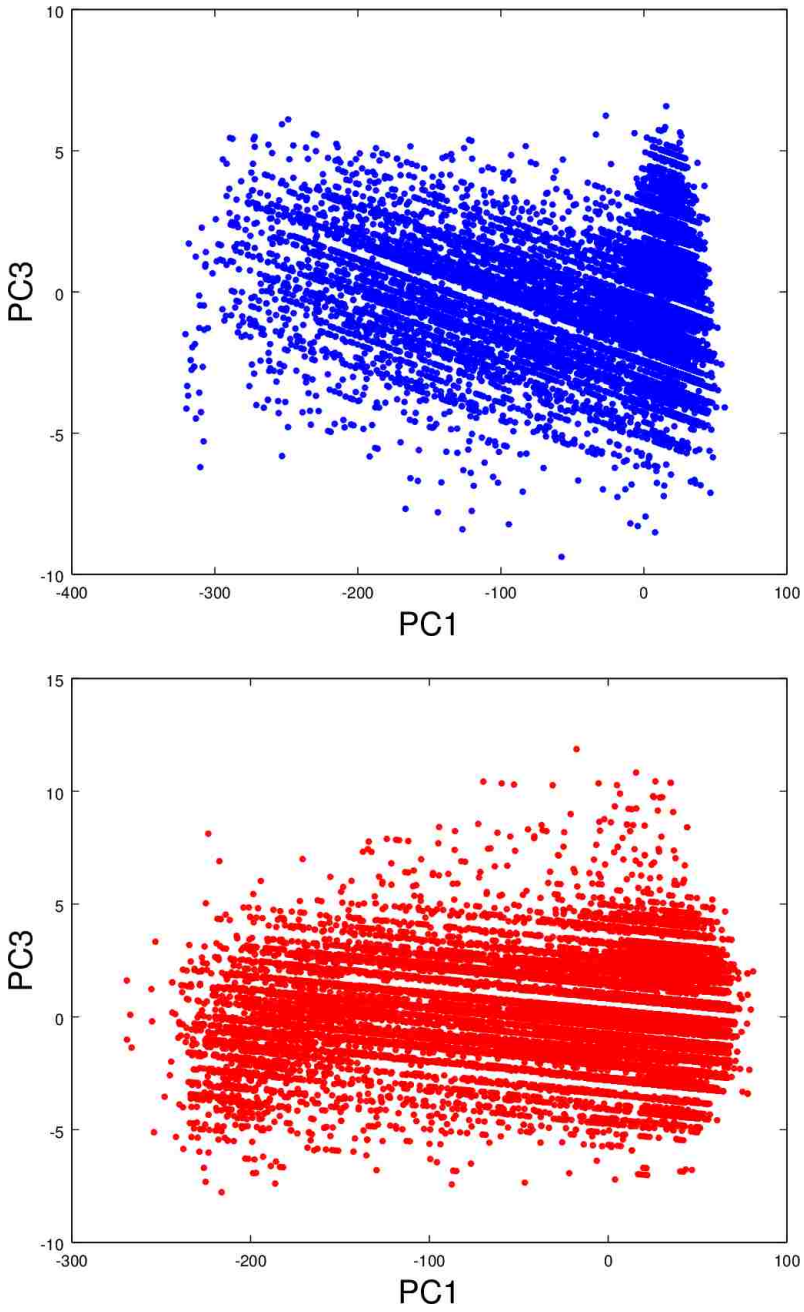


Figure 3.64: First and third principal components for group 3 sample set 2.

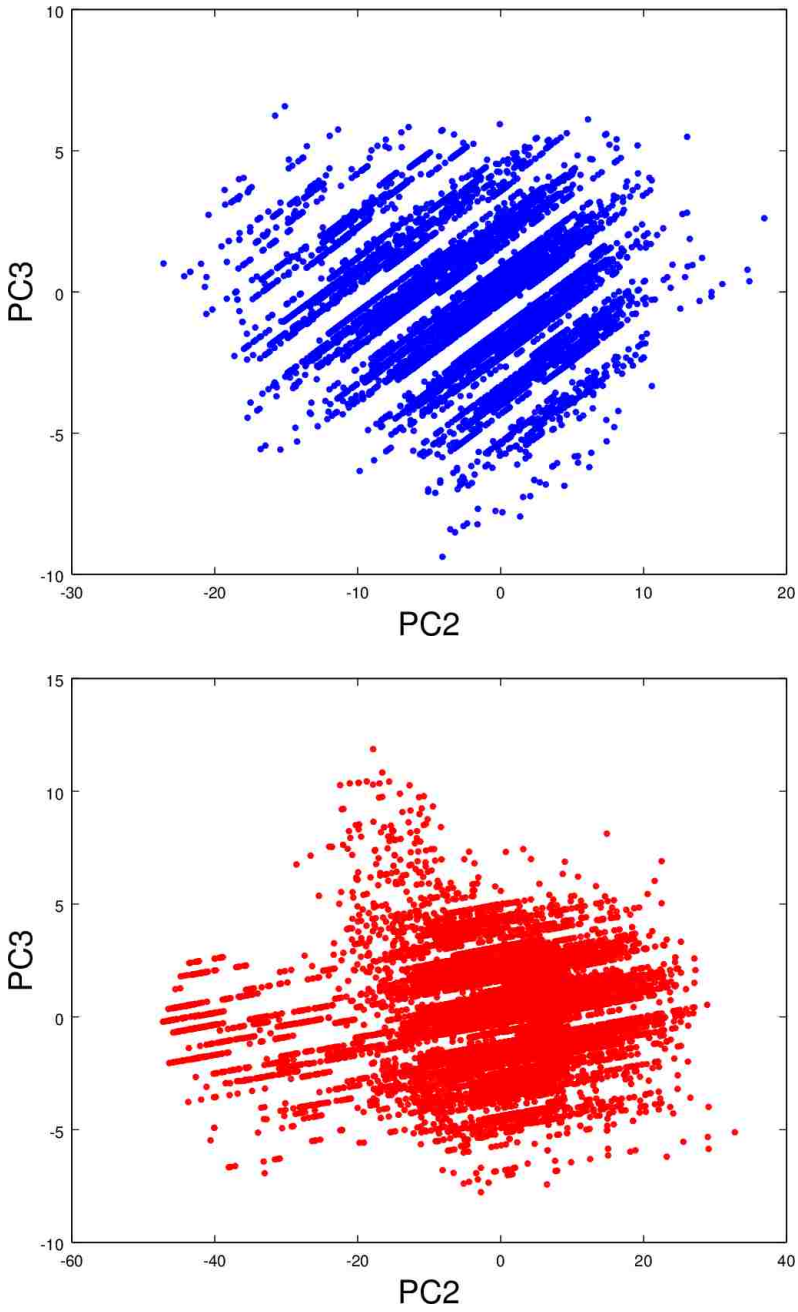


Figure 3.65: Second and third principal components for group 3 sample set 2.

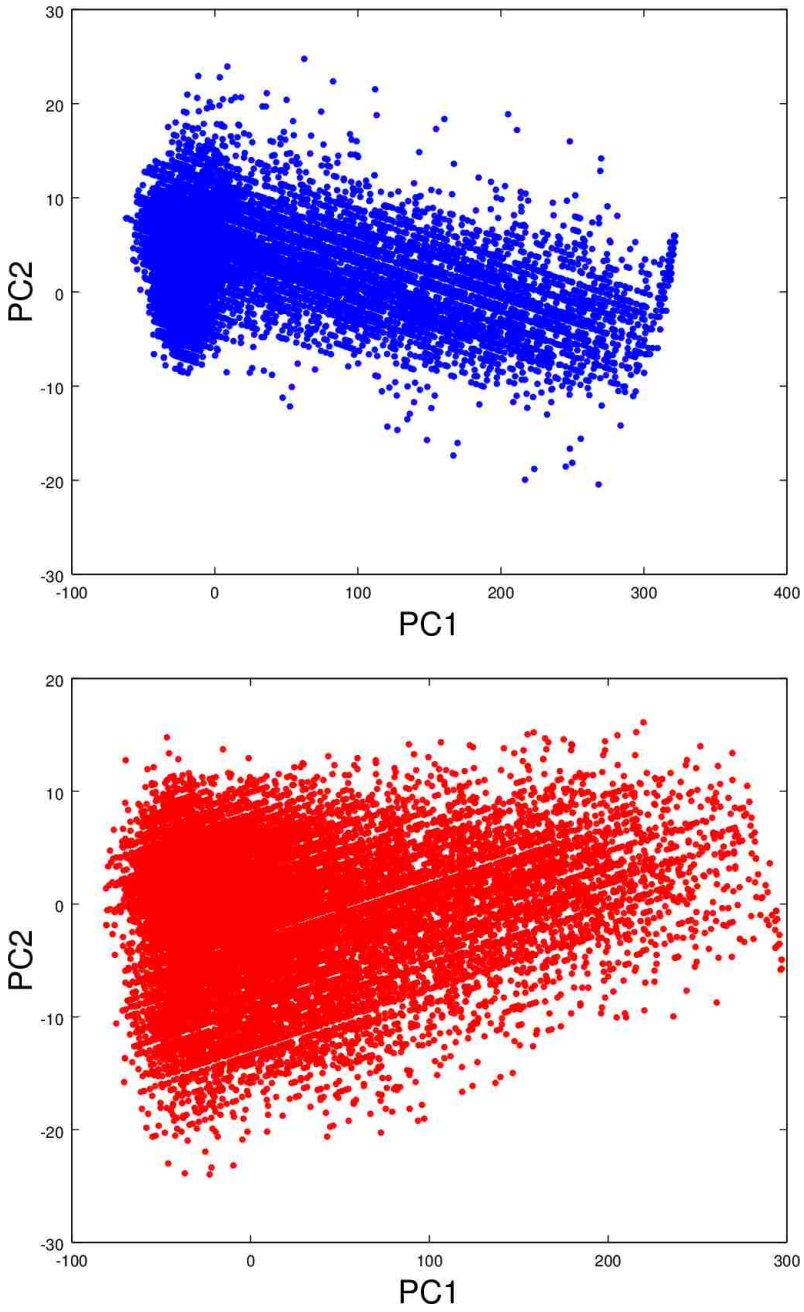


Figure 3.66: First and second principal components for group 3 sample set 3.

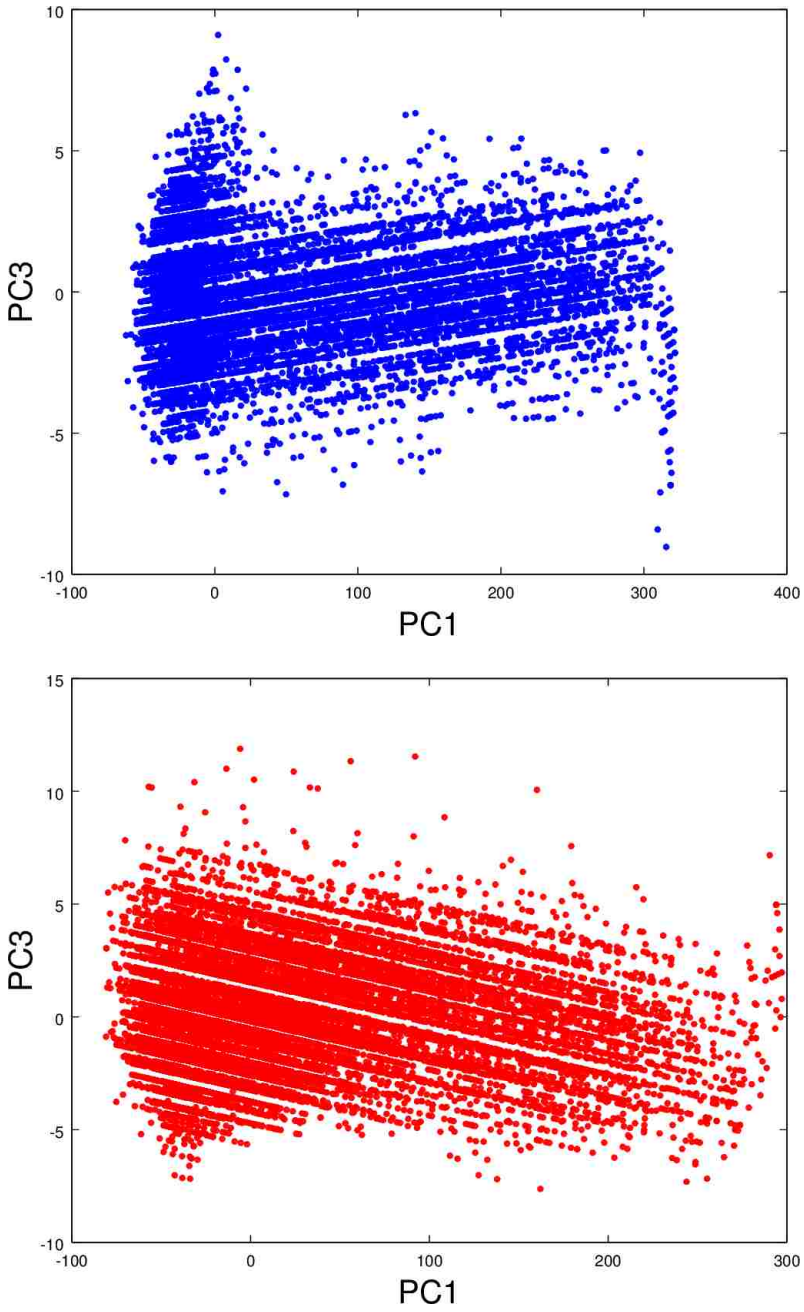


Figure 3.67: First and third principal components for group 3 sample set 3.

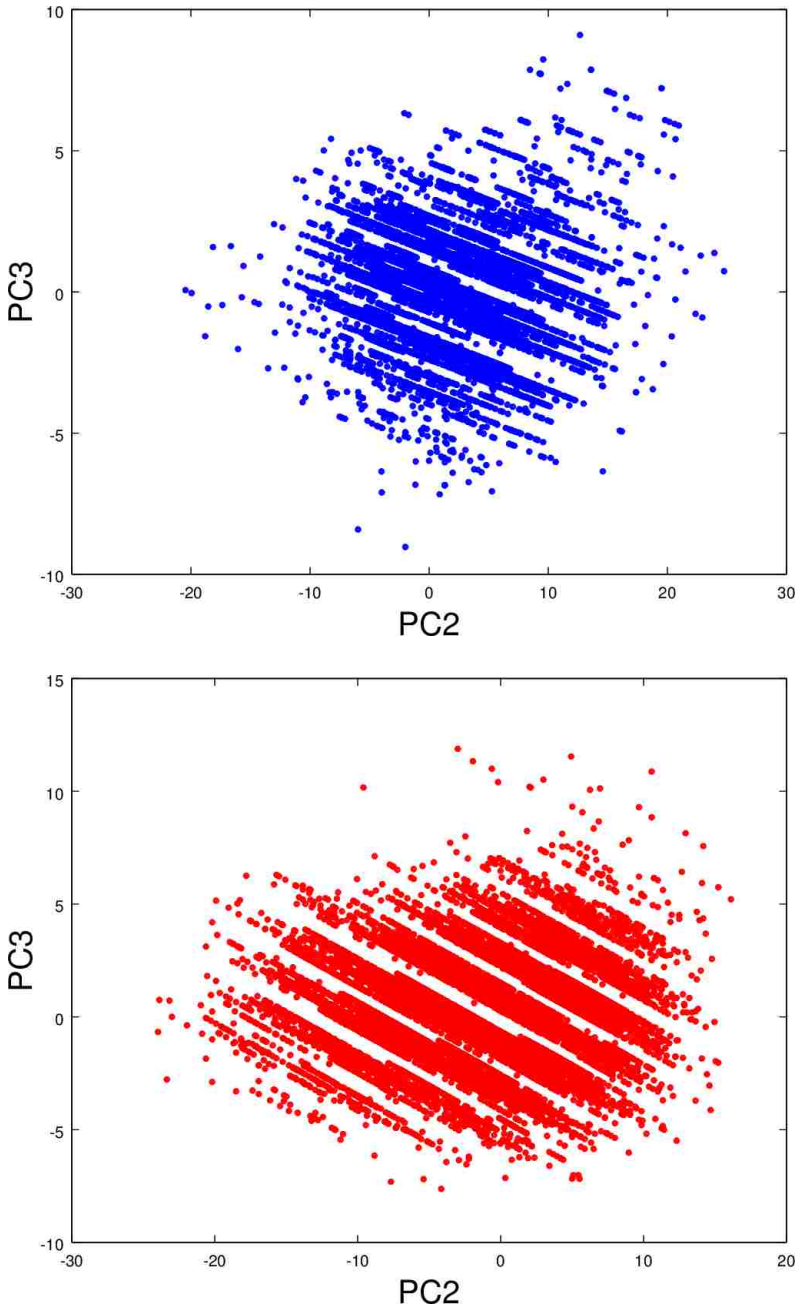


Figure 3.68: Second and third principal components for group 3 sample set 3.

3.6 Future Work

There is plenty of future work to be done. We must create a more comprehensive experimental design and more extensive sampling strategy. Moreover, we need to handle the noise in the data. There are various solutions to eliminate noise, and we should explore them. We can obtain the voltage output and current from the panels using a Voltometer and Amperometer, respectively, and knowing the voltage and current we can find the power, which is a product of the two. This data also gives insight to noise, and we can devise filters to get rid of that noise. For starters, since noise is a random phenomenon expressed by a probability distribution function, and it is a high frequency, it can be eliminated by applying a low pass filter to the original data. Furthermore, we can look at the cross-correlation between wind velocity and voltage and current output, because the cooler the panels remain, the more voltage there is, so the more power there is. Additionally, a common theme of our research so far has been to look at qualitative, pictoral results, but the next step is to quantify the results against a statistical model. Lastly, we can apply this algorithm in real time to detect when a panel is dirty and to trigger automated cleaning mechanisms.

Appendix A

Source Code

The source code can be found on this remote git repository:
`https://bitbucket.org/suziee/unlv-thesis-code/`

Bibliography

[BF85] Craig F. Bohren and Alistair B. Fraser. Colors of the Sky. *The Physics Teacher*, (5):269–272, 1985.

[FPWW00] Robert Fisher, Simon Perkins, Ashley Walker, and Erik Wolfart. Fourier Transform. `http://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm`, 2000. [Online; accessed 18-May-2014].

[GW08] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Pearson Prentice Hall, Upper Saddle River, New Jersey, third edition, 2008.

[HP08] Panagiotis E. Haralabidis and Christodoulos Pilinis. Skylight Color Shifts due to Variations of Urban-Industrial Aerosol Properties: Observer Color Difference Sensitivity Compared to a Digital Camera. *Aerosol Science and Technology*, 8(42):658–673, 2008.

[Mai12] Ranjan Maitra. Discrimination and Classification - Introduction. `http://www.public.iastate.edu/~maitra/stat501/lectures/Classification-I.pdf`, 2012. [Online; accessed 30-May-2014].

[MBY+06] Malay K. Mazumder, Alexandru S. Biris, Caner U. Yurteri, Robert A. Sims, Charles E. Johnson, Rajesh Sharma, Karin Pruessner, Carlos I. Calle, Steve Trigwell, Charles R. Buhler, and Sid Clements. Solar Panel Obscuration by Dust and its Mitigation in the Martian Atmosphere. *Particles on Surfaces 9: Detection, Adhesion and Removal*, pages 1–29, 2006.

[PTVF92] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Natural Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, New York, second edition, 1992.

[Ram11] Ernesto Zamora Ramos. Using Image Processing Techniques to Estimate the Air Quality. *Journal of McNair Scholars Institute*, pages 189–194, 2011.

[RVM89] C. P. Ryan, Frank Vignola, and David K. McDaniels. Solar cell arrays: Degradation due to dirt. *Proceedings of 1989 Annual Conference of The American Solar Energy Society*, pages 234–237, 1989.

[Shl03] Jon Shlens. A Tutorial on Principal Component Analysis; Derivation, Discussion and Singular Value Decomposition. `http://www.cs.princeton.edu/picasso/mats/PCA-Tutorial-Intuition_jp.pdf`, 2003. [Online; accessed 18-May-2014].

[TSG91] Si-Chee Tsay, Graeme L. Stephens, and Thomas J. Greenwald. An investigation of aerosol microstructure on visual air quality. *Atmospheric Environment*, 25A(5/6):1039–1053, 1991.

[Wei] Eric W. Weisstein. Eigen Decomposition. `http://mathworld.wolfram.com/EigenDecomposition.html`. [Online; accessed 18-May-2014; From MathWorld–A Wolfram Web Resource].

[WMMY12] Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, and Keying Ye. *Probability & Statistics for Engineers & Scientists*. Prentice Hall, ninth edition, 2012.

Vita

Graduate College
University of Nevada, Las Vegas

Suzanna Ho

Degrees:

Bachelor of Science in Computer Science 2010
University of Nevada Las Vegas

Thesis Title: Spectral Decomposition of the Scattered Light due to Deposits on the Solar Panel Surface, and Cross Correlated to Power Loss

Thesis Examination Committee:

Chairperson, Dr. Evangelos A. Yfantis, Ph.D.
Committee Member, Dr. John T. Minor, Ph.D.
Committee Member, Dr. Jan B. Pedersen, Ph.D.
Graduate Faculty Representative, Dr. Robert F. Boehm, Ph.D., P.E.