Understanding High Resolution Aerial Imagery Using Computer Vision Techniques

by

Fan Wang

B.S. Harbin Institute of Technology, 2010

M.S. Harbin Institute of Technology, 2012

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Chester F. Carlson Center for Imaging Science
College of Science
Rochester Institute of Technology

August, 2017

Signature of the Author ______________________________

Accepted by ______________________________
Coordinator, Ph.D. Degree Program   Date

CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE
COLLEGE OF SCIENCE
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

Ph.D. DEGREE DISSERTATION

The Ph.D. Degree Dissertation of Fan Wang
has been examined and approved by the
dissertation committee as satisfactory for the
dissertation required for the
Ph.D. degree in Imaging Science

____
Dr. John P. Kerekes, Dissertation Advisor

____
Dr. Pengcheng Shi

____
Dr. Carl Salvaggio

____
Dr. Yandong Wang

____
Date

Understanding High Resolution Aerial Imagery Using Computer Vision Techniques

by

Fan Wang

Submitted to the
Chester F. Carlson Center for Imaging Science
in partial fulfillment of the requirements
for the Doctor of Philosophy Degree
at the Rochester Institute of Technology

Abstract

Computer vision can make important contributions to the analysis of remote sensing satellite or aerial imagery. However, the resolution of early satellite imagery was not sufficient to provide useful spatial features. The situation is changing with the advent of very-high-spatial-resolution (VHR) imaging sensors. This change makes it possible to use computer vision techniques to perform analysis of man-made structures. Meanwhile, the development of multi-view imaging techniques allows the generation of accurate point clouds as ancillary knowledge.

This dissertation aims at developing computer vision and machine learning algorithms for high resolution aerial imagery analysis in the context of application problems including debris detection, building detection and roof condition assessment. High resolution aerial imagery and point clouds were provided by Pictometry International for this study.

Debris detection after natural disasters such as tornadoes, hurricanes or tsunamis, is needed for effective debris removal and allocation of limited resources. Significant advances in aerial image acquisition have greatly enabled the possibilities for rapid and automated detection of debris. In this dissertation, a robust debris detection algorithm is proposed. Large scale aerial images are partitioned into homogeneous regions by interactive segmentation. Debris areas are identified based on extracted texture features.

Robust building detection is another important part of high resolution aerial imagery understanding. This dissertation develops a 3D scene classification algorithm for building

detection using point clouds derived from multi-view imagery. Point clouds are divided into point clusters using Euclidean clustering. Individual point clusters are identified based on extracted spectral and 3D structural features.

The inspection of roof condition is an important step in damage claim processing in the insurance industry. Automated roof condition assessment from remotely sensed images is proposed in this dissertation. Initially, texture classification and a bag-of-words model were applied to assess the roof condition using features derived from the whole rooftop. However, considering the complexity of residential rooftop, a more sophisticated method is proposed to divide the task into two stages: 1) roof segmentation, followed by 2) classification of segmented roof regions. Deep learning techniques are investigated for both segmentation and classification. A deep learned feature is proposed and applied in a region merging segmentation algorithm. A fine-tuned deep network is adopted for roof segment classification and found to achieve higher accuracy than traditional methods using hand-crafted features.

Contributions of this study include the development of algorithms for debris detection using 2D images and building detection using 3D point clouds. For roof condition assessment, the solutions to this problem are explored in two directions: features derived from the whole rooftop and features extracted from each roof segments. Through our research, roof segmentation followed by segments classification was found to be a more promising method and the workflow processing developed and tested. Deep learning techniques are also investigated for both roof segmentation and segments classification. More unsupervised feature extraction techniques using deep learning can be explored in future work.

Acknowledgements

First of all, I would like to express my sincere gratitude to my advisor, Dr. John Kerekes, for the help, support and guidance of my research. You are a great advisor in balancing guidance and giving me the freedom to various projects of my interest. I cannot reach this point without your mentoring.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Pengcheng Shi, Dr. Carl Salvaggio and Dr. Yandong Wang, for taking time to provide insightful comments and valuable advice.

In addition, there are several individuals I would like to express my appreciation. First, besides Dr. Yandong Wang, I would like to thank Frank Giuffrida, Stephen Schultz and Charles Mondello from Pictometry International Corporation for their guidance and discussion in completing these projects. Also I would like to thank Dr. Harvey Rhody and Michael Richardson for their help and useful suggestions for my projects. A special thank to Sue Chan, Cindy Schultz and Beth Lockwood for their help and keeping things well organized. I also wish to thank my friends: Bo Ding, Ming Li, Can Jin, Chao Zhang, Gajendra J. Katuwal, Zhaoyu Cui, Xuewen Zhang, Zhenlin Xu and Runchen Zhao for their help and companion. Last but not least, I want to devote my gratitude to my girlfriend Zhuoyi Xu and my family. Thank you for your unconditional support and encouragement in my life and study.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

High resolution aerial imagery contains plenty of information which is not able to be directly understood by a computer while it can be easily understood by the human vision system. Since computer vision aims to model, duplicate and exceed the abilities of human vision system through electrical hardware and computational models [2], it can make important contributions to the analysis of remote sensing or aerial imagery [3].

However, the resolution of early satellite imagery was not sufficient to provide useful spatial features [4]. For example, the spatial resolution of Landsat 7 (panchromatic band) is 15 m. Its pixel size is bigger than the size of many man-made or natural objects. Thus, pixel based or even sub-pixel based methods are the main trend of image analysis techniques in traditional remote sensing [5].

The situation is changing with the advent of very-high-resolution (VHR) imaging sensors [3]. The change makes it possible to use computer vision techniques to analyze of man-made structures. Meanwhile, the development of multi-view imaging techniques allows the generation of accurate point clouds as ancillary knowledge for analysis [3].

This dissertation aims at developing computer vision and machine learning algorithms for high resolution aerial imagery analysis in the context of application problems including debris detection, building detection and roof condition assessment. High resolution aerial imagery and point clouds were provided by Pictometry International for this study.

Debris detection after natural disasters such as tornadoes, hurricanes or tsunamis, is needed for effective debris removal and allocation of limited resources. It can be performed manually, but such effort is labor intensive and hinders the quick response needed in large hurricane impact zones. Significant advances in aerial image acquisition have greatly promoted the possibilities for rapid and automated detection of debris. In this dissertation, a robust debris detection algorithm using texture features of debris areas is proposed.

Robust building detection is an important part of high resolution aerial imagery understanding [6]. It also serves as pre-processing for building modeling and roof condition assessment. This dissertation develops a 3D scene classification algorithm for building detection using point clouds derived from multi-view imagery.

The inspection of roof condition is an important step of damage claim processing in the insurance industry. Currently, roof inspections are done by humans and are an expensive, time-consuming and unsafe process. Thus, automated roof condition assessment from remotely sensed images is proposed in this dissertation. Initially, texture classification and bag-of-words models are developed to assess the roof condition using features covering the whole rooftop. However, considering the complexity of residential rooftops, a more sophisticated method is proposed to divide the task into two stages: 1) roof segmentation, followed by 2) classification of the segmented roof regions. Deep learning techniques are investigated for both segmentation and classification. A deep learned feature is proposed and applied in the region merging segmentation algorithm. A fine-tuned deep network is adopted for roof segments classification and compared to traditional method using hand-crafted features.

The rest of this dissertation is organized as follows. Chapter 2 briefly states the objectives of this research, and relevant background is reviewed in Chapter 3. The approach and results of debris detection, building detection and roof condition assessment are stated in Chapters 4, 5 and 6, separately. Finally, a summary and recommendations for future work are presented in Chapter 7.

Chapter 2

Objectives

The purpose of this research is to develop computer vision and machine learning algorithms for understanding high resolution aerial imagery. More specifically, the objectives can be divided into three parts based on the applications:

- To investigate methods for debris detection using very-high-resolution (VHR) aerial images provided by Pictometry;
- To develop algorithms for 3D point cloud classification. The point clouds used are derived from multiple high resolution aerial images. The ultimate goal is to divide the scene into three categories: vegetative areas, terrain and building footprints;
- To investigate automated roof condition assessment techniques using VHR aerial imagery. The assessment includes preliminary roof condition assessment, roof segmentation and roof segments classification.

This study makes contributions to the use of computer vision and machine learning techniques to understand high resolution aerial imagery in the context of debris detection, building detection and roof condition assessment. An algorithm for debris detection is proposed and tested on large scale data. Building detection algorithms are developed using 3D structure features. The possibility is explored to assess the roof condition using features covered the whole rooftop image or features inside each roof segments. Algorithms for preliminary roof condition assessment, roof segmentation and roof segments classification are proposed and developed. Deep learning techniques are investigated for roof condition assessment. As a fixed feature extractor, a deep learned feature is proposed for region representation in the roof segmentation. For roof segments classification, both traditional method using hand-crafted features and deep learning method are investigated. Fine-tuning a pre-trained deep network is used for roof segments classification and achieves better result than traditional method.

Chapter 3

Background

In this chapter, the background of our research is reviewed. Section 3.1 reviews the development of high resolution remotely sensed imagery. A brief history of computer vision is stated. Computer vision and deep learning applied in remote sensing are described in the following sections.

3.1 High Resolution Remotely Sensed Imagery

The first moderate-resolution civilian earth observation satellite, Landsat 1, was launched in July 1972 as a new way of monitoring land cover and land use globally [7]. It carried the Multispectral Scanner System (MSS) with a spatial resolution of 80 m. The spatial resolution number specifies the ground distance covered by one pixel in the image.

In 1999, Landsat 7 was successfully launched which contained the Enhanced Thematic Mapper Plus (ETM+) with a 15 m panchromatic band and 30 m multispectral bands. Each pixel in a Landsat 7 image may represent an area as large as a house. The resolution is too coarse to be useful for analysis of a particular man made object of interest. Thus, image analysis techniques for Landsat 7 imagery are confined to pixel-based analysis or even sub-pixel analysis and the applications are limited to land use and land cover classification.

The spatial resolution of these early satellite images were not sufficient to identify useful spatial features. This situation is changing with the increasing spatial resolution of new generation sensors such as SPOT 5, launched in 2002, which provided a panchromatic band with spatial resolution as small as 2.5 m and 10 m multispectral images. Very-High-resolution (VHR) satellite imagery even offers sub-meter resolution from commercial remote sensing satellites. GeoEye's IKONOS is the world's first sub-meter commercial remote sensing satellite. It was launched in September 1999 with panchromatic data at 0.82 m at nadir. The multispectral imagery are collected at 3.2 m at nadir. DigitalGlobe's QuickBird was launched in October of 2001 collecting 0.61m panchromatic band and 2.44m multispectral imagery. With the dramatic increase of satellite image resolution, a number of new applications could be tackled by remote sensing such as road detection and building detection [8].

In the last ten years we have seen spectacular developments as the very high resolution images taken from optical satellites reached a spatial resolution down to half a meter. WorldView-1 was launched in September 2007 and is owned by DigitalGlobe. It provides a panchromatic image with 0.5 m resolution at nadir. GeoEye-1 was launched in September 2008. It provides a panchromatic band with 0.41m resolution at nadir and four multispectral bands with a 1.65 m resolution at nadir. WorldView-2 was launched in October 2009. The panchromatic imaging system has a 0.46 m resolution at nadir. The multispectral imaging system has a 1.85 m resolution at nadir. With the resolution reaching down to half a meter, we see new application fields including security applications, vehicle detection and many urban applications.

Currently, the world's highest resolution commercial earth imaging satellite is WorldView-3 which was launched in August 2014 with ability to capture panchromatic imagery at 0.31m resolution, 1.24m multispectral and shortwave infrared (SWIR) imagery. Some satellites are able to produce higher resolution images. However, the Federal law restricts the resolution of commercially available satellite images.

Aerial photography is another widely used method of capturing remotely sensed images of the planet. Aerial imagery is gathered through specialized cameras or sensors mounted on platforms flying between 200 and 15000 m [9]. The sensors can be multispectral, hyperspectral, thermal, lidar or other survey sensors. Most aerial cameras offer a fourth near-infra-red band of imagery as well as standard R,G,B bands. Camera and platform configurations can be grouped in terms of oblique and vertical.

The first air photo was taken from a balloon by a Parisian photographer in 1858. With the advent of new generation commercial digital aerial cameras, an aerial platform offers superior image quality over even very-high-resolution satellite imagery. Meanwhile, aerial images are not subject to the resolution limits imposed on satellites. They are available at resolutions down to 10 to 15 cm per pixel for most of the populated areas in the US.

Furthermore, Pictometry International can capture a image with resolution down to 2.5cm today. These superior resolution air photos provide a large amount of detail information and results in a wide range of value-added products such as 3D modeling and automated information extraction.

Nowadays, there is a rise in use of Unmanned Aircraft Systems, or drones, with super-high-resolution for low cost. Military drones can achieve super-high-resolution of under 10 centimeters. Drones for civilian use are not far behind. Thus, from satellites to drones, there is an increasing availability of imagery with higher resolution at lower cost.

A comparison of commercial satellite and aerial images is shown in Table. 3.1. The change of spatial resolution can be seen visually in Fig. 3.1.

Table 3.1: Comparison of Spatial Resolution of Commercial Satellite and Aerial Images

Sensor Name	Sensor Type	Spectral Bands	Sensor Resolution
Pictometry Data	Airplane	RGB	0.025 m
WorldView-3	Satellite	Pan	0.31 m
		8-band MS	1.24 m
		8-band SWIR	1.24 m
GeoEye-1	Satellite	Pan	0.41 m
		4-band MS	1.65 m
WorldView-2	Satellite	Pan	0.46 m
		8-band MS	1.85 m
WorldView-1	Satellite	Pan	0.50 m
QuickBird	Satellite	Pan	0.61 m
		4-band MS	2.44 m
IKONOS	Satellite	Pan	0.82 m
		4-band MS	3.20 m
SPOT5	Satellite	Pan	2.5 m or 5.0 m
		3-band MS	10.0 m
		SWIR	20.0 m

3.2 A 'Brief' History of Computer Vision

Computer vision is a discipline that aims to analyze and interpret scenes of the real world [18]. Its ultimate goal is to model, duplicate and exceed the abilities of human vision system through electrical hardware and computational models [2]. Computer vision overlaps significantly with image processing and pattern recognition, and typical tasks of computer vision include object recognition, motion analysis and 3D scene reconstruction.

In 1960, for the first time, Larry Robert's Ph.D. thesis discussed the possibility of extracting 3D geometrical information from 2D perspective views [19]. Researchers started to follow this work and studied computer vision. In 1966, Marvin Minsky at MIT assigned computer vision as an undergrad summer project. Later, much research was done in "low-level" image processing such as edge detection and segmentation.

The most widely cited article in computer vision history is the Scale Invariant Feature Transform (SIFT) paper [20]. It was introduced by David Lowe in 1999 to provide local feature descriptors [21]. SIFT was widespread in the geometric field of computer vision and it served later as the basis of the Bag of Words (BoW) model for object recognition.

In 2003, Caltech-101 as a dataset of images was created by Feifei Li. It opened the door to an era of large-scale datasets. Researchers started to run and evaluate their algorithms on these publicly available datasets. It offered researchers an objective and standard way to compare their algorithms with state-of-the-art methods.

Visual words were also introduced in 2003. It was an algorithm from text recognition and has been applied to object recognition. Visual words are a fairly robust way to represent the content of an image and is still heavily utilized. Another factor for rise of computer vision was the widespread use of the support vector machine (SVM) as a robust, accurate and easy to use learning algorithm.

In 2005, Histogram of Oriented Gradients (HOG) was introduced to solve the problem of pedestrian detection [22]. In 2006, spatial pyramid was applied as an extension of bag-of-features image representation [23]. In 2008, the strength of the HOG was reinforced by Deformable Part Models [24]. At this stage, researchers poured a lot of effort into the design of better features.

With the development of the Internet, researchers could acquire a cheap and limitless source of images. Labeled datasets with millions of images became available. For exmaple, ImageNet consists over 15 million labeled high-resolution images in over 22000 categories [25].

Neural networks are a very old technique but they were overshadowed by SVM for a long time. However, with the significant increase of GPU computing capability, Big Data started to unlock the capabilities of neural networks. They began to rival SVM. Convolutional neural network have produced stellar results and their capabilities are illustrated in the ImageNet Large Scale Visual Recognition Challenge. The performance is now close to humans.

3.3 Computer Vision in Remote Sensing

Since the remote sensing and computer vision communities share a common goal of extracting useful information from raw imagery [26], computer vision can make important contribution to the analysis of remote sensing imagery.

However, the resolution of early satellites was too coarse. Thus, image analysis techniques for those imagery are confined to pixel-based analysis or even sub-pixel analysis and the applications are limited to land use and land cover classification using spectral information. Then, computer vision techniques had not much importance except in low-level processing such as image filtering, contrast enhancement, edge detection and region segmentation [3].

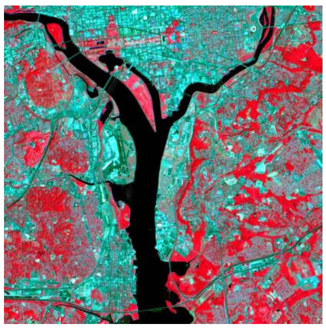
With the emergence of imaging sensors with very high spatial resolution, the rapid development of computer vision had enormous impact on processing and interpretation of remotely sensed data. A number of application areas have evolved linking the two fields, such as road detection, building modeling and vehicle tracking. Recognition of limitations with pixel-based image approaches [8], there was a hype in applications beyond pixels [5]. A segmentation algorithm is used to divide the image into homogeneous regions which could be recognized by shape, texture and context information extraction by techniques from computer vision.

Today, more researchers are adopting computer vision methods to remote sensing and working at the forefront of combining knowledge from both fields. The gap between these two fields is getting smaller as remote sensing has benefited from great improvements in computer vision.

3.4 Deep Learning in Remote Sensing

Recently, deep learning techniques have demonstrated excellent performance on various tasks and have drawn increased attention from remote sensing community.

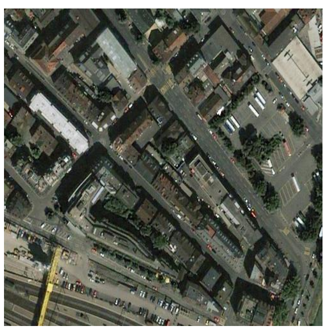
Convolutional neural network (CNN) was employed to classify hyperspectral data using spectral information and experimental results demonstrate that the method achieved better performance than some traditional methods, such as support vector machines (SVM) [27]. An saliency-guided unsupervised feature learning framework using deep network was proposed in [28] for scene classification. In [29], Romero et al. proposed a greedy layerwise unsupervised sparse features using CNN for pixel classification. A systematically review of the state-of-the-art deep learning-based methods in remote sensing image analysis was made by Zhang et al. in 2016 [30].



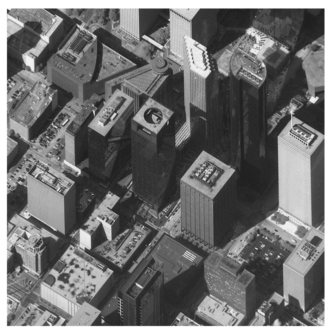
(a) Landsat 7 [10]

(b) SPOT 5 [11]

(c) IKONOS [12]



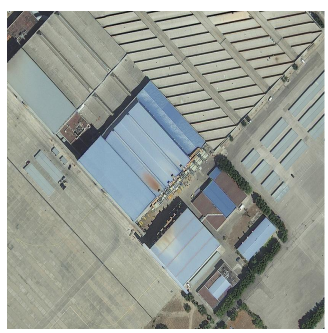
(d) QuickBird [13]



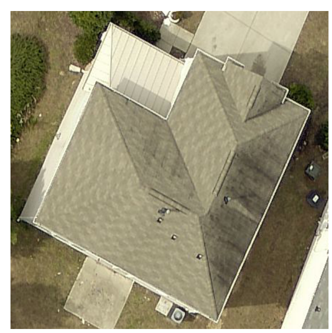
(e) WorldView-1 [14]

(f) WorldView-2 [15]

(g) GeoEye-1 [16]



(h) WorldView-3 [17]



(i) Pictometry Image

Figure 3.1: Comparison of commercial satellite and aerial images.

Chapter 4

Debris Detection

Debris detection after natural disasters such as tornadoes, hurricanes or tsunamis, is needed for effective debris removal and allocation of limited resources. It can be performed manually, but such effort is labor intensive and hinders the quick response needed in large hurricane impact zones. Significant advances in aerial image acquisition have greatly promoted the possibilities for rapid and automated detection of debris.

In this chapter, a robust debris detection algorithm using the texture feature of debris area is proposed. Section 4.1 reviews the previous research for debris detection. Section 4.2 states the method proposed. Results are provided in section 4.3 and conclusions are drawn in section 4.4.

4.1 Previous Research

In this dissertation, debris detection means delineation of debris areas in the aerial imagery. Debris piles contain concrete, asphalt and wood material from damaged roof caused by hurricanes or tornados.

Debris detection is part of disaster assessment. In the literature, many methods have been proposed to address this problem.

Pixel-based methods have been proposed for debris detection. For example, Zoltan Szantoi et al. developed an algorithm to locate downed tree debris using Leica Airborne Digital Sensor(ADS40) data[31]. A sobel edge detection algorithm was combined with spectral information based on color filtering.

With the increase in data resolution, object based methods became a popular choice for debris detection. In 2008, Fumio Yamazaki et al. used imagery acquired before and after the 2007 Off-Mid-Niigata earthquake to detect building damage [32]. Supervised object based classification was performed to extract the debris of collapsed buildings. In 2011, Takumi Fukuoka et al. conducted an analysis to estimate the distribution and the amount of tsunami debris [33]. A supervised classification was performed on the post-tsunami aerial photos. In 2011, Shunichi Koshimura et al. developed an object-based method for tsunami impact mapping using QuickBird data [34]. Ground objects are classified into six classes: vegetation, water, soil, building, road and debris. The debris was labeled with a larger standard deviation than other objects of NIR band. Threshold $\sigma > 55.0$ was applied.

In some literature, pixel and object based methods are compared. In 2007, Jae Sung Kim et al. implemented a hurricane damage assessment using before and after Katrina image data [35]. Landsat 7, Quickbird and IKONOS satellite imagery were first classified using both pixel and object based approaches. Change detection was then performed to compare each class before and after Katrina to identify and quantify the damaged area. In 2008, Myint et al. compared pixel based and object oriented classification approaches on identification of tornado damaged areas [36]. Accuracy assessment revealed that the object based image analysis(OBIA) approach outperforms pixel-by-pixel analysis in damage detection.

With the increase in data resolution, pixel-based analysis has been replaced by object based method. Features extracted from object level are considered for better performance. Lots of research have been done about debris mapping. However, these techniques have never been tested on very high resolution Pictometry data. The aim of our research is to develop a robust debris detection for the one-inch-resolution data.

4.2 Approach

Traditional remote sensing techniques like change detection, anomaly detection and supervised classification were first tested. A Fast Fourier transform (FFT) based debris detection algorithm was proposed. However, these methods did not provide promising results.

Since the pixel-by-pixel analysis did not work, OBIA framework was chosen for debris detection on Pictometry one-inch-resolution data. The image is first grouped into homogeneous objects. Texture and spatial features are extracted from each object. A debris detection algorithm using texture variance is proposed. Meanwhile, the parameter selection for segmentation scale and kernel size is discussed. Robust assessment is performed on large scale images. The data are 8-bit values ranging from 0 to 255.

4.2.1 Data

The data used for experiments was provided by Pictometry International. The spatial resolution is around one inch. Two sets of data are provided as shown in Table 4.1.

Table 4.1: Data Sets Provided by Pictometry International for Debris Detection

Name of Storm	Date	Location	Data Type
Hurricane Ike	Sep, 2008	Galveston, Texas	Pre and Post
Moore Tornado	May, 2013	Moore, Oklahoma	Post

Hurricane Ike was the third-costliest hurricane ever to make landfall in the Unites States and the costliest hurricane in Texas history. The Moore tornado struck Moore, Oklahoma, and adjacent areas on May 20, 2013, killing 23 people and injuring 377 others.

The Pictometry Online Interface (POL) provides web based access to Pictometry imagery. Pictometry images every location multiple times, with different views. In this study, only the Ortho images are used. Twenty large scale images of each data set are tested in the experiment. Fig. 4.1 shows samples of before-and-after aerial imagery collected by Pictometry.

4.2.2 Statistical and frequency analysis of debris area

Statistical and spatial frequency analysis on the debris area is performed.

As shown in Fig. 4.2, before-and-after hurricane Ike images are pixel to pixel registered. Mean radiance and standard deviation of before-and-after hurricane Ike images are calculated.

Table 4.2 shows that the mean radiance value is increased after Ike. The standard deviation is also increased. The value is band averaged. The average radiance values are increased because the main materials of debris are concrete and wood with relative higher radiance values than other materials. The standard deviation is increased because of the complexity of debris area. Standard deviation can be used for debris detection as proposed

(a) Pre-Ike data (b) Post-Ike data

Figure 4.1: Before-and-After hurricane Ike images

by Shunichi Koshimura et al. [34]. They used the threshold of standard deviation $\sigma > 55.0$ to detect the debris area.

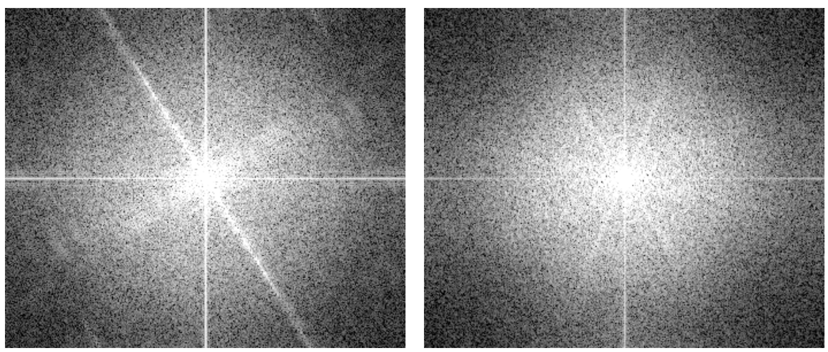
Table 4.2: Mean and Standard Deviation (Std) Values of Before-and-After Hurricane Ike Subimages.

	Pre-Ike data	Post-Ike data
Mean	154	176
Std	45	59

Frequency analysis of before-and-after hurricane Ike images are shown in Fig. 4.3. The Fourier spectrum of the no-debris area is relatively structured compared to the spectrum of the debris area, especially in the high frequency domain. It can be seen more clearly through the Fourier spectrum of the images as shown in Fig. 4.4.

(a) Pre-Ike data (b) Post-Ike data

Figure 4.2: Before-and-After hurricane Ike images



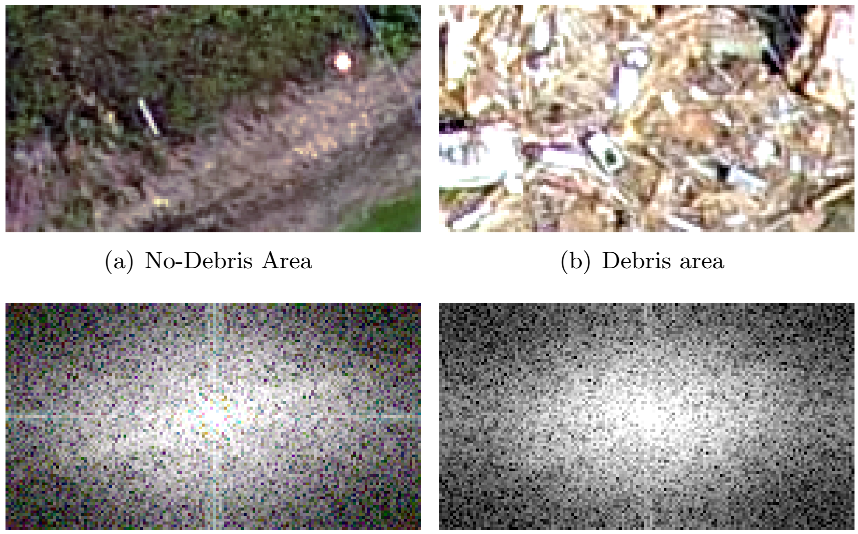
(a) FFT of no debris area (b) FFT of debris area

Figure 4.3: Fourier spectrum of no-debris and debris area

4.2.3 FFT based debris detection algorithm

A detection method is proposed based on the frequency analysis of debris area. The method is established on the fact that, for a no debris area with directional or periodic texture, the magnitudes of its Fourier spectrum will concentrate on a certain direction or several directions. For debris area with random texture spreads out over all directions, i.e., for random textures, the distributions of the responses of spectra are not restricted to certain directions.

First, a Fourier transform is applied to the local region extracted by a $w \times w$ sliding window. For a square image of size $w \times w$, the two dimensional Discrete Fourier Transform (DFT) is given by:



(a) No-Debris Area (b) Debris area

(c) Fourier spectrum of no-debris area (d) Fourier spectrum of debris area

Figure 4.4: Fourier spectrum of no-debris and debris area

$$F(k,l) = \sum_{i=0}^{w-1} \sum_{j=0}^{w-1} f(i,j) e^{-i2\pi(\frac{ki}{w} + \frac{lj}{w})} \tag{4.1}$$

where $f(a,b)$ is the image in the spatial domain and exponential term is the basis function corresponding to each point $F(k,l)$ in the Fourier space. Then, each spectrum window is smoothed by convolving with a 2D Gaussian kernel. Then, as shown in Fig. 4.5, the spectrum "pie" is divided into 8 pieces and the summation of pixel value in each piece is calculated. The standard deviation of these summations is calculated and saved for the center pixel of the sliding window. According to the frequency analysis, the standard deviation should be lower for debris area. A threshold can be applied to extract the debris. The results are provided in Section 4.3.

4.2.4 Object based debris detection algorithm

Object based imagery analysis (OBIA) works on the object level instead of the pixel level. In contrast to traditional pixel-based methods that classify individual pixels directly, object-based classification first groups pixels into homogeneous objects. The feature of each segment can be extracted and used for the following processing.

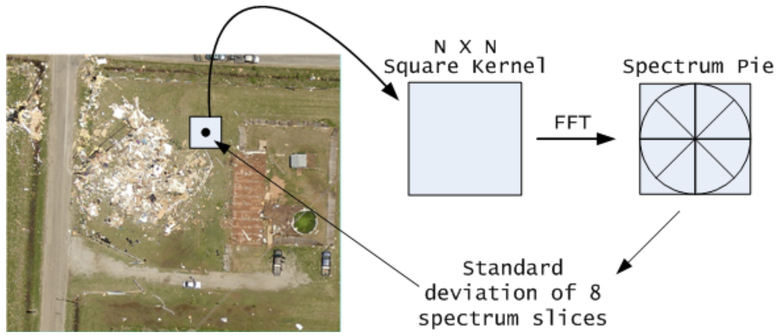


Figure 4.5: Fourier spectrum analysis

There are several software packages available to perform OBIA, such as eCognition, Feature Analyst and Feature Extraction. Our algorithm is fulfilled using the Feature Extraction module in ENVI software from ITT Visual Information Solution.

As shown in Fig. 4.6, the algorithm can be divided into three steps: interactive segmentation, feature extraction and thresholding.

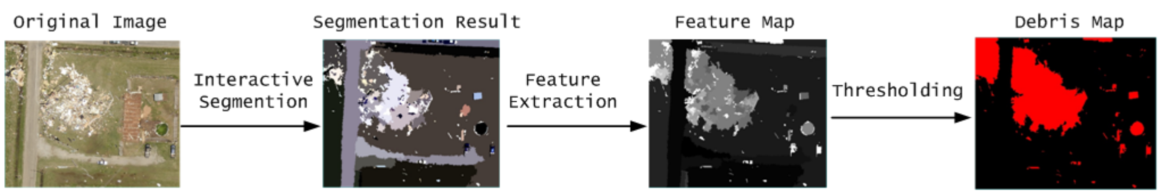


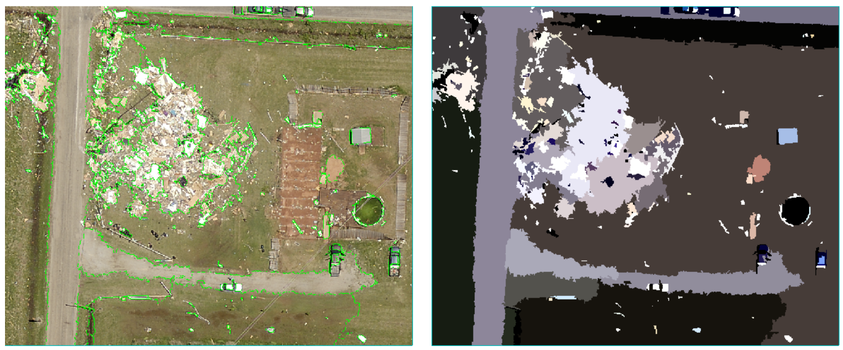
Figure 4.6: Framework of OBIA debris detection

The first step is segmentation with a proper scale. If the segment is too small, it can not capture the texture feature of debris. If the object is too large, it can not separate the debris from others.

The segmentation in ENVI Feature Extraction is controlled by two parameters: scale level and merge level. ENVI uses two numbers between 0 to 100 to represent the degree of segmentation and mergence. The scale level controls relative segment size. The segment size should adjusted to be able to represent the minimum-sized debris. The merge level controls the degree of merging contiguous segments into larger objects. A scale level of 25 followed by merge level of 99.7 is selected by trail and error.

The segmentation inside ENVI Feature Extraction includes two steps: a gradient map is first computed using a sobel edge detector and watershed algorithm is then applied on the gradient map to get the segmentation result.

Fig. 4.7(a) shows that values of 25 for Scale level and 99.7 for Merge Level effectively delineate the boundaries of debris area. Fig. 4.7(b) is the region mean display of the segmentation result.



(a) Boundary image (b) Region mean display

Figure 4.7: Segmentation result with Scale Level 25 and Merge Level 99.7

After the segmentation, the image is separated into individual objects as shown in Fig. 4.7(b), then spectral, texture and spatial features of each object can be computed and used for the following classification.

The most obvious difference between debris and no-debris area is texture pattern. The texture pattern of debris is random while others is relative uniform. Thus, texture feature should be extracted for debris detection. Four texture attributes are available in Feature Extraction: texture range, texture mean, texture variance and texture entropy.

As shown in Fig. 4.8, the texture features are calculated in the following steps. A $k \times k$ square kernel is applied on the image. Range, mean, variance and entropy are calculated for pixels inside the kernel window. Result is referenced to the center pixel. Average the values inside each segment to create the feature map.

The texture range attribute (kernel size 3) images are shown in Fig. 4.9 . A higher gray level means higher texture range value. From Fig. 4.9, we can easily identify debris with high texture range value. Meanwhile, we find that in the very-high-resolution(VHR) aerial image, the texture feature of cars is similar to debris object. The aerial image provided by Pictometry is a 3 band image. The attribute image of each band is compared. There is no obvious difference among the R, G and B bands.

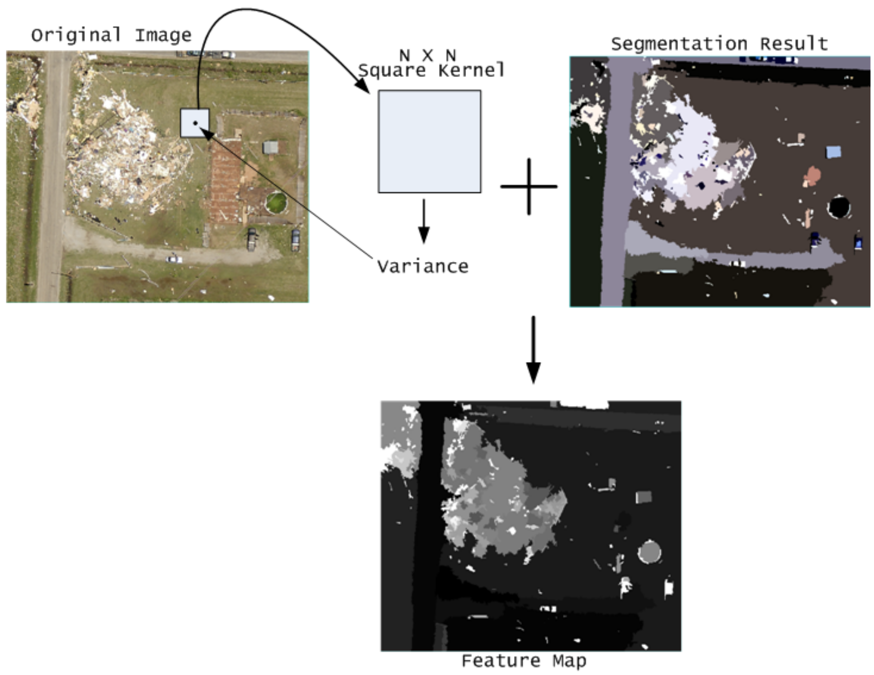
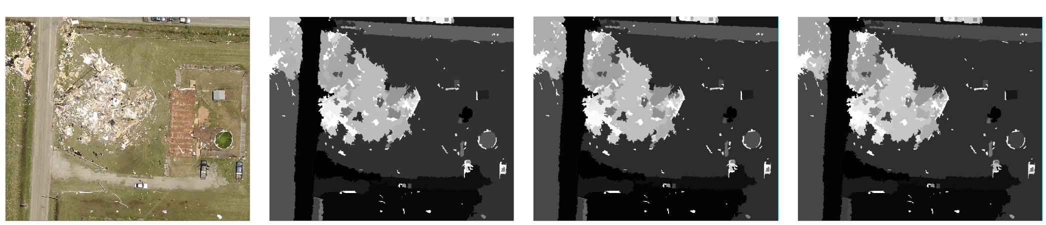


Figure 4.8: Texture feature calculation

Texture mean was rejected as a feature because it does not work for asphalt debris.The texture mean attribute image is shown in Fig. 4.10. Texture entropy is also rejected because it can not distinguish debris and no-debris area. This attribute image is shown in Fig. 4.11.

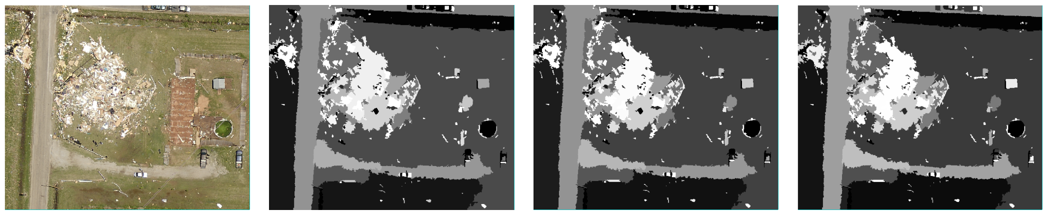
Texture variance attribute images are shown in Fig. 4.12. Now, we need to make a choice between texture range and variance.

Another parameter, kernel size, is considered. Attribute images of texture range with different kernel sizes are shown in Fig. 4.13. With the increase of kernel size, the texture



(a) Original image (b) R band (c) G band (d) B band

Figure 4.9: Attribute image of texture range



(a) Original image (b) R band (c) G band (d) B band

Figure 4.10: Attribute image of texture mean

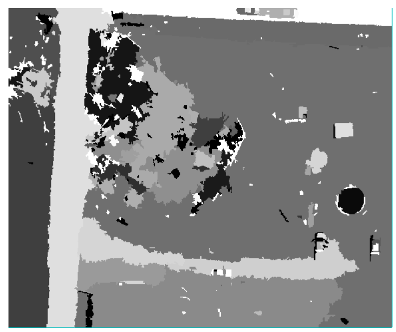
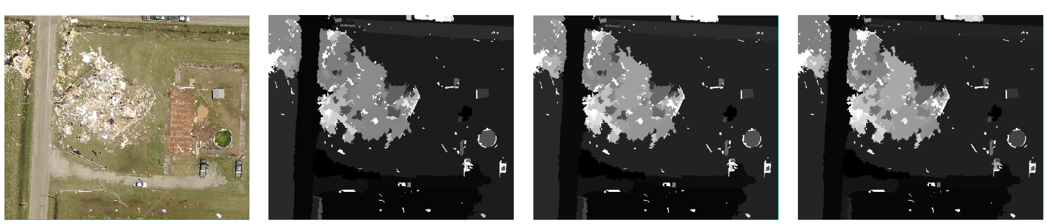


Figure 4.11: Attribute image of texture entropy (R band)

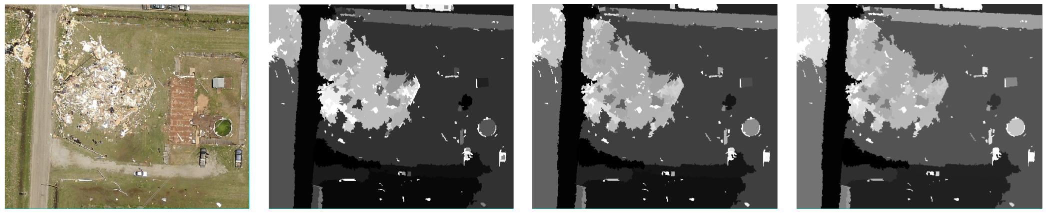
range of the undamaged roof would increase. It may result in false alarm.

Attribute images of texture variance with different kernel sizes are shown in Figs. 4.14. The obvious difference between Figs. 4.13 and Figs. 4.14 is that the attribute value of land and road would not change when kernel size is increased. Thus, texture variance is selected. Kernel size is set to 15, to increase the probability of detection of large concrete debris. A threshold of 1800 for texture variance is set to obtain the final result. More



(a) Original image (b) R band (c) G band (d) B band

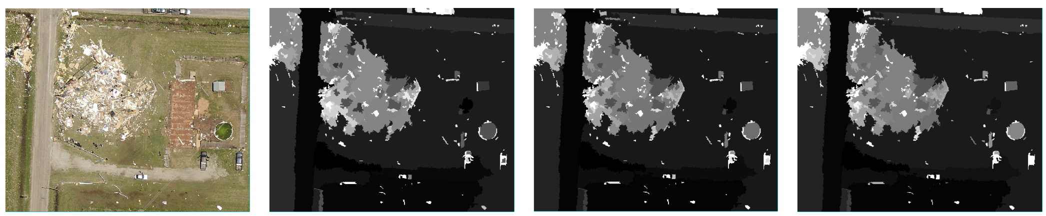
Figure 4.12: Attribute image of texture variance



(a) Original image (b) Kernel size = 3 (c) Kernel size = 9 (d) Kernel size = 15

Figure 4.13: Attribute images of texture range with different kernel sizes

results are provided in Chapter 5.



(a) Original image (b) Kernel size = 3 (c) Kernel size = 9 (d) Kernel size = 15

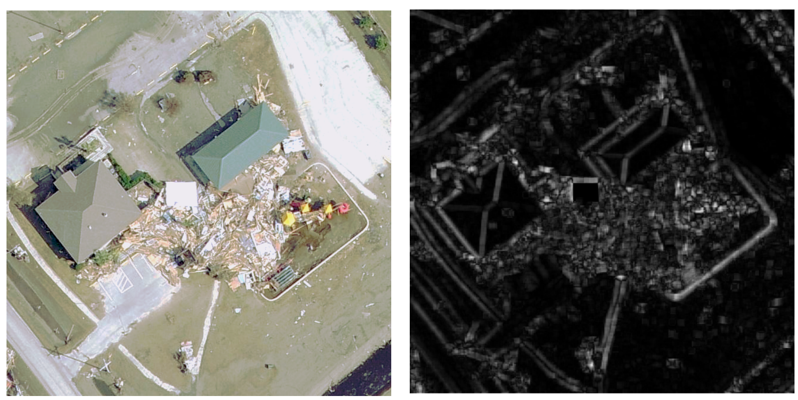
Figure 4.14: Attribute images of texture variance with different kernel sizes

4.3 Results

Two methods are proposed for debris detection: Fast Fourier transform (FFT) based method and object based image analysis (OBIA) method as described in section 4.2.

4.3.1 FFT based debris detection algorithm

The result of the FFT based debris detection algorithm is shown in Figs. 4.15. According to the frequency analysis, the standard deviation of summations of pixel values in each spectrum piece should be lower for debris area. A threshold can be used to extract the debris.



(a) Original image (b) Debris detection result

Figure 4.15: FFT based debris detection Result

However, in the result, uniform areas show lower standard deviation than debris areas. Thus, this method can not be used for debris detection.

4.3.2 OBIA debris detection algorithm

The OBIA debris detection framework is shown in Fig. 4.16. The detection algorithm using texture variance provides us a general location of debris.

Texture variance with kernel size 15 is extracted from each segment. The segment scale is set to 25 and merge scale is 99.7. The debris detection results just based on the texture variance are shown in Fig. 4.17. The detection result is promising with threshold 1800.

Most of the debris is detected, even small areas. However, the car and the edge of a pool are misclassified as debris.

More data are involved in the robust assessment and the results are shown in Figs. 4.18-4.20. We have compared results performed on all bands and found that the blue band gives the best results.

In Fig. 4.18, the algorithm detects the uprooted trees around the building. The debris pile at the lower right corner consisting of concrete and wood is also detected. The trunks of uprooted trees at the left part of the image are located. Tree crown would be detected if red and green band are used with threshold of texture variance 1300. The edge of the road is misclassified as debris. Compared to data 1, most of the data 2 is no-debris area. In the result shown in Figs. 4.19, most part of the image 2 is labeled as no-debris area. It detected the debris at the upper left corner of the image. Cars are misclassified as debris because of their complex texture pattern. Most debris at the right part of data 3 are detected as shown in Figs. 4.20. However, the algorithm fails to find the the debris pile in the forest at the left part of image. In general, the results are acceptable on large scale data.

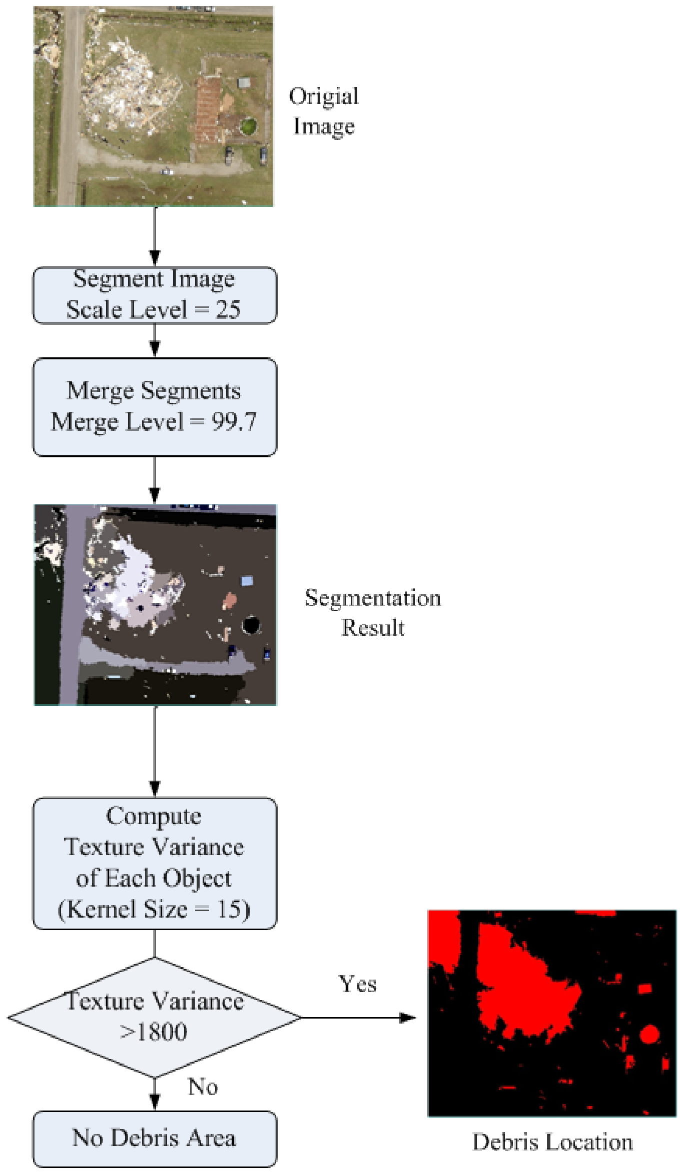
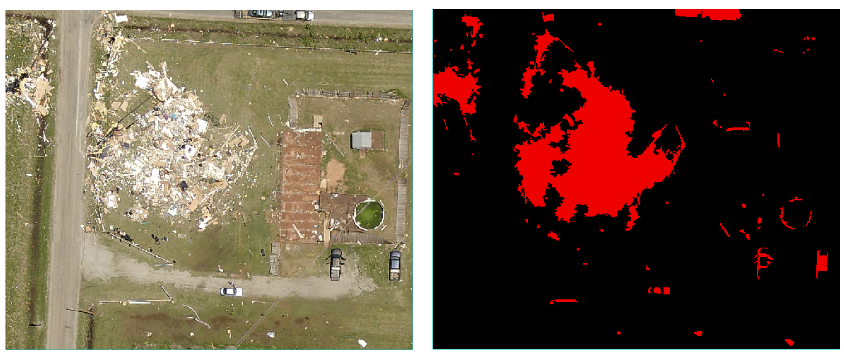


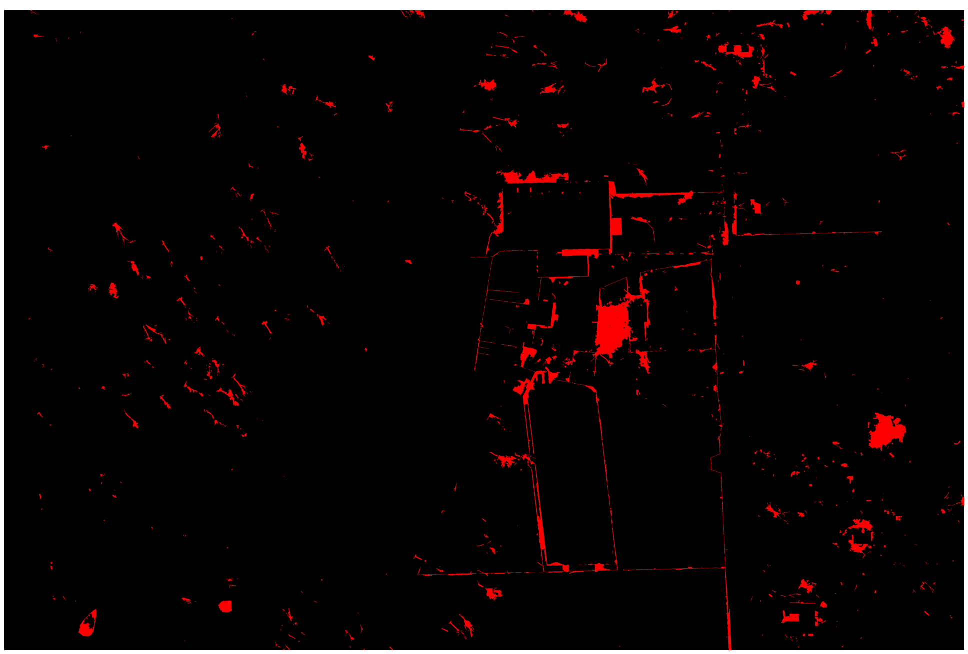
Figure 4.16: OBIA based debris detection Framework



(a) Original image (b) Detection result

Figure 4.17: OBIA based debris detection result

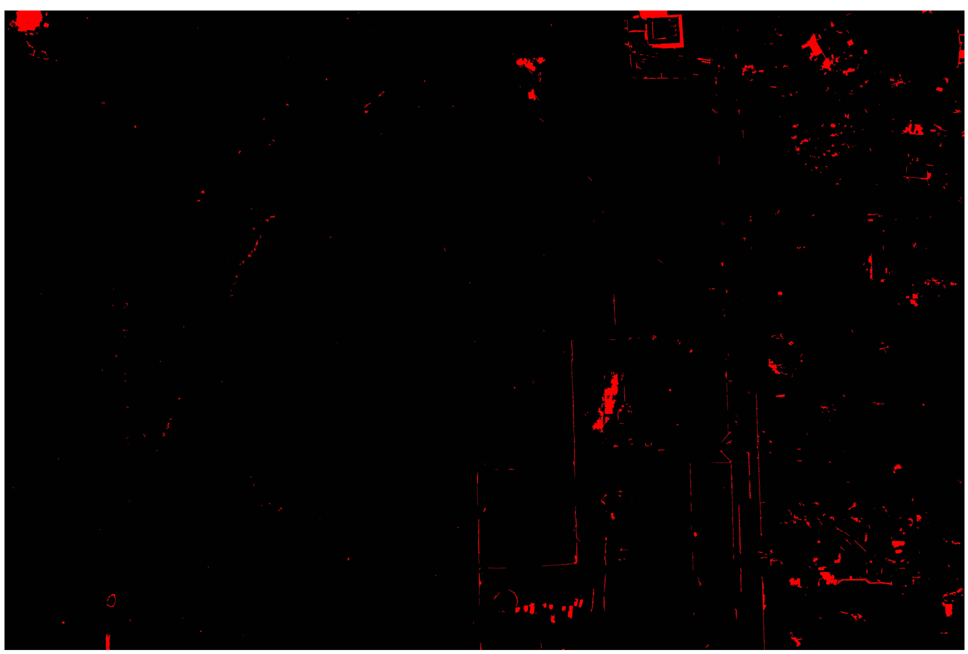
(a) Original image



(b) Debris detection result on blue band

Figure 4.18: Robust experiment on large scale image 1

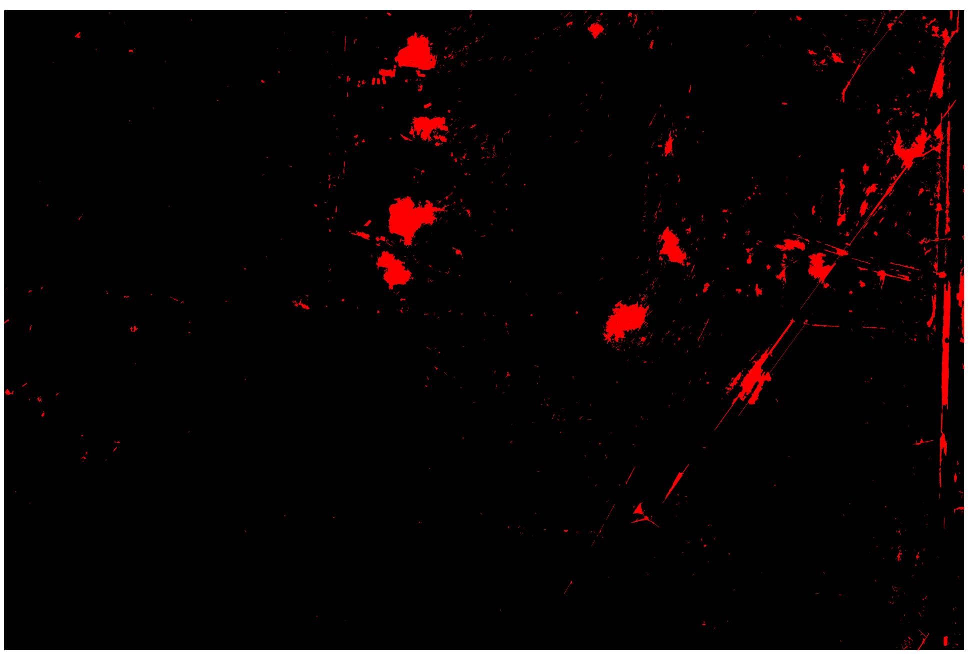
(a) Original image



(b) Debris detection result on blue band

Figure 4.19: Robust experiment on large scale image 2

(a) Original image



(b) Debris detection result on blue band

Figure 4.20: Robust experiment on large scale image 3

4.4 Conclusion

Debris detection is needed for effective debris removal and allocation of limited resource. The frequency analysis of no-debris and debris areas is performed and it shows that the frequency spectrum of non-debris area is relatively structured compared to debris area. Thus, a Fast Fourier transform (FFT) based debris detection algorithm is proposed. Although the method did not provide promising results, it sets a direction for future research. A more effective way to use the frequency feature of debris area could be investigated. Another debris detection algorithm under object based image analysis(OBIA) framework was proposed. The algorithm can be divided into three steps: interactive segmentation, feature extraction and thresholding. The interactive segmentation is performed using ENVI Feature Extraction module. Texture variance inside each segment is extracted to create the feature map. Thresholding is applied to get the debris detection result. According to the visual inspection, the performance is promising and robust on large scale data. Compared with traditional pixel-based methods which can require pre-and post-disaster imagery, the proposed method utilizes the texture feature more efficiently at the object level. Compared with traditional disaster assessment methods, only post-disaster imagery is needed to fulfill the debris detection with methods proposed here providing robust results.

Chapter 5

Building Detection

Robust building detection is an important part of high resolution aerial imagery understanding [6]. Building detection also serves as pre-processing for building modeling and roof condition assessment applications.

In this chapter, a 3D scene classification algorithm for building detection using point clouds derived from multi-view imagery is proposed. Section 5.1 reviews the previous research for building detection. Section 5.2 states the method proposed. Results are provided in section 5.3 and conclusions are drawn in section 5.4.

5.1 Previous Research

Buildings are one of most important man-made objects in aerial images. Many studies have explored robust building detection. In this dissertation, building detection means labeling the building objects in point clouds derived from multi-view imagery. Plenty of approaches have been proposed and can be grouped into two basic categories [37].

The first group use purely mathematical models and geometric reasoning techniques [37]. In 2001, G.Sithole et al. assumed that the natural terrain surface does not have slopes above certain threshold. Based on this assumption, a slope adaptive filter was proposed to remove non-ground objects[38]. In 2007, Schnabel et al. presented an algorithm to detect basic shapes in point clouds based on RANSAC [39]. The algorithm is fast while still maintaining accuracy. In 2009, Sirmacek et al. used scale invariant feature transform (SIFT) and graph theoretical tools to extract separate buildings in the urban area [40]. In 2013, Sun et al. used a graph cut based method to segment vegetative areas using Lidar data [41]. A hierarchical Euclidean clustering was used to extract the ground terrain and building rooftop patches. Running time of these approaches is fast. However, they may be not robust for complex scenes [37]..

Another category uses machine learning theory [37]. 3D structure features are extracted for classification. Promising result can be provided even in complex scenes. In 2005, Muller et al. performed a seeded region growing algorithm [6]. Photometric and geometric features were extracted from each segment. Classification was performed on the feature vector to get the building class. In 2006, Pu et al. divided the terrestrial laser data into individual segments using the planar surface growing algorithms [42]. Properties are computed for each segment and a rule based recognition performed. In 2008, Biosca et al. presented an unsupervised robust clustering approach for terrestrial laser data based on fuzzy methods [43]. The drawback of these methods is that the feature extraction processing is time consuming [37].

For this study, the data we use are point clouds derived from multi-view imagery provided by Pictometry International. The techniques of point cloud generation in Pictometry are still under development. The quality of data is far from perfect. Lots of existing methods may not work for our data. We need to investigate methods specific for the Pictometry point cloud data. For the future, spectra, texture, shape or morphological features [44] from one-inch-resolution imagery should also be considered to improve the accuracy.

5.2 Approach

There are many segmentation methods available for 3D scene segmentation. However, at this point, the quality of point clouds provided by Pictometry International is limited. The goal of our research is to investigate methods that will work with limited quality point clouds.

5.2.1 Data

Point clouds derived from multi-view imagery were provided by Pictometry International. Sample data are shown in Fig. 5.1 and were derived from image shown in Fig. 5.2. A detail is shown in Fig. 5.3. We can see that two small rooftops are noisy with some outliers located near the boundary of the rooftop. Fig. 5.4 shows a large scale scene of point clouds.

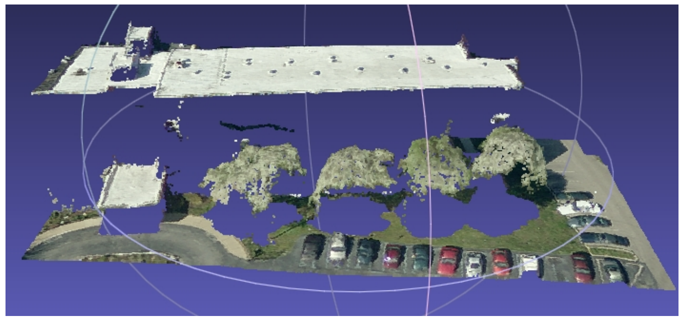


Figure 5.1: Point cloud provided by Pictometry International

Figure 5.2: Image provided by Pictometry International

5.2.2 Simple algorithm exploration

We started the project with a simple scene as shown in Fig. 5.1. The objective was to divide the data into tree categories: roofs, trees and ground.

The point cloud generation technique now is still under development and results in sparse outliers. Thus, pre-processing is needed to remove these outliers.

Statistical analysis on each point is performed. For each point, the mean distance from it to all its k nearest neighbors is computed. Assuming the distribution of all points' mean distances is Gaussian, those mean distances outside one standard deviation are labeled as outliers and rejected. The before and after outlier removal data are shown in Figs. 5.5. After the outlier removal, the point cloud is relatively clear.The canopy is not integrated compared with the data before outlier removal. Man-made objects are our main interest, thus the result is acceptable.

The following processing is plane model segmentation. The random sample consensus

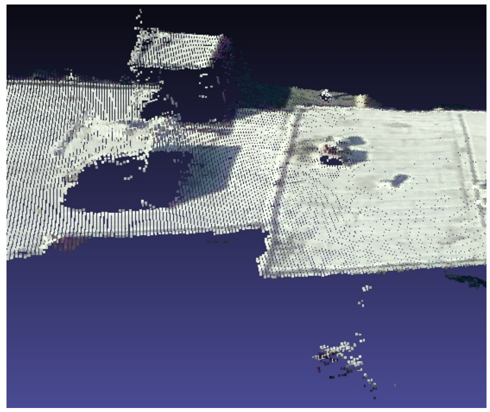


Figure 5.3: Detail of point cloud

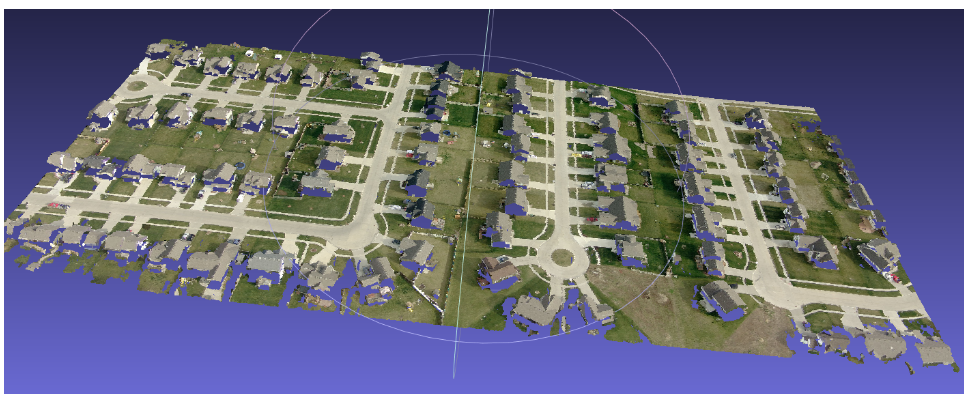


Figure 5.4: Large scale scene of point cloud

(RANSAC)[45] algorithm is used to fit all the points that support a plane model in the selected scene. The result of the RANSAC algorithm is shown in Fig. 5.6. Ground and large roofs in the simple scene are extracted. We can label them based on their average elevation. First, the average elevation of the entire scene is calculated. Then, the average heights of plane 1 and plane 2 are calculated. Assuming the elevation of roof is higher than the average height of the scene while the height of ground is lower than the average, we can label them as roof or ground. However, the small rooftops can not be extracted through RANSAC plane model segmentation since they are not flat enough to fit the plane model.

The remaining areas after plane model segmentation are shown in Fig. 5.7. Three small roofs and four tree canopies are left. Also, we can see some noisy parts which belong to the ground. These noisy parts can be removed through post-processing.

Now, the mission is to distinguish small roofs from trees. The first step is still segmentation. Different objects in the leftover scene are spatially separated, therefore it is reasonable to apply Euclidean clustering for segmentation.

After the segmentation, the next step is to extract features from individual clusters and label them based on the extracted features. A structure feature, the average difference of normals (DoN) [46] is selected. For each point, normals with two different radius are calculated and then the difference of these two normals is computed. The average DoN of a tree cluster should be higher than that of a relative flat rooftop. A threshold is set to distinguish trees and roofs. The result of DoN based classification is shown in Fig. 5.8.

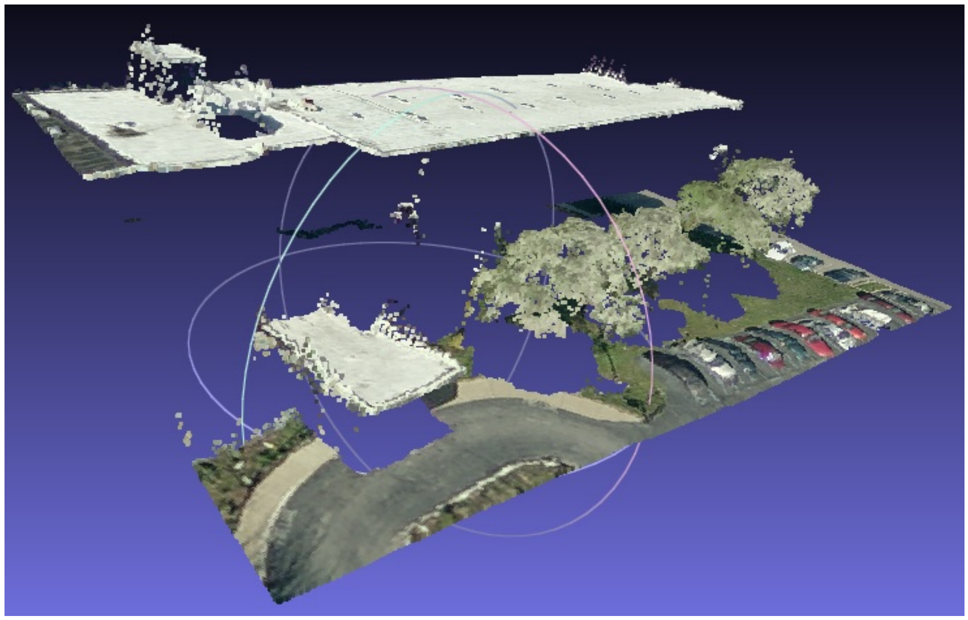
5.2.3 Robust building detection using computer vision technique

A robust building detection using computer vision algorithm is proposed.

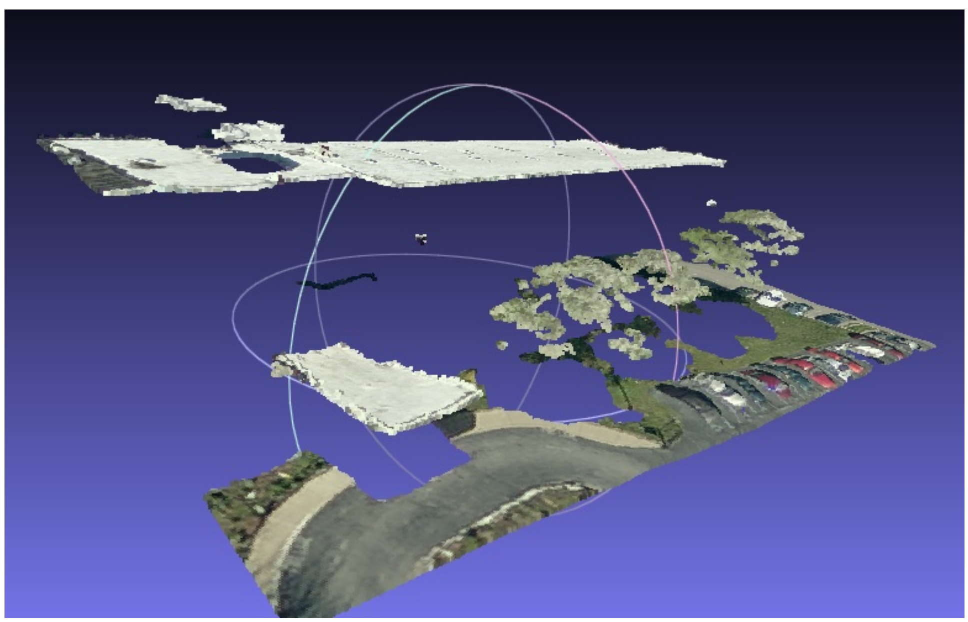
As shown in Fig. 5.9, after the outlier removal and Euclidean cluster, the input point cloud is segmented into individual point clusters. The main ground can be easily identified based on the number of points it contains.

Four kinds of features are extracted from individual point clusters as shown in Fig. 5.10. For each point cluster, the number and density of point cluster are computed. The height, area and also average RGB value are calculated. The curvature and DoNs are extracted as normal related features to describe the structure of point clusters. Among these features, the radius-search-based normal calculation takes most of time. To speed up the calculation, point clusters are downsampled and multi-threaded computation is applied.

With point clusters and its feature vectors, the next step is SVM classification as shown in Fig. 5.11. These manually labeled clusters are automatically divided into 3 folders. Folder 1 and 2 are used to train the SVM classifier with radius basis function (RBF) kernel. V-fold cross validation and grid search is used to find the best parameter pair for RBF kernel. Then, folder 3 is used to test the performance of trained classifier. Results are provided in Section 5.3.



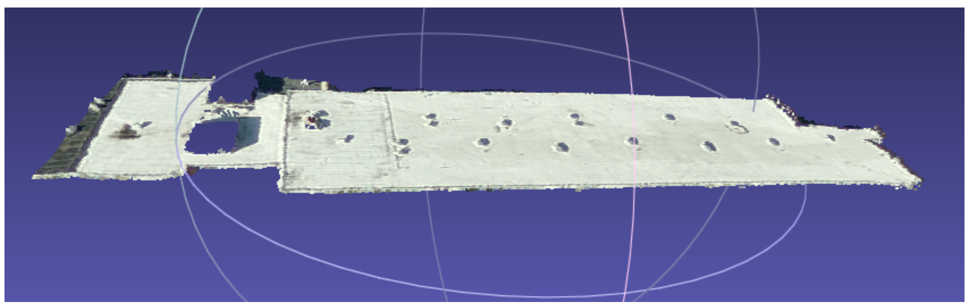
(a) Point cloud before outlier removal



(b) Point cloud after outlier removal

Figure 5.5: Point cloud before and after outlier removal with $k = 50$

(a) Plane 1



(b) Plane 2

Figure 5.6: Plane model segmentation result

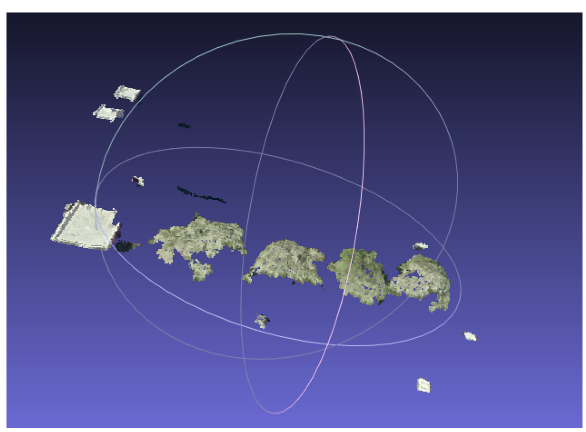
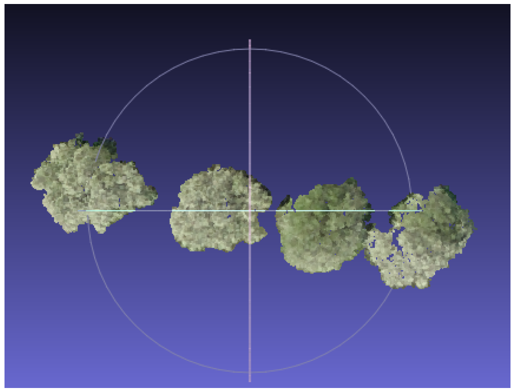
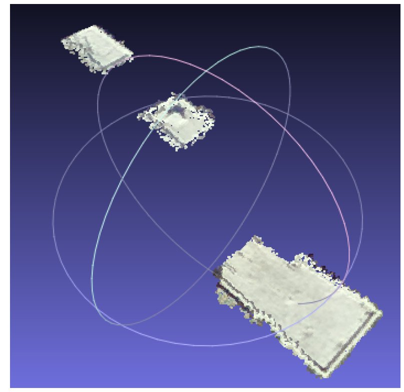


Figure 5.7: Leftover after plane model segmentation



(a) Plane 1



(b) Plane 2

Figure 5.8: DoN classification result

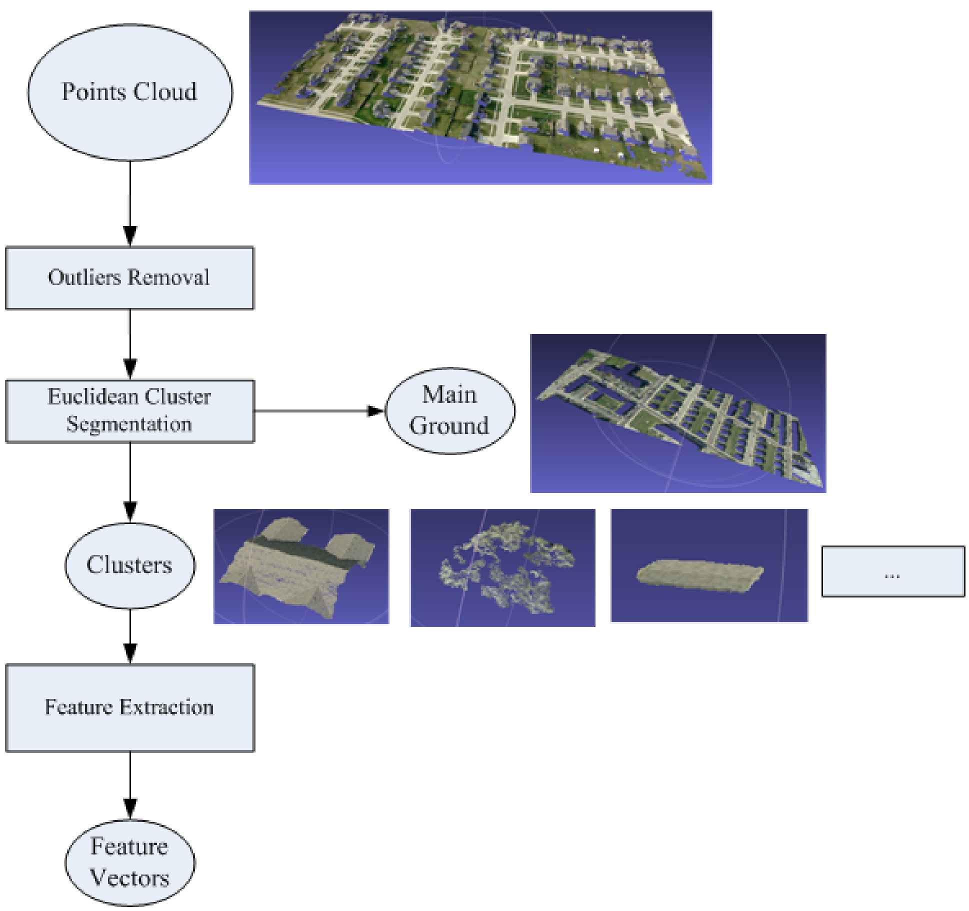


Figure 5.9: Segmentation and feature extraction

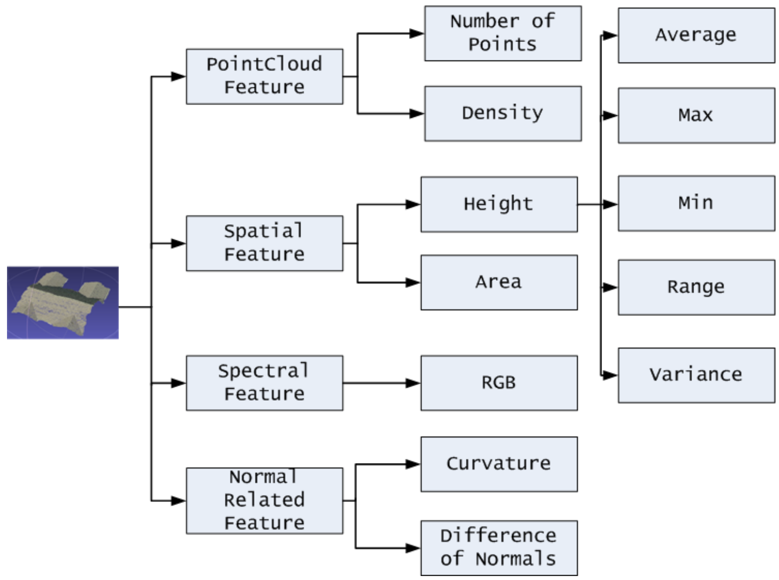


Figure 5.10: Feature extraction

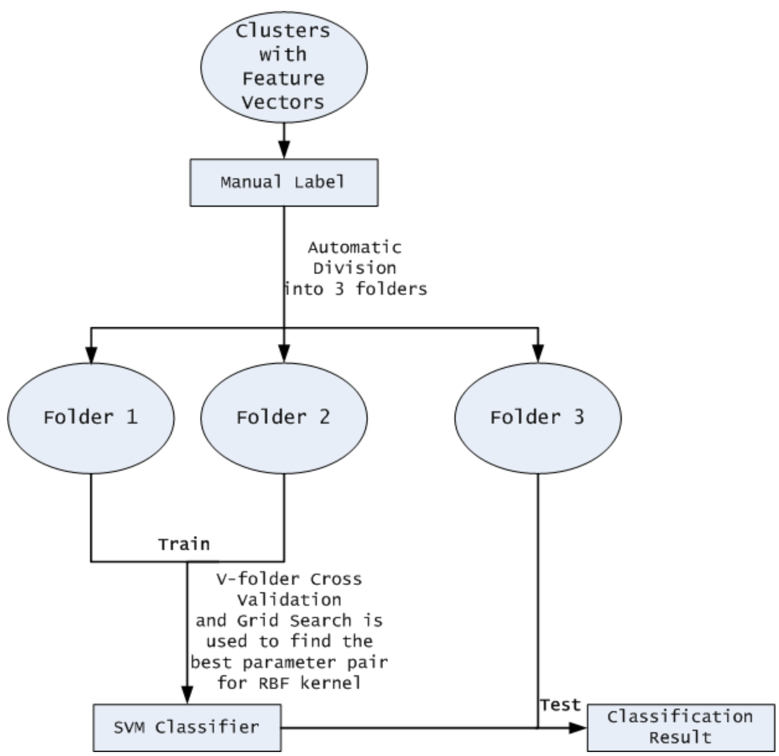


Figure 5.11: SVM classifier training and testing

5.3 Results

The results of the simple algorithm and the machine learning based methods for building detection are provided and discussed in the following sections.

5.3.1 Simple algorithm exploration

The framework of proposed simple algorithm for point cloud segmentation is shown in Fig. 5.12.

The result of the simple algorithm is shown in Figs. 5.13. The result is perfect for the selected scene. However, this algorithm is not robust on large scale scenes like Fig. 5.4.

The first problem is that the classical RANSAC algorithm can not detect the individual plane roofs in large scale scene where the plane roofs are very small part of the entire scene. Another problem is that RANSAC would not give us the main ground for some data shown in Fig. 5.14(a). The main ground of this data is not flat enough. Even with a loose limitation, RANSAC would output a result shown in Fig. 5.14(b) which is clearly not the ground.

In summary, the proposed unsupervised algorithm works perfect on the selected scene. However, it is not robust enough for large scale data.

5.3.2 Robust building detection using machine learning technique

The experiment is performed on the large scale scene of point cloud shown in Fig. 5.4. The accuracy of trained Support Vector Machine (SVM) classifier is 89% with 7% false detections. The result is acceptable. Mistakes happen when the trees and rooftops are spatially connected. The Euclidean clustering can not separate them and this case results in a mixed cluster.

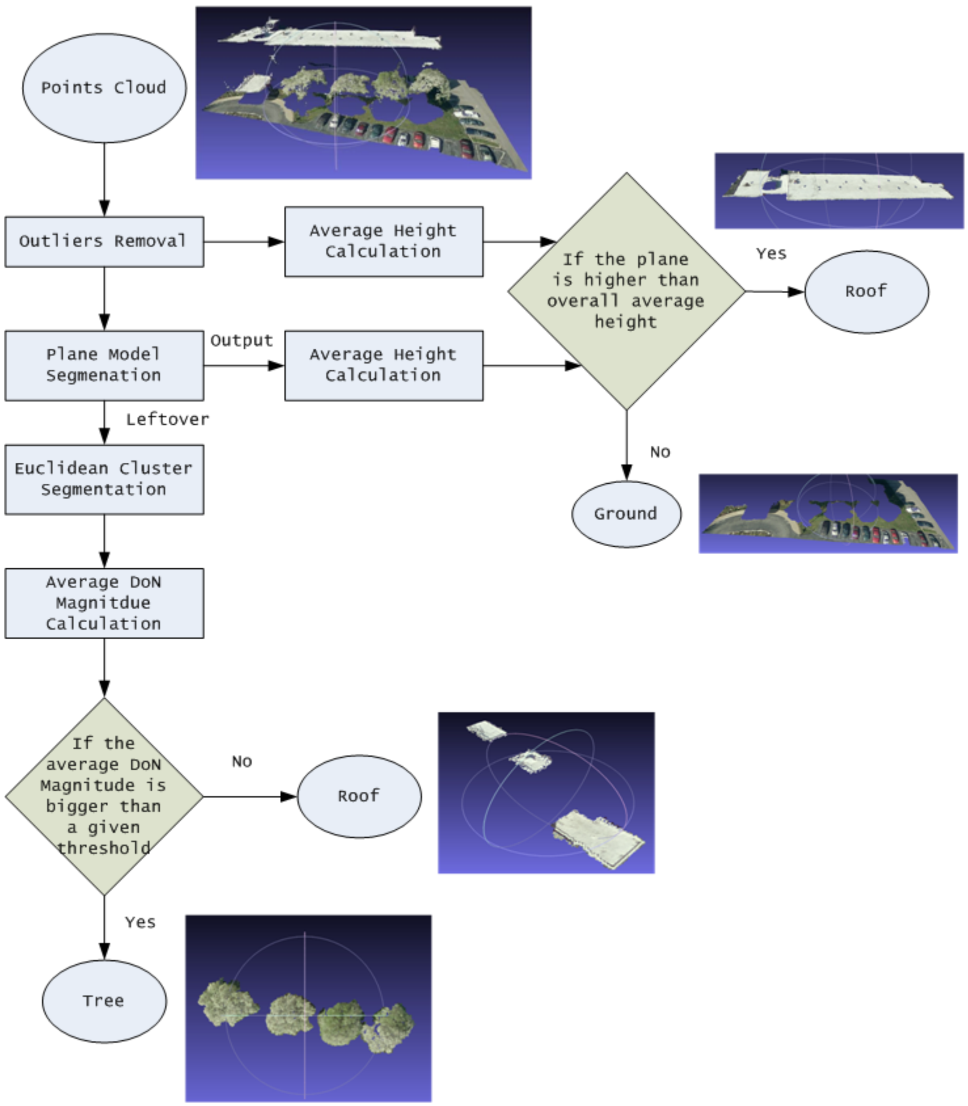
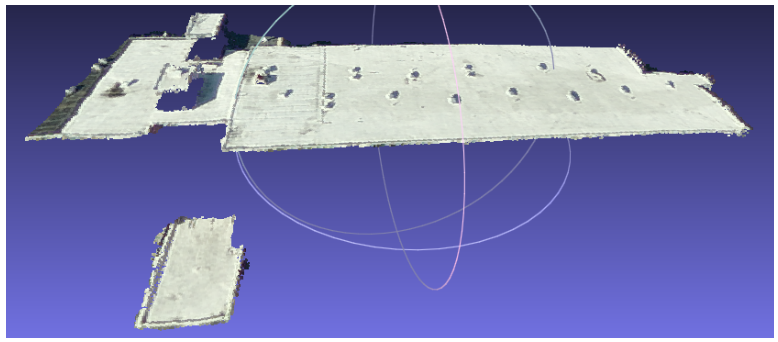
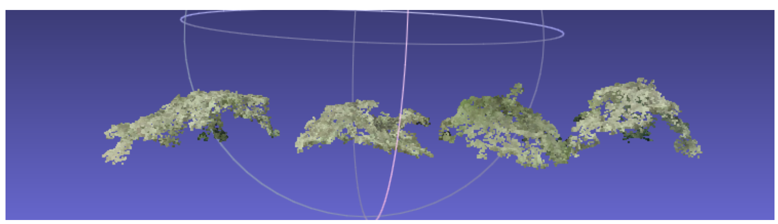


Figure 5.12: Framework of proposed simple algorithm

(a) Ground



(b) Roof



(c) Tree

Figure 5.13: Final result of our simple algorithm

(a) Data

(b) Output of RANSAC

Figure 5.14: Data sample which RANSAC can not detect the main ground

5.4 Conclusion

Building detection is an important part of high resolution aerial imagery understanding and it also serves as pre-processing for other applications. A building detection algorithm using 3D point cloud is proposed in this thesis. Point clouds are first segmented into individual point clusters using Euclidean clustering. Spectral and 3D structure features are extracted to represent each cluster. The extracted feature vectors are used to train the support vector machine (SVM) classifier. The accuracy of the proposed algorithm is 89% with 7% false detections.

Chapter 6

Roof Condition Assessment

The inspection of roof condition is an important step of damage claim processing in the insurance industry. Automated roof condition assessment methods from remotely sensed images are proposed in this Chapter. In section 6.1, previous research is reviewed. In section 6.2, texture classification and bag-of-words model are first performed to assess the roof condition using features covering the whole rooftop. Then, considering the complexity of residential rooftop, a more sophisticated method is proposed to divide the task into two stages: 1) roof segmentation (section 6.3), followed by 2) segments classification (section 6.4).

6.1 Previous Research

In this dissertation, roof condition assessment is defined as grading the roof condition based on the area of missing shingles or cosmetic damage on the roof using very-high-resolution(VHR) imagery. Very little research has been published specifically on roof condition assessment. There is some research for roof condition using spectral information. In 2016, Samsudin et al. proposed spectral indices to generate degradation status maps of concrete and metal roofing materials using multispectral imagery [47].

In the literature, the closest field we found is building damage assessment. The algorithms on building damage assessment are divided into two categories: methods using both pre- and post-event data or only post-event data. The methods working with pre- and post-event data provide more promising results. However, pre-event data are not always available.

As shown in Fig. 6.1, building damage can be classified into five damage grades: slight damage, moderate damage, heavy damage, very heavy damage and destruction in the European Macroseismic Scale 1998 (EMS98) [1].

(a) Grade 1: Negligible to slight damage

(b) Grade 2: Moderate damage

(c) Grade 3: Substantial to heavy damage

(d) Grade 4: Very heavy damage

(e) Grade 5: Destruction

Figure 6.1: Classification of damage to masonry buildings [1]

In 2009, Sirmacek et al. extracted building rooftop based on color invariants and shadow region using grayscale histogram. A damage measure derived from rooftop and shadow was proposed [48].

In 2010, Brunner et al. proposed a damage detection algorithm based on the similarity between predicted and actual synthetic aperture radar (SAR) data [49]. 3D parameters are derived from pre-event optical imagery and used to predict signature of the building without damage in the post-event SAR scene.

In 2011, a supervised classification algorithm was proposed by Gerke et al. to divide the scene into facades, intact roofs, destroyed roofs and vegetation using oblique Pictometry data [50]. 22 features were used. The role of features in classification was discussed. Damage score was derived from classification results from different viewing direction. EMS 98 standard was adopted. The accuracy of 70 percent for scene classification and 63 percent for building damage assessment were achieved. The benefit of using oblique Pictometry data is that the condition of facades can be assessed.

In 2011, Bignami et al. studied the sensitivity of objects textural feature with respect to damage levels using pre and post-event QuickBird data. Four texture features: contrast, dissimilarity, entropy and homogeneity were extracted from the building object [51] and tested regarding to different damage scale in EMS98.

According to Dong et al.'s review [52], heavy damage grades such as grad 5 in EMS98 are detectable. The challenge is identification of lower damage grades, even with a sub-meter resolution data [52]. In other words, most literature on damage assessment is about damage detection. Labeling the exact damage grade of a building is still a barrier. Compared to building damage assessment, roof condition assessment is a more sophisticated task because the emphasis of roof condition assessment is placed on grading lower damage grades.

In this dissertation, we aim to assess the roof condition using features covered the whole roof at first. Then, a novel method containing roof segmentation and segments recognition is proposed to provide a more subtle assessment.

6.2 Preliminary roof condition assessment

In this section, the preliminary roof condition assessment using feature extracted from the entire roof sample is performed. Texture classification and bag-of-words (BoW) model are applied.

6.2.1 Approach

6.2.1.1 Data

The data for this research consisted of one inch ground resolution color airborne imagery collected by Pictometry International. For this study, we manually extracted the roof sample images as shown in Fig. 6.2. Rooftops of interest were cut out from the imagery and a rotation was followed by additional cropping to obtain the final experimental sample images.

110 intact roofs and 164 damaged roofs were manually extracted from 469 high resolution images provided by Pictometry. Typical roof sample images are shown in Fig. 6.3. Cosmetic damage and missing shingles are the most obvious features of damaged roofs.

A challenge with these data is the high within-class diversity in the intact roofs and damaged roofs as shown in Figs. 6.4- 6.5. Features extracted from the entire rooftop may not be sufficient for automated condition assessment.

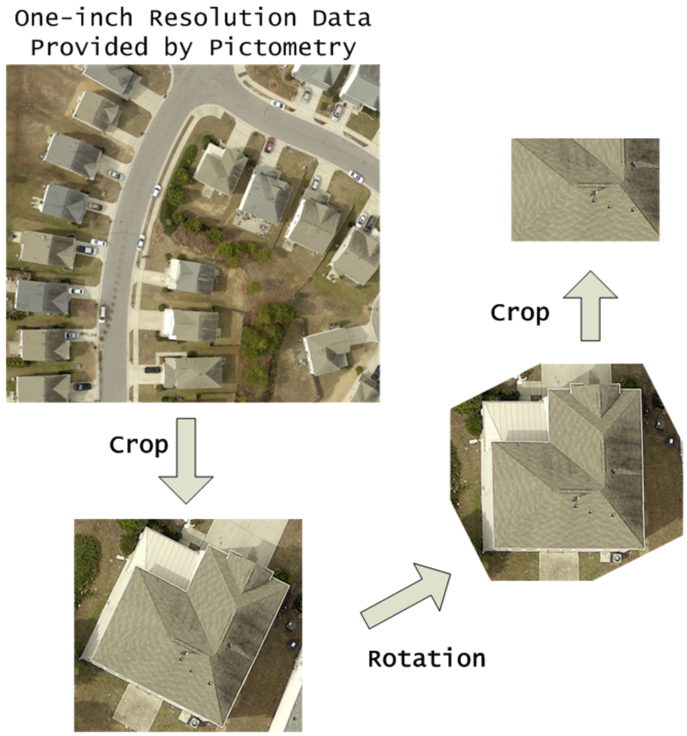


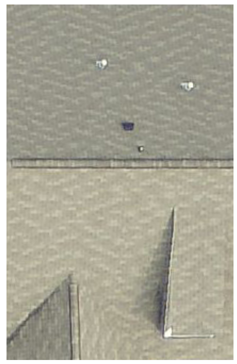
Figure 6.2: Manual data extraction

6.2.1.2 Texture classification

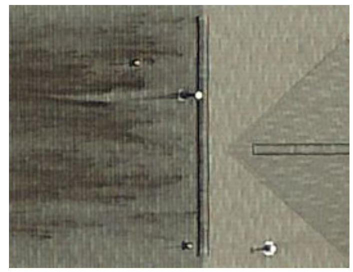
Texture features: Gray-Level Co-occurrence Matrix(GLCM), Local binary patterns(LBP) and Gabor filter have achieved great success in texture classification. Since the resolution of Pictometry data is around one inch, texture information is a reasonable choice to distinguish intact roofs from damaged ones.

The framework is shown in Fig. 6.6. We manually extract the experiment sample images and label them. As shown in Fig. 6.7, texture features are extracted. Statistical features: variance, standard deviation, skewness, kurtoisis, uniformity and entropy of each roof sample image are computed. Dissimilarity, correlation, homogeneity, contrast, ASM and energy are extracted from GLCM computation. LBP histogram is added to the feature vector. Meanwhile, mean and variance of Gabor filtered image are added. In summary, for each roof sample image, 72 features are calculated.

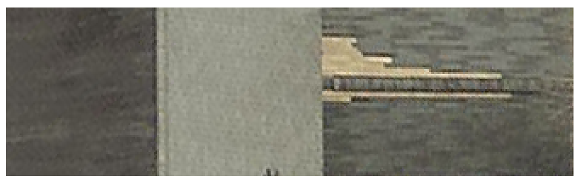
Recursive feature elimination (RFE) on training data is performed to select the most



(a) Intact roof



(b) Damaged roof



(c) Damaged roof

Figure 6.3: Pictometry rooftop sample images

useful features. A SVM linear kernel is trained on the initial set of features and weights (coefficients) are assigned to each one of them. Then, the features with smallest absolute weight are removed. The procedures are recursively repeated. The processing of RFE is shown in Fig. 6.8. Finally, 16 features which give the best classification accuracy are selected. The 16 selected features are used for the following SVM classification.

Feature vectors with labels are randomly divided into 2 folders. Folder 1 are used to train the SVM classifier with Polynomial kernel. Folder 2 are used to test the trained SVM classifier. Results are provided in Section 6.2.2.

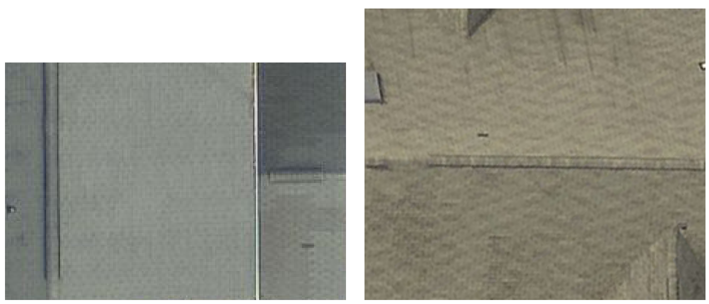
6.2.1.3 Bag of Words model

The BoW model is a classical computer vision technique used for content based image retrieval. Here, BoW is used for preliminary roof condition assessment. The basic flow for BoW is:

- Use Dense SIFT to collect lots of features from training images.
- Use K-means to cluster those features into a visual vocabulary.

(a) Shadow (b) Dust (c) Tree (d) Window

Figure 6.4: Intact rooftop sample images with high within-class diversity



(a) Sample 1 (b) Sample 2

Figure 6.5: Damaged rooftop sample images with tiny missing shingles

- For each of training images build a histogram of word frequency (assigning each feature found in the training image to the nearest world in the vocabulary).
- Feed these histogram to train an SVM classifier.
- Build a histogram for each of test images and classify them with the trained SVM.

The framework of BoW is shown in Fig. 6.9. Results are provided in Section 6.2.2.

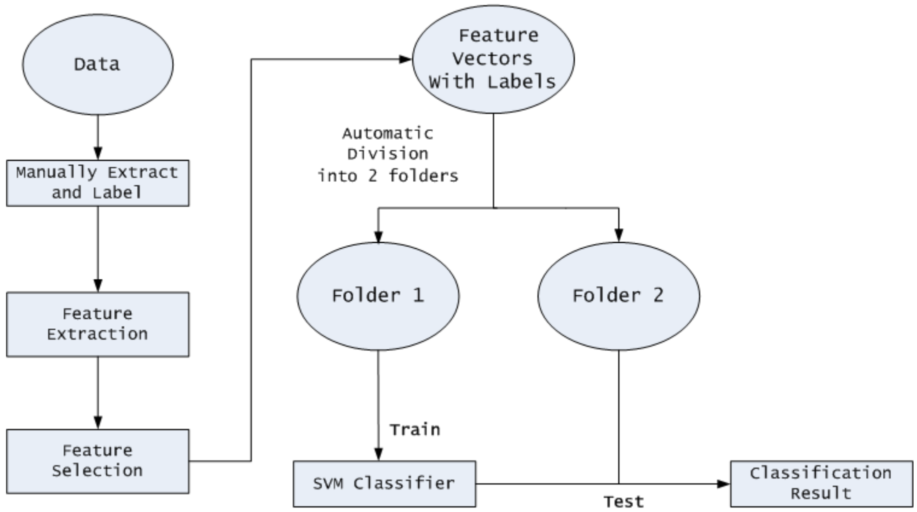


Figure 6.6: Traditional texture features based classification framework

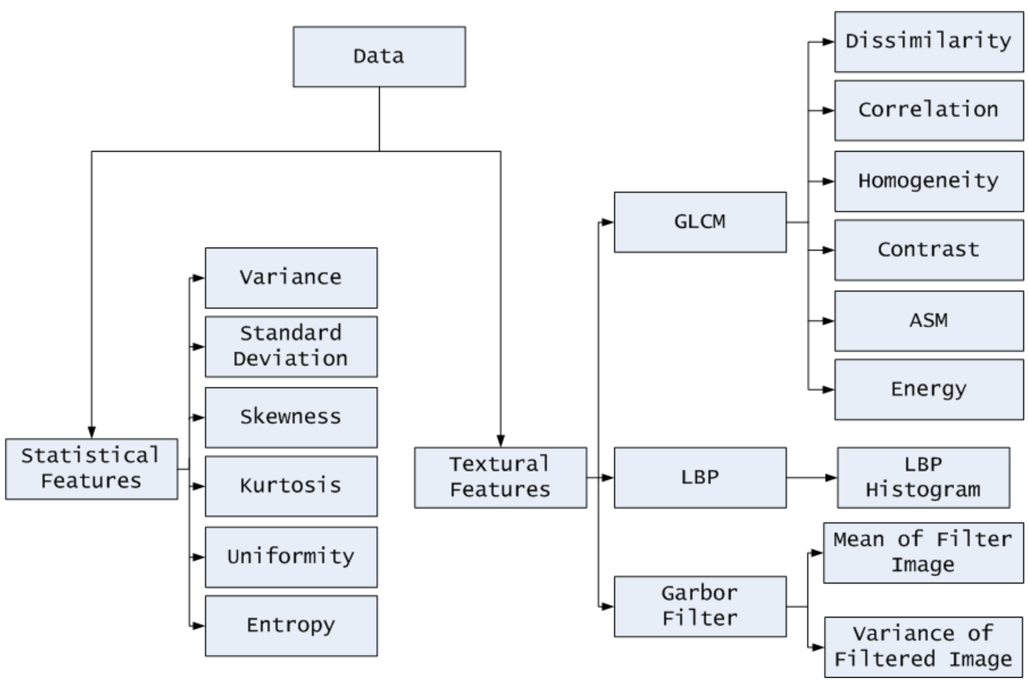


Figure 6.7: Traditional texture feature computation

Figure 6.8: Recursive feature elimination

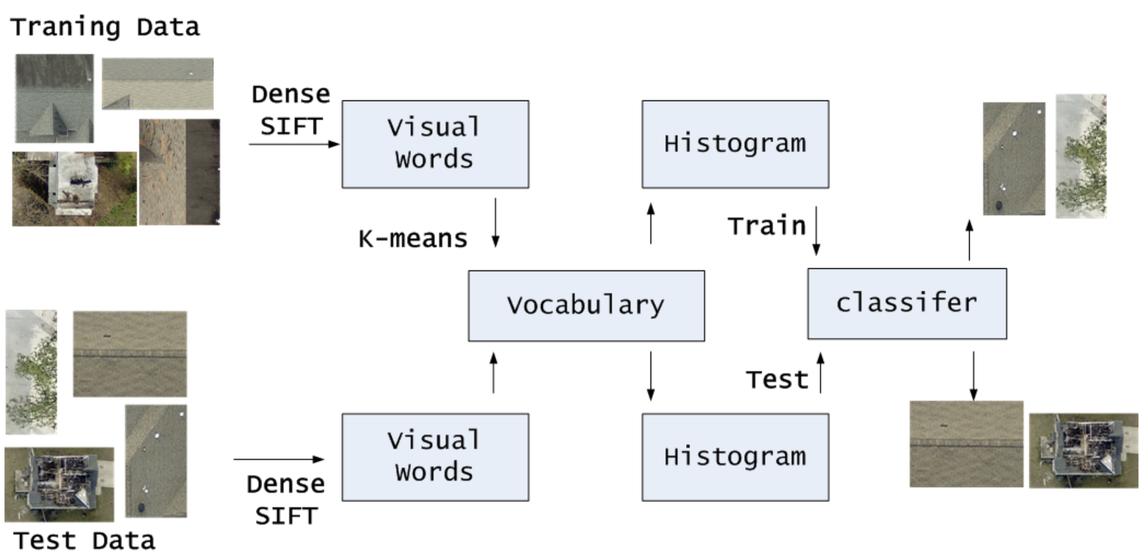


Figure 6.9: BoW framework

6.2.2 Results

The objective of preliminary roof condition assessment is to divide the roof images into intact roofs and damaged ones. The results of two proposed methods are presented and discussed bellow.

6.2.2.1 Results of texture classification

Experiments are performed on the 110 intact roofs and 164 damaged roofs which were manually extracted from 469 high resolution images provided by Pictometry International.

The result of texture classification is shown Table 6.1. The definitions of precision, recall and F1 are: $Precision = \frac{tp}{tp+fp}$, $Recall = \frac{tp}{tp+fn}$ and $F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision+Recall}$ where t is for true, f is for false, p is for positive and n is for negative.

Table 6.1: Results of Texture Classification

Class	Precision	Recall	F1-Score
Intact roof	0.83	0.55	0.66
Damaged roof	0.75	0.93	0.83

16 selected features derived from Recursive feature elimination (RFE) are used. For the trained SVM classifier, a polynomial kernel are selected. Penalty parameter C of error term is set to 10. The degree of polynomial is set to 5. From the result, we can see that the high within-class diversity in intact roofs class leads to relatively low classification performance.

If all the features are used, the result is shown Table 6.2. For the trained SVM, polynomial kernel is used. C is 10 and degree is 4. There are no false positives in this result. It can serve as a pre-processing step to eliminate part of intact roofs. The recall is low due to the complexity of our selected experiment sample images.

6.2.2.2 Results of bag of words (BoW) based classification

The best result of BoW is shown in Table 6.3. After RFE, only 2 features are selected. The linear kernel is used.

Table 6.2: Result of Texture Classification without Feature Selection

Class	Precision	Recall	F1-Score
Intact roof	1.00	0.15	0.25
Damaged roof	0.64	1.00	0.78

Table 6.3: Result of BoW

Class	Precision	Recall	F1-Score
Intact roof	0.77	0.73	0.75
Damaged roof	0.82	0.85	0.84

According to the **F1-Score**, it is the best result we achieved for preliminary roof condition assessment. Consider the complexity of residential roof, this result is not that bad. However, BoW method suffers random problem caused by K-means.

6.2.3 Conclusion

The inspection of roof condition is an important step of damage claim processing in insurance industry. Currently, roof inspection is done by humans which is expensive, time-consuming and unsafe. Thus, an automated roof condition assessment is of great interest. Preliminary roof condition assessment is proposed to assess roof condition using the feature covered the entire rooftop. Two methods are proposed in this thesis. The first method is based on the texture classification using Gray Level Co-Occurrence Matrix (GLCM), Local Binary Pattern (LBP) and Gabor filter. Another method is based on the classical bag of words (BoW) method. Method 1 gives a relatively lower performance but produces stable results. Method 2 gives better results but suffers from random problem cased by K-means.

6.3 Roof Segmentation

Considering the complexity of residential roofs, roof condition assessment methods using features covering the entire rooftop may not provide a promising result. A better approach is to divide the task into two stages: 1) roof segmentation; followed by 2) recognition of roof segments. In this section, we will detail the steps of our proposed segmentation algorithm. First, the simple linear iterative clustering (SLIC) algorithm was applied to partition an image into homogeneous regions [53]. Then, our proposed features using Holistcally-nested edge detection (HED), Color-HED histogram were extracted for region representation. A similarity measure of Color-HED features was defined. The image was represented by a region adjacency graph (RAG). The next step was region merging where the most similar neighboring regions were merged at each iteration. The optimal result was selected by an unsupervised index, Q.

6.3.1 Approach

Conventional region merging methods would stop when the similarity between any two adjacent regions is less than a preset threshold [54]. The preset threshold is fixed to a certain value and is expected to give satisfactory performance for images similar to those that were used to tune the parameter [55]. However, for our study, the high diversity of roof images causes there to be no single parameter value which will result in the best possible segmentation for all the images. It is impossible and unwise to perform manual parameter adjustment for each roof sample image. Thus, a self-tuning region merging segmentation method is proposed.

Our region merging segmentation is divided into pre and post-region-merging segmentation. Pre-region-merging stopped when the region number was equal to 25. Post-region-merging started from the pre-merging result and stopped at a preset similarity threshold. An unsupervised evaluation metric quantified the post-merging steps into a score list. The result of our algorithm corresponded to the segmentation achieved at the post-merging step with the minimal Q.

6.3.1.1 Over-segmentation using SLIC Superpixel

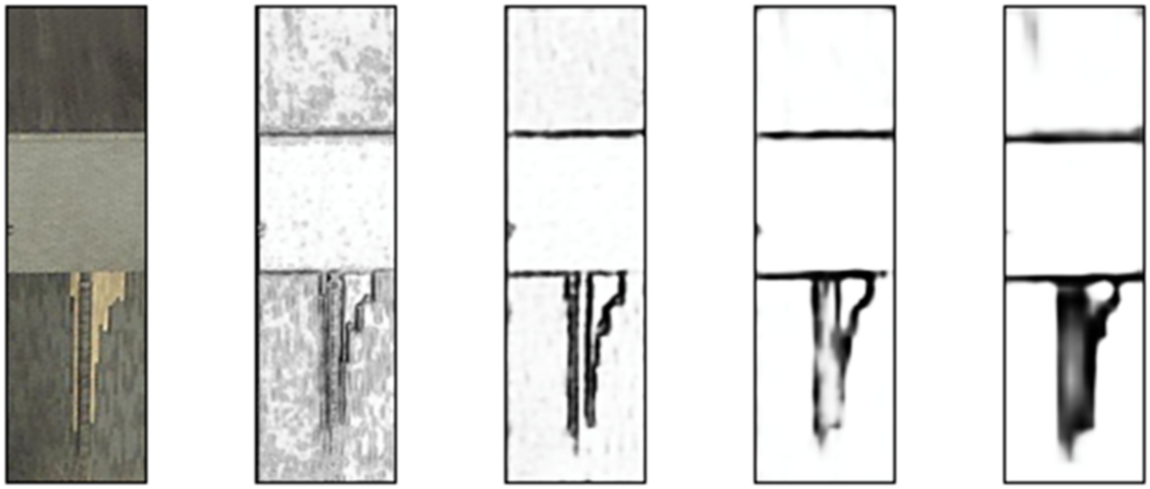
The proposed region merging method is based on an initial over-segmentation using the SLIC superpixel method which groups pixels into small homogeneous regions. The SLIC superpixel algorithm is selected because of its excellent boundary adherence [53] which is a necessary prerequisite to eventually obtain the accurate shape and area of missing shingles and cosmetic damage area. Meanwhile, SLIC is fast and easy to use [53].

SLIC is often intended to be applied to images in the CIELAB color space[53]. In our algorithm, the roof image was first converted to CIELAB format before SLIC superpixel was performed. Instead of determining the number of desired superpixels in each roof image, we set the nominal size of the superpixel to 15×15. The compactness parameter is used to control the tradeoff between superpixel compactness and boundary adherence [53] which was empirically set to 7 in this study.

6.3.1.2 Color-HED Feature Extraction

The feature representations through deep learning often outperform traditional hand-engineered features. Thus, a deep learned feature: Color-HED histogram is proposed for region representation.

Instead of extracting image features using traditional methods, feature can be extracted using a CNN. Each layer of CNN produces a response to an input image. The layer at the beginning of the network learn features similar to Gabor filter and color blobs [56]. Thus, those features are not specific to a particular task. The features computed by the last layer combine all the basic features into a richer one and thus depend greatly on the chosen task.



(a) (b) (c) (d) (e)

Figure 6.10: Examples of HED side outputs on roof image : (a) original image, (b) HED side output 1, (c) HED side output 2, (d) HED side output 3, (e) HED side output 4.

Our study has limited data, thus a fully supervised deep architectures will generally overfit the training data. Rather than learning a full deep representation, an easy way is to use a pre-trained CNN learned from related tasks as feature extractor. Holistcally-nested edge detection (HED)[57] is chosen for our task. It was first proposed as a deep learning architecture for edge detection in natural images [57]. It is a system inspired by fully convolutional neural networks with additional fine-tuning on top of VGGNet [58]. The HED networks comprise a single stream deep network with multiple side outputs. Each side-output produces a corresponding edge map at different scale level. Combining the side outputs together, the weighted-fusion layer yields final result.

As shown in Fig 6.10, the pre-trained HED network is applied on a roof image. As the HED network goes deeper, later outputs, such as side output 3 and 4 are more like the edge probability maps and cannot be used as region feature representation. Activation early in the network, such as side output 1, is more suitable for feature representation.

Thus, a novel feature, Color-HED histogram is defined in this study. The color channels in RGB space and side output 1 of HED network constitute a 4 dimension feature map. In our algorithm, the image was divided into many primitive regions after the SLIC segmentation. Color-HED histograms were then computed to represent each region. Each channel in Color-HED space was quantized into 16 bins. Each region was then represented by a vector of dimension $16^4 = 65536$.

6.3.1.3 Similarity Measure

The order of merging is one of the essential issues in region merging segmentation [59]. It is controlled by the similarity measure between adjacent regions.

In our algorithm, the Bhattacharyya coefficient [60] was adopted to measure the Color-HED feature similarity Sim between adjacent regions p and q

$$\mathrm{Sim} = \mathrm{BC}(p, q) = \sum_{i=1}^{n} \sqrt{\mathrm{CH}_p^i \cdot \mathrm{CH}_q^i} \tag{6.1}$$

where CH_p^i and CH_q^i are the normalized Color-HED histograms of region p and q. The superscript i represents the ith element and n is the dimension of the Color-HED histogram.

6.3.1.4 Maximal Similarity Merging Process

After the distance measure Sim between all adjacent region pairs were computed, the image was represented as a region adjacency graph (RAG) $G = (V, E, W)$ [61]. Regions produced by SLIC superpixel algorithm were denoted as a set of nodes $v_i \in V$. The edge $(v_i, v_j) \in E$ between adjacent nodes had a corresponding weight $w(v_i, v_j) \in W$ to measure the dissimilarity of two nodes. The minimum dissimilarity corresponds to the maximal similarity denoted by distance Sim. The region merging segmentation were then performed by iteratively merging the most similar connected regions. After each merging, the Color-HED features, RAG and similarity ranking were updated.

6.3.1.5 Self-tuning Segmentation with Unsupervised Segmentation Evaluation

To design a self-tuning segmentation algorithm, the first thing we need to know is what is a good segmentation. Segmentation evaluation is usually done by visual inspection or supervised evaluation using a manually-derived reference [62]. However, unsupervised segmentation evaluation is needed for a self-tuning segmentation algorithm. In the literature, relatively little research effort has been devoted to segmentation evaluation as compared to the development of segmentation algorithms [63]. There is some fundamental research about unsupervised segmentation evaluation using intra-region homogeneity and inter-region disparity to access the segmentation result. However, no metric could handle well the semantic relationships present in a complex scene. Thus, researchers rarely use unsupervised evaluation compared to supervised methods. For our study, the semantic relationships are not necessary to be considered which makes it possible for us to incorporate unsupervised evaluation into our segmentation algorithm.

The next problem is which metric to use. An extensive evaluation of unsupervised evaluation metrics is presented in Zhang et al.'s survey [64]. In one of Zhang et al.'s experiments which examines the performance of the metrics on segmentation produced by the same algorithm with varying numbers of segments, the best performing metric, Q [65] is selected for our algorithm.

Q is defined by

$$Q = \frac{\sqrt{R}}{10000(N \times M)} \sum_{i=1}^{R} \Big[\frac{e_i^2}{1 + \log A_i} + (\frac{R(A_i)}{A_i}) \Big] \tag{6.2}$$

where $N \times M$ is the size of the image, R is the number of regions, A_i and e_i^2 are the area in number of pixels and the squared color error of the ith region v_i, respectively. The squared color error of the ith region v_i is defined as

$$e_i^2 = \sum_{p \in v_i} \big(C(p) - \hat{C}_i \big)^2 \tag{6.3}$$

where $C(p)$ denotes the value of pixel p and $\hat{C}_i$ is the average value of ith region. $R(A_i)$ represents the number of regions that have an area equal to A_i[65]. Since $R(A_i)/A_i$ typically has a very small value as compared to the first term in the summation[66], $R(A_i)$ is fixed as 1 during the implementation. And 10000 is replaced by 1000 for display. Lower Q value means better segmentation quality.

Then, we will demonstrate how to incorporate Q into our algorithm. The framework of the algorithm is shown in Fig. 6.11. The roof sample image was segmented into superpixels using the SLIC algorithm. The following region merging process was divided into two stages for computational efficiency. Pre-region-merging stopped when the number of regions was equal to 25. 25 was selected empirically to provide sufficient regions for the next stage. During the post-region-merging, the quantitative score Q was computed and updated during each merging step. The merging stopped when an empirically selected sim-

Figure 6.11: Framework of self-tuning segmentation algorithm.

ilarity threshold of 0.4 was reached. The series of post-region-merging steps were compiled into a score list. The selected segmentation result corresponded to the step with minimal Q value.

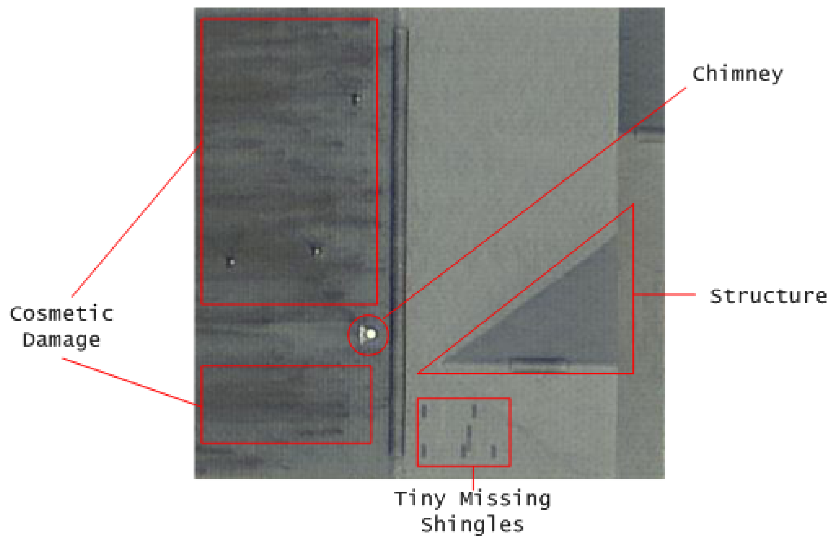


Figure 6.12: Typical impaired roof.

6.3.2 Results

6.3.2.1 Segmentation Results

In this section, we will demonstrate the segmentation results of our proposed algorithm on manually extracted roof sample images from the one-inch-resolution Pictometry data set.

The segmentation algorithm is divided into steps of forming the SLIC superpixel, pre-region-merging and post-region-merging segmentation. To demonstrate the results of each step, a typical impaired roof sample image shown in Fig. 6.12 is selected as an example.

As shown in Fig. 6.12, the roof contains 5 tiny missing shingles, a cosmetic damage region, a white chimney and a triangular "structure". An ideal segmentation would be able to depict the general boundary of the cosmetic damage area. Meanwhile, it would also isolate the tiny missing shingles in detail.

The SLIC superpixel algorithm is applied in the CIELAB color space. The nominal size of the superpixel is set to 15×15. To isolate tiny missing shingles, the compactness, which control the ability of boundary adherence, was set to 7. As shown in Fig. 6.13, the boundaries of the five tiny missing shingles are well depicted.

Pre-region-merging processing started from the SLIC superpixel result. Color-HED histograms were computed to represent each region. The Bhattacharyya coefficient was

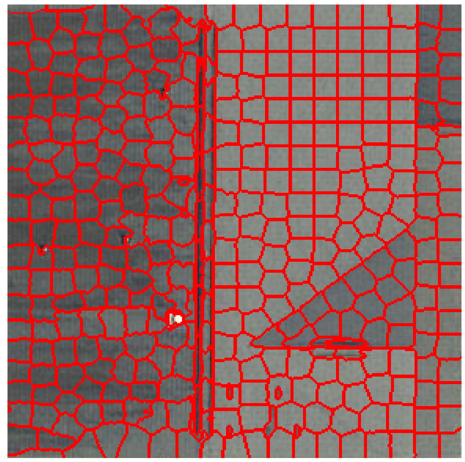


Figure 6.13: SLIC Superpixel result.

used to measure the similarity between all adjacent region pairs. The most similar connected regions were merged iteratively until the number of regions was equal to 25. 25 is selected because we assume that the final segmentation result should contain no more than 25 regions. The roof sample image shown in Fig. 6.12 is a relatively complex roof scene containing tiny missing shingles, cosmetic damage, chimney and "structure". As shown in Fig. 6.14, the pre-region merging result is still an overly-segmented result.

Post-region merging segmentation started from the pre-merging result. Unlike pre-merging processing, the unsupervised segmentation evaluation score, Q was computed and updated during each step. The merging steps were quantified by the Q score as shown in Fig. 6.14. Like conventional region merging segmentation approaches, the merging stopped when the preset similarity threshold of 0.4 was reached. The result of the last step is also shown in Fig. 6.14. The preset threshold was fixed to 0.4 because it gave satisfactory performance on most images that were used for tuning. However, for this specific roof, the result achieved by the conventional method is not perfect. Only 2 tiny missing shingles are isolated. The result of our proposed initial algorithm corresponds to the step with minimal Q value. At this step, all five tiny missing shingles are isolated. The boundary of the cosmetic damage area is also depicted.

To better illustrate the results of proposed algorithm, additional representative results are provided in Fig.6.15. The proposed modified algorithm is compared to the well-

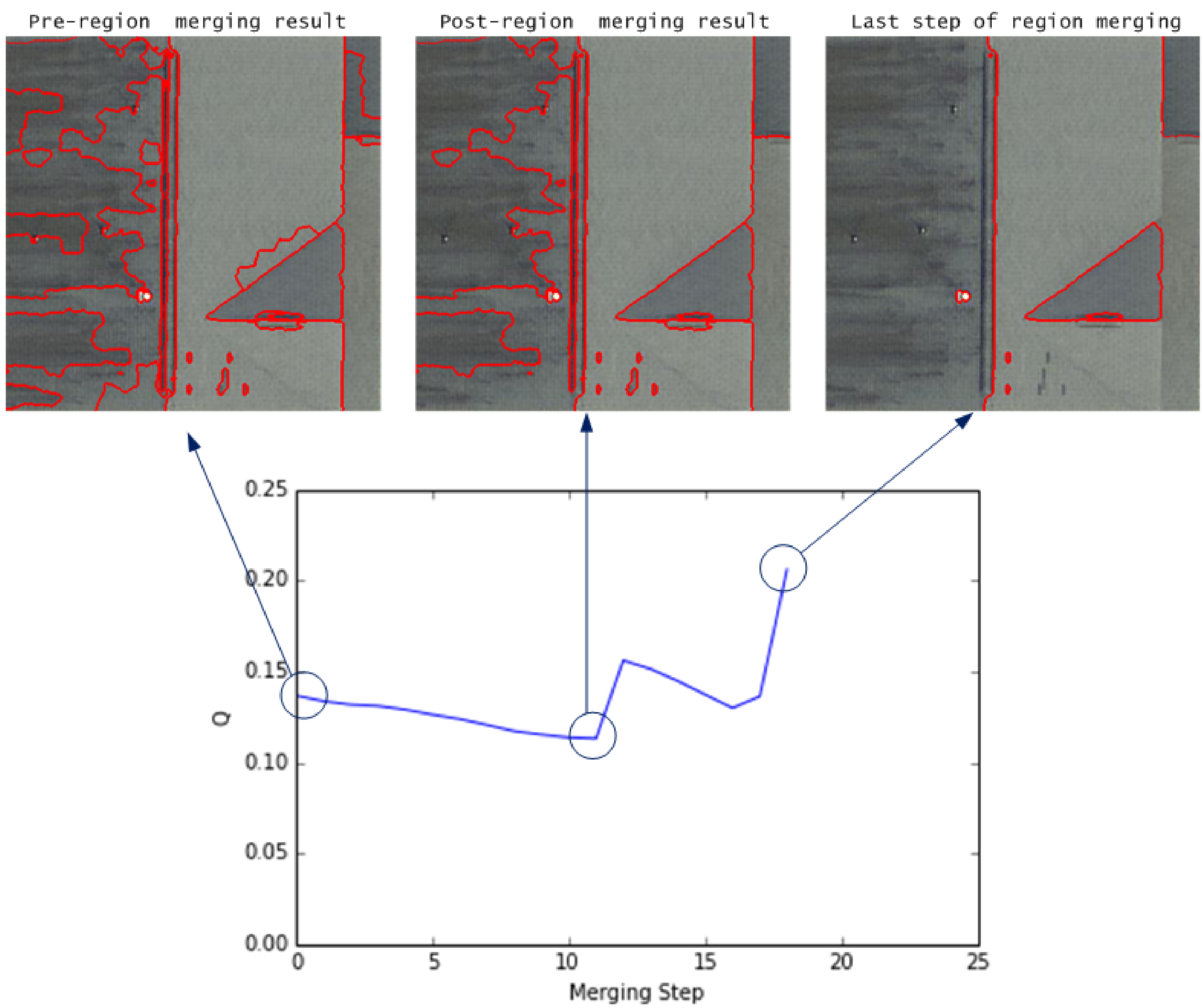


Figure 6.14: Pre and post-region-merging results.

known compression-based texture merging (CTM) method [67]. Human segmentation truth boundary maps are also provided. To fairly compare these methods, the original NCuts superpixel method in the CTM software was replaced by SLIC for better performance. For CTM, we need to adjust the threshold γ for a satisfactory segmentation. We ran CTM with parameters γ chosen at intervals in $[2, 7]$ and found that $\gamma = 5$ gives a good overall performance.

A “texture” roof is shown in the first row of Fig.6.15. The CTM algorithm has an over-segmentation problem around the white chimney. CTM produces a sinuous edge around the ridge. The modified algorithm produces a clear and reasonable result.

An intact roof example is shown in the second row of Fig.6.15. The modified algorithm produces a clearer boundary map compared to the CTM algorithm. CTM generates weird boundaries around the white object on the roof.

A “structure” roof is shown in the third row of Fig. 6.15. CTM fails to isolate the cosmetic damage area and produces a weird boundary around the chimney. The modified algorithm depicts the boundary of the cosmetic damage area better.

A roof with cosmetic damage and a missing shingle is shown in the fourth row of Fig. 6.15. Both algorithms isolate the missing shingle well. Our algorithm produces a better shape of the cosmetic damage area.

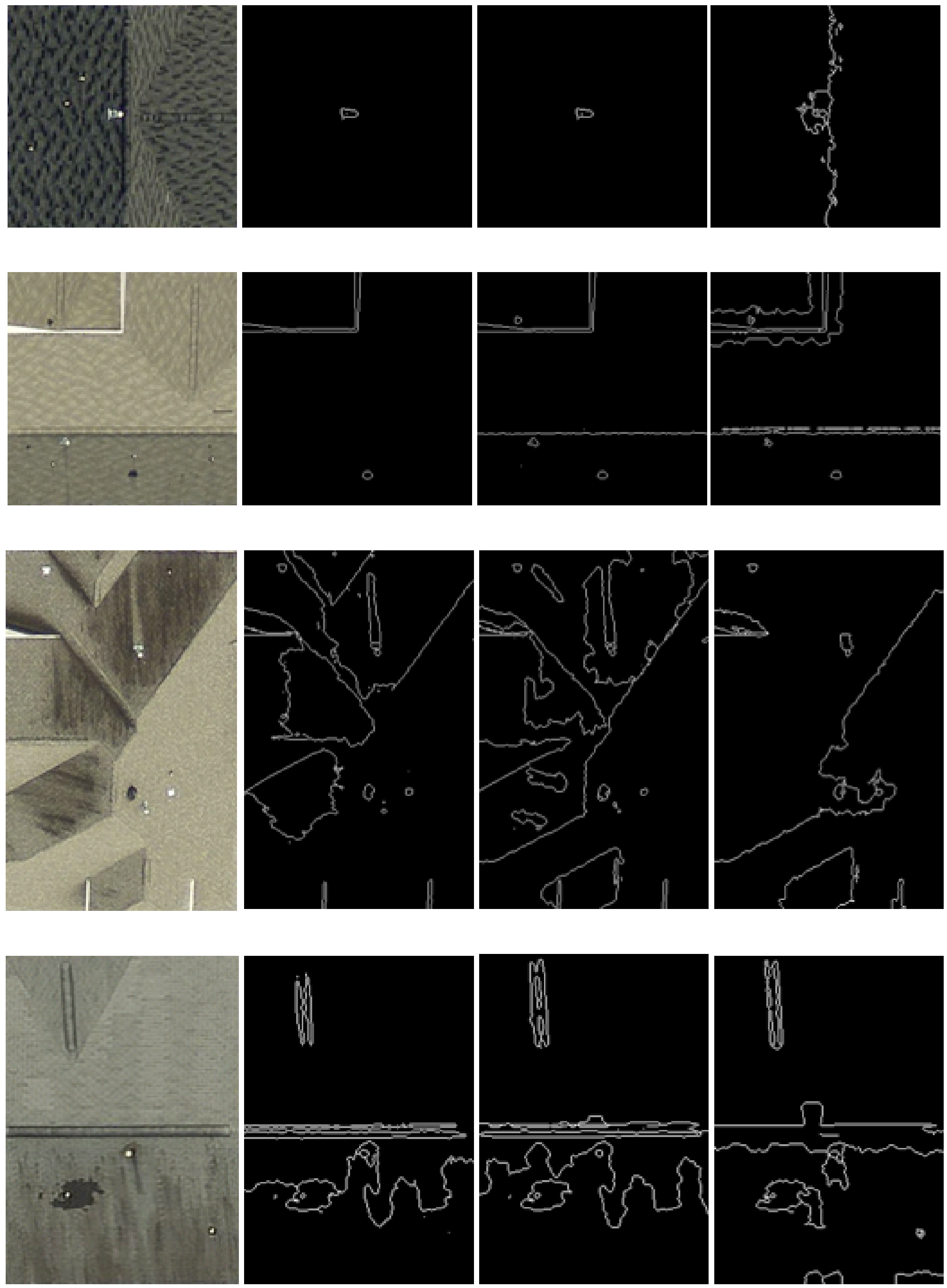
In summary, the proposed algorithm produced better visual results compared with the CTM method. The CTM algorithm suffers an over-segmentation problem and produces sinuous boundaries around ridges and chimneys. Meanwhile, CTM is not robust. It isolated cosmetic damage areas on some data and ignored them on other data.

6.3.2.2 Discussion

In this section several topics are discussed including the implementation and limitations of Q and failure cases of our algorithm.

6.3.2.2.1 Implementation of Q

Another topic we want to discuss is that whether performing an unsupervised segmentation evaluation for each step of the merging processing is time consuming. A similar method was used by Pichel *et al.* to adjust the parameters of the weight function of the edges of the RAG[68]. However, the implementation detail about how they assess the segmentation quality at each iteration is not provided clearly in their paper.



(a) (b) (c) (d)

Figure 6.15: Segmentation results by human, proposed algorithms and CTM: (a) original image, (b) human segmentation, (c) proposed algorithm, (d) CTM Result.

For our study, to save time, the initial algorithm was divided into pre- and post-region-merging and only post-region-merging processing was evaluated by Q. Another trick is that Q was only computed on the pre-region-merging result once.

To explain how it works, Q is rewritten as

$$Q = \frac{\sqrt{R}}{10000(N \times M)} Q_{sum} \tag{6.4}$$

where

$$Q_{sum} = \sum_{i=1}^{R} Q_{region} \tag{6.5}$$

and

$$Q_{region} = \frac{e_i^2}{1 + \log A_i} + (\frac{R(A_i)}{A_i}). \tag{6.6}$$

At the beginning of post-region-merging, Q was calculated on the pre-region-merging result. For each region, Q_{region} was computed and stored. Then, Q_{sum} was achieved and stored by summing up all the Q_{region}. Q for per-merging result was obtained by $\frac{\sqrt{R}}{10000(N \times M)}$ times Q_{sum}.

Now, assuming that region a and b were merged into region c, we do not need to compute Q again on the new merged result. The only thing we need to do is calculate and store the value of Q^c_{region} for new region c. Since Q^a_{region} and Q^b_{region} for region a and b were already calculated and stored, we get

$$Q_{offset} = Q^c_{region} - (Q^a_{region} + Q^b_{region}). \tag{6.7}$$

Since Q_{sum} is stored, the new $Q_{sum} = Q_{sum} + Q_{offset}$. Finally, we get Q by $\frac{\sqrt{R-1}}{10000(N \times M)}$ times Q_{sum}.

The computational complexity did not increase much. For each merging step, we just need to compute Q_{region} for the new region. All other operations are simple and fast. In summary, the merging processing is monitored by Q on the fly. We ran the proposed algorithm on a computer with Intel Core i7-4770 CPU and 8G RAM. The execution time to segment a test image of size 320×280 is 13.5 seconds.

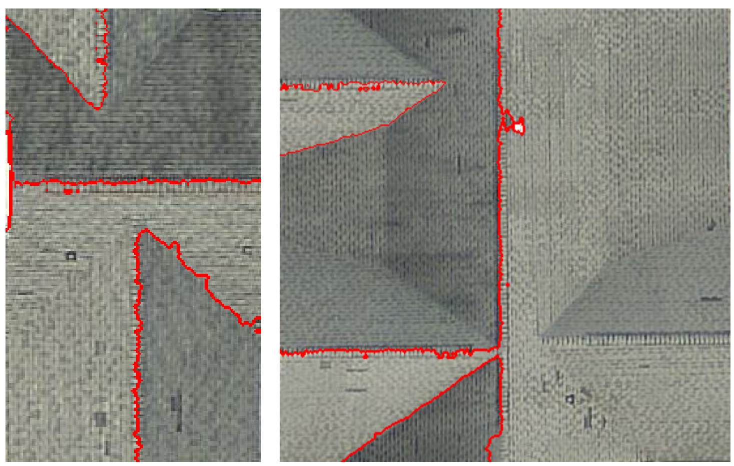
6.3.2.2.2 Limitations of Q

Both initial and modified algorithms rely on the assumption that the optimal segmentation is the one that gives the best Q score. However, Q is designed for natural scene image segmentation evaluation. Different from natural image segmentation, roof segmentation values small regions like chimneys or even tiny regions like missing shingles. During the study, we found that Q was designed to be insensitive to these small regions. For example, when a tiny missing shingle was merged into the background, we expect an increase in Q. However, we had a opposite result sometimes. When a tiny missing shingle is merged, $\sum_{i=1}^{R} \frac{e_i^2}{1+\log A_i} + \left(\frac{R(A_i)}{A_i}\right)$ rises because of the increase of intra-region diversity. The problem is that $\frac{\sqrt{R}}{10000(N\times M)}$ declines. The increase of $\sum_{i=1}^{R} \frac{e_i^2}{1+\log A_i} + \left(\frac{R(A_i)}{A_i}\right)$ is not enough, *i.e.*, the penalty for merging tiny missing shingle is not enough compared with the decrease of $\frac{\sqrt{R}}{10000(N\times M)}$. It results in a drop in Q.

To solve this problem, we tried to increase the penalty for this situation. During the region merging when there are less than 20 regions left, if any region of less than 100 pixels is merged, the penalty is increased several times. This method works for some data. However, regions with less than 100 pixels are not always missing shingles. It could be a tiny object in the complex texture background. Thus, it would produce an over-segmentation result sometimes. The limitation can be explored further in the future.

6.3.2.2.3 Failure cases

A particular case that our algorithm has trouble with is tiny missing shingles on a "textured" roof as shown in Fig 6.16. The SLIC compactness parameter is identified based the the roof texture. Coarse superpixels are generated for this "texture" roof, and the final result will miss these tiny missing shingles. Future work will explore segmentation on these "textured" roofs.



(a) Failure case 1 (b) Failure case 2

Figure 6.16: Failure cases

6.3.3 Conclusion

As a step of roof condition assessment, the roof segmentation algorithm began with an over-segmentation result yielded by SLIC superpixel method. Our proposed Color-HED histogram features were extracted to represent each superpixel. The region merging process merged the most similar adjacent regions iteratively. An unsupervised evaluation metric Q was incorporated into the merging process to select the optimal result.

The deep learning technique is investigated, we used a pre-trained HED network as a fixed feature extractor. Early output of HED network is extracted from each superpixel and combined with RGB information to construct a novel feature: Color-HED histogram for region representation.

6.4 Roof Segment Classification

After roof segmentation, roof images were divided into homogeneous regions. To obtain a overall roof condition report for the specific roof image, roof segment classification should be applied. Then, the roof condition assessment can be performed based on the area of cosmetic damage or missing shingles, if there are any.

6.4.1 Approach

In this study, roof segment classification is performed in two ways: traditional method using hand-crafted feature and deep learning method. We will first introduce the data we used for classification and then describe the two methods respectively.

6.4.1.1 Data

For this study, we only consider those segments with an area larger than 400 pixels. The reason is that the average area of our residential roof data is around 60000 (200×300) pixels. Those segments with area less than 400 pixels take less than 0.67 percent of the whole roof area and thus can be ignored in this study. Furthermore, those tiny segments also may not be able to provide enough information for classification.

Our proposed segmentation algorithm was performed on 274 roof images provided by Pictometry International. 1107 roof segments were generated and used for this study. Typical roof segment images are shown in Fig. 6.17.

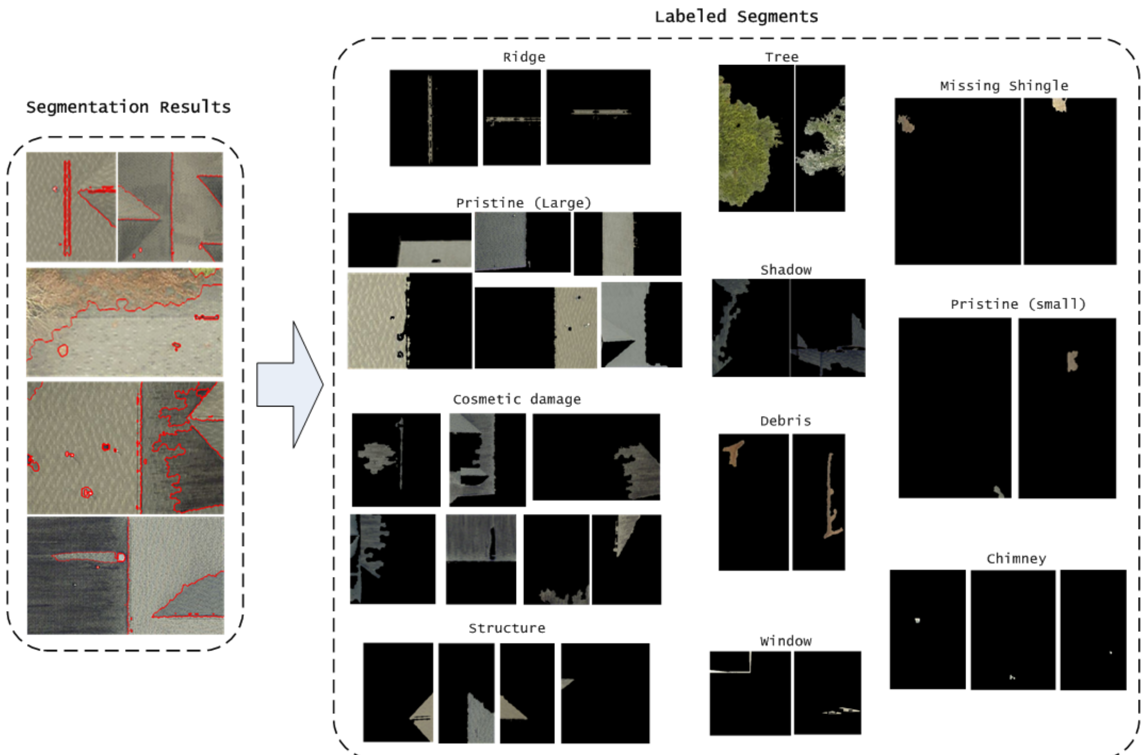


Figure 6.17: Labeled roof segment images.

The 1107 roof segments can be divided into 11 classes including pristine regions (large), cosmetic damage regions, "structure" regions, ridges, tree regions, shadow, debris regions, windows, missing shingles, chimneys and pristine regions (small).

For roof condition assessment, we are more concerned if the segment is intact or not. Thus, pristine regions (large and small), chimney, structure, shadow, tree and window can be merged into one class: intact segments. Cosmetic damage and missing shingle are merged into one class: impaired segments. We keep the ridge class because its shape is special. So, there are three classes for experiment: intact region, impaired region and ridge.

6.4.1.2 Roof Classification via Hand-crafted Features

Feature extraction is an important step of roof segments classification. A traditional method is to design hand-crafted features. The performance of classification relies on how well the designed features represent the image segments.

6.4.1.2.1 Feature Extraction

The first thing for hand-crafted feature design is to understand how does our brain recognize the roof segments. As shown in Fig. 6.17, we recognized pristine regions because of their uniform texture and large area. Cosmetic damage areas are recognized for their irregular shape and random texture. Meanwhile, cosmetic damage areas usually have darker color compared to their neighbors. Structure segments have triangular shape while ridges have slender shape. Trees are green, debris areas are brown and windows are white. Chimneys can be recognized because of their white color, tiny area and round shape.

The next step is to transform human perceptions into computer vision techniques. As shown in Figs. 6.18- 6.19, different features are designed.

For our color feature, we used 16 bins color histogram which is also used in our incremental learning segmentation.

Gabor filter is a Gaussian kernel function which is modulated by a sinusoidal plane wave [69]. It is widely used as a texture discriminator [70]. Gabor filter [71] was adopted to extract texture feature from each segment. The spatial frequency of the harmonic function was set to 0.25 $cycle/pixel$. Orientation was set between $[0, 150]$ degrees in steps of 30 degrees. Standard deviation of Gaussian function was set to 2 and 3. Thus, 12 Gabor filters were generated and applied to each roof image. The mean and variance of the filtered roof segments were used as features for classification.

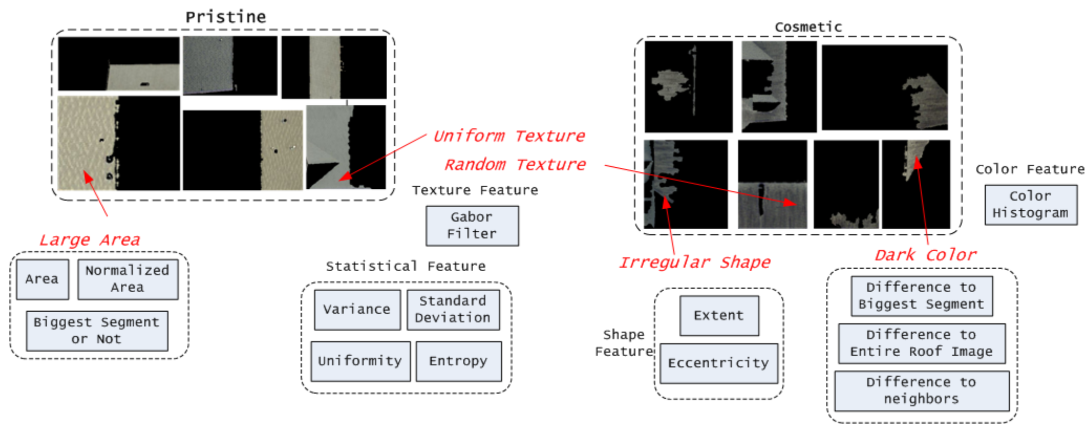


Figure 6.18: Features for pristine and cosmetic damage segments

Several statistical indices were calculated to represent the statistical features inside each segment. Statistical features included variance, standard deviation, median, maximum, minimum, range, uniformity and entropy.

We also considered the location information for each image segment. Normalized X and Y (row and column indices) means were computed. The number of pixels located on the edge of roof image were also calculated for each segment.

Shape features, extent and eccentricity were also considered. Extent is the ratio of pixels in the segment to pixels in the total bounding box. Bounding box is the smallest rectangle containing the segment. The eccentricity of the ellipse that has the same covariance matrix as the segments was also used as one of our shape features.

For each segment, we designed two sets of similarity features including neighbor similarity and biggest segment similarity. The Bhattacharyya coefficient was adopted to measure the similarity of color histograms of two regions. We went through all the adjacent segments for the segment being processed. The maximal and minimal similarity between the current segment to its neighbors were used as features for classification as well as the number of its neighbors. For biggest segment similarity, we calculated the area of the segments in the image by pixel. Normalized area was obtained by ratio of segment area to the image area. The similarity of the current segment to the biggest segment in the image was calculated. Is the current segment the biggest segment in the image or not was also added as the feature for classification.

So, color features, texture features, statistical features, shape features and similarity

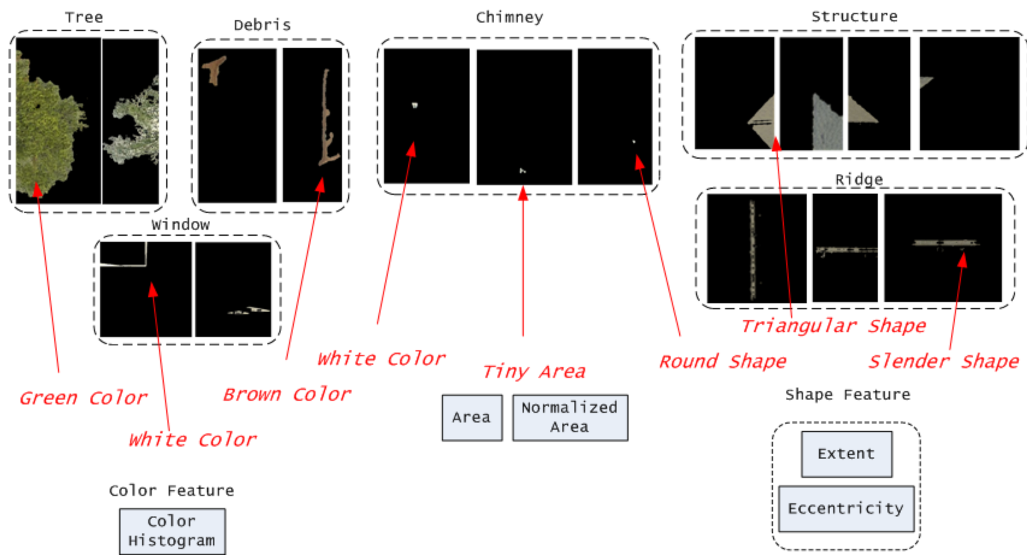


Figure 6.19: Features for tree, debris, window, chimney, structure and ridge segments

features were extracted from each segment. Before classification, feature scaling was applied to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges [72]. Each attribute was linearly scaled to the range $[-1, +1]$.

6.4.1.2.2 Model Selection and Performance Estimation

Support vector machine (SVM) as a powerful machine learning technique for classification [72] is selected for this study.

With a training set $(x_i, y_i), i = 1, ..., N$ where attributes $x_i \in R^n$ and class label $y \in \{1, -1\}^N$, SVM solves the solution of the optimization problem[72]:

$$\min_{w,b,\xi} \quad \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{N} \xi_i$$

$$subject\ to \quad y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i, \ \ \xi_i \geq 0 \ \ for i = 1, ..., N \tag{6.8}$$

The original input attributes x_i are mapped into a higher dimensional feature space by function ϕ. SVM tries to find a linear separating hyperplane with the maximal margin in the higher dimensional space. C is the regulation parameter which trades off training error against model complexity. A small C makes the decision surface simple and smooth, while a large C aims at classifying all the training examples correctly.

Real-world classification problems involve data that can only be separated using a non-linear discision surface[73] and thus kernel function is defined as $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$. Different kernels were considered in our study. An extremely popular kernel is the radial basis function (RBF): $K(x_i, x_j) = exp(-\gamma \parallel x_i - x_j \parallel^2), \gamma > 0$ were finally selected by its performance. γ can be seen as the inverse of the RBF kernel width. When γ is very small, the model cannnot capture the complexity of the the data. As γ increases, the number of support vectors increases and a narrower support region is produced, which leads to a complex decision surface [73].

Since a SVM classifier with RBF kernel is selected, the next problem is to identify good parameters (C, γ) for our problem.

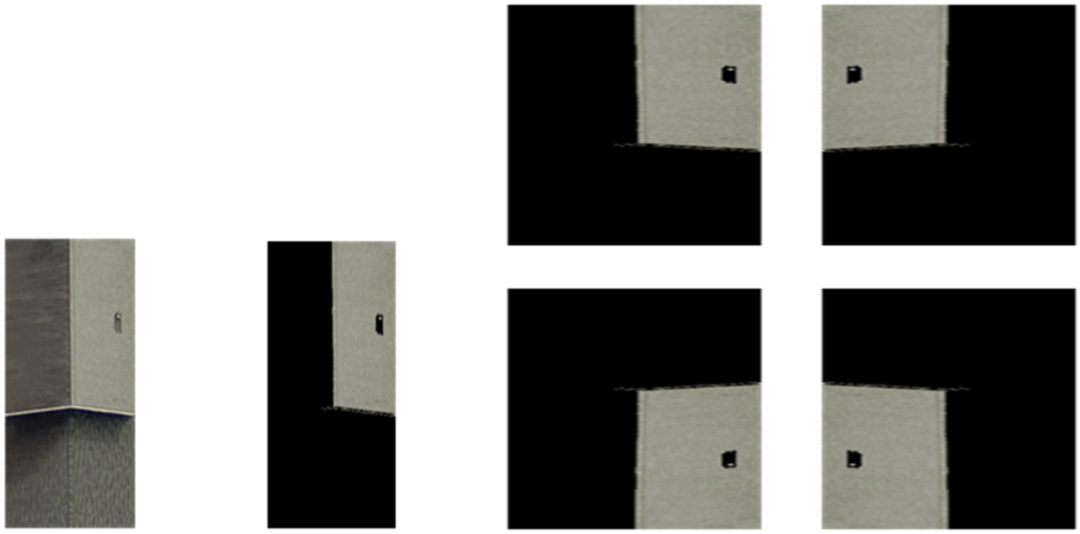
$k-$ folder cross-validation is used. Our data set was divided into training and testing sets. The training data is partitioned into k equal sized folders. The folds preserves the percentage of samples for each class. A model is trained using $k - 1$ folds and validated on the remaining single fold. The cross-validation process is repeated k times. Each of the k folds is used once as the validation set. The performance estimation is the average of k results obtained in the loop.

Grid-search on C and γ is performed using cross-validation. Exponentially growing sequences of C and γ values are tried and the one with the best cross-validation score is picked. Then, with the best (C, γ), the whole training set is used for training again to generate the final classifier. Performance is achieved by applying final classifier on the test set.

6.4.1.3 Roof Classification via Deep Learning

In recent years, CNNs have produced stellar results and their capabilities are illustrated in the ImageNet Large Scale Visual Recognition Challenge [74]. The performance is now close to humans. Thus, in this study, deep learning techniques are also investigated for roof segments classification.

In 2015, Deep Residual Networks (ResNets) took the deep learning world by storm and won the ImageNet competition with a 3.57 % error score [75]. By adding skip connections that bypass a few convolution layers and learning a residual mapping, ResNet ensures a fluent information flow, allowing neural networks that are over 100-layers deep to be



(a) (b) (c)

Figure 6.20: Training data augmentation: (a) original roof image, (b) roof segments with background (c) resized segments after horizontal and vertical flipping.

effectively trained [75]. Thus, ResNet is selected for our study. Actually, before ResNet is adopted for roof segments classification, VGG net [58] and GoogLeNet [76] were also investigated. However, the result was not comparable with ResNet.

After roof segmentation, roof images were segmented into homogeneous regions. 683 segments are used for training and 424 segments for testing. As shown in Fig. 6.20, for training data, all the roof segments with background are interpolate to 321×321 following horizontal and vertical flipping, leading to an augmented training set that is a factor of 4 larger. Note that ResNet is fully convolutional and the implementation allows inputs to be any size.

Since we have a small mount of training data, learning a full deep representation from scratch can be ineffective and time consuming [77]. Instead, it is common to pre-train a base network on a very large dataset like ImageNet, which contains 1.2 million images with 1000 categories [56]. Earlier features of a network have sufficient representational power and generalization ability. Thus, we can keep some of the earlier layers fixed for a target network, the higher-level portion of the network are then randomly initialized and trained toward the target task [56].

For our study, we use the ResNet with 50 layers pre-trained on ImageNet as a starting point. We replace the 1000-way ImageNet classifier in the last layer with a classifier having as many targets as the number of roof segment classes. The models are trained for up to 5000 iterations. Then, we run back propagation on the network to fine-tune all layers for 1000 steps.

6.4.2 Results

In this section, we will demonstrate the roof segment classification results. So, after roof segmentation, roof images were segmented into homogeneous regions. Our data was divided into training and testing sets. 683 segments are used for training and 424 segments for testing.

For traditional method, hand-crafted features: color, texture, statistical, shape and similarity features were extracted to represent each roof segment. $k-fold$ cross-validation was applied on the training set to identify the best parameter pair for SVM classifier with RBF kernel. The performance of trained classifer was test on the test set. Through $4-fold$ cross-validation, $C = 1328.8$ and $\gamma = 0.00264$ were found to be the optimal parameter pair for the SVM with RBF kernel on our training data. The performance is measured by applying trained SVM classifier on training set. The overall accuracy is 0.86.

For deep learning method, data augmentation is applied with the training data. All the roof segments with background are interpolated to 321×321 following horizontal and vertical flipping. We used the pre-trained ResNet with 50 layers as initialization. The 1000-way ImageNet classifier in the last layer was replaced with a 3-class classifier. RMSprop with a batch size of 32 is used. The learning rate starts from 0.01 and a weight decay of 0.00004 is selected. The models are trained for 5000 steps. Then, we fine-tuned all the layers for another 1000 steps with learning rate 0.001 and weight decay 0.00004. The experiments were run using a single NVIDIA GeForce GTX 1080Ti GPU and implemented with tensorflow. This fine-tuning process took approximately 4 hours. The final accuracy is 0.91.

The confusion matrix of segments classification via ResNet and traditional method are shown in Table 6.4 and 6.5. From the confusion matrix, we can see that 17 intact segments are misclassified as impaired region by ResNet compared to 33 misclassification by tradition method. 21 impaired segments are misclassified as intact region by ResNet compared to 27 misclassification by traditional method. For ridge class, there is no error for ResNet while 4 of 21 ridges are misclassified.

Table 6.4: Confusion Matrix for Roof Segments Classification via ResNet

Class	Intact	Impaired	Ridge
Intact	287	17	0
Impaired	21	78	0
Ridge	0	0	21

Table 6.5: Confusion Matrix for Roof Segments Classification via Traditional Method

Class	Intact	Impaired	Ridge
Intact	271	30	3
Impaired	27	72	0
Ridge	3	1	17

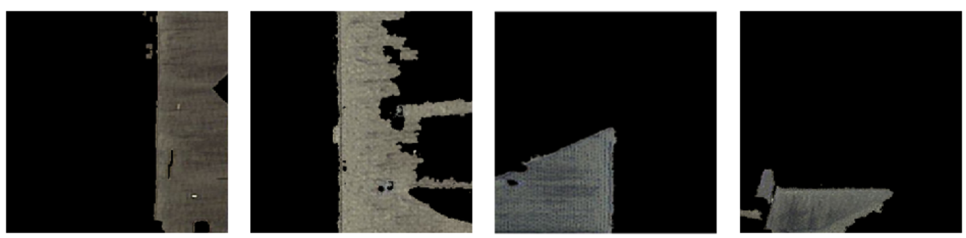
Some examples which are correctly classified by ResNet are shown in Fig 6.21-6.22. Fig 6.21 shows examples of intact segments which are correctly classified. Fig 6.21 (a) is a typical intact segment with large area and uniform texture. It is the most common pristine roof segment in our dataset. The texture of Fig 6.21 (b) is rough which caused some trouble for method using hand-crafted features. Fig 6.21 (c) is intact segment covered by shadow. It is a problem for the traditional method and even for a human. Fig 6.21 (d) is a small intact segment with regular shape. Its texture is similar to Fig 6.21 (b). Fig 6.21 (e) is a intact segment with random shape. These kind of segments are most difficult to identify. When we manually labeled it, we had to rely on the background information. So all of them are correctly classified by our fine-tuned ResNet.

Fig 6.22 shows examples of impaired segments which are correctly classified by ResNet. Fig 6.22 (a) is a typical impaired segment with cosmetic damage. Its shape is regular and takes half of the rooftop. Fig 6.22 (b) is an impaired segment with random shape. It suffers less cosmetic damage than Fig 6.22 (a). Fig 6.22 (c) and (d) are relative small segments with cosmetic damage. All of them are identified correctly by ResNet.

Some failure cases of our fine-tuned ResNet are shown in Fig 6.23-6.24. The intact

(a) (b) (c) (d) (e)

Figure 6.21: Examples of intact segments which are correctly classified by ResNet: (a)-(c) large intact roof segments, (d)-(e) small intact roof segments.



(a) (b) (c) (d)

Figure 6.22: Examples of impaired segments which are correctly classified by ResNet: (a)-(b) large impaired roof segments, (c)-(d) small impaired roof segments.

(a) (b) (c) (d)

Figure 6.23: Examples of intact segments which are misclassified as impaired segments by ResNet: (a) large intact roof segments, (b)-(d) small intact roof segments.

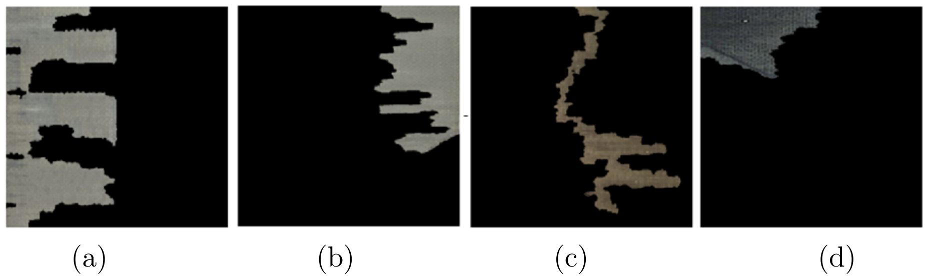


Figure 6.24: Examples of impaired segments which are misclassified as intact segments by ResNet: (a) large impaired roof segments, (b)-(d) small impaired roof segments.

segment shown in Fig 6.23 (a) is misclassified as impaired region. The texture of Fig 6.23 (a) and Fig 6.21 (b) are both rough compared to typical intact region shown in Fig 6.21 (a). However, the texture pattern of this region Fig 6.23 (a) is more irregular and our fine-tuned ResNet is misguided. The intact segment shown in Fig 6.23 (b) is also misclassified as impaired region. Its shape is irregular. Furthermore, it is covered by shadow which makes it remarkably similar to region with cosmetic damage. It is very difficult to identify even for human. The area of Figs 6.23 (c) is too small. Whether labeling it as intact region or impaired region is ambiguous. We manually label this segment as intact region while the fine-tuned ResNet label it as impaired region. The same goes for Figs 6.23(d), the area of the roof segment is too small to provide enough information for ResNet.

Fig 6.24 shows failure cases where impaired segments are classified as intact regions. Those four roof segments have one thing in common: the damage is minor. These regions are neither perfect nor seriously damaged. Take Fig 6.24 (a) for example, you cannot say whether it is wrong to label it as an intact roof segment.

In summary, the fine-tuned ResNet has done very well in roof segment classification with limited training data. It achieved better results compared to a traditional method using hand-crafted feature.

6.4.3 Conclusion

After the roof segmentation, roof images were divided into homogeneous regions. Pre-trained ResNet are fine tuned on the augmented dataset. The overall accuracy is 0.91 compared 0.86 achieved by traditional method.

The deep learning technique is investigated for roof segment classification. We use the pre-trained ResNet as an initialization and fine-tuned the network by two stages: keeping the earlier layers fixed and only training the last layer for 5000 iterations, followed by fine-tuning all the layers for 1000 steps.

Chapter 7

Summary and Future Work

In this chapter, the work in each topic is reviewed and possible future work is presented.

7.1 Debris detection

Debris detection is needed for effective debris removal and allocation of limited resources in times of natural disasters. In this dissertation, a debris detection algorithm under object based image analysis(OBIA) framework is proposed. The algorithm can be divided into three steps: interactive segmentation, feature extraction and thresholding. The interactive segmentation is performed using ENVI Feature Extraction module. Texture variance inside each segment is extracted to create the feature map. Thresholding is applied to get the debris detection result. According to the visual inspection, the performance is promising and robust on large scale data. Future work could be done to evaluate the algorithm quantitatively by comparing the result with expert-created debris map.

7.2 Building detection

Building detection is an important part of high resolution aerial imagery understanding and it also serves as pre-processing for other applications. A building detection algorithm using 3D point clouds is proposed in this thesis. Point clouds are first segmented into individual point clusters using Euclidean Clustering. Spectral and 3D structure features are extracted to represent each cluster. Then the extracted feature vectors are used to train the support vector machine (SVM) classifier. The accuracy of proposed algorithm is 89% with 7% false detection. Future work could be done to combine features from 2D image and 3D point cloud.

7.3 Roof condition assessment

7.3.1 Preliminary roof condition assessment

Inspection of roof condition is an important step of damage claim processing in the insurance industry. Currently, roof inspection is done by human which is expensive, time-consuming and unsafe. Thus, an automated roof condition assessment is of great interest. Preliminary roof condition assessment was investigated to assess roof condition using features derived from the entire rooftop. Two methods are proposed in this thesis. The first method is based on the texture classification using Gray Level Co-Occurrence Matrix (GLCM), Local Binary Pattern (LBP) and Gabor filter. Another method is based on the classical bag of words (BoW) method. Method 1 gives a relatively lower performance but produces stable results. Method 2 gives better results but suffers from problems caused by the random initialization in K-means.

7.3.2 Roof segmentation

Considering the complexity of residential roofs, roof condition assessment methods using features covering the entire rooftop were not observed to provide a promising result. Thus, a better approach for roof condition assessment is proposed to divide the task into two stages: 1) roof segmentation, followed by 2) roof segment classification.

A novel self-tuning segmentation method for roof condition assessment is proposed. The algorithm began with an over-segmentation result yielded by the simple linear iterative clustering (SLIC) superpixel method [53]. Our proposed deep learned feature: Color-HED histogram were extracted to represent each superpixel. A similarity measure was defined to measure the feature similarity. The region merging process merged the most similar adjacent regions iteratively. An unsupervised evaluation metric Q was incorporated into the merging process to select the optimal result.

7.3.3 Roof segment classification

After the roof segmentation, roof images were divided into homogeneous regions. Two methods are performed: traditional method using hand-crafted features and deep learning method. Pre-trained Deep Residual Networks (ResNets) [75] were fine-tuned on the augmented roof segments regions to yield final classification result for roof condition assessment. The algorithm was evaluated on imagery collected by Pictometry International. The overall accuracy achieved by ResNet is 91% compared 86% achieved by traditional method.

Future work could be done on roof segments classification using deep learning techniques. The obstacle is the number of data samples. There are 2000 labeled roof images available. Data augmentation should be applied. It is believed that the deep learning technique would provide better performance compared to hand-crafted features.

7.4 Specific Contributions

The following are specific contributions:

- Exploration of a variety of computer vision approaches applied to very-high-resolution (VHR) aerial image in the context of three specific applications.

- The frequency analysis of no-debris and debris areas is performed and it shows that the frequency spectrum of non-debris area is relatively structured compared to debris area. Thus, a Fast Fourier transform (FFT) based debris detection algorithm is proposed. Although the method did not provide promising results, it sets a direction for future research. A more effective way to use the frequency feature of debris area could be investigated.

- Considering the random texture observed in debris areas, another debris detection algorithm using a texture feature is proposed. Using the framework of object based image analysis (OBIA). Feature selection is investigated and texture variance is chosen to obtain the optimal performance. Compared with traditional pixel-based methods, the proposed method utilizes texture features more efficiently at the object level. Compared with traditional disaster assessment methods, only post-disaster imagery is needed to fulfill the debris detection with robust results.

- The building detection algorithm starts from a simple algorithm on a selected scene. Classical random sample consensus (RANSAC) algorithm is used to extract main ground plane and flat roofs from point clouds. However, it is found not work on a large scale scene. The classical RANSAC cannot detect extract individual roof plane from the large scale scene. Meanwhile, RANSAC cannot identify the main ground plane which is often not flat enough.

- The proposed robust building detection algorithm expand OBIA to the 3D point clouds. Euclidean cluster enabled cluster-level processing rather than the point-level processing. Spectral, spatial and 3D structure features are extracted for classification.

- The possibility to assess roof condition using features derived the whole rooftop is investigated. However, because of the complexity of residential roofs, these methods do not provide promising result. Thus, roof condition assessment using features for segmented roof regions is proposed.

- A pre-trained HED network is used as a fixed feature extractor. The early output of the HED network is extracted from each superpixel and combined with RGB information to construct a novel feature: Color-HED histogram for region representation.

- An unsupervised segmentation evaluation metric Q is incorporated into the region merging segmentation processing to select the optimal segmentation result. Q is calculated on the superpixel and then updated every merging step. It is simple and fast. The merging processing is monitored by Q on the fly.

- A labeled dataset for roof regions containing 2000 data samples are created for this research. It is also valuable for future research.

- After the segmentation, a pre-trained residual network (ResNet) is fine tuned on the augmented roof segments. The algorithm was evaluated on imagery collected by Pictometry International. The overall accuracy is 91% compared 86% achieved by traditional method.

- Through the investigation of deep learning techniques, traditional hand-engineered features are replaced with trainable multilayer networks for both roof segmentation and roof segments classification.

Bibliography

[1] G Grünthal. Conseil de l' europe. *European macroseismic scale 1998 (ems-98)*, 1998.

[2] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.

[3] GG Wilkinson. Computer vision in earth observation. `http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/ECVNET/Earth.Observation.html`, 1997.

[4] WS Hartley. Topographic mapping and satellite remote sensing: is there an economic link? *International Journal of Remote Sensing*, 12(9):1799–1810, 1991.

[5] Thomas Blaschke. Object based image analysis for remote sensing. *ISPRS journal of photogrammetry and remote sensing*, 65(1):2–16, 2010.

[6] Sonke Muller and Daniel Wilhelm Zaum. Robust building detection in aerial images. *International Archives of Photogrammetry and Remote Sensing*, 36(B2/W24):143–148, 2005.

[7] Samuel N Goward, Jeffrey G Masek, Darrel L Williams, James R Irons, and RJ Thompson. The landsat 7 mission: terrestrial research and applications for the 21st century. *Remote Sensing of Environment*, 78(1):3–12, 2001.

[8] GJ Hay and G Castilla. Object-based image analysis: strengths, weaknesses, opportunities and threats (swot). In *Proc. 1st Int. Conf. OBIA*, pages 4–5, 2006.

[9] JR Eastman. Introduction to remote sensing and image processing. *Idrisi for Windows User's Guide. Cap*, 3, 1995.

[10] Landsat 7. `en.wikipedia.org/wiki/Landsat_7`. Accessed: 2016-04-22.

[11] Spot-5 satellite sensor. http://www.satimagingcorp.com/satellite-sensors/other-satellite-sensors/spot-5/. Accessed: 2016-04-22.

[12] Ikonos satellite sensor. http://www.satimagingcorp.com/satellite-sensors/ikonos/. Accessed: 2016-04-22.

[13] Quickbird satellite sensor. http://www.satimagingcorp.com/satellite-sensors/quickbird/. Accessed: 2016-04-22.

[14] Worldview-1 satellite sensor. http://www.satimagingcorp.com/satellite-sensors/worldview-1/. Accessed: 2016-04-22.

[15] Worldview-2 satellite sensor. http://www.satimagingcorp.com/satellite-sensors/worldview-2/. Accessed: 2016-04-22.

[16] Geoeye-1 satellite sensor. http://www.satimagingcorp.com/satellite-sensors/geoeye-1/. Accessed: 2016-04-22.

[17] Worldview-3 satellite sensor. http://www.satimagingcorp.com/satellite-sensors/worldview-3/. Accessed: 2016-04-22.

[18] Reinhard Klette. *Concise computer vision*. Springer, 2014.

[19] Linda G. Shapiro. Special issue on purposive, qualitative, active vision. *CVGIP: Image Underst.*, 56(1), 1992.

[20] Scott Krig. *Computer Vision Metrics: Survey, Taxonomy, and Analysis*. Apress, 2014.

[21] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[22] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[23] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.

[24] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[25] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[26] Dingding Liu, Bilge Soran, Gregg Petrie, and Linda Shapiro. A review of computer vision segmentation algorithms. *Lecture notes*, 53, 2012.

[27] Wei Hu, Yangyu Huang, Li Wei, Fan Zhang, and Hengchao Li. Deep convolutional neural networks for hyperspectral image classification. *Journal of Sensors*, 2015.

[28] Fan Zhang, Bo Du, and Liangpei Zhang. Saliency-guided unsupervised feature learning for scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 53(4):2175–2184, 2015.

[29] Adriana Romero, Carlo Gatta, and Gustau Camps-Valls. Unsupervised deep feature extraction for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 54(3):1349–1362, 2016.

[30] Liangpei Zhang, Lefei Zhang, and Bo Du. Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 4(2):22–40, 2016.

[31] Francisco Escobedo Zoltan Szntoi, Orlando Misas and Scot Smith. Rapid debris estimation after hurricane damage in urban areas using high resolution aerial imagery. In *San Antonio 2009 ASPRS-MAPPS Fall Conference Proceedings*, November 16–19, 2009.

[32] Fumio Yamazaki, Daisuke Suzuki, and Yoshihisa Maruyama. Detection of damages due to earthquakes using digital aerial images. In *6th International Workshop on Remote Sensing for Disaster Applications*, 2008.

[33] Shunichi Koshimura Takumi Fukuoka. Quantitative analysis of tsunami debris by object-based image classification of aerial photos. In *9th International Workshop on Remote Sensing for Disaster Response*, September 15–16, 2011.

[34] S. Koshimura, S. Kayaba, and H. Gokon. Object-based image analysis of post-tsunami high-resolution satellite images for mapping the impact of tsunami disaster. In *Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International*, pages 1993–1996, 2011.

[35] Jae Sung Kim and Jie Shan. Hurricane damage assessment using remote sensing techniques – a case study in new orleans. In *American Society for Photogrammetry and Remote Sensing Annual Conference*, May 7–11, 2007.

[36] Randall S. Cerveny Soe W. Myint, May Yuan and Chandra P. Giri. Comparison of remote sensing image processing techniques to identify tornado damage areas from Landsat TM data. In *Sensors 8(2)*, pages 1128–1156, 2008.

[37] Anh Nguyen and Bac Le. 3d point cloud segmentation: A survey. In *Robotics, Automation and Mechatronics (RAM), 2013 6th IEEE Conference on*, pages 225–230, Nov 2013.

[38] G Sithole. Filtering of laser altimetry data using a slope adaptive filter. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/W4):203–210, 2001.

[39] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient RANSAC for point-cloud shape detection. In *Computer Graphics Forum*, volume 26, pages 214–226. Wiley Online Library, 2007.

[40] Beril Sirmacek and Cem Unsalan. Urban-area and building detection using sift keypoints and graph theory. *Geoscience and Remote Sensing, IEEE Transactions on*, 47(4):1156–1167, 2009.

[41] Shaohui Sun and Carl Salvaggio. Aerial 3d building detection and modeling from airborne lidar point clouds. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 6(3):1440–1449, 2013.

[42] Shi Pu, George Vosselman, and Commission Vi. Automatic extraction of building features from terrestrial laser scanning. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 33–39, 2006.

[43] Josep Miquel Biosca and José Luis Lerma. Unsupervised robust planar segmentation of terrestrial laser scanner point clouds based on fuzzy clustering methods. *{ISPRS} Journal of Photogrammetry and Remote Sensing*, 63(1):84 – 98, 2008. Theme Issue: Terrestrial Laser Scanning.

[44] Ellen M Rathje, Melba Crawford, Kyuseok Woo, and Amy Neuenschwander. Damage patterns from satellite images of the 2003 bam, iran, earthquake. *Earthquake Spectra*, 21(S1):295–307, 2005.

[45] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[46] Yani Ioannou, Babak Taati, Robin Harrap, and Michael Greenspan. Difference of normals as a multi-scale operator in unorganized point clouds. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 501–508. IEEE, 2012.

[47] Sarah Hanim Samsudin, Helmi ZM Shafri, and Alireza Hamedianfar. Development of spectral indices for roofing material condition status detection using field spectroscopy and worldview-3 data. *Journal of Applied Remote Sensing*, 10(2):025021–025021, 2016.

[48] B Sirmacek and Cem Unsalan. Damaged building detection in aerial images using shadow information. In *Recent Advances in Space Technologies, 2009. RAST'09. 4th International Conference on*, pages 249–252. IEEE, 2009.

[49] Dominik Brunner, Guido Lemoine, and Lorenzo Bruzzone. Earthquake damage assessment of buildings using vhr optical and sar imagery. *Geoscience and Remote Sensing, IEEE Transactions on*, 48(5):2403–2420, 2010.

[50] Markus Gerke and Norman Kerle. Automatic structural seismic damage assessment with airborne oblique pictometry© imagery. *Photogrammetric Engineering & Remote Sensing*, 77(9):885–898, 2011.

[51] C Bignami, M Chini, S Stramondo, William J Emery, and Nazzareno Pierdicca. Objects textural features sensitivity for earthquake damage mapping. In *Urban Remote Sensing Event (JURSE), 2011 Joint*, pages 333–336. IEEE, 2011.

[52] Laigen Dong and Jie Shan. A comprehensive review of earthquake-induced building damage detection with remote sensing techniques. *ISPRS Journal of Photogrammetry and Remote Sensing*, 84:85–99, 2013.

[53] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2274–2282, 2012.

[54] Saket Navlakha, Parvez Ahammad, and Eugene W Myers. Unsupervised segmentation of noisy electron microscopy images using salient watersheds and region merging. *BMC Bioinformatics*, 14(1):294, 2013.

[55] Bo Peng and Olga Veksler. Parameter selection for graph cut based image segmentation. In *The British Machine Vision Conference (BMVC)*, volume 32, pages 42–44, 2008.

[56] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, pages 3320–3328, 2014.

[57] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1395–1403, 2015.

[58] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[59] Bo Peng and David Zhang. Automatic image segmentation by dynamic region merging. *Image Processing, IEEE Transactions on*, 20(12):3592–3605, 2011.

[60] Thomas Kailath. The divergence and bhattacharyya distance measures in signal selection. *Communication Technology, IEEE Transactions on*, 15(1):52–60, 1967.

[61] Robert M Haralick and Linda G Shapiro. Image segmentation techniques. In *1985 Technical Symposium East*, pages 2–9. International Society for Optics and Photonics, 1985.

[62] Ranjith Unnikrishnan, Caroline Pantofaru, and Martial Hebert. Toward objective evaluation of image segmentation algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):929–944, 2007.

[63] Paulo L Correia and Fernando Pereira. Objective evaluation of video segmentation quality. *Image Processing, IEEE Transactions on*, 12(2):186–200, 2003.

[64] Hui Zhang, Jason E Fritts, and Sally A Goldman. Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260–280, 2008.

[65] M Borsotti, Paola Campadelli, and Raimondo Schettini. Quantitative evaluation of color image segmentation results. *Pattern Recognition Letters*, 19(8):741–747, 1998.

[66] Hui Zhang, Jason E Fritts, and Sally A Goldman. An entropy-based objective evaluation method for image segmentation. In *Electronic Imaging 2004*, pages 38–49. International Society for Optics and Photonics, 2003.

[67] Allen Y Yang, John Wright, Yi Ma, and S Shankar Sastry. Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding*, 110(2):212–225, 2008.

[68] Juan C Pichel, David E Singh, and Francisco F Rivera. Image segmentation based on merging of sub-optimal segmentations. *Pattern Recognition Letters*, 27(10):1105–1116, 2006.

[69] Linlin Shen and Li Bai. Mutualboost learning for selecting gabor features for face recognition. *Pattern Recognition Letters*, 27(15):1758–1767, 2006.

[70] Itzhak Fogel and Dov Sagi. Gabor filters as texture discriminator. *Biological cybernetics*, 61(2):103–113, 1989.

[71] Javier R Movellan. Tutorial on gabor filters. *Open Source Document*, 2002.

[72] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[73] Aravind Ganapathiraju, Jonathan E Hamaker, and Joseph Picone. Applications of support vector machines to speech recognition. *Signal Processing, IEEE Transactions on*, 52(8):2348–2355, 2004.

[74] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[75] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[76] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[77] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning*, volume 32, pages 647–655, 2014.