

**FUSION OF HYPERSPECTRAL AND DEPTH DATA USING
MORPHOLOGICAL IMAGE PROCESSING
FOR PIXEL-BASED CLASSIFICATION**

by

Abdullah Goker

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering

Summer 2018

© 2018 Abdullah Goker
All Rights Reserved

**FUSION OF HYPERSPECTRAL AND DEPTH DATA USING
MORPHOLOGICAL IMAGE PROCESSING
FOR PIXEL-BASED CLASSIFICATION**

by

Abdullah Goker

Approved: _____
Gonzalo R. Arce, Ph.D.
Professor in charge of thesis on behalf of the Advisory Committee

Approved: _____
Kenneth E. Barner, Ph.D.
Chair of the Department of Electrical and Computer Engineering

Approved: _____
Babatunde A. Ogunnaike, Ph.D.
Dean of the College Engineering

Approved: _____
Ann L. Ardis, Ph.D.
Senior Vice Provost for Graduate and Professional Education

ACKNOWLEDGMENTS

First and foremost, I thank Allah for endowing me with the health, strength, patience, and knowledge to complete this work. Without the help of Allah, I am nothing.

Special thanks to the Turkish Educational Ministry that has supported me throughout my graduate studies.

I acknowledge the inspiration, encouragement, valuable time and guidance given to me by Dr. Gonzalo R. Arce, who has served as my advisor. His dynamism, vision, sincerity, and motivation have deeply inspired me. It has been a great privilege to work and study under his guidance.

I would like to express my deepest gratitude to my loving mother, father, sister, brother-in-law, and my cute niece for their emotional and moral support throughout my academic career, and also for their love, patience, encouragement, and prayers. Special thanks to my beautiful wife for being there for me through difficult times.

Thanks are due to my colleagues Juan, Yuksel, Carlos, Xiao, Edgar, Hoover, Claudia, Chen, Karelia, Laura, Angela, and Alejandro for their sincere and untiring efforts in assisting me during all stages of my research. Sincere thanks to my friends Abdul Rehman and Emily who helped me with the writing process. Finally, I would like to offer my deepest apologies for having unintentionally missed naming anyone who has contributed to this project.

TABLE OF CONTENTS

LIST OF FIGURES	vi
ABSTRACT	viii
Chapter	
1 INTRODUCTION	1
1.1 Literature	2
1.2 Structure	3
2 HYPERSPECTRAL IMAGING	4
2.1 Image Acquisition	6
2.2 Reflectance Spectrum	7
3 MORPHOLOGICAL IMAGING	8
3.1 Basics of Mathematical Morphology	8
3.1.1 Structural Element	9
3.1.2 Set Operations	9
3.1.3 Dilation and Erosion	10
3.1.4 Opening and Closing	12
3.1.5 Some Types of Structural Elements	15
3.2 Reconstruction	15
3.2.1 Partial Reconstruction	18
4 DIMENSION REDUCTION AND DATA FUSION	21
4.1 Principal Component Analysis	21
4.2 Kernel Principal Component Analysis	22
4.3 Locality Preserving Projections	25

5 DATA SETS, PROPOSED METHOD AND EXPERIMENTAL RESULTS	27
5.1 Data Sets	27
5.2 Experiment	30
5.2.1 Principal Component Analysis	30
5.2.2 Morphological Profile	30
5.2.3 Dimension Normalization with Kernel Principal Component Analysis	32
5.2.4 Feature Fusion with Locality Preserving Projections	32
5.2.5 Classification	33
6 CONCLUSION	36
BIBLIOGRAPHY	37

LIST OF FIGURES

2.1	The cube is a 3D example of a hyperspectral image. Plots are the reflectance spectrum for different materials. The image is taken from [1].	5
2.2	Data is taken for every location as a 2D image with axes x (location) and λ (wavelength) in a pushbroom spectrometer. On the other hand, one 2D image is taken at a time at a specific wavelength using a tunable filter. This image is taken from [2].	6
3.1	Stepwise growing and subsequent shrinking. Image is taken from [3]	9
3.2	Fitting and hitting positions of structuring element. Image is taken from [4].	11
3.3	Dilation and erosion by a square SE with the origin at the lower-left corner[5].	11
3.4	When a disk-shaped opening with radius, $R=1$, is applied, some of the small white dots are removed from the image. When a disk-shaped opening with $R=2$ is applied, all of the small dots, most of the lines, and some of the numbers are deleted. When $R=8$ is applied, all of the white pixels that are smaller than SE disappear. Only the borderline of the football field remains.	13
3.5	A linear opening operator behaves slightly differently than a disk-shaped SE. When the length, $L=10$, all of the small dots are removed. The rest of the objects remain unchanged. When $L=20$, some of the numbers disappear because they do not have any orientation in the same direction that is bigger than 20 pixels. When $L=50$, only the lines of the football field remain in the image, because all of the lines are larger than 50 pixels.	14

3.6	Original Lidar image is taken from [6]. An opening applied image. More objects disappear with the increasing size of the SE, and more degradation effects can be observed. While the blobby effect is mostly on the objects that have disappeared, the rounding effect of rectangular shapes can be seen on the remaining objects.	16
3.7	A closing works on objects that are darker when compared to their surroundings. Degradation effects are easily observed, as with the opening operator in Figure 3.6.	17
3.8	Original image, disk-shaped SE applied image and partially reconstructed image.	19
3.9	Original image, linear SE applied image and partially reconstructed image.	20
5.1	Image under the white light.	27
5.2	Imaging System	28
5.3	18 bands of hyperspectral image.	29
5.4	Depth image.	30
5.5	First 3 principal component of hyperspectral image	31
5.6	Thematic map using the proposed fused features with color code(numbers at the color codes represent the depth level). SVM classifier works well except some regions. These are caused by two reasons. First, the stereo vision system is not precise in the boundaries. Second, pixels such as those in the real apple (upper right) are caused by specular reflection. The surface of the real apple is very much shiny that the camera sees reflection of the light source's light.	35

ABSTRACT

The availability of multi-sensor data from the same field of view has increased drastically with recent developments in sensor technologies. There are many image processing algorithms to extract different features of objects from sensors, but no single-sensor technology is sufficient to provide dependable classification. Extracting features from multiple sources with morphological operations gives rise to problems like the curse of dimensionality, which degrades the performance of the classifier and considerably increases the computational time. In order to overcome this problem, in this project the features are fused in a lower dimensional space, while as much information as possible about the features of the pixels is preserved. In this way, the classification performance of the given system can be enhanced.

Chapter 1

INTRODUCTION

With the development of new sensor technologies in imaging systems, data fusion has become a promising approach to classify objects. In particular, hyperspectral imagers can supply detailed information about the materials because different materials have unique spectral signatures in a different wavelength. On the other hand, height information (depth maps) is widely used in many applications for characterizing topographical information. Since fusion of these hyperspectral imagers with depth information is often used in remote sensing applications [7], [8], the necessity of both of the data sets is exemplified as follows.

Automated pixel based detection systems are well suited to distinguish man-made objects in natural environments by using hyperspectral information. For example, it is possible to detect the difference between concrete-based parking space and asphalt-based parking space via hyperspectral imagers. However, it is not possible to distinguish between objects made of the same material with this method. Hence a parking lot made of concrete and the concrete based roof of a building cannot be separated without height knowledge. On the other hand, height information cannot separate objects with the same elevation.

In order to have a better classification of objects, geometrical features of objects should be involved in the process of classification, because spectral and height characteristics of some objects such as road surfaces, parking lots, and open areas could be very similar. Therefore, using morphological features for improving classification accuracies is essential. For improved classification of very high-resolution images in urban areas, using the morphological features to classify geometrical features is studied in [9].

In order to overcome the limitations of one particular modality, fused data sets can carry additional information about pixels. When received through a variety of different sensors and preprocesses, the fusion of these data may look like a stacking of all the features. Although this fusion method appears simple, when the classifiers are run, the system may not perform better than it does with single features. This is because information from the features could be redundant, and not have an equal effect on the system. Another issue about stacking these features is the curse of dimensionality, which can cause overfitting of classifiers, which in turn can cause reduced classification accuracies. The solution to these problems is using methods which can reduce the dimensionality while keeping as much intrinsic information as possible. In this work, Principal Component Analysis, Kernel Principal Component Analysis and Locality Preserving Projections (LPP) are applied to the system to reduce the dimensionality without losing necessary information.

This paper will explore the fusion of hyperspectral and depth data acquired in University of Delaware Computational Image Processing Laboratory. Morphological processing will be applied to the data in order to produce information about the geometrical features. After the spatial, spectral, and height information is collected from the data, the dimension normalization and fusion techniques are applied. The Support Vector Machines (SVM) will be used to classify the data. This classification will work for different materials, shapes, and depths. In this paper it will be applied to a scene created from both synthetic and real materials at different depths.

1.1 Literature

Multi-sensor data fusion with hyperspectral imaging (HSI) is a common research area, and fusion with HSI has generated much interest due to its many applications. The fusion of HSI and Light Detection and Ranging (LiDAR) is a common challenge in remote sensing applications. In [10], a novel graph-based fusion technique was proposed to couple dimensionality reduction and data fusion of the topological features of HSI and LiDAR. In [11], a two stream fusion framework was proposed to find

pixel-based and object-based classification. In [12], a new methodology was presented which fuses LiDAR and HSI for landscape visual quality. In [13], hyperspectral data and bathymetric laser data was used in a complementary fashion in Kaneohe Bay to achieve bottom classification and increase the density of depth measurements. It was shown in [14] that fusing data from HSI sensors with data from foliage-penetration synthetic aperture radar (FOPEN SAR) can increase overall detection and classification performance. The enhanced performance can also be achieved when we fuse data of HSI sensor with high-resolution imaging (HRI). In [15], while synthetic aperture radar (SAR) was used to penetrate tree canopies for detection, HSI was used to distinguish spectral features between background and targets in a study with military support applications.

1.2 Structure

The organization of the paper is as follows. Chapter 2 gives a review of HSI and how hyperspectral imagers can be used to distinguish between objects. In Chapter 3, mathematical morphology and reconstruction methods for preventing the degradation of images are presented. In Chapter 4, an exploration of dimension reduction and the fusion of data using Principal Component Analysis, Kernel Principal Component Analysis, and Locality Preserving Projections is presented. Data sets, proposed method and experimental results are presented in Chapter 5 and the conclusions are presented in Chapter 6.

Chapter 2

HYPERSPETRAL IMAGING

Recent advances in computer technologies, high speed, and memory, have opened the door to the study of a variety of applications in multispectral imaging. Imaging systems today can reconstruct 3D images of human organs through computer tomography, show a 3D display of an aircraft approaching an airport, and supply enhanced and stored digital photographs before printing [16]. Multispectral imaging also overcomes the problem of conventional cameras being poor identifiers of similar color surfaces.

HSI, which acquires images of many more narrow continuous spectral bands than multispectral imaging, was originally developed for remote sensing applications [17]. HSI enhances research in a diverse range of fields with the detailed information of data acquired by imaging spectrometers. These fields include: agriculture and forestry [18], food quality and safety control [19], astronomy and space surveillance [20], and even classification for pigment identification and mapping in painting [21].

A hyperspectral imager consists of a light source, a monochromator or filter system to scatter the light into wavelengths, and a detection system [22]. Acquired images actually consist of hundreds of images with the third dimension of wavelength. Different absorption and reflectance characteristics of materials in different wavelengths provide identifier information about the materials. An example figure is shown in Figure 2.1.

In the figure 2.1, it is clear that the spectral signatures of different materials are clear identifiers.

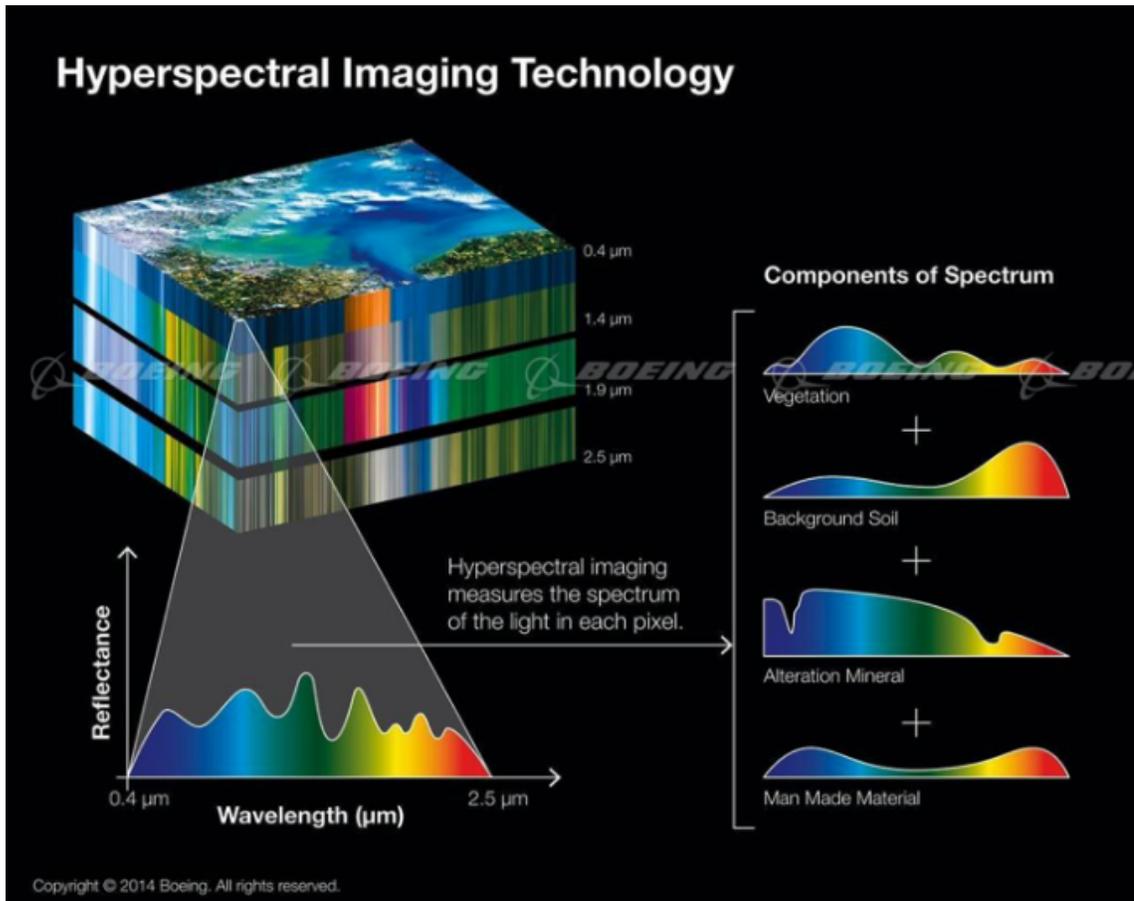


Figure 2.1: The cube is a 3D example of a hyperspectral image. Plots are the reflectance spectrum for different materials. The image is taken from [1].

2.1 Image Acquisition

Two main approaches to hyperspectral image acquisition are introduced in [19]. One is known as the Staring Imager (SI). This technique is not capable of obtaining hyper-cube images simultaneously. In this technique, the field of view must be fixed, and one 2D image is taken at a time at a specific wavelength by using a tunable filter. After that, images taken for every wavelength are stacked in sequence and stored in Band Sequential Format (BSQ).

The other method is known as Pushbroom Imager which requires relative movement between the detector and the object. In this technique, simultaneous spectral measurements from adjacent spatial positions are taken. The spectral response of a point is taken, and then the camera or object is moved to the next position and another spectral response is taken. Data taken from every location is stacked and stored in Band Interleaved Format. The acquisition time of hypercube has been reduced with the improvement of detector technology. Recording lines with spectral information by an array detector and storing them in the Band Interleaved by Line(BIL) format is also a good option. The aforementioned two methods are shown in Figure 2.2.

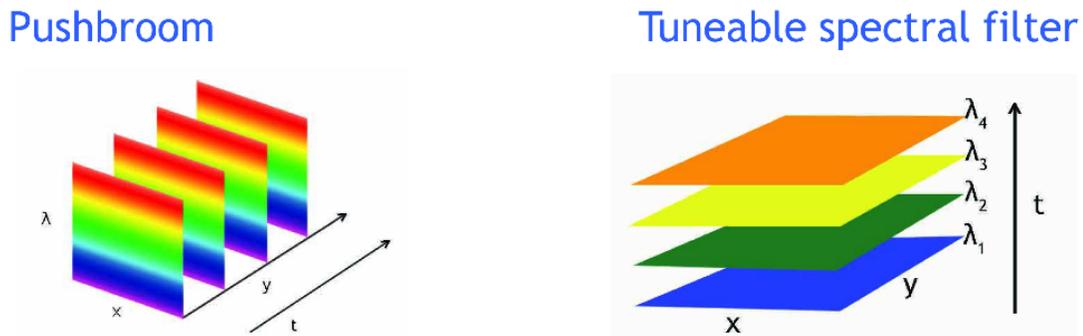


Figure 2.2: Data is taken for every location as a 2D image with axes x (location) and λ (wavelength) in a pushbroom spectrometer. On the other hand, one 2D image is taken at a time at a specific wavelength using a tunable filter. This image is taken from [2].

The system for this project used the SI method to acquire the hypercube. Contiguous filters were then used to tune the light source for each image.

2.2 Reflectance Spectrum

As similar materials have similar spectral responses, it is simple to classify materials with hyperspectral information. Once the spectral signature of a material is established, the classifiers can be trained based on known spectral signatures and used to classify unknown materials. The radiance spectra obtained by an HSI system in a different time or place, or with different illumination sources, cannot be compared because of outer or inner effects. In [23], the effects of the atmosphere were shown as an example of this situation and a solution was introduced. Reflectance spectrum, which indicates the portion of incident energy that is reflected as a function of wavelength, was explained. Thus, the properties of the illumination source and outer effects can be removed. Once the data is corrected by changeable factors, it can be compared to data in spectral libraries.

The method for calculating the corrected reflectance value (R) was explained in [19]. In this calculation, it is necessary to find background reflectance spectrum and dark camera response. Background reflectance is acquired from a scene which is from high reflectance standard or white ceramic. Dark Camera response is recorded with no light by covering the lens with a cap for complete darkness. Corrected reflectance value is found using equation 2.1.

$$R = (sample - dark)/(background - dark) \quad (2.1)$$

As the data for this project was taken at the same time, in the same place, and with the same illumination source, it was not affected by any of these variables. The only constraint was the non-uniform distribution of the light source. To overcome this problem without taking reflectance spectrum, an advanced classifier, Support Vector Machines, was used. To eliminate the illumination source non-uniform distribution problem, training samples from different spaces which were under different illumination coefficients were taken.

Chapter 3

MORPHOLOGICAL IMAGING

Mathematical morphology is a study of objects based on shape, geometry, topology, color, and neighborhood information. As explained in [5], researchers Matheron and Serra, who work in petrography and mineralogy, were pioneers of mathematical morphology. Due to their pioneering work, mathematical morphology has become a powerful tool in microscopic imaging, medical imaging, material science, pattern recognition, and computer vision.

As mentioned in Chapter 2, HSI has become a diverse field with many research applications. Improvement of sensor technologies also made hyperspectral data at very high resolution available. Spatial information, as well as spectral, is of great import to image analysis. Thus, mathematical morphology is a useful tool to analyze spatial information, and plays a fundamental role in both segmentation and classification.

In the remainder of this chapter, first, basic mathematical morphological operations are explained from the help of [3], [5], [9] and [24]. Secondly, in order to avoid the degeneration effects of morphological features, reconstruction methods are introduced from [9].

3.1 Basics of Mathematical Morphology

Filtering an image from noise can involve removing small objects from the image. It is challenging, however, to accomplish this without altering the larger objects in the image. Stepwise shrinking and subsequent growing operations filters do not leave the larger objects unaltered. This is shown in Figure 3.1.

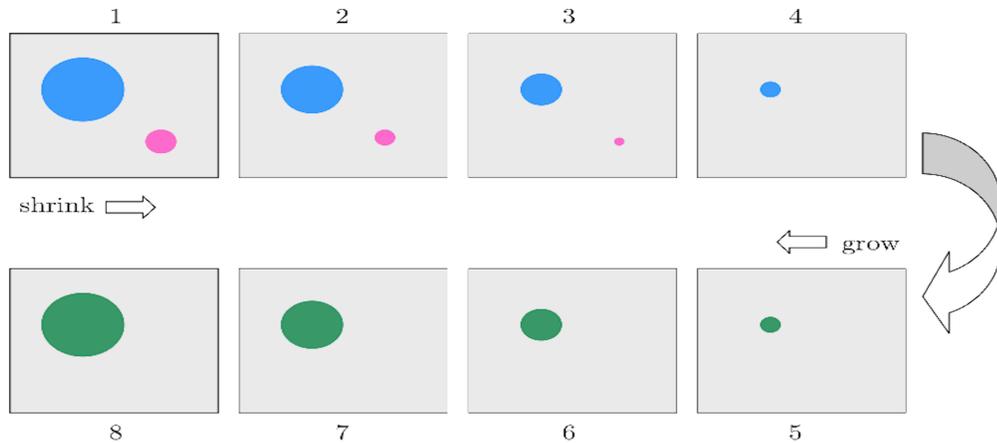


Figure 3.1: Stepwise growing and subsequent shrinking. Image is taken from [3]

The most basic morphological operations are similar to shrinking and growing but they can do much more complicated operations than removing or attaching single pixel layers.

3.1.1 Structural Element

Structural element specifies the Properties of a morphological filter and consists of the values 0 and 1. It is shown as $H(i, j) \in \{0, 1\}$. The size and the shape of the structural element can be chosen according to the application.

3.1.2 Set Operations

Set theory is the language used to understand mathematical morphology. Describing binary images as sets of two-dimensional coordinate points helps to specify symbolic notations of morphological operations. A binary image is shown as $I(u, v) \in \{0, 1\}$. Coordinate pairs $p = (u, v)$ of foreground pixels create the corresponding point set.

$$\mathcal{Q}_I = \{p \mid I(p) = 1\}. \quad (3.1)$$

The structuring element is also represented by point sets. With a given description of point sets, it is possible to perform operations on binary images with set operations. For example, building a complementary set is equal to inverting a binary image.

$$\mathcal{Q}_{\bar{I}} = \bar{\mathcal{Q}} = \{p \in \mathbb{Z}^2 \mid p \notin \mathcal{Q}_I\} \quad (3.2)$$

Since \mathcal{Q}_I is a description of an image, I will be used instead of \mathcal{Q}_I directly for point sets.

The other set operations used in describing morphological operations are translation and reflection. Translation, $I_d(p+d) = I(p)$, is equal to all coordinates in point set \mathcal{Q}_I being shifted by d .

$$I_d \equiv \{ (p+d) \mid p \in I \} \quad (3.3)$$

Reflection is mirroring of a point set about its origin.

$$H^* \equiv \{-p \mid p \in H\} \quad (3.4)$$

3.1.3 Dilation and Erosion

Dilation is a basic growing operator. As a set operation, it is defined as Minkowski addition.

$$I \oplus H \equiv \{(p+q) \mid \text{for every } p \in I, q \in H\} \quad (3.5)$$

Applying a dilation operator to an image produces a point set which is the sum of all possible pairs of coordinate points from the original sets.

Erosion is quasi-inverse of dilation and can be defined as Minkowski subtraction.

$$I \ominus H \equiv \{p \in \mathbb{Z}^2 \mid (p+q) \in I, \text{ for every } q \in H\} \quad (3.6)$$

These definitions can seem complex, however, they can be easily understood with the help of hit and fit operators, as shown in Figure 3.2.

The structural element can be seen as a window. If fitting or hitting occurs while it is shifting around the image, the dilated resulting point set is the sum of pairs

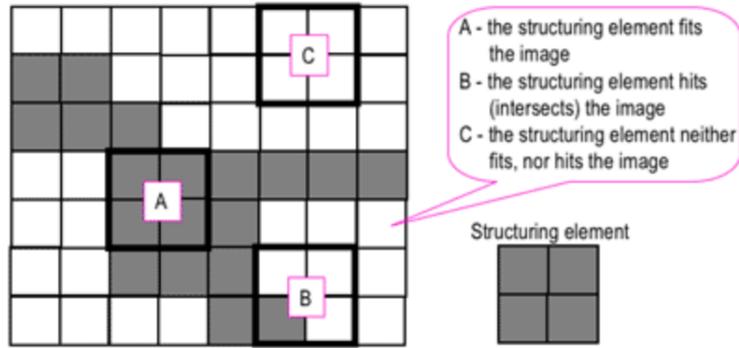


Figure 3.2: Fitting and hitting positions of structuring element. Image is taken from [4].

of the structural element and image. On the other hand, the erosion of a binary image simply produces a new image and keeps the ones in all locations (p, q) of a structuring element's origin if the structuring element fits the image. Examples of erosion and dilation operators are shown in Figure 3.3.

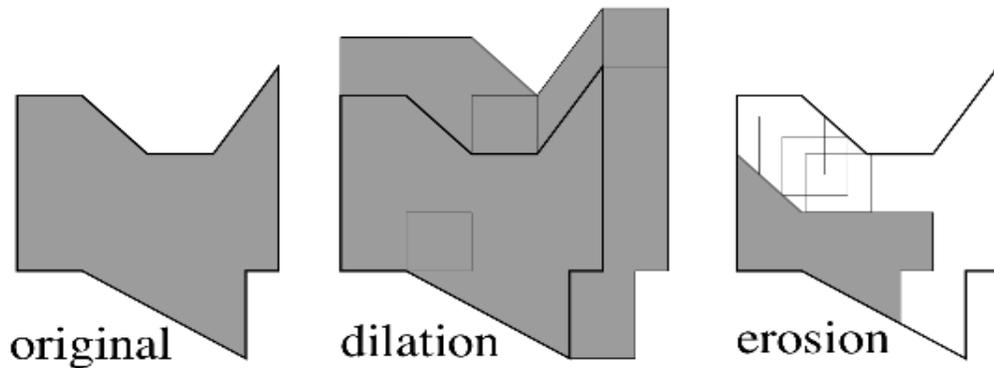


Figure 3.3: Dilation and erosion by a square SE with the origin at the lower-left corner[5].

Grayscale dilation and erosion is a generalization of binary dilation and erosion. The dilated image takes the maximum value of hitting point sets, while eroded image takes the minimum value of the hitting point sets. Gray scale dilation (equation 3.7)

and erosion (equation 3.8) are shown mathematically below.

$$I \oplus H \equiv \max_{(i+j) \in H} \{I(u+i, v+j) + H(i, j)\} \quad (3.7)$$

$$I \ominus H \equiv \min_{(i+j) \in H} \{I(u+i, v+j) - H(i, j)\} \quad (3.8)$$

3.1.4 Opening and Closing

An opening is an operation which is denoted by $I \circ H$. It is an “erosion followed by a dilation” [3] with the same structuring element.

$$A \circ B = (A \ominus B) \oplus B \quad (3.9)$$

This operation is named opening because it can open up a gap between objects connected by a thin bridge of pixels when we assume that the objects are brighter than their surroundings.

A closing is an operation which is denoted by $I \bullet H$. It is a dilation followed by erosion with the same structuring element.

$$A \bullet B = (A \oplus B) \ominus B \quad (3.10)$$

A closing operation can fill the holes while preserving the size of the initial region.

Consequently, an opening results in a new image where small bright objects are deleted. Small bright objects are assigned by their surrounding pixels. An opening does not have any effect on dark objects. A closing operator has the same effect on dark objects that an opening operator has on bright objects.

More and more objects are deleted by increasing the size of SE. Opening and closing with different sized SE is applied and the final images obtained are then stacked together in a matrix \mathbf{X} which is called Morphological profile. A morphological profile can be formed by using different SE, such as disk-shaped SE or linear SE.

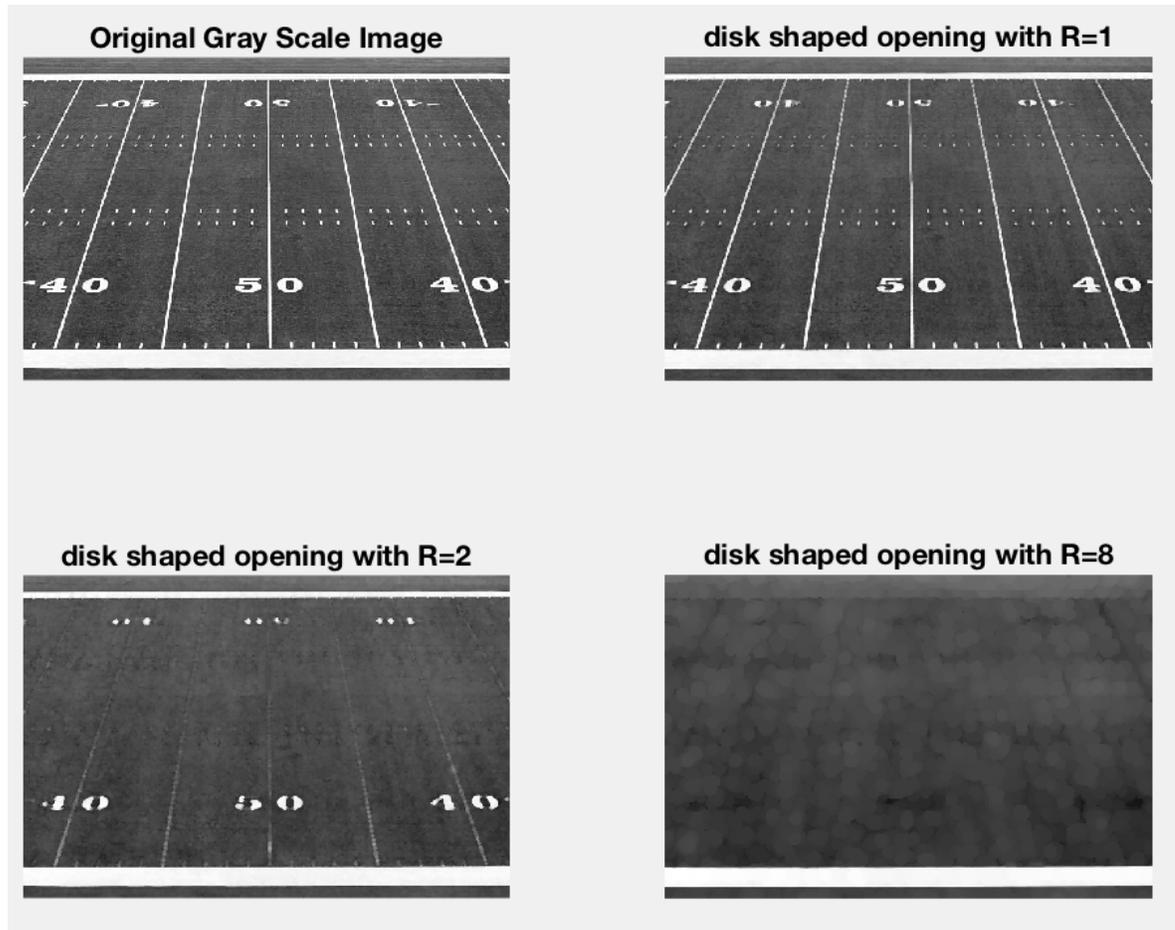


Figure 3.4: When a disk-shaped opening with radius, $R=1$, is applied, some of the small white dots are removed from the image. When a disk-shaped opening with $R=2$ is applied, all of the small dots, most of the lines, and some of the numbers are deleted. When $R=8$ is applied, all of the white pixels that are smaller than SE disappear. Only the borderline of the football field remains.

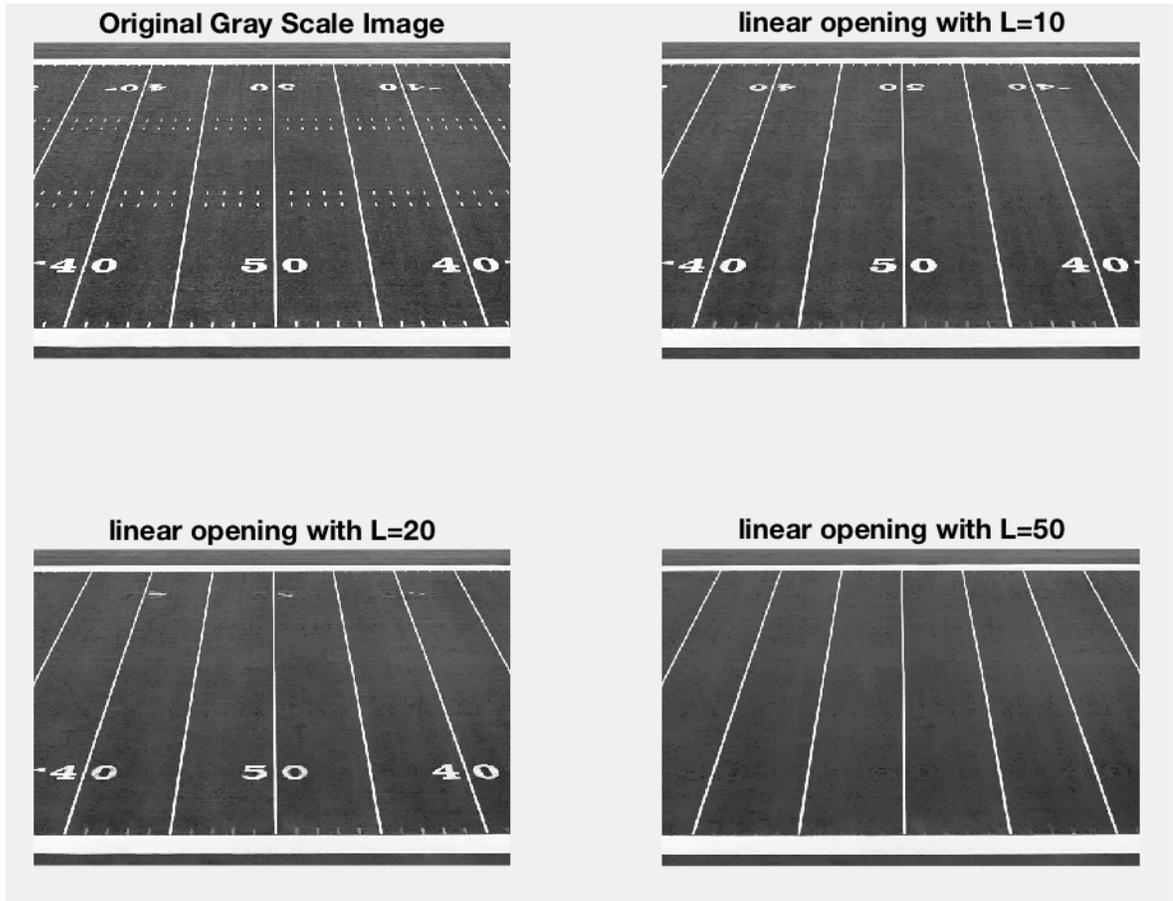


Figure 3.5: A linear opening operator behaves slightly differently than a disk-shaped SE. When the length, $L=10$, all of the small dots are removed. The rest of the objects remain unchanged. When $L=20$, some of the numbers disappear because they do not have any orientation in the same direction that is bigger than 20 pixels. When $L=50$, only the lines of the football field remain in the image, because all of the lines are larger than 50 pixels.

3.1.5 Some Types of Structural Elements

As mentioned before, the structural element can be selected with different sizes and shapes. A disk-shaped structural element with a radius R is a common SE. Applying a disk-shaped opening and closing preserves the objects bigger than the SE while deleting the objects smaller than the SE (Figure 3.4).

Another common structural element is directional one (figure 3.5). “A line with length L and orientation θ is used as a structural element. A pixel is deleted if there exists no line of length L and orientation θ that goes through that pixel” [9]. Opening works on objects that are brighter when compared to their surroundings. Closing works on objects that are darker when compared to their surroundings. Applying different sizes of disk-shaped SEs forms a morphological profile which indicates the minimum dimension of objects, while a morphological profile with a linear SE indicates the minimum length of the objects. In order to remove more details from the objects, the scale of the SE increases.

Whereas openings and closings preserve the objects bigger than the SE, these operations degrade the image. Figure 3.6 shows the effects of openings. Opening operations affects bright objects. While objects smaller than the SE disappear, the image gets some blobby effects and rounding effects. Figure 3.7 shows the effects of closing. Since closing affects objects that are darker than their surroundings, in order to show the effects of closing, the original image was inverted and used as a new original image.

3.2 Reconstruction

Reconstruction methods are used to avoid the degenerative effects of morphological operations. Reconstruction processes work as follows. When morphological operations were applied if at least one pixel of the object is left unchanged, the whole object is reconstructed again according to the original image (mask). Two pixels are considered in the same object if they are connected. Connectivity is decided according



Figure 3.6: Original Lidar image is taken from [6]. An opening applied image. More objects disappear with the increasing size of the SE, and more degradation effects can be observed. While the blobby effect is mostly on the objects that have disappeared, the rounding effect of rectangular shapes can be seen on the remaining objects.

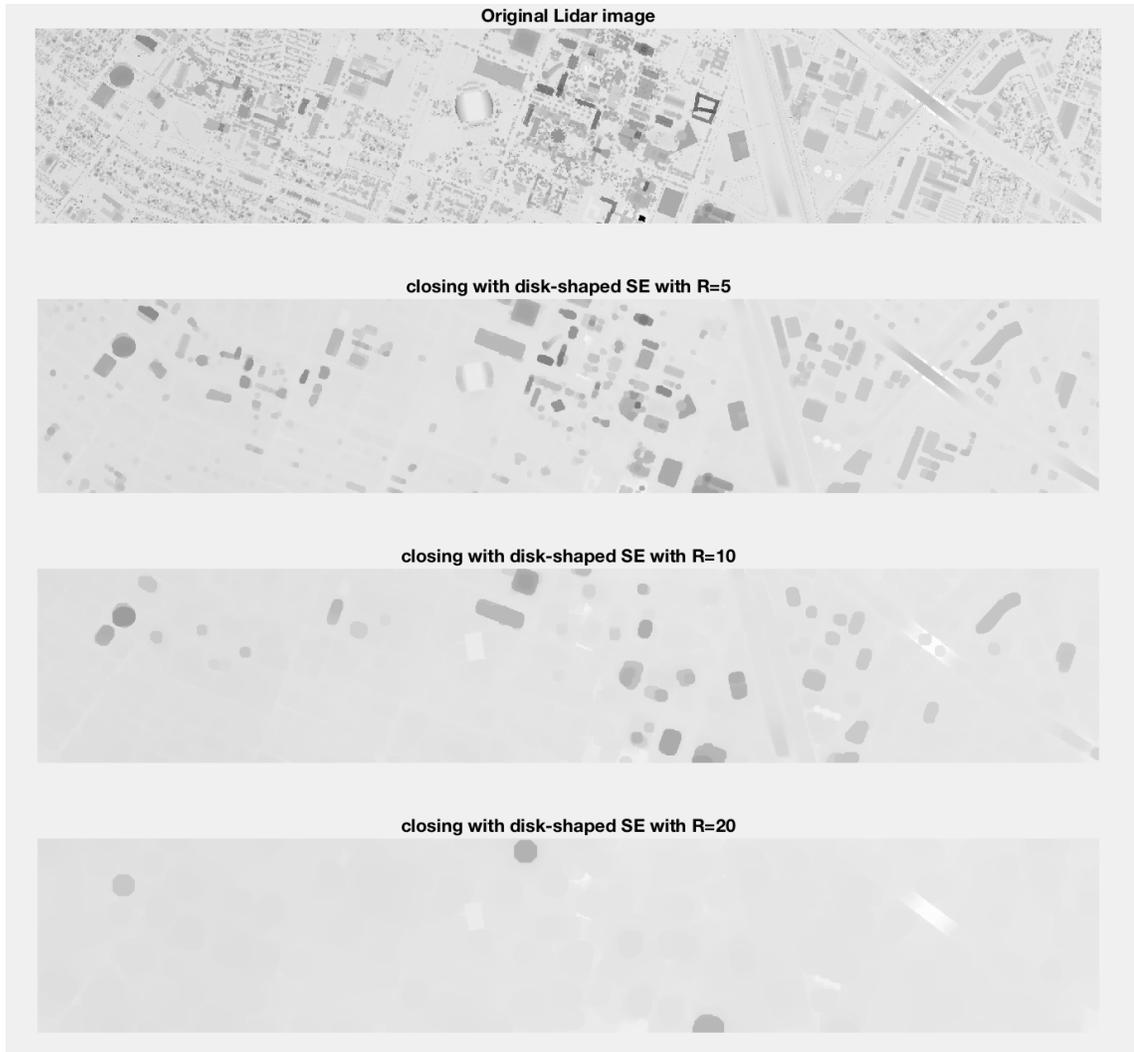


Figure 3.7: A closing works on objects that are darker when compared to their surroundings. Degradation effects are easily observed, as with the opening operator in Figure 3.6.

to a threshold value. If two pixel's intensity value difference is less than a particular threshold, they are considered connected.

Thus, by using reconstruction processes, blobby effects are avoided. However, reconstruction processes may lead to some other undesirable effects. Objects which disappeared without reconstruction can remain present with reconstruction. The reason for this is as follows. For example, in a remote sensing application, objects, such as roads, parking lots and shadows, may appear to be connected to each other even when they really are not. This misperception can lead to reduced classification performance.

3.2.1 Partial Reconstruction

In order to avoid undesirable effects of reconstruction, the method of partial reconstruction was proposed in [9]. In the partial reconstruction process “a pixel is reconstructed if it is connected to another pixel that was not deleted after the opening or closing” [9]. Partial reconstruction provides a second condition for reconstruction of a pixel. The connection of pixels is measured by geodesic distance. If pixel lies in a certain geodesic distance from another pixel, the pixels are considered to be connected. Geodesic distance is the length of the distance between two pixels in an object.

The partial reconstruction process is applied by doing elementary geodesic dilation of size d . This essentially consists of applying elementary geodesic dilation d times. “This is a dilation with an elementary SE followed by an intersection with the mask” [9]. Elementary SE in practice can be chosen eight-neighborhood SE. The intersection of two images is found by taking the minimum grayscale value of the corresponding two pixels. The mask which is used for intersection in the geodesic dilation process is obtained as follows. First, an opening with a disk-shaped SE is applied to the original image, and dilation with disk-shaped SE is applied to the opening applied image. And the intersection of this image and the original image produces the mask which is used in the geodesic dilation.

The d -value defines the amount of reconstruction. It is chosen according to the application. If a low d -value is selected, the corners of objects are not reconstructed

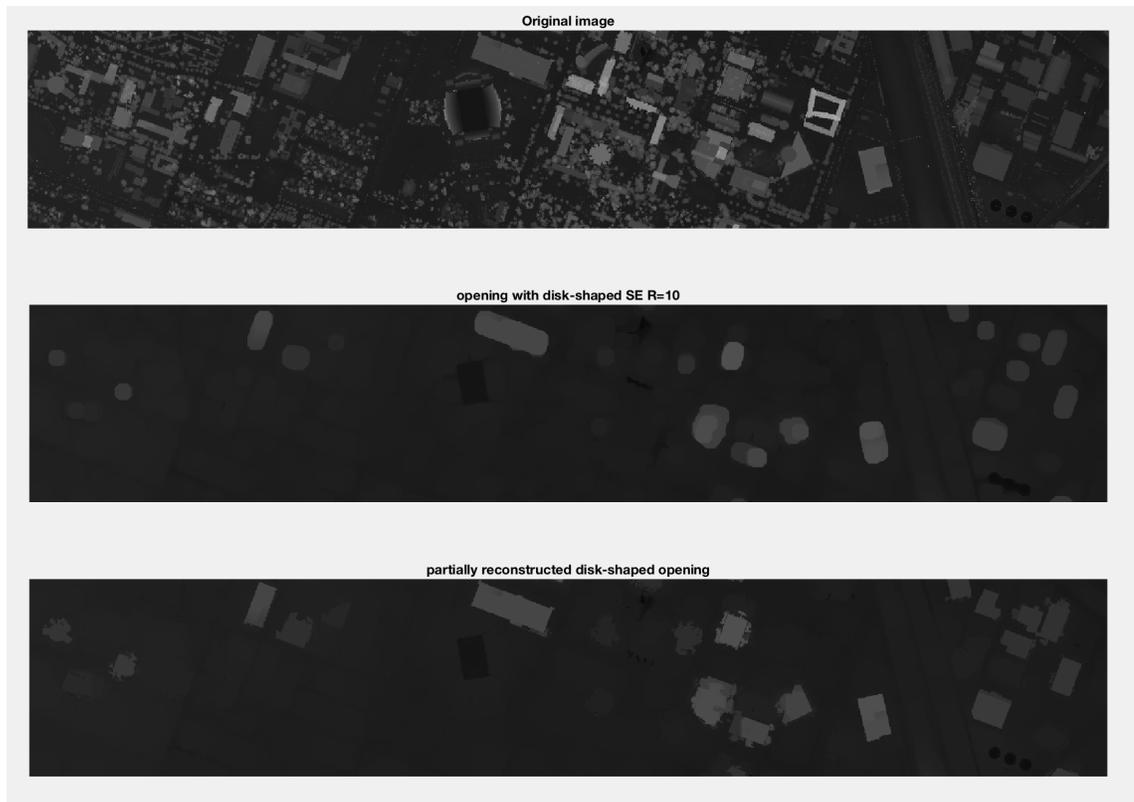


Figure 3.8: Original image, disk-shaped SE applied image and partially reconstructed image.

fully. For higher values of d , connected objects start to affect each other. A detailed explanation of choosing d for different applications is also found in [9].

Examples of openings and closings with and without partial reconstruction are shown in the figures 3.8 and 3.9.



Figure 3.9: Original image, linear SE applied image and partially reconstructed image.

Chapter 4

DIMENSION REDUCTION AND DATA FUSION

When different features from different sensors and preprocesses are used in a classification problem, it is vital that dimension reduction techniques be used to prevent the Hughes Phenomenon [25]. In this chapter, an exploration of dimension reduction and fusion of data using Principal Component Analysis, Kernel Principal Component Analysis, and Locality Preserving Projections is presented.

4.1 Principal Component Analysis

Principal Component Analysis (PCA) for hyperspectral image classification is explained in [26]. PCA uses the fact that hyperspectral images carry similar information in neighboring bands to reduce the dimension of hyperspectral images. This highly correlated information is redundant and can be turned into a linear combination of spectral bands according to the variation of pixel values. A brief formulation of PCA and how to use it to solve the problem of dimension reduction is given below.

A pixel vector in an N -dimensional hyperspectral image with m rows and n columns is shown as:

$$\mathbf{x}_i = [x_1, x_2, x_3, \dots, x_N]_i^T \quad (4.1)$$

The covariance matrix of the data is used in the calculations. In order to find the covariance matrix \mathbf{C}_x , first the data is centered using the following formula:

$$\mathbf{m} = \mathbf{E}(x) = \frac{1}{M} \sum_{i=1}^M [x_1 \ x_2 \ x_3 \ \dots \ x_N]_i^T \quad (4.2)$$

And the covariance matrix is found by:

$$\mathbf{C}_x = \frac{1}{M} \sum_{i=1}^M (\mathbf{x}_i - \mathbf{m}) (\mathbf{x}_i - \mathbf{m})^T \quad (4.3)$$

The covariance matrix is decomposed using the eigenvalue technique and this is then used to find the principal components, which takes the following form:

$$\mathbf{C}_x = \mathbf{A}\mathbf{D}\mathbf{A}^T \quad (4.4)$$

\mathbf{D} is a diagonal matrix, $diag(\lambda_1, \lambda_2, \dots, \lambda_N)$, which consist of eigenvalues of the covariance matrix. \mathbf{A} consist of corresponding eigenvectors for every eigenvalue. Eigenvalues are organized a descending order ($\lambda_1 \geq \lambda_2 \geq \dots, \lambda_N$) and corresponding eigenvectors are also replaced in \mathbf{A} .

$$\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N) \quad (4.5)$$

For the transformation, first k columns of \mathbf{A} is chosen, and the projected matrix is found.

$$\mathbf{y}_i = \mathbf{A}^T \mathbf{x}_i \quad (4.6)$$

Columns of \mathbf{Y} is the principal components of the original hyperspectral image. Such PCA bands have the highest variance in the first band and the lowest variance in the last band.

4.2 Kernel Principal Component Analysis

Kernel Principle Component Analysis (KPCA) for feature reduction is clearly clarified in [27]. While PCA is a linear method of statistical analysis, which uses second-order statistics. KPCA is actually the nonlinear version of PCA which uses higher order statistics, and can provide more information from original data. In order to explain KPCA, first PCA formulations are rewritten in a simpler way. Data \mathbf{x}_i with M observation was formulized in equation 4.1. The centered version of data is represented by as $\mathbf{x} \leftarrow \mathbf{x} - E(\mathbf{x})$. The covariance matrix now is shown as:

$$\mathbf{C}_x = \frac{1}{M} \sum_{i=1}^M \mathbf{x}_i \mathbf{x}_i^T. \quad (4.7)$$

The solution for 4.7 is eigenvalue problem, which is:

$$\begin{aligned} \lambda \mathbf{v} &= \mathbf{C}_x \mathbf{v} \\ \|\mathbf{v}\| &= 1 \end{aligned} \quad (4.8)$$

The transformation matrix is shown simply as:

$$\mathbf{x}_{pc}^k = \mathbf{v}^k \cdot \mathbf{x}. \quad (4.9)$$

In order to start KPCA process, first the data is mapped onto \mathbb{H} space, such that:

$$\begin{aligned} \Phi &: \mathbb{R}^n \rightarrow \mathbb{H} \\ \mathbf{x} &\rightarrow \Phi(\mathbf{x}) \end{aligned} \quad (4.10)$$

Where \mathbb{H} could be infinitely dimensional space with Φ as a nonlinear function. Using the same procedures as before, PCA can be performed in \mathbb{H} . The new covariance matrix can be written in this form:

$$\mathbf{C}_{\Phi(\mathbf{x})} = \frac{1}{M} \sum_{i=1}^M \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_i)^T. \quad (4.11)$$

and this problem can be solved by using eigenvalue equation.

$$\begin{aligned} \lambda \mathbf{v}_{\Phi} = \mathbf{C}_{\Phi(\mathbf{x})} \mathbf{v}_{\Phi} &= \left(\frac{1}{M} \sum_{i=1}^M \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_i)^T \right) \mathbf{v}_{\Phi} \\ &= \frac{1}{M} \sum_{i=1}^M (\Phi(\mathbf{x}_i) \cdot \mathbf{v}_{\Phi}) \Phi(\mathbf{x}_i). \end{aligned} \quad (4.12)$$

Equation 4.12 shows that function \mathbf{v}_{Φ} lies within the span of $\Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2), \dots, \Phi(\mathbf{x}_N)$, therefore each eigenvector can be written in the following way.

$$\mathbf{v}_{\Phi} = \sum_{i=1}^N \alpha_i \Phi(\mathbf{x}_i). \quad (4.13)$$

Now, multiplying 4.12 from left with $\Phi(\mathbf{x}_k)$ and plugging in equation 4.13, we finally get:

$$\lambda \sum_{i=1}^M \alpha_i (\Phi(\mathbf{x}_k) \cdot \Phi(\mathbf{x}_i)) = \frac{1}{M} \sum_{i=1}^M \alpha_i \left(\Phi(\mathbf{x}_k) \cdot \sum_{j=1}^M \Phi(\mathbf{x}_j) \right) \quad (4.14)$$

Gram matrix \mathbf{K} of $M * M$ dimension can be defined as follows:

$$K_{ij} := (\Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}_i)) \quad (4.15)$$

Equation 4.14 takes the form:

$$\lambda \mathbf{K} \boldsymbol{\alpha} = \frac{1}{M} \mathbf{K}^2 \boldsymbol{\alpha} \quad (4.16)$$

Above equation is simply an eigenvalue problem which can be solved for all non-zero eigenvalues as:

$$M \lambda \boldsymbol{\alpha} = \mathbf{K} \boldsymbol{\alpha} \quad (4.17)$$

On the other hand, zero eigenvalue solutions are irrelevant for this problem and it can be shown that these solutions will lead us to the null expansion of 4.13. The solution for $\mathbf{C}_{\Phi(\mathbf{x})}$'s eigenvalue equation is same as solving 4.17. Thus, the dimension reduced projection in \mathbb{H} is achieved by:

$$\Phi(\mathbf{x})_{kpc}^k = \mathbf{v}_{\Phi}^k \cdot \Phi(\mathbf{x}) = \sum_{i=1}^M \alpha_i^k (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x})). \quad (4.18)$$

The computing of PCA in \mathbb{H} space required high computation. In order to resolve this problem, kernel trick can be used, working implicitly in \mathbb{H} space is possible while the computation is done in the input space. The dot product in equation 4.14 can be reduced in feature space using the kernel function.

$$\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) = \mathbf{K}(\mathbf{x}_i, \mathbf{x}_j). \quad (4.19)$$

The most common kernel functions are polynomial and Gaussian kernel functions.

$$\begin{aligned} \mathbf{K}_{poly}(\mathbf{x}_i, \mathbf{x}_j) &= (\mathbf{x}_i \cdot \mathbf{x}_j + r)^d \\ \mathbf{K}_{guass}(\mathbf{x}_i, \mathbf{x}_j) &= \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2). \end{aligned} \quad (4.20)$$

To summarize, the KPCA algorithm works as follows. First kernel matrix \mathbf{K}_{ij} is found, and these \mathbf{K} values are centered. $M \lambda \boldsymbol{\alpha} = \mathbf{K} \boldsymbol{\alpha}$ is solved and then eigenvectors are normalized $\lambda_k (\boldsymbol{\alpha}^k \cdot \boldsymbol{\alpha}^k) = 1$. As a result, first k principal components are found by using 4.18 which is $\Phi(\mathbf{x})_{kpc}^k = \sum_{i=1}^M \alpha_i^k (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}))$.

4.3 Locality Preserving Projections

A linear dimensionality reduction algorithm called Locality Preserving Projections (LPP) was proposed in [28]. The purpose of LPP is finding a transformation matrix which maps the data points to a subspace. This method uses the Laplacian of the graph and aims to preserve local neighborhood information in the transformation.

The general dimension reduction problem is as follows. The goal is to find the transformation matrix \mathbf{A} , which maps the given set $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m \in \mathbb{R}^n$ into set of points $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_m \in \mathbb{R}^l$ ($k \ll l$). The set of points \mathbf{y}_i is found with:

$$\mathbf{y}_i = \mathbf{A}^T \mathbf{x}_i \quad (4.21)$$

The way of finding transformation matrix \mathbf{A} with LPP can be explained in three steps.

1. First, the adjacency graph G with m nodes is constructed. Connectivity of nodes i and j is defined only when x_i and x_j are **close** or not. There are two variations in order to decide whether they are close or not.
 - Epsilon neighborhood. If $\|x_i - x_j\|^2 < \epsilon$, nodes i and j are assumed to be connected by an edge.
 - k nearest neighbors. If node i is in the k nearest neighbor of node j , x_i and x_j are connected by an edge.
2. After establishing the connection of nodes, the weight of the connected edge is considered. If there is no connection, the edge weight is zero. Otherwise, if the nodes are connected, there are two variations of deciding the weights of edges.
 - Heat kernel. Weight of connected nodes i and j is defined as $W_{ij} = \exp -\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{t}$ ($t \in \mathbb{R}$).
 - Simple minded. The weight of edge is 1 if there is a connection.
3. Eigenmaps: \mathbf{W} is the adjacency matrix which is found in the above steps. \mathbf{D} is the diagonal matrix which is $\mathbf{D}_{ii} = \sum_i \mathbf{W}_{ii}$. \mathbf{L} is the Laplacian of graph which is $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Finally, we can compute the eigenvectors and eigenvalues of the generalized eigenvalue problem.

$$\mathbf{X}\mathbf{L}\mathbf{X}^T \mathbf{a} = \lambda \mathbf{X}\mathbf{D}\mathbf{X}^T \mathbf{a} \quad (4.22)$$

The solution for the equation 4.22 is eigenvectors $\mathbf{a}_0, \mathbf{a}_1, \dots, \mathbf{a}_n$ which are put in an ascending order according to corresponding eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n$. So the transformation matrix \mathbf{A} is selected with first l eigenvectors ($l < n$). The dimension reduced projected data matrix y is described as:

$$\mathbf{x} \rightarrow \mathbf{y}_i = \mathbf{A}^T \mathbf{x}_i, \quad \mathbf{A} = (\mathbf{a}_0, \mathbf{a}_1, \dots, \mathbf{a}_{l-1}) \quad (4.23)$$

Justification of the aforementioned algorithm also can be found in [\[28\]](#).

Chapter 5

DATA SETS, PROPOSED METHOD AND EXPERIMENTAL RESULTS

5.1 Data Sets

In this experiment, data is captured using an *IntelRealSense410* camera. This camera uses stereo vision to calculate depth. The camera system consists of two cameras and an infrared projector that enhances depth accuracy in scenes with low texture. The camera can also capture RGB images. A scene from real and synthetic objects was built in the University of Delaware Computational Image Processing Laboratory. Figure 5.1 shows the scene under the white light. In order to obtain the hyperspectral

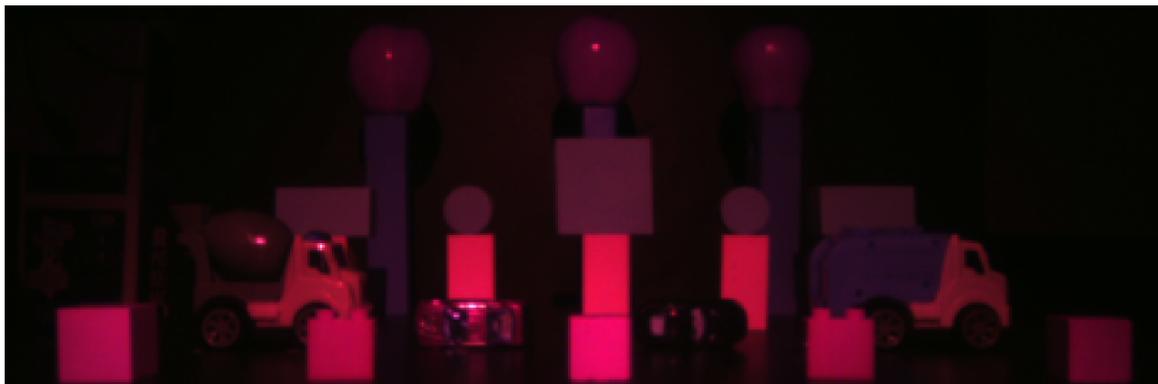


Figure 5.1: Image under the white light.

image, the field of view was fixed. For every image in the hypercube, the light source was filtered with a specific filter. The system setup can be seen in Figure 5.2. The hyperspectral image has 19 bands with wavelength ranges from 467nm to 694nm. Figure 5.3 shows the bands of hyperspectral image. Without changing the field of view, the depth image was also taken, and is depicted in Figure 5.4.

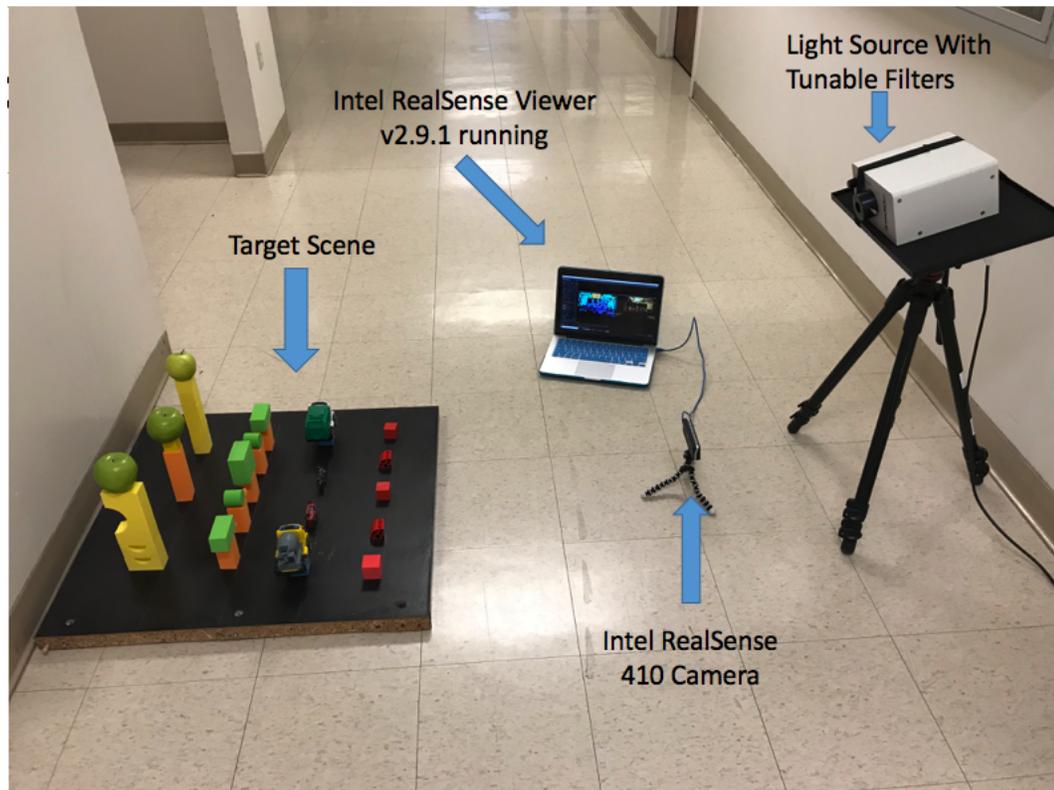


Figure 5.2: Imaging System

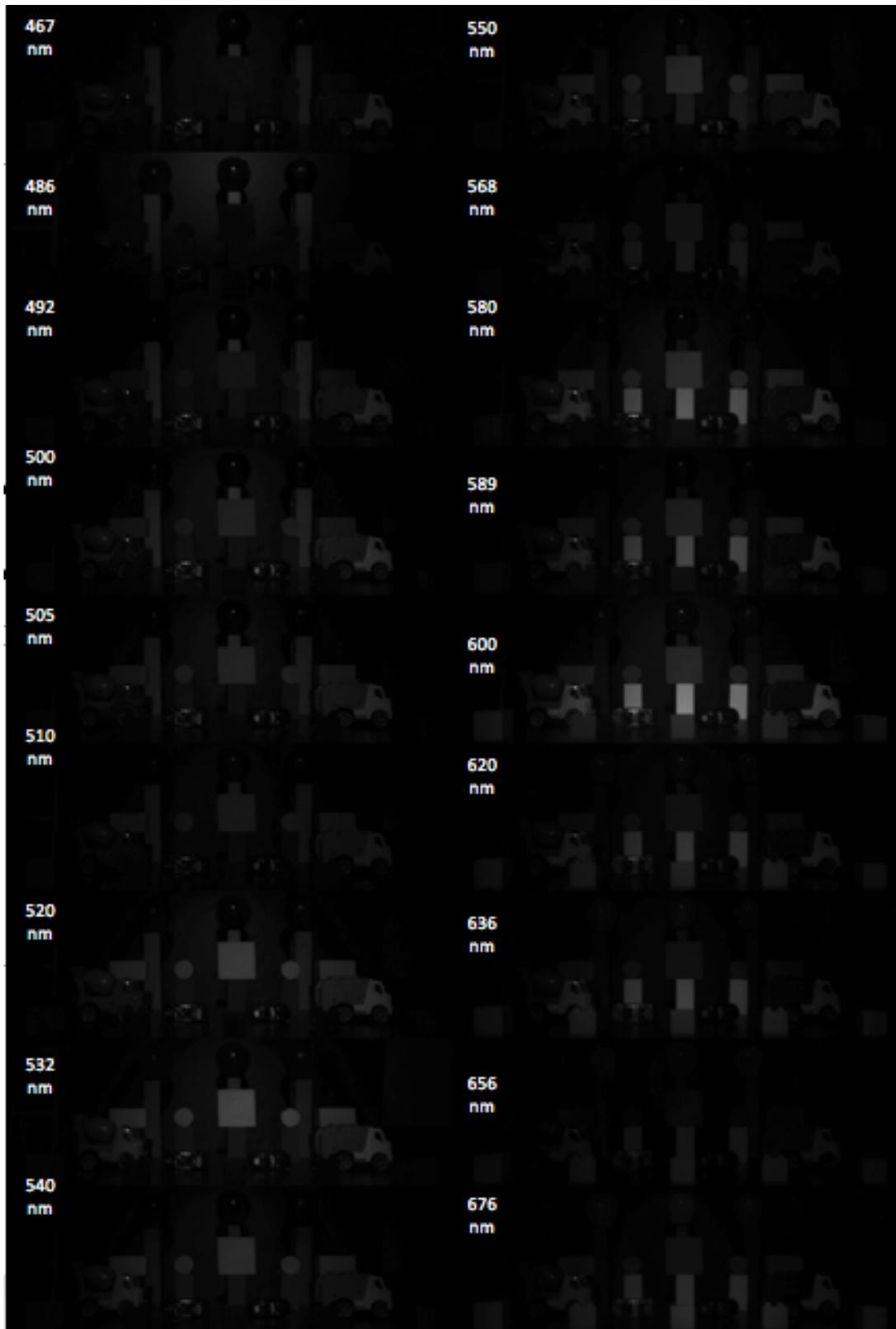


Figure 5.3: 18 bands of hyperspectral image.

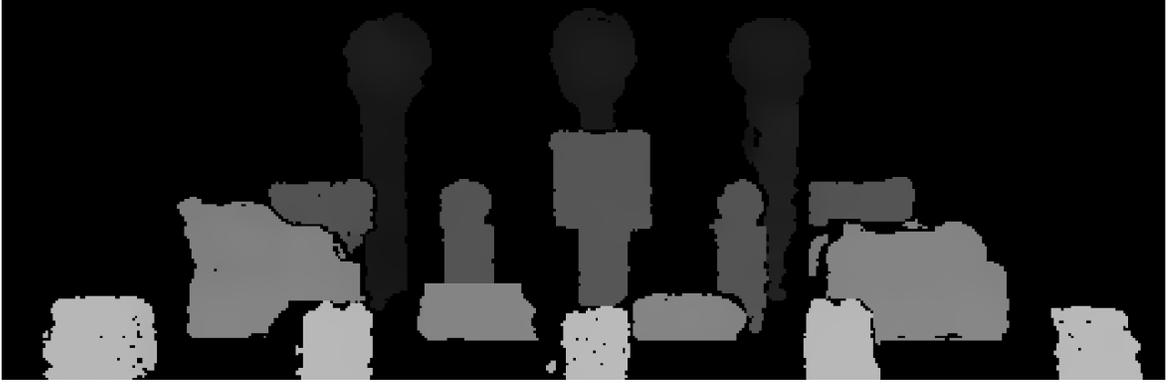


Figure 5.4: Depth image.

5.2 Experiment

The aim of the experiment is classifying the pixels in 14 classes according to depth values, material type, and object shape. This work is similar to the one in [10]. The algorithm followed for the study is broken into the following sections:

5.2.1 Principal Component Analysis

In order to use the morphological information from the hyperspectral data, the first step is applying PCA to the hyperspectral data, after which the first three principal components are selected, which represent 97.4% of the cumulative variance. Fig 5.5 shows the first 3 PCs. The higher number PCs carry less information. The last 16 PCs of the data-cube represents only 2.6% of the cumulative variance.

5.2.2 Morphological Profile

The 3 PCs in the previous section carry the spatial information of the data, while the depth image contains the distance information from pixels to the camera. Morphological openings and closings are applied to these 3 PCs and depth image to construct a morphological profile. For disk-shaped structuring elements, morphological profiles with 10 openings (with partial reconstruction) and 10 closings (with partial reconstruction) are computed for each PC and depth image. The radius of disk-shaped

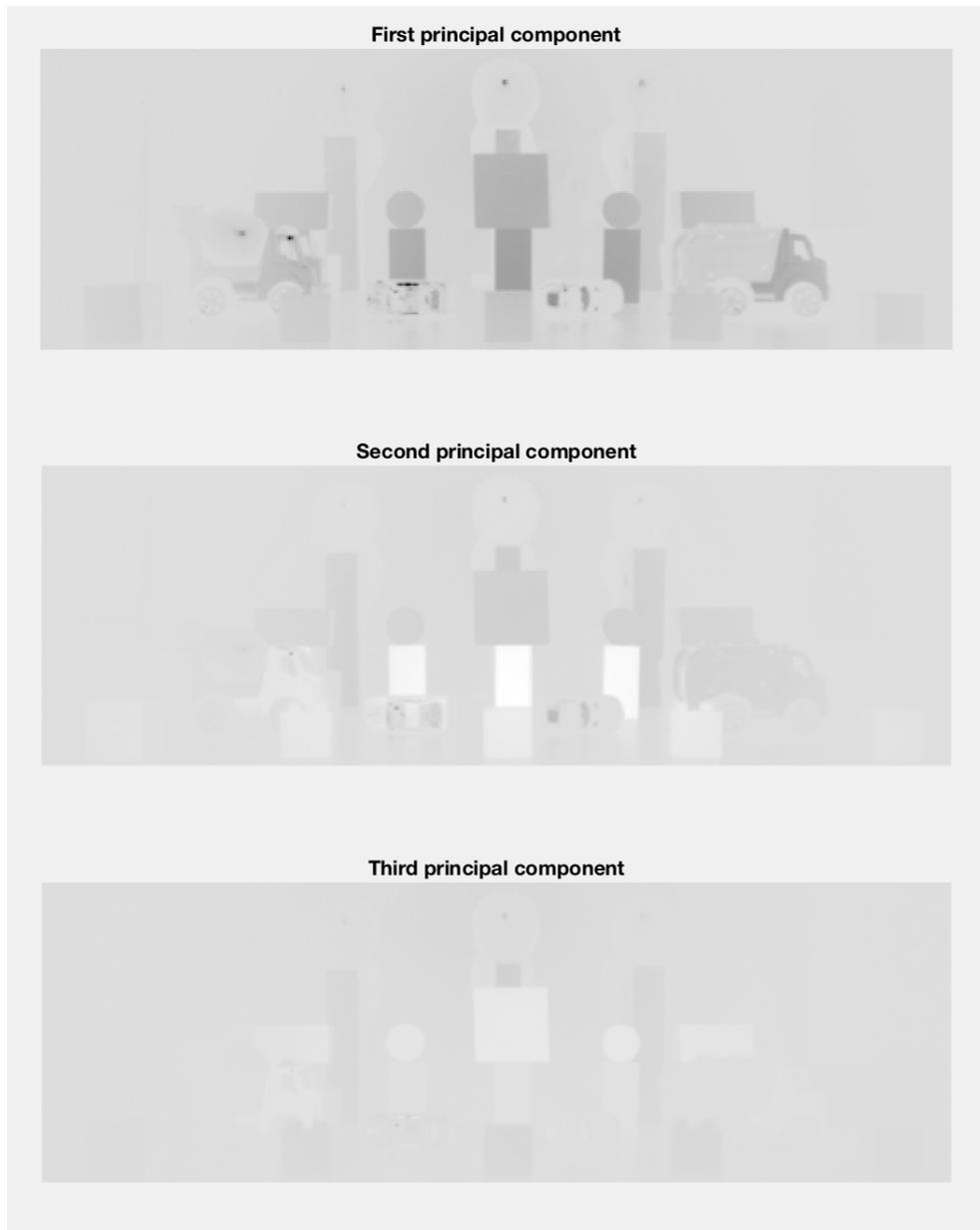


Figure 5.5: First 3 principal component of hyperspectral image

structuring elements is chosen in the range of 2 to 20 pixels with step size increments of 2 pixels. For linear structuring elements, the length of the structuring element is chosen in the range of 10 to 55 pixels with step size increments of 5 pixels. From this, morphological profiles are computed. The final morphological profile, which is based on both disk-shaped and linear structuring elements, is thus produced. The morphological profile of spatial data has 120 bands, 20 from disk-shaped and 20 from linear structuring elements for each PC. Likewise, the morphological profile which is based on depth information contains 40 bands, 20 from disk-shaped and 20 from linear structuring element.

5.2.3 Dimension Normalization with Kernel Principal Component Analysis

Now, the original 19-dimensional hyperspectral, 120-dimensional spatial data, and 40-dimensional depth data are ready to be fused. Before fusing all the data, feature dimension normalization is applied to reduce the computational cost and to allow the use of the graph fusion method in LPP. The KPCA is applied to normalize the features. The lowest dimension of the 3 data-cubes is 19, so 19 PCs are extracted for every data set.

5.2.4 Feature Fusion with Locality Preserving Projections

We define $\mathbf{X}^{Spectral} = \{\mathbf{x}_i^{Spectral}\}_{i=1}^m$, $\mathbf{X}^{Spatial} = \{\mathbf{x}_i^{Spatial}\}_{i=1}^m$ and $\mathbf{X}^{Depth} = \{\mathbf{x}_i^{Depth}\}_{i=1}^m$, where m is the number of pixels and $\mathbf{X}^{Spectral}$, $\mathbf{X}^{Spatial}$ and \mathbf{X}^{Depth} all belongs to Real space of dimension 19. The purpose of fusion is to find the transformation matrix which can transform the matrix $\mathbf{x}^{total} = [\mathbf{X}^{Spectral}; \mathbf{X}^{Spatial}; \mathbf{X}^{Depth}]$ in to a lower dimensional space i.e

$$\mathbf{y}_i = \mathbf{A}^T \mathbf{x}_i.$$

The solution for this equation was explained in Chapter 4. First, adjacency matrices for $\mathbf{X}^{Spectral}$, $\mathbf{X}^{Spatial}$ and \mathbf{X}^{Depth} which are $\mathbf{W}^{Spectral}$, $\mathbf{W}^{Spatial}$ and \mathbf{W}^{Depth} are

found. Finding these adjacency matrices is computationally difficult due to the large size of the matrix $m \times m$, (72850×72850), where m is the total number of pixels. To overcome this hurdle, only 10% of the data is randomly sampled and used in adjacency matrices.

In order to find adjacency matrices, the edges of the graphs where all the pixels are represented by a graph $G = (W, A)$ need to be defined. The edges are chosen according to k nearest neighbors which are 200 in this case. The Weight of the edges is 1 if they are connected and is 0 otherwise. The matrices $\mathbf{W}^{Spectral}$, $\mathbf{W}^{Spatial}$ and \mathbf{W}^{Depth} are all found separately. Since neighborhood information from spatial, spectral, and depth data is required, these adjacency matrices are fused together with element wise multiplication in the following way

$$\mathbf{W}^F = \mathbf{W}^{Spectral} \odot \mathbf{W}^{Spatial} \odot \mathbf{W}^{Depth}.$$

\mathbf{W}^F is found in the above steps. \mathbf{D} is the diagonal matrix which is defined as $D_{ij} = \sum_j W_{ij}$. The Laplacian of the graph is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Finally, the transformation matrix \mathbf{A} can be found by computing the eigenvalues and eigenvectors of the generalized eigenvalue problem.

$$\mathbf{X}\mathbf{L}\mathbf{X}^T \mathbf{a} = \lambda \mathbf{X}\mathbf{D}\mathbf{X}^T \mathbf{a}.$$

The transformation matrix $\mathbf{A} \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_l\}$ consists of l eigenvectors which are put in ascending order according to their corresponding eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_l$ with $l < 3d$, where d is the normalized dimension of spatial, spectral and depth matrices. In this experiment $l = 15$ was chosen.

5.2.5 Classification

After projecting all the features from spatial, spectral and depth data into a linear subspace \mathbf{y}_i^l , (where i can take values from 1 to m (total number of pixels) and l is the dimension of each pixel which is 15 in our case) we are set to classify the pixels. Sample regions were taken from each class, and training and test sets were randomly

selected from these samples. Sample regions were chosen from the center of objects in order to avoid depth image imperfections at the boundaries. Support vector machine is used as a classifier. The Penalty factor, C , is chosen to be 20, and radial basis function (RBF) kernel width, γ , is chosen to be equal to 5. A precision accuracy of more than 97% is achieved for a training set of size 2000 pixels and a test set of a similar size. The result of the classification is shown in Figure 5.6.

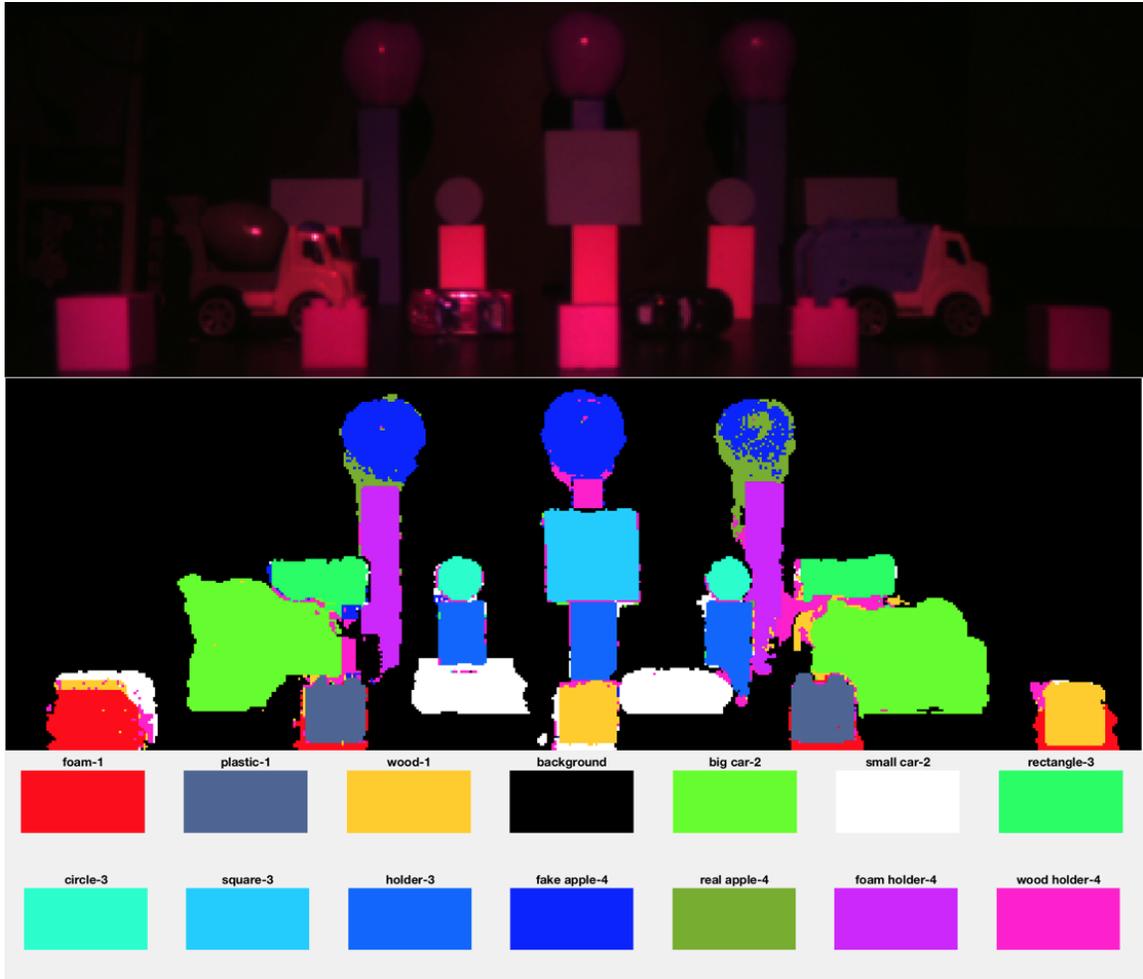


Figure 5.6: Thematic map using the proposed fused features with color code (numbers at the color codes represent the depth level). SVM classifier works well except some regions. These are caused by two reasons. First, the stereo vision system is not precise in the boundaries. Second, pixels such as those in the real apple (upper right) are caused by specular reflection. The surface of the real apple is very much shiny that the camera sees reflection of the light source's light.

Chapter 6

CONCLUSION

In this study, background information about hyperspectral imaging, morphological image processing, and dimension reduction techniques was first presented. Image processing and dimension reduction techniques were then used to extract spectral, spatial, and depth features from original hyperspectral and depth images. After the extraction of spectral, spatial, and depth information, a transformation matrix was found by using the LPP method to fuse the data. Finally, a new data-cube in low-dimensional subspace is obtained and used to classify pixels. Classification results showed that new data cube, which was obtained by feature extraction and fusing techniques, works well with a SVM classifier.

BIBLIOGRAPHY

- [1] <http://www.boeingimages.com/archive/Hyperspectral-Imaging-Technology-Chart-2JRSXLJ8BQW1.html>.
- [2] http://www.cosch.info/documents/14030/52106/Hyvriinen_presentation.pdf/021725c5-a6af-4003-8bb2-4cfd2c70ccc4.
- [3] W. Burger, M. J. Burge, M. J. Burge, and M. J. Burge, *Principles of digital image processing*. Springer, 2009.
- [4] <https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.html>.
- [5] H. Heijmans, “Mathematical morphology: basic principles,” in *Proceedings of Summer School on Morphological Image and Signal Processing*, 1995.
- [6] C. Debes, A. Merentitis, R. Heremans, J. Hahn, N. Frangiadakis, T. van Kasteren, W. Liao, R. Bellens, A. Pižurica, S. Gautama *et al.*, “Hyperspectral and lidar data fusion: Outcome of the 2013 grss data fusion contest,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2405–2418, 2014.
- [7] U. Heiden, W. Heldens, S. Roessner, K. Segl, T. Esch, and A. Mueller, “Urban structure type characterization using hyperspectral remote sensing and height information,” *Landscape and Urban Planning*, vol. 105, no. 4, pp. 361–375, 2012.
- [8] M. Dalponte, L. Bruzzone, and D. Gianelle, “Fusion of hyperspectral and lidar remote sensing data for classification of complex forest areas,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 5, pp. 1416–1427, 2008.
- [9] R. Bellens, S. Gautama, L. Martinez-Fonte, W. Philips, J. C.-W. Chan, and F. Canters, “Improved classification of vhr images of urban areas using directional morphological profiles,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 10, pp. 2803–2813, 2008.
- [10] W. Liao, R. Bellens, A. Pizurica, S. Gautama, and W. Philips, “Graph-based feature fusion of hyperspectral and lidar remote sensing data using morphological features,” in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, vol. 7, 2013, pp. 4942–4945.

- [11] C. Debes, A. Merentitis, R. Heremans, J. Hahn, N. Frangiadakis, and T. van Kasteren, “A two-stream fusion framework for lidar and hyperspectral imagery.”
- [12] N. Yokoya, S. Nakazawa, T. Matsuki, and A. Iwasaki, “Fusion of hyperspectral and lidar data for landscape visual quality assessment,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2419–2425, 2014.
- [13] J. M. Wozencraft, M. Lee, G. H. Tuell, and W. D. Philpot, “Fusion of hyperspectral and bathymetric laser data in kaneohe bay, hawaii,” in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery IX*, vol. 5093. International Society for Optics and Photonics, 2003, pp. 517–528.
- [14] S. M. Hsu and H.-h. Burke, “Multisensor fusion with hyperspectral imaging data: detection and classification,” in *Handbook Of Pattern Recognition And Computer Vision*. World Scientific, 2005, pp. 347–364.
- [15] S. M. Hsu, J. P. Kerekes, H.-H. Berke, and S. Crooks, “Sar and hsi data fusion for counter cc&d,” in *Radar Conference, 1999. The Record of the 1999 IEEE*. IEEE, 1999, pp. 218–220.
- [16] S. Jun and Z. Tianxu, *Multispectral image processing and pattern recognition*. World Scientific, 2001, vol. 44.
- [17] A. F. Goetz, G. Vane, J. E. Solomon, and B. N. Rock, “Imaging spectrometry for earth remote sensing,” *Science*, vol. 228, no. 4704, pp. 1147–1153, 1985.
- [18] T. Adão, J. Hruška, L. Pádua, J. Bessa, E. Peres, R. Morais, and J. J. Sousa, “Hyperspectral imaging: A review on uav-based sensors, data processing and applications for agriculture and forestry,” *Remote Sensing*, vol. 9, no. 11, p. 1110, 2017.
- [19] A. Gowen, C. O’Donnell, P. Cullen, G. Downey, and J. Frias, “Hyperspectral imaging—an emerging process analytical tool for food quality and safety control,” *Trends in Food Science & Technology*, vol. 18, no. 12, pp. 590–598, 2007.
- [20] E. K. Hege, D. O’Connell, W. Johnson, S. Basty, and E. L. Dereniak, “Hyperspectral imaging for astronomy and space surveillance,” in *Imaging Spectrometry IX*, vol. 5159. International Society for Optics and Photonics, 2004, pp. 380–392.
- [21] C. Balas, G. Epitropou, A. Tsapras, and N. Hadjinicolaou, “Hyperspectral imaging and spectral classification for pigment identification and mapping in paintings by el greco and his workshop,” *Multimedia Tools and Applications*, pp. 1–15, 2018.
- [22] H. Grahn and P. Geladi, *Techniques and applications of hyperspectral image analysis*. John Wiley & Sons, 2007.

- [23] D. Manolakis and G. Shaw, “Detection algorithms for hyperspectral imaging applications,” *IEEE signal processing magazine*, vol. 19, no. 1, pp. 29–43, 2002.
- [24] J. Serra, *Image analysis and mathematical morphology*. Academic Press, Inc., 1983.
- [25] G. Hughes, “On the mean accuracy of statistical pattern recognizers,” *IEEE transactions on information theory*, vol. 14, no. 1, pp. 55–63, 1968.
- [26] C. Rodarmel and J. Shan, “Principal component analysis for hyperspectral image classification,” *Surveying and Land Information Science*, vol. 62, no. 2, p. 115, 2002.
- [27] M. Fauvel, J. Chanussot, and J. A. Benediktsson, “Kernel principal component analysis for feature reduction in hyperspectral images analysis,” in *Signal Processing Symposium, 2006. NORSIG 2006. Proceedings of the 7th Nordic*. IEEE, 2006, pp. 238–241.
- [28] X. He and P. Niyogi, “Locality preserving projections,” in *Advances in neural information processing systems*, 2004, pp. 153–160.