AUTOMATED DAMAGE ASSESSMENT FROM HIGH RESOLUTION
REMOTE SENSING IMAGERY

A Thesis

Submitted to the Graduate School
of the University of Notre Dame
in Partial Fulfillment of the Requirements
for the Degree of

Master of Science
in
Computer Science and Engineering

by

Jim Thomas

_________________________
Kevin Bowyer, Co-Director

_________________________
Ahsan Kareem, Co-Director

Graduate Program in Computer Science and Engineering
Notre Dame, Indiana
July 2010
AUTOMATED DAMAGE ASSESSMENT FROM HIGH RESOLUTION REMOTE SENSING IMAGERY

Abstract

by

Jim Thomas

Estimating the extent of damage caused by natural disasters is necessary for implementing effective recovery measures. Damage detection from high-resolution satellite or aerial imagery for post-disaster analysis has been a major research effort in the past decade. A careful analysis of images from before and after an event facilitates rapid detection and assessment of building damage. This work presents a first-of-its-kind system for automatic damage assessment. The proposed framework for damage estimation consists of three steps. First the pre-event and post-event images are registered automatically. A SURF-based feature extraction and matching technique is used for automatic image registration. Next, the objects of interests such as buildings are extracted from pre-storm images. A novel robust algorithm for building detection is proposed and evaluated. Lastly, change detection is performed and damage is classified using supervised learning algorithms. Relevant features that reflect the spectral properties of damaged buildings are identified and used to classify the damage level into various states.
To my parents
with much love
and gratitude
CONTENTS

FIGURES ................................................................. v

TABLES ................................................................. viii

ACKNOWLEDGMENTS ................................................... ix

CHAPTER 1: INTRODUCTION ........................................ 1
  1.1 Motivation ......................................................... 1
  1.2 Contribution ...................................................... 3
  1.3 Outline ............................................................ 4

CHAPTER 2: RELATED WORK ......................................... 5
  2.1 Earthquake Damage ............................................ 5
  2.2 Windstorm Damage ............................................. 7

CHAPTER 3: DATA ....................................................... 11

CHAPTER 4: IMAGE REGISTRATION .............................. 12
  4.1 Steps in Image Registration ................................... 12
  4.2 SURF-based feature Extraction .............................. 13
    4.2.1 Interest Point Detectors ................................ 14
    4.2.2 Feature Descriptors ..................................... 16
  4.3 Interest Point Matching ..................................... 18
  4.4 Results .......................................................... 22

CHAPTER 5: BUILDING DETECTION ............................ 28
  5.1 Approaches ....................................................... 28
  5.2 Algorithm Steps ................................................ 30
    5.2.1 Segmentation .............................................. 31
    5.2.2 Shape Measure .......................................... 37
    5.2.3 Measure of Estimated and Predicted Shadow (MEPS) ... 39
FIGURES

4.1 Features extracted from (a) before and (b) after disaster images shown in red circles. 

4.2 Matching between 10 features in (a) before and (b) after disaster shown in red lines. 

4.3 Registered (a) before and (b) after disaster images. Non-overlap area is filled with black pixels. 

4.4 Unregistered (a) before and (b) after disaster images. The images taken from different camera viewpoints and with lighting conditions. 

4.5 Registered (a) before and (b) after disaster images. The images taken from different camera viewpoints, overlap percentage and lighting conditions. 

5.1 A $4077 \times 4092$ NOAA image from Pensacola Florida. 

5.2 Result of maximum-likelihood classification with $M = 11$, $N = 3$. Classes included ocean, grass, vegetation, road, white roof, blue roof, dark roof, gray roof, pavement, land area and shadow. These are shown in different colors. Detection Percentage = 28% . . . . 

5.3 Result of k-means clustering followed by maximum-likelihood classification. $M = 11$, $N = 3$. Classes included ocean, grass, vegetation, road, white roof, blue roof, dark roof, gray roof, pavement, land area and shadow. These are shown in different colors. Detection Percentage = 32.3% . . . . 

5.4 (a) An example image with 3 bands (b) The segmented image . . 

5.5 (a) Segments classified as buildings in Figure 5.4 (b) Result of Canny detection followed by morphological dilation. . . . . 

5.6 (a) An image with 3 bands. (b) Detected buildings and roads are shown as green contours and the shadows are shown in white. The few red arrows placed on buildings show on which side the shadow is cast. . . . . 

v
5.7 Rectangular buildings found from segments classified as buildings in Figure 5.3 are outlined in green.

5.8 (a) A building (b) The arrow indicates $\vec{v}_{sh}$. The perimeter of the building which is visible to a line perpendicular to this vector is where we would expect the shadow to be present.

5.9 The segments classified as buildings are shown in white. The pixels along the perimeter predicted for shadows are marked in red.

5.10 (a) A parking lot (b) Fragments of the parking lot which were wrongly classified as buildings. The labeled numbers indicate the $MEPS$ values for each fragment. Note that $MEPS \leq 0.07$ for all fragments.

5.11 (a) A building (b) The labeled numbers indicate the $MEPS$ value for the segment classified as the building. Note that $MEPS = 0.96$ for the building.

5.12 (a) A few buildings (b) Segmented image shows shadows in yellow, roads in pink and buildings in different shades of green, pink and blue. This is an example of over-segmentation.

5.13 A before storm NOAA image from Pensacola, Florida.

5.14 Buildings extracted from Figure 5.13 is shown in white. $C_{th} = 0.4$, $M_{th} = 0.4$ and $A_{th} = 100$ pixels.

5.15 Another before storm NOAA image from Pensacola, Florida. The buildings hidden behind trees and variations in rooftops makes this a difficult example. $C_{th} = 0.3$, $M_{th} = 0.3$ and $A_{th} = 100$ pixels.

5.16 Buildings extracted from Figure 5.15 is shown in white. $C_{th} = 0.3$, $M_{th} = 0.3$ and $A_{th} = 100$ pixels.

6.1 Pre- and post-storm images (grayscale). QuickBird imagery from DigitalGlobe, Inc.

6.2 Damage regions, in red, manually marked from pre- and post-storm images in Figure 6.1.

6.3 The pictures show the comparison process for (a) ground truth (GT) and (b) change detection (CD). The different connected components in the change masks are computed and labeled by using different pixel intensities. Then the connected components are numbered and compared to obtain the quantitative measures.
6.4 A windstorm damage profile for damage metrics obtained from (a) simple differencing and (b) for statistical hypothesis testing. The damage metrics are based on our own assigned ratings for the Remote-Sensing Damage Scale suggested in Womble (2005) based on pre- and post-storm comparison of 16 buildings in our image dataset.

6.5 A grid is placed over (a) before event building image and (b) after disaster building. (c) Features are extracted for the building. (d) Damage measure is calculated for each cell of the grid.

6.6 Google Earth images of National Palace, Haiti from (a) before and (b) after 2009 Haiti earthquake.

6.7 Buildings extracted from (a) before and (b) after 2009 Haiti earthquake images shown in Figure 6.6.

6.8 (a) Absolute difference of the before and after images. (b) Result of Canny edge detection.

6.9 The first row shows a collapsed building and the corresponding false color images for edge density, V Histogram and H means. Notice that the edge density values are significantly higher in this case. The second row corresponds to a partially damaged building with a cavity in the rooftop. V histogram indicates a significant change in the cavity area and H means shows a minor damage on the roof.

6.10 Performance of RandomForests with different number of trees for (a) 3-scale damage metric and (b) 4-scale damage metric.

6.11 Performance of AdaBoost with different number of iterations for (a) 3-scale damage metric and (b) 4-scale damage metric.
### TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>REGISTRATION SUCCESS AT RESOLUTIONS 50CM, 100CM, 200CM, 400CM AND 500CM</td>
<td>27</td>
</tr>
<tr>
<td>5.1</td>
<td>SOME OF THE COMPACTNESS MEASURES OF BUILDINGS IN FIG 5.5 (b) AFTER MORPHOLOGICAL DILATION</td>
<td>42</td>
</tr>
<tr>
<td>5.2</td>
<td>PERFORMANCE OF BUILDING DETECTION</td>
<td>54</td>
</tr>
<tr>
<td>6.1</td>
<td>AVERAGE REGION CLASSIFICATIONS FOR 6 PAIRS OF BEFORE-AND-AFTER IMAGES, THE AVERAGE OF REGIONS CLASSIFIED AS CORRECT, UNDER-SEGMENTED, OVER-SEGMENTED, MISSED AND NOISE WERE COMPUTED FOR EACH CHANGE DETECTION ALGORITHM EVALUATED</td>
<td>63</td>
</tr>
<tr>
<td>6.2</td>
<td>A 3-SCALE DAMAGE METRIC</td>
<td>75</td>
</tr>
<tr>
<td>6.3</td>
<td>A 4-SCALE DAMAGE METRIC</td>
<td>76</td>
</tr>
<tr>
<td>6.4</td>
<td>A CONFUSION MATRIX FOR 3-SCALE METRIC WITH RAND-FOREST, NUMBER OF TREES = 7</td>
<td>76</td>
</tr>
<tr>
<td>6.5</td>
<td>A CONFUSION MATRIX FOR 4-SCALE METRIC WITH AD-ABOOST, NUMBER OF ITERATIONS = 3</td>
<td>77</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

I would like to thank my advisors, Dr. Kevin Bowyer and Dr. Ahsan Kareem; without their guidance, support and encouragement this work would not have been possible. I would also like to thank the members of my committee, Dr. Christian Poellabauer and Dr. Soma Biswas, for their help and support in this process. I would also like to thank my labmates, friends and family for their suggestions and support during my graduate studies. I’m grateful to the Global Center for Excellence funded by MEXT, Japan, for providing the grant that allowed me to undertake this research.

And above all, my thanks to God... Who makes all things possible!
CHAPTER 1

INTRODUCTION

1.1 Motivation

In the event of a natural disaster such as hurricane or earthquake, estimating the extent of damage is necessary for implementing fast and effective recovery measures. Images of affected areas are easily obtained through satellite or aerial sensors. A damage assessment from such images can assist in providing loss estimates to benefit the federal government, the insurance industry, researchers, engineers, residents of high-risk areas, and policymakers involved in mitigation decisions.

The rapid detection and assessment of damage is essential for effective emergency-management efforts, as assessment of the geographic extent of relative levels of damage is of principal importance in prioritizing relief efforts and in estimating economic losses. Estimating the magnitude of losses helps in prioritizing the dispatch of emergency response teams and supplies. A reliable system for assessment of region-wide infrastructure conditions following disasters is therefore critical for effective immediate response and for long-term planning. Such a system should ideally include effective methods for rapidly assessing the extent and severity of damage to urban areas; for providing a detailed per-building analysis; and for
extending the results to an entire disaster-affected region.

For post-disaster damage analysis, engineers and scientists have employed remote-sensing technology in the form of aerial photogrammetric surveys or satellite imagery to obtain views of overall damage conditions in an affected region. With the rapid explosion of computer technology and digital-imaging technologies over the past two decades, remote sensing has progressed from a qualitative discipline (based on the manual interpretation of aerial photographs) to a quantitative discipline founded on the computer-automated analysis of multispectral digital images [Atkinson (1999)]. Prior to 1999, earth-observing satellites boasted spatial image resolutions of only 10-30 m or larger; such resolutions were generally adequate for many applications in which assessments could be made on a regional basis (e.g., agriculture, forestry, meteorology etc) but were neither able to resolve components of individual buildings nor to discern temporal changes (damage) in individual buildings.

Since 1999, a new generation of high-resolution commercial imaging satellites (with resolutions of 1 m or finer) has enabled the rapid and automated detection of physical changes in individual buildings by comparing pre-event and post-event images, as well as the rapid preservation of pre-cleanup damage scenarios to provide a permanent record of the damage for further analysis [Womble (2005)]. With such refined spatial resolutions, remote-sensing technology is presently experiencing a progression from regional, aggregated assessments of building damage to the assessment of damage to specific buildings.
1.2 Contribution

Work done in the past decade [Yamazaki (2001), Adams et al. (2004), Matsuoaka et al. (2004)] has demonstrated the effectiveness of modern remote-sensing technologies for the thorough and consistent assessment of earthquake-damaged infrastructure. Pre- and post-storm aerial images have also been used for a rapid and thorough assessment of damage following hurricanes. The first known opportunity for using this technology was with Hurricanes Charley and Ivan in 2004 [Womble (2005)].

The key objects of interest in such images are buildings, as their destruction directly impacts lives. Qualitative characterization of building damage can be done based on a classification of roof structure damage visible from satellite images. Certain visual signatures of damage could be used to categorize damage into categories such as no damage, removed tiles, removed decking, partially collapsed, fully collapsed etc. This work attempts at building a system capable of fine-grained damage analysis with minimal manual supervision. While we mainly focus on damage caused by hurricanes, most of what we present can easily be extended to earthquake or other forms of disasters.

We make minimal assumptions about the quality of data-set available. We do not assume that geo-referenced data is available or that the images have more than 3 bands (red, green and blue). We deal with the challenge of creating an automatic system that can work with any real world image repository; aerial or satellite. Part of this challenge deals with identifying objects of interest- mainly buildings. We introduce a novel algorithm that detects buildings of a broad range
of variations in shape, texture or color. Introducing this robust object extraction technique into the framework of damage detection has opened up new possibilities in analysis from high-resolution imagery. Finer details of damage at a per-building basis can now be captured. A highly accurate image registration is required for damage analysis from before-and-after event images. We introduce a reliable and fast automatic technique that computes a large number of landmark points. This system almost always guarantees to register images with a much higher precision than possible manually. Finally, we identify relevant features that are intuitive and useful for classification into qualitative states.

1.3 Outline

Chapter 2 describes some of the previous work that has been completed in related areas. Chapter 3 briefly discusses the sources and specifications of data used in this work. The steps used in automatic image registration, the proposed SURF-based feature detection for identifying landmark points and point matching algorithms are discussed in Chapter 4. The shortcomings of previous approaches and the proposed building detection algorithm is presented in Chapter 5. Chapter 6 elaborates on existing gray-level and proposed object-based change detection techniques. The features extracted, their relevance and the use of learning algorithms for damage classification is also discussed. Finally, Chapter 7 concludes the work that has been presented in this thesis.
 CHAPTER 2

RELATED WORK

Previous work in damage assessment from images vary in the nature of the disasters analysed (earthquakes, hurricanes, tornadoes, landslides etc), approaches to classification (pixel-based or object-based) and type of images used (Aerial TV, Optical, SAR etc.)

2.1 Earthquake Damage

Some of the earlier work takes a pixel-oriented approach where the damage map is created by a pixel-based analysis between before and after damage images. This is useful for a high-level analysis of the region. The use of optical images for the pixel-oriented approach was discussed in Yamazaki (2001). Pixel-based approach uses digital numbers (DN) of each band in the multi spectral imagery. Because the digitized values in the satellite images were different depending on the observation and surface conditions, they suggested the use of digital number (DN) normalization. The characteristics of the reflection of electromagnetic waves from the surface differ depending on material with which it is made. With proper thresholding of normalized DN, clouds vegetation etc were removed. Slight damage and no damage of buildings were selected from the images to characterize the DN in the damaged areas.
Yamazaki (2001) also discusses the use of color indices and edge elements from aerial television images to identify severely damaged buildings. Another alternative discussed is the use of SAR (Synthetic Aperture Radar) imagery and comparing the backscattering intensity images in the HH, HV, VH, and VV polarizations. The intensities of the co-polarization (HH, VV) were found to be larger than those of the cross-polarization (HV, VH), and the HH polarization intensity was largest. High-rise buildings were indicated as a cause of strong backscattering reflection whereas low-rise buildings, forests and ponds in parks were found to have weak backscattering reflection.

Other pixel-based approaches involve the use of edge-based measures to analyze textural dissimilarity. In Adams et al. (2004) damage was represented by a 9×9 pixel Laplacian edge detection filter. It was initially applied to each co-registered scene, followed by a 25×25 dissimilarity texture measure. The resulting images were differenced on a per-pixel basis and the mean standard deviation about the image mean computed. An average standard deviation was then plotted within a 200×200 cell window. Mapping these block statistics in intervals of 1 standard deviation highlighted areas of potential building collapse where textural change was consistently high.

In contrast to the pixel-based approaches, an object-based approach detects areas of interest such as buildings and then performs change detection. The object-based approach is discussed in Bitelli et al. (2004). The authors used object-based classification in eCognition 3.0 (a software by Definiens Imaging). The first op-
eration performed in eCognition is the segmentation of post-event image. This is followed by a definition of objects of interest. An analysis was done using the area that was classified as damaged. Based on accuracy in classifying built area into various damage states, object-based approach was found to be superior to the pixel-based approach.

Another object-based technique was proposed in Chen and Hutchinson (2005). In this approach the recognition of urban structures was obtained by performing morphological filtering and intensity thresholding, which is further optimized through a statistical procedure. By overlaying the recognized structures with the pre- and post-event images, three object-based change detection methods were presented. The change indices used include correlation analysis, principal component analysis and boundary compactness index. A boundary compactness index is defined by thresholding the local spatial variances of the pre- and post-event images, which effectively characterizes the structural change of urban structures due to seismic damage. The performance of change indices resulting from the three change detection methods was evaluated by means of a histogram-based classification approach. It was concluded that of the three change detection methods considered, the damage map based on boundary compactness change detection displays the best agreement with a manually prepared damage map Chen and Hutchinson (2005).

2.2 Windstorm Damage

Recent studies have investigated the use of modern remote-sensing technology for windstorm damage assessment. Studies Womble (2005), Womble et al. (2007)
have proposed remote sensing classification schemes and assessed their validity with actual field data. Womble (2005), Womble et al. (2007), Friedland et al. (2008), Friedland (2009) suggested the use of measures obtained from spectral bands for a per-building classification of windstorm damage.

In Womble et al. (2007) the qualitative characterization of building damage, structures within the Hurricane Charley and Ivan study areas were classified according to their roofing system (type of roof construction) rather than building occupancy because (quoted from Womble et al. (2007)) (1) post-storm conditions of roofing components are most distinguishable via overhead remote sensing and (2) because the type of roofing construction is also closely linked to the damage mechanisms of buildings, and thus roofs of a similar construction type tend to exhibit similar visible damage characteristics. The library of damage data consists primarily of low-rise buildings, enabling the qualitative analysis of four building (roofing) types: (1) Wood-Frame Roofs (nominally termed Residential, including single-family homes, some apartment buildings, and a few small offices buildings, and specifically those buildings constructed with wood-frame roofs covered with wood decking and either tile or asphalt shingles); (2) Metal Warehouses; (3) Built-Up Roofs; and (4) Manufactured Housing. Of these, the Residential category comprised the majority of the buildings; this category is selected for the case study described herein. Womble (2005) provides detailed descriptions of the visual damage characteristics of all four roofing categories. The results of the qualitative examination were expressed in the Remote-Sensing Damage Scale (RS-A to RS-D) for Residential Construction.
In Friedland et al. (2008), Friedland (2009) per-building analysis was completed using information previously collected in a structure database developed from detailed review of high definition video collected after Hurricane Katrina. Inventory information such as number of stories, construction types, and damage state were assigned to each building. The damage state was assigned according to a proposed Wind and Flood (WF) Damage Scale that was created to describe damage caused by the long duration flooding of New Orleans, Louisiana in the aftermath of Hurricane Katrina. The incorporation of flood damage metrics into the wind damage scale was done in an attempt to combine the multi-hazard nature of hurricane events. After the remote sensing classification was completed, the ground-based WF damage states were correlated with the remote sensing damage signatures. Using the WF Damage Scale criteria for roof damage only, a mapping of damage states (WF-0 to WF-4) was proposed.

Works Womble et al. (2007), Womble et al. (2006) involving damage assessment from Hurricane images mainly deploy measures from digital numbers (DN) of spectral bands. In Womble et al. (2007) DN values of pixels comprising each roof-facet object were extracted from the before-and-after digital image pairs for each of the four multispectral bands available in QuickBird satellite imagery. Object-level statistics were computed for each set of DN values, including standard deviation, variance, average deviation, skewness, uniformity, and entropy. Comparison of before-and-after object-level statistics (such as by differencing or ratioing) results in damage metrics, which numerically describe temporal changes in the roof facets. For this case study, nine separate damage metrics were examined: standard deviation (ratio and difference), variance (ratio), skewness (difference),
average deviation (ratio), uniformity (ratio and difference), and entropy (ratio and difference). Complete results of this study are provided by Womble (2005).

A unique and different approach was proposed in Barnes et al. (2007) and it used a system-level methodology. An image-driven data mining with sigma-tree structures was demonstrated and evaluated. Results show a capability to detect hurricane debris fields and storm-impacted near shore features (such as wind-damaged buildings, sand deposits, standing water, etc.) and an ability to detect and classify non-impacted features (such as buildings, vegetation, roadways, railways, etc.). The sigma-tree-based image information mining capability was demonstrated to be useful in disaster response planning by detecting blocked access routes and autonomously discovering candidate rescue/recovery staging areas.
CHAPTER 3

DATA

The aerial imagery used for this research was acquired by the NOAA Remote Sensing Division to support NOAA national security and emergency response requirements. These images are available publicly for ongoing researches. The images are uncorrected and not rotated. The approximate ground sample distance (GSD) for each pixel is 37 cm (1.2 feet). The images have 60% forward overlap, and sidelap unknown. Image file size is between 2 MB and 3 MB. Each image is 4077×4092 pixels. The NOAA images used were Hurricane Dennis (2005) and Hurricane Ivan (2004) images of Pensacola, Florida. A total of 30 NOAA images were used in this work. All images were in 3-channel 24 bit JPEG format.

In addition, 2 Google Earth images of the 2010 Haiti Earthquake and 6 QuickBird satellite images of Punta Gorda, Florida from Digital Globe were also used. The QuickBird images contained 4 spectral bands—blue (450-520nm), green (520-600nm), red (630-690nm), near-IR (760-900nm) in GeoTiff format. QuickBird collects panchromatic (black and white) imagery at 60-70 cm resolution and multispectral imagery at 2.4- and 2.8-meter resolutions. All images were pansharpened in the preprocessing stage.
CHAPTER 4

IMAGE REGISTRATION

Image registration is the process of determining the point-by-point correspondence between two images of a scene. It is the process of transforming the different sets of images into one coordinate system. The satellite or aerial images from before and after a disaster may have been taken by different cameras at different altitudes, angles and positions. While image providers may provide registered or georeferenced images, often this is not the case. As a result, images of the same location may vary in overlap and viewpoint. By registering two images, such differences can be corrected.

4.1 Steps in Image Registration

One of the images in a set of before and after images is used as a reference image. This image is kept unchanged. The reference image is also known as the base or source image. The second image in the set of two is known as the sensed image or target image. This image is resampled to register with the reference image. Given two images of a scene, the following steps are usually taken to register the images.

1. **Preprocessing**: This involves preparing the images for feature selection and correspondence using methods such as scale adjustment, noise removal, contrast normalization and segmentation. When pixel sizes in the images to be
registered are different but known, one image is resampled to the scale of the other image. This scale adjustment facilitates feature correspondence. If the given images are known to be noisy, they are smoothed to reduce the noise. Image segmentation is the process of partitioning an image into regions so that features can be extracted.

2. **Feature Selection**: To register two images, a number of features are selected from the images and correspondence is established between them. Knowing the correspondences, a transformation function is then found to resample the sensed image to the geometry of the reference image. The features used in image registration could be corners, blobs, lines, curves, templates, regions, or patches. The type of features selected in an image depends on the type of image provided and the detector used.

3. **Feature Correspondence**: This can be achieved either by selecting features in the reference image and searching for them in the sensed image or by selecting features in both images independently and then determining the correspondence between them. The former method is chosen when the features contain considerable information, such as image regions or templates. The latter method is used when individual features, such as points and lines, do not contain sufficient information.

4. **Determination of a Transformation Function**: Knowing the coordinates of a set of corresponding points in the images, a transformation function is determined to resample the sensed image to the geometry of the reference image. The type of transformation function used should depend on the type of geometric difference between the images. If geometric difference between the images is not known, a transformation that can easily adapt to the geometric difference between the images should be used.

5. **Resampling**: Knowing the transformation function, the sensed image is resampled to the geometry of the reference image. The transformed images can then be cropped to include only overlap region. This enables detection of changes in the scene.

4.2 **SURF-based feature Extraction**

For feature extraction we propose the use of scale- and rotation-invariant interest point detector and descriptor, coined SURF (Speeded Up Robust Feature) first introduced in Bay et al. (2006). In SURF-based registration, firstly the in-
Interest points are selected at distinctive locations in the grayscale image, mainly blobs. The most valuable property of an interest point detector is its repeatability, i.e. whether it reliably finds the same interest points under different viewing conditions. Next, the neighborhood of every interest point is represented by a feature vector. This descriptor has to be distinctive and, at the same time, robust to noise, detection errors, and geometric and photometric deformations. Finally, the descriptor vectors are matched between different images. In case of registering before and after damage images, the interest point detector should find points which are repeated in both the images irrespective of variations due to damage, lighting conditions or camera viewpoint. The time taken to register the images should also be small ideally. The choice of SURF features for this application is inspired by the fact that they are more robust and faster than other state-of-the-art detectors and descriptors [Herbert et al. (2008)] while maintaining high accuracy in matching. SURF focuses on scale-invariant and in-plane rotation-invariant detectors and descriptors. These seem to offer a good compromise between feature complexity and robustness to commonly occurring photometric deformations. As quoted in [Herbert et al. (2008)] the skew, anisotropic scaling, and perspective effects are assumed to be second order effects, that are covered to some degree by the overall robustness of the descriptor.

4.2.1 Interest Point Detectors

The most widely used detectors are either Harris-based detectors or Hessian-based detectors [Harris and Stephens (1988), Lowe and G (1999)]. They find corners or blob-like structures. From previously published comparisons [Krystian and Cordelia (2005)], we can conclude that Hessian-based detectors are more
stable and repeatable than their Harris-based counterparts. The SURF-based approach for interest point detection uses a very basic Hessian matrix approximation. Integral images are used to reduce computation time. They allow for the fast implementation of box type convolution filters. The entry of an integral image \( I_{\Sigma}(X) \) at a location \( X = (x, y)^T \) represents the sum of all pixels in the input image \( I \) within a rectangular region formed by the origin and \( X \).

\[
I_{\Sigma}(X) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (4.1)
\]

Once the integral image has been computed, it takes three additions to calculate the sum of the intensities over any upright, rectangular area. Hence, the calculation time is independent of its size. The detector is based on the Hessian matrix because of its good performance in accuracy. Using the Hessian matrix, we detect blob-like structures at locations where the determinant is maximum. Given a point \( X = (x, y)^T \) in an image \( I \), the Hessian matrix \( H(X, \sigma) \) in \( X \) at scale \( \sigma \) is defined as follows

\[
H(X, \sigma) = \begin{bmatrix}
L_{xx}(X, \sigma) & L_{xy}(X, \sigma) \\
L_{xy}(X, \sigma) & L_{yy}(X, \sigma)
\end{bmatrix} \quad (4.2)
\]

where \( L_{xx}(X, \sigma) \) is the convolution of the Gaussian second order derivative \( \frac{\partial^2}{\partial x^2} g(\sigma) \) with the image \( I \) in point \( X \), and similarly for \( L_{xy}(X, \sigma) \) and \( L_{yy}(X, \sigma) \). These approximate second order Gaussian derivatives and can be evaluated at a very low computational cost using integral images. The calculation time therefore is independent of the filter size. The filters that are used are \( 9 \times 9 \) in size. They approximate a Gaussian with \( \sigma = 1.2 \) and represent the lowest scale (i.e.
highest spatial resolution) for computing the blob response maps. We will denote them by $D_{xx}$, $D_{yy}$, and $D_{xy}$. The weights applied to the rectangular regions are kept simple for computational efficiency. This yields

$$\det(H_{\text{approx}}) = D_{xx}D_{yy} - (wD_{xy})^2 \quad (4.3)$$

The relative weight $w$ of the filter responses is used to balance the expression for the Hessians determinant. The approximated determinant of the Hessian represents the blob response in the image at location X. These responses are stored in a blob response map over different scales, and local maxima are detected. For more details refer to Herbert et al. (2008). Figure 4.1 shows an example of the detected interest points using the Fast-Hessian detector.

4.2.2 Feature Descriptors

SURF descriptor describes the distribution of the intensity content within the interest point neighborhood, similar to the gradient information extracted by SIFT Lowe and G (2004) and its variants. For SURF, we build on the distribution of first order Haar wavelet responses in x and y direction rather than the gradient, and exploit integral images for speed. This reduces the time for feature computation and matching, and has proven to simultaneously increase the robustness Herbert et al. (2008).

The first step consists of fixing a reproducible orientation based on information from a circular region around the interest point. This is critical in order for the descriptors to be rotation invariant. For that purpose, we first calculate the Haar wavelet responses in x and y direction within a circular neighborhood of radius 6s.
around the interest point, with $s$ the scale at which the interest point was detected. The sampling step is scale dependent and chosen to be $s$. In keeping with the rest, also the size of the wavelets are scale dependent and set to a side length of $4s$. Since integral images are used only six operations are needed to compute the response in $x$ or $y$ direction at any scale. Once the wavelet responses are calculated and weighted with a Gaussian ($\sigma = 2s$) centered at the interest point, the responses are represented as points in a space with the horizontal response strength along the abscissa and the vertical response strength along the ordinate. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window. The horizontal and vertical responses within the window are
summed. The two summed responses then yield a local orientation vector. The longest such vector over all windows defines the orientation of the interest point.

The next step consists of constructing a square region centered around the interest point and oriented along the orientation selected. The region is split up regularly into smaller $4 \times 4$ square sub-regions. This preserves important spatial information. For each sub-region, we compute Haar wavelet responses at $5 \times 5$ regularly spaced sample points. We call $d_x$ the Haar wavelet response in horizontal direction and $d_y$ the Haar wavelet response in vertical direction. Then, the wavelet responses $d_x$ and $d_y$ are summed up over each sub-region and form a first set of entries in the feature vector. In order to bring in information about the polarity of the intensity changes, we also extract the sum of the absolute values of the responses, $|d_x|$ and $|d_y|$ . Hence, each sub-region has a 4D descriptor vector $v$ where $v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$. Concatenating this for all $4 \times 4$ subregions, this results in a descriptor vector of length 64. The wavelet responses extracted are invariant to a bias in illumination and contrast. The matching of feature descriptors is then done using matching techniques described in the next section.

4.3 Interest Point Matching

Once the interest points are detected, we need to find a one-to-one correspondence between the detected points. This is further complicated by the fact that there maybe outliers. The point matching problem is formally defined as: given two sets of points in 2D space, we need to determine whether there is a transformation among a specified group of space transformations, that maps the first set
onto (or satisfactorily close to) the second set of points. The transformation can consist of a combination of affine transforms like rotation, scaling and translation. Given points $P_1 = (x, y)$ and $P_2 = (x', y')$, the point $P_1$ can be transformed to $P_2$ by scaling in x and y directions using scaling factors $s_x$ and $s_y$.

$$
\begin{bmatrix}
  x' \\
  y'
\end{bmatrix} = \begin{bmatrix}
  s_x & 0 \\
  0 & s_y
\end{bmatrix} \begin{bmatrix}
  x \\
  y
\end{bmatrix}
$$

(4.4)

Similarly, rotation can be performed by

$$
\begin{bmatrix}
  x' \\
  y'
\end{bmatrix} = \begin{bmatrix}
  \cos \theta & -\sin \theta \\
  \sin \theta & \cos \theta
\end{bmatrix} \begin{bmatrix}
  x \\
  y
\end{bmatrix}
$$

(4.5)

And, translation is given by

$$
\begin{bmatrix}
  x' \\
  y' \\
  1
\end{bmatrix} = \begin{bmatrix}
  1 & 0 & t_x \\
  0 & 1 & t_y \\
  0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}
$$

(4.6)

One approach is to simply use the geometric properties of interest points to perform the matching. Such techniques use matching using scene coherence, matching using clustering or matching using invariance. Matching using scene coherence uses a technique where three pairs of points are chosen from the point sets and are used to find the affine parameters. This process is done repeatedly until a transform that causes a good matching is found. In the clustering approach, the transform parameters are estimated through a voting process. Clusters of points are found using a k-nearest neighborhood clustering process and voting decides whether a group of close points correspond to a group in the other set. Another approach uses affine invariant property of three point pairs to find combinations of
three point pairs from the images and determine the relation between remaining points in the images with respect to the three points.

Another and more widely used approach is to compare an interest point in the test image with that in the reference image by calculating the Euclidean or Mahalanobis distance between their descriptors [Baumberg (2000)]. There are three such matching strategies [Krystian and Cordelia (2005)]. In the case of threshold based matching two regions are matched if the distance between their descriptors is below a threshold. A descriptor can have several matches and several of them may be correct. In the case of nearest neighbor based matching two regions $A$ and $B$ are matched if the descriptor of $A$ is the nearest neighbor to that of $B$ and if the distance between them is below a threshold. With this approach a descriptor has only one match. The third matching strategy is similar to nearest neighbor matching except that the thresholding is applied to the distance ratio between the first and the second nearest neighbor.

As the precision is higher for nearest neighbor based matching [Krystian and Cordelia (2005)], we use it on the SURF descriptors described earlier. Consider interest points $A$, $B$ and $C$ with descriptors $D_A$, $D_B$ and $D_C$. The points $A$ and $B$ are matched if $|D_A - D_B|/|D_A - D_C| < t$ where $D_B$ is the nearest and $D_C$ is the second nearest neighbor to $D_A$. The threshold $t$ decides the precision of the matching. We compare each descriptor of the reference image with each descriptor of the transformed image to find a one-to-one correspondence. Clearly, this is a $O(n^2)$ algorithm. To make the process faster we use the sign of the Laplacian (i.e. the trace of the Hessian matrix) for the underlying interest
point. Typically, the interest points are found at blob-type structures. The sign of the Laplacian distinguishes bright blobs on dark backgrounds from the reverse situation. This feature is available at no extra computational cost, as it was already computed during the detection phase. In the matching stage, we only compare features if they have the same type of contrast; i.e., the sign of the Laplacian is the same [Herbert et al. (2008)]. The algorithm is formally given in Algorithm [1]. Figure 4.2 shows an example of matched feature points using the nearest neighborhood algorithm.

![Figure 4.2](image_url)

Figure 4.2. Matching between 10 features in (a) before and (b) after disaster shown in red lines.
Algorithm 1 naiveNearestNeighbor

1:  EXTRACT DESCRIPTORS $A[a_1 \ldots a_m]$
2:  EXTRACT DESCRIPTORS $B[b_1 \ldots b_n]$
3:  for $i \leftarrow 1 \text{ to } m$ do
4:      for $j \leftarrow 1 \text{ to } n$ do
5:          if $\text{signLaplacian}(a_i) = \text{signLaplacian}(b_j)$ then
6:              if $\text{dist1} < \text{Distance}(a_i, b_j)$ then
7:                  $\text{dist1} = \text{Distance}(a_i, b_j)$
8:              else
9:                  if $\text{dist2} < \text{Distance}(a_i, b_j)$ then
10:                     $\text{dist2} = \text{Distance}(a_i, b_j)$
11:             end if
12:         end if
13:     end for
14:  end for
15:  if $\text{dist1} \div \text{dist2} < t$ then
16:      PUSH(matchedPoints, $i$, $j$)
17:  end if
18: end for

4.4 Results

Each interest point in the test image is compared to an interest point in the reference image by calculating the Euclidean distance between their descriptor vectors. A matching pair is detected if its distance is closer than the threshold ($t = 0.7$) times the distance of the second nearest neighbor. Once the matching is done, we find the transformation matrix. Satellite images can be treated as 2D projections being viewed through a camera viewfinder. Because the camera’s position, orientation, and field of view may change for before-and-after disaster images, we consider perspective transformation on the matched points. Given a perspective transformation matrix $H$ with elements $h_{ij}$ we determine the value of the matrix by minimizing the back projection error [Zhang (1999)]:

$$
\min \sum_i \left( (x'_i - (h_{11} \ast x_i + h_{12} \ast y_i + h_{13}) \div (h_{31} \ast x_i + h_{32} \ast y_i + h_{33}))^2 + (y'_i - (h_{21} \ast x_i + h_{22} \ast y_i + h_{23}) \div (h_{31} \ast x_i + h_{32} \ast y_i + h_{33}))^2 \right)
$$

This is a nonlinear minimization problem, which is solved with the Levenberg-Marquardt Algorithm as implemented in

Next, the perspective transformation is applied to the test image. To crop out the non-overlapping region from reference image, we construct a mask of size equal to that of the test image, with all pixels set to 255. The transformation is then applied to the mask and an AND operation with the reference image is performed. Figure 4.3 shows images from Figure 4.2 registered and cropped to fit only the overlapping region. To study the effect of image resolution on the SURF-based registration process, we used images from 15 locations in coastal Florida before and after Hurricane Dennis. These images were $4077 \times 4092$ and 50cm resolution originally. The image pairs vary in overlap, lighting conditions and viewpoint. They were downsampled to smaller versions with 1m, 2m, 4m and 5m resolutions. The original images were 24 bit color images.

For best performance, the color to grayscale conversion was done by extracting the V component of HSV color images. The effect of registering at various resolutions is shown in Table 4.1. While 14 of the 15 images registered successfully at 50cm and 1m, the performance dropped at coarser resolutions. The registration error for all registered images was found to be $\leq 1$ pixel. The registration performed well even with low overlap and varying lighting conditions (See Figure 4.4 and Figure 4.5). Experiments show that repeatability of the SURF detector was affected at lower resolutions. However the results show that proposed scheme is well suited for high-resolution satellite imagery.
Figure 4.3. Registered (a) before and (b) after disaster images. Non-overlap area is filled with black pixels.
Figure 4.4. Unregistered (a) before and (b) after disaster images. The images taken from different camera viewpoints and with lighting conditions.
Figure 4.5. Registered (a) before and (b) after disaster images. The images taken from different camera viewpoints, overlap percentage and lighting conditions.
### TABLE 4.1

REGISTRATION SUCCESS AT RESOLUTIONS 50CM, 100CM, 200CM, 400CM AND 500CM

<table>
<thead>
<tr>
<th>Images</th>
<th>500cm</th>
<th>400cm</th>
<th>200cm</th>
<th>100cm</th>
<th>50cm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image-pair1</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Registered</td>
<td>Registered</td>
</tr>
<tr>
<td>Image-pair2</td>
<td>Failed</td>
<td>Failed</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
</tr>
<tr>
<td>Image-pair3</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
</tr>
<tr>
<td>Image-pair4</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
</tr>
<tr>
<td>Image-pair5</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Registered</td>
<td>Registered</td>
</tr>
<tr>
<td>Image-pair6</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
</tr>
<tr>
<td>Image-pair7</td>
<td>Failed</td>
<td>Failed</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
</tr>
<tr>
<td>Image-pair8</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
</tr>
<tr>
<td>Image-pair9</td>
<td>Failed</td>
<td>Failed</td>
<td>Failed</td>
<td>Registered</td>
<td>Registered</td>
</tr>
<tr>
<td>Image-pair10</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
</tr>
<tr>
<td>Image-pair11</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
</tr>
<tr>
<td>Image-pair12</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
</tr>
<tr>
<td>Image-pair13</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
</tr>
<tr>
<td>Image-pair14</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
</tr>
<tr>
<td>Image-pair15</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
<td>Registered</td>
</tr>
</tbody>
</table>
CHAPTER 5

BUILDING DETECTION

Many image repositories do not contain meta data, therefore we require automatic building detection from post-disaster images. Numerous methods address building extraction at a single time instance. There are a variety of approaches, depending on the type, quality and number of the input images. It is common to use multiview inputs [Noronha and Nevatia (2001)] to exploit 3-D information in building modeling. The detection can be significantly facilitated by working on stereo-based Digital Elevation/Surface Models (DEM/DSM), where the building rooftops can be separated from the ground planes by the estimated height data. However several image repositories lack stereo or multi-sensor information. This case is addressed in this section, thus building identification becomes here a challenging monocular object recognition task based on purely optical data.

5.1 Approaches

An object oriented change detection approach is introduced in [Tanathong et al. (2008)] and applied for the extraction of damaged buildings after a Tsunami disaster. This method uses independent building detection processes in the two images which is followed by object level comparison. The later step is based on matching the geometry and spectral characteristics of the corresponding building
candidates in the two time instances. However the object detection phase can be corrupted by image noise, irregular structures or occlusion by vegetation, which may present missing or only partially extracted buildings to the object matching module. Moreover the comparison may be affected by further intensity artifacts caused by shadows or altered illumination conditions.

A SIFT keypoint based approach has been introduced in Sirmacek and Unsalan (2009) for urban area extraction and building detection. This method assumes that the building structures in a given image can be efficiently characterized by a couple of template buildings (here two templates: a bright and a dark one) which are used for training. The goal is localization, but the accurate bounding boxes of the buildings are not extracted. This makes it difficult to apply the method for change detection. As well, images containing a high variety of buildings may need a huge template library, where the overlap between the buildings and background in the descriptor domain can be hardly controlled.

Many extraction schemes employ edge-based techniques Noronha and Nevatia (2001). This approach is based on hierarchical grouping of extracted edge segments to form continuous lines, junctions and finally closed curve hypothesizes. However, several restrictions are used for the buildings, assuming that they have uniform height, they are composed of planar surfaces with parallel sides and each building casts its shadow on a locally flat surface. The method needs a reasonable edge map because the edge grouping process maybe corrupted by missing large side parts, or plenty of false edges inside and around the buildings.
Following a different approach from edge based techniques, building detection is often considered as a region level or image segmentation problem. In Song et al. (2006) the authors assume that buildings are homogeneous areas either in color or in texture, which can be used for training-based background subtraction. Thereafter elementary constraints for shape and size are used to group the candidate regions into building objects. This method can fail, if due to the weak contrast several building and background parts are merged in the same region of the oversegmented map, or the background and building areas are strongly overlapped in the chosen feature domain. Also, this method may only extract part of the rooftop.

5.2 Algorithm Steps

The common property of the previous techniques is that they are based on one or more specific assumptions (like presence of unique roof colors, shadows and shadow filters, strong edges, homogeneous roofs, only a few typical building structures, or simple 3-D models can be fit), but they fail if the features used are missing or less discriminative for the input data. For a robust performance in the real world, we need to have sufficient generalization and yet highly reliable features. Due to the several restrictions of edge-based techniques as discussed earlier, we adopt a segmentation-based scheme. However unlike Song et al. (2006) we consider buildings to be heterogeneous areas in color and texture. We consider building areas to be a set of multiple homogeneous regions. Rooftops can thus be extracted by trying different combinations of regions and using shadow, shape and spectral content to accept or reject a combination. Our underlying assumption is that even though buildings can be considered to be made of multiple segments,
the number of distinct colors or textures that these segments belong to is very few. This assumption is important for ensuring that the combination process is computationally tractable. Further, unlike previous approaches, each extracted building is to be considered as an area enclosed by a closed contour instead of polygons with linear edges.

5.2.1 Segmentation

The segmentation step groups together pixels of similar spectral content. We assume an N dimensional multispectral space. While the images used in this evaluation have N = 3 (R, G and B channels), the use of additional bands can improve the segmentation performance. In this section we compare various supervised and unsupervised classification schemes for segmentation.

Maximum likelihood classification is the most common supervised classification method used with remote sensing image data. Let the spectral classes for an image be represented by

\[ \omega_i, i = 1, \ldots M \]  

(5.1)

where M is the total number of classes. In trying to determine the class or category to which a pixel vector \( x \) belongs, it is strictly the conditional probabilities

\[ p(\omega_i|x), i = 1, \ldots M \]  

(5.2)

that are of interest. The measurement vector \( x \) is a column of N dimensional multispectral space for the pixel. The probability \( p(\omega_i|x) \) gives the likelihood that the correct class is \( \omega_i \) for a pixel at position \( x \). Classification is performed
according to

\[ x \in \omega_i \quad if \quad p(\omega_i|x) > p(\omega_j|x) \quad for \ all \ j \neq i \]  \hspace{1cm} (5.3)

i.e., the pixel at \( x \) belongs to class \( \omega_i \) if \( p(\omega_i|x) \) is the largest. This intuitive decision rule is a special case of a more general rule in which the decisions can be biased according to different degrees of significance being attached to different incorrect classifications. The general approach is called Bayes classification.

The desired \( p(\omega_i|x) \) in (5.3) and the available \( p(x|\omega_i) \), estimated from training data, are related by Bayes theorem:

\[ p(\omega_i|x) = \frac{p(x|\omega_i)p(\omega_i)}{p(x)} \]  \hspace{1cm} (5.4)

where \( p(\omega_i) \) is the probability that class \( \omega_i \) occurs in the image. It is of interest to note in passing that

\[ p(x) = \sum_{i=1}^{M} p(x|\omega_i)p(\omega_i) \]  \hspace{1cm} (5.5)

although \( p(x) \) itself is not important in the following. The \( p(\omega_i) \) are called a priori or prior probabilities, since they are the probabilities with which class membership of a pixel could be guessed before classification. By comparison the \( p(\omega_i|x) \) are posterior probabilities. Using (5.4) it can be seen that the classification rule of (5.3) is:

\[ x \in \omega_i \quad if \quad p(x|\omega_i)p(\omega_i) > p(x|\omega_j)p(\omega_j) \quad for \ all \ j \neq i \]  \hspace{1cm} (5.6)

where \( p(x) \) has been removed as a common factor. The rule of (5.6) is more acceptable than that of (5.3) since the \( p(x|\omega_i) \) are known from training data, and it is conceivable that the \( p(\omega_i) \) are also known or can be estimated from the
analyst knowledge of the image. Mathematical convenience results if in (5.6) the definition
\[
g_i(x) = \ln p(x|\omega_i)p(\omega_i) = \ln p(x|\omega_i) + \ln p(\omega_i)
\]
is used, where \(\ln\) is the natural logarithm, so that (5.6) is restated as
\[
x \in \omega_i \text{ if } g_i(x) > g_j(x) \text{ for all } j \neq i
\]
This is, with one modification to follow, the decision rule used in maximum likelihood classification. The \(g_i(x)\) are referred to as discriminant functions.

At this stage it is assumed that the probability distributions for the classes are of the form of multivariate normal models. This is an assumption, rather than a demonstrable property of natural spectral or information classes. However it leads to mathematical simplifications in the following. Moreover, it is one distribution for which properties of the multivariate form are well-known.

In (5.7) therefore, it is now assumed for \(N\) bands that
\[
p(x|\omega_i) = (2\pi)^{-N/2} |\Sigma_i|^{-1/2} \exp \left\{ -1/2(x - m_i)^t \Sigma_i^{-1}(x - m_i) \right\}
\]
where \(m_i\) and \(\Sigma_i\) are the mean vector and covariance matrix of the data in class \(\omega_i\). The resulting term \(N/2 \ln(2\pi)\) is common to all \(g_i(x)\) and does not aid discrimination. Consequently it is ignored and the final form of the discriminant function for maximum likelihood classification, based upon the assumption of normal statis-
statistics, is:

\[ g_i(x) = \ln p(\omega_i) - 1/2 \ln |\Sigma_i| - 1/2(x - m_i)^t \Sigma_i^{-1} (x - m_i) \] (5.10)

Often we have no useful information about the \( p(\omega_i) \), in which case a situation of equal prior probabilities is assumed. As a result \( \ln p(\omega_i) \) can be removed from (5.10) since it is then the same for all \( i \). In that case the \( 1/2 \) common factor can also be removed leaving, as the discriminant function:

\[ g_i(x) = -1/2 \ln |\Sigma_i| - 1/2(x - m_i)^t \Sigma_i^{-1} (x - m_i) \] (5.11)

Implementation of the maximum likelihood decision rule involves using (5.11) in (5.8). Sufficient training pixels for each spectral class must be available to allow reasonable estimates to be obtained of the elements of the class conditional mean vector and covariance matrix. For an \( N \) dimensional multispectral space the covariance matrix is symmetric of size \( N \times N \). It has, therefore, \( 1/2N(N+1) \) distinct elements that need to be estimated from the training data. To avoid the matrix being singular at least \( N(N+1) \) independent samples are needed. Fortunately, each \( N \) dimensional pixel vector in fact contains \( N \) samples (one in each band); thus the minimum number of independent training pixels required is \( (N+1) \). For practical purpose as many as 100\( N \) training pixels per class is recommended. An image from coastal Florida is shown in Figure 5.1. From this image, regions belonging to 11 classes were manually labeled. For each class, 4 regions with at least 100 pixels each were used for training. Classes included ocean, grass, vegetation, road, white roof, blue roof, dark roof, gray roof, pavement, land area and shadow. The result of maximum-likelihood classification is shown in
As seen in Figure 5.2, the maximum-likelihood classification results in salt and pepper noise. At this stage a comparison with ground truth shows that the detection percentage for segments classified as buildings is 28%. To reduce the noise, misclassification, over-segmentation and under-segmentation, we require more training pixels. However, to minimize supervision in the detection process, we instead use a k-means clustering algorithm to improve the results of the maximum likelihood classifier. Using the means of the labeled training classes as seeds, a clustering is performed over pixels of the entire image. The means and covari-
Figure 5.2. Result of maximum-likelihood classification with $M = 11$, $N = 3$. Classes included ocean, grass, vegetation, road, white roof, blue roof, dark roof, gray roof, pavement, land area and shadow. These are shown in different colors. Detection Percentage = 28%

The clustering process is described below.

1. The procedure is initialized by selecting $M$ points in multispectral space to serve as candidate cluster centres. Let these be called

$$\hat{m}_i, i = 1, \ldots, M. \quad (5.12)$$

The selection of the $\hat{m}_i$ at this stage is done by setting $\hat{m}_i = m_i$, where $m_i$ are the means computed from the data in supervised learning process.

2. The location $x$ of each pixel in the segment of the image to be clustered is examined and the pixel is assigned to the nearest candidate cluster. This assignment would be made on the basis of the Euclidean distance measure.

3. The new set of means and covariances that result from the grouping pro-
duced in Step 2 are computed. Let these be denoted as
\[ \tilde{m}_i, i = 1, \ldots, M. \] (5.13)

4. If \( \tilde{m}_i = \hat{m}_i \) for all \( i \), the procedure is terminated. Otherwise \( \hat{m}_i \) is redefined as the current value of \( \tilde{m}_i \) and the procedure returns to Step 2.

The means and covariance vectors computed from Step 3 are used as \( m_i \) and \( \Sigma_i \) for the maximum-likelihood classifier. The result is shown in Figure 5.3. Clearly there is a decrease in salt and pepper noise and the detection percentage increases to 32.3%. Step 4 may not terminate always and hence other termination conditions can be specified, such as maximum number of iterations, percentage of convergence etc. We can avoid any supervision in the classification by using random seeds in step 1.

5.2.2 Shape Measure

A common challenge in remote sensing is the classification of objects that are spectrally similar but represent physically different types of structures. After the image segmentation, the obtained image objects need to be characterized by their shape. In [Glasbey and Horgan (1996)](Glasbey and Horgan (1996)), the following shape parameters were used: compactness, roundness and convexity. These shape measures are based on relations between the surface area of an object, perimeter length of the object and perimeter length of the object’s convex hull. Object areas were calculated by a summation of all pixels in an object. For measuring the perimeter of a convex hull, the so-called convex perimeter, only pixels belonging to the object’s outer edge were counted.

For the purpose of extracting buildings we use only the compactness measure
Figure 5.3. Result of k-means clustering followed by maximum-likelihood classification. M = 11, N = 3. Classes included ocean, grass, vegetation, road, white roof, blue roof, dark roof, gray roof, pavement, land area and shadow. These are shown in different colors.
Detection Percentage = 32.3%

which uses the perimeter length of the object. The perimeter of the original object may also include pixels inside the object when boundaries exist within an object. Perimeter lengths have been approximated by:

\[
Perimeter \approx \frac{\sum N_8 + \pi}{0.900} \quad (5.14)
\]

where \(\sum N_8\) is a summation of all 8-connected edge pixels that belong to the object. The measured perimeters are approximated by counting edge pixels of an object. This gives the so-called internal perimeter, which is shorter than the actual perimeter. Although this difference is mainly noticeable for small objects,
\( \pi \) is added to the perimeter length as a correction \(^{\text{Glasbey and Horgan}} (1996)\). Furthermore, the perimeter is corrected with factor 0.900, which in an image raster is the number of 8-connected pixels per unit length. Compactness is defined as:

\[
\text{Compactness} = \frac{4\pi \text{area}}{\text{perimeter}^2}
\]  

(5.15)

and has a maximum value of 1 for a circle, being the most compact object. Both a change of overall shape as well as the presence of local irregular borders decrease this measure to lower values. Shape measures are pose-invariant: change of orientation, location and size leave a measure unchanged. However, the spatial resolution does influence shape measurements. As we deal with only high-resolution satellite imagery, this can be ignored. An example image and its segmented version are shown in Figure 5.4. Figure 5.5 (a) shows the segments classified as buildings in Figure 5.4 (b). However the high irregularity of borders detected increases the perimeter and distorts the compactness measure. To fix this we use the Canny edge detector to find the outer edges of each building and use a morphological dilation operation \(^{\text{Serra}} (1983)\). The result is shown in Figure 5.5 (b). Table 5.1 shows some of the corresponding compactness measures. Shape 2 is a parking lot that is wrongly classified as a building and has a low compactness value of 0.069. On the other hand, all correctly detected buildings have high compactness values.

5.2.3 Measure of Estimated and Predicted Shadow (MEPS)

Shadows are widely used in the building localization process. This step needs principally the extraction of the shadowed regions which is itself a hot topic of research \(^{\text{Benedek and Szirnyi}} (2010)\) and has its own literature for remote sensing.
The large variety of the input data calls for different approaches to tackle the problem. Techniques addressing high-resolution satellite imagery usually deal with single channel images. Since here the only available pixel information is the intensity, one can, at pixel level, solely rely on the assumption that shadows correspond to dark image areas. It is also important to note that for building detection the cast shadows (i.e. shadows on the ground) are relevant, while self shadows (i.e. weakly or not illuminated building parts) should be ignored. However, as pointed out in Dare (2005), in most cases cast and self shadows have different intensity values, since the shadowed object parts are illuminated more by secondary light sources such as reflection from surrounding buildings.
Performing shadow detection via pixel brightness filtering faces two main challenges. Firstly, we usually find a significant overlap between the intensity domains of the shadowed and non-shadowed areas, therefore some misclassified regions are expected. Secondly, the separation commonly thresholding, must be appropriately parametrized. There have been a few methods proposed for automated threshold estimation, most frequently based on global image histograms. According to the experiments of Tsai (2006), the intensity based shadow filtering may be notably improved considering color values, however, they can only decrease but not eliminate the problem of domain-overlapping, which still means challenges for higher level model elements.

In our approach we detect the shadow by the segmentation process described...
TABLE 5.1

SOME OF THE COMPACTNESS MEASURES OF BUILDINGS IN FIG 5.5 (b) AFTER MORPHOLOGICAL DILATION

<table>
<thead>
<tr>
<th>Shape</th>
<th>Compactness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.613670</td>
</tr>
<tr>
<td>2</td>
<td>0.069449</td>
</tr>
<tr>
<td>6</td>
<td>0.623588</td>
</tr>
<tr>
<td>7</td>
<td>0.271278</td>
</tr>
<tr>
<td>8</td>
<td>0.620232</td>
</tr>
<tr>
<td>11</td>
<td>0.419952</td>
</tr>
</tbody>
</table>

previously. We detect shadows using the $N$ dimensional multispectral space of pixels rather than simply using the brightness values of a single channel. An image with 3 spectral bands is shown in Figure 5.6 (a) and the detected shadows, buildings and roads are shown in Figure 5.6 (b). Note that shadows are cast in certain directions (with respect to the buildings) depending on the time of the day. In the remainder of this section we propose a feature, which is the measure of estimated and predicted shadow (MEPS) used to describe the ”shadow strength” of an object.

The first step in computing the MEPS feature is finding an approximate estimation of the direction in which we expect the shadow to be present. Given any image, the direction of shadows of the buildings depends on the time of the
Figure 5.6. (a) An image with 3 bands. (b) Detected buildings and roads are shown as green contours and the shadows are shown in white. The few red arrows placed on buildings show on which side the shadow is cast.

day in which the image was captured. This shadow direction can be denoted by a direction vector $\vec{v}_{sh}$. If the time at which the image was acquired is available then computing $\vec{v}_{sh}$ is trivial. But often this information is not available. One alternative is to use segments that were classified as buildings to estimate $\vec{v}_{sh}$. Due to noise and misclassifications there maybe disagreements between building segments on the direction of $\vec{v}_{sh}$. This can be overcome by a voting process. In the second step, based on $\vec{v}_{sh}$ we predict the pixels that could be marked as shadow along the perimeter of any object. The MEPS feature is then computed based on the number of predicted pixels. The process can be summarized as follows.

1. Perform a Canny edge detection on the binary image of segments detected as buildings during the segmentation process. Find the outer contours for
each building. Approximate the contours to polygons using Douglas-Peucker algorithm \cite{Douglas:1973}. Find polygons with 4 sides and 90° angles between the sides, i.e find buildings that are rectangles. See Figure 5.7 for an example.

![Figure 5.7](image)

Figure 5.7. Rectangular buildings found from segments classified as buildings in Figure 5.3 are outlined in green.

2. For each rectangular building, find the normals $\vec{v}_i$ ($i = 1...4$) to each of the 4 sides. Take votes for all the normals by counting the number of normals that point in same direction. If a single direction $\vec{v}_i$ is found to be the winner,

$$\vec{v}_{sh} = \vec{v}_i$$

(5.16)

else, at the most two directions $\vec{v}_i$ and $\vec{v}_j$ can be winners. In this case,

$$\vec{v}_{sh} = \vec{v}_i + \vec{v}_j$$

(5.17)
3. Once $\vec{v}_{sh}$ is found, the perimeter along which a shadow is predicted to be cast is the convex perimeter that if "visible" to a line perpendicular to $\vec{v}_{sh}$. See Figure 5.8 for an illustration of this concept. For any object, the MEPS feature can be calculated as follows. Let $x = (x_1...x_n)$ be the outer contour of the object and $\vec{y}_1...\vec{y}_n$ be the respective normals.

$$\theta_i = \arccos \left( \frac{\vec{v}_{sh} \cdot \vec{y}_i}{|\vec{v}_{sh}| |\vec{y}_i|} \right) \times \frac{180}{\pi}$$ (5.18)

Figure 5.8. (a) A building (b) The arrow indicates $\vec{v}_{sh}$. The perimeter of the building which is visible to a line perpendicular to this vector is where we would expect the shadow to be present.

If $0^\circ \leq \theta_i \leq 90^\circ$, then we predict that a shadow pixel will be found for $x_i$. An example of such a prediction is shown in Figure 5.9. We search for a shadow pixel along a profile of length $t$ in the direction $\vec{y}_i$ at all such predicted $x_i$. Let the total number of such $x_i$’s be $N_x$ and the total number
for which at least a shadow pixel is found be $N_{sh}$. Now, we can define MEPS feature $MEPS_x$ for a contour $x$ as

$$MEPS_x = \frac{N_{sh}}{N_x}$$

(5.19)

Figure 5.9. The segments classified as buildings are shown in white. The pixels along the perimeter predicted for shadows are marked in red.

Steps 1 and 2 can be replaced by a manual identification of $\vec{v}_{sh}$ by visual inspection of the image. Alternately, both the steps can be modified to include non-rectangular objects. $MEPS_x$ takes up a value between 0 and 1. Even though $MEPS_x$ is expected to be close to 1 for buildings, it could be lesser as the shadow vector identified by rectangular buildings may only be approximations. Inaccurate
segmentation is another factor that could affect $MEPS_z$. The MEPS values for a parking lot and a building are shown in Figure 5.10 and Figure 5.11 respectively. Higher MEPS values are expected for elevated surfaces like rooftops and lower values for flat areas like roads, pavements and parking lots.

Figure 5.10. (a) A parking lot (b) Fragments of the parking lot which were wrongly classified as buildings. The labeled numbers indicate the $MEPS$ values for each fragment. Note that $MEPS \leq 0.07$ for all fragments.
Figure 5.11. (a) A building (b) The labeled numbers indicate the \textit{MEPS} value for the segment classified as the building. Note that \textit{MEPS} = 0.96 for the building.

5.2.4 Segment Combination

As described earlier the segmentation process divides the image into regions called segments which may belong to various building or non-building classes. Buildings are considered to be objects composed of one or more building segments, and which fulfills certain properties such as presence of shadows in a certain direction, compactness, meeting area threshold etc. However, in practice due to variations in rooftop texture and color, segments in the image tend to be misclassified often and buildings are usually composed of multiple non building and building segments (see Figure 5.12). This results in a poorly performing extraction process if we consider only building segments.
Figure 5.12. (a) A few buildings (b) Segmented image shows shadows in yellow, roads in pink and buildings in different shades of green, pink and blue. This is an example of over-segmentation.

A highly robust way to overcome this problem is to consider buildings as objects consisting of at most $r$ segment classes which may include non-building classes as well. For $n$ classes of segments we need to consider $\binom{n}{r} + \binom{n}{r-1} + \ldots + \binom{n}{1}$ combinations of classes. Each segment formed from each combination of classes can then be tested with threshold values for compactness $C_{th}$, area $A_{th}$ and MEPS value $M_{th}$. In our experiments we considered $n = 9$ and $r = 4$ (we ignored shadow and grass classes as their omission did not affect performance). This is still a computationally intensive process and can affect computation time significantly depending on the size/dimensions of the image. One way to make the process faster is start with the highest value of $r$ and to eliminate all segments that are found to be buildings before decrementing $r$. 
The algorithm is formally summarized in Algorithm 2. In step 1 \( S_1 \ldots S_n \) contain the segments that belong to class 1 \ldots class n respectively. Several other useful constraints can be used in step 12 of the algorithm. The hue histogram values (see section on HSV Measures for description) of individual segments that make up the segment could be compared against each other to make sure that segments with completely different colors are not combined together. Also, combinations containing certain hue values for segments could be omitted (for example all combinations with segments having mean hue value of light green can be omitted). An initial pass eliminating all individual segments greater than the higher threshold for area can be useful in speeding up the process. The value of \( r \) can be changed depending on the quality of the segmentation and type of buildings in the image.

**Algorithm 2 segmentCombination**

1: \( S \leftarrow S_1 \ldots S_m \)
2: Image\(_{final} \leftarrow \emptyset \)
3: Image\(_{temp1} \leftarrow \emptyset \)
4: Image\(_{temp1} \leftarrow \emptyset \)
5: for \( \bar{r} \leftarrow r \) to 1 do
6: for each combination Image\(_{temp1} \) composed of \( \bar{r} \) elements in \( S \) do
7: Image\(_{temp1} \leftarrow Image\(_{temp1} \) − Image\(_{temp2} \)
8: for each segment \( s \in Image\(_{temp1} \) \) do
9: \( A_s \leftarrow \text{Area}(s) \)
10: \( M_s \leftarrow \text{MEPS}(s) \)
11: \( C_s \leftarrow \text{Compactness}(s) \)
12: if \( (A_s \geq A_{th} \text{ and } M_s \geq M_{th} \text{ and } C_s \geq C_{th}) \) then
13: \( Image\(_{final} \leftarrow Image\(_{final} \) + s \)
14: end if
15: end for
16: end for
17: Image\(_{temp2} \leftarrow Image\(_{final} \)
18: end for
19: return Image\(_{final} \)
5.3 Results

Building detection experiments were performed on 3 images of coastal Florida before Hurricane Dennis. Figure 5.13 and Figure 5.14 show an example with 50 buildings in the Ground Truth. Another image with 35 buildings in Ground Truth is shown in Figure 5.15 and Figure 5.16. The buildings hidden behind trees, variations in rooftop color and shape makes this a difficult example. Let TP be the true positives found by the detection and FP be the false positives. Let AP or actual positives be the number of buildings in the Ground Truth, we define DP (Detection Percentage) and Accuracy as

\[ DP = \frac{TP}{AP} \]  
\[ Accuracy = \frac{TP}{TP + FP} \]

Table 5.2 show the results of the detection after comparison with the ground truth. A high DP ranging between 85% to 91% was observed for all images. However the Accuracy values were lower, with image 3 having the lowest value of 76%. This is due to higher FP with lower threshold values of \( C_{th} = 0.3 \) and \( M_{th} = 0.3 \). The largest running time observed was 25 mins (for image 2).
Figure 5.13. A before storm NOAA image from Pensacola, Florida

Figure 5.14. Buildings extracted from Figure 5.13 is shown in white. $C_{th} = 0.4$, $M_{th} = 0.4$ and $A_{th} = 100$ pixels
Figure 5.15. Another before storm NOAA image from Pensacola, Florida. The buildings hidden behind trees and variations in rooftops makes this a difficult example. $C_{th} = 0.3$, $M_{th} = 0.3$ and $A_{th} = 100$ pixels

Figure 5.16. Buildings extracted from Figure 5.15 is shown in white. $C_{th} = 0.3$, $M_{th} = 0.3$ and $A_{th} = 100$ pixels
TABLE 5.2

PERFORMANCE OF BUILDING DETECTION

<table>
<thead>
<tr>
<th>Image</th>
<th>Size</th>
<th>Buildings</th>
<th>DP</th>
<th>Accuracy</th>
<th>Time (mins)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>785 × 719</td>
<td>50</td>
<td>86%</td>
<td>88%</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>2056 × 1320</td>
<td>250</td>
<td>88%</td>
<td>79%</td>
<td>25</td>
</tr>
<tr>
<td>3</td>
<td>659 × 496</td>
<td>35</td>
<td>91%</td>
<td>76%</td>
<td>2</td>
</tr>
</tbody>
</table>
CHAPTER 6
DAMAGE ESTIMATION

Damage estimation can be considered from two aspects—change detection and damage classification. Building damage and debris spread can be identified and classified by applying change detection algorithms to pre- and post-storm image pairs. In this comparison process, building damage appears as changes in shape, lines, colors, texture, or other image properties. Previous research has shown that the severity of damage to buildings can be estimated from the extent of change in the roof structure [Womble (2005)]. The severity is reflected in damage metrics that quantify the extent of change. The damage metrics are then used to classify regions of the image into categories ranging from no damage to severely damaged. While change detection is a means to find the most relevant changes and omitting the non-relevant ones, damage classification is used to categorize damage into different qualitative states. Both aspects are of interest for post-disaster analysis. In this chapter we compare existing pixel-based change detection techniques for grayscale images, propose new building-based change measures and explore damage classification using learning algorithms.

6.1 Gray-level Change Detection

Change detection algorithms use different criteria to classify pixels as representing change or not. A survey of existing gray-level change detection algorithms,
though not from the specific viewpoint of windstorm damage assessment, is presented in [Radke et al. (2005)](Radke et al. (2005)). The simplest change detection algorithm is based on a plain differencing of corresponding pixels in the pre- and post-storm images. This technique thresholds a difference image to obtain a change mask. A pixel is marked as either "change" or "no change". A more complicated approach uses significance tests to make a decision at each pixel based on values in a local neighborhood of pixels, so that change/no change threshold can vary across the image. Algorithms based on shading models use ratio of intensities at the pixels. In this study we compare change detection algorithms of various types. We performed our study on 6 before-and-after image pairs, taken from Hurricane Charley (2004), Hurricane Dennis (2005) and Hurricane Ivan (2004). The images were taken from Pensacola, Fort Walton Beach, and Punta Gorda, Florida. Figure [6.1](#) shows a pair of pre- and post-storm Hurricane Charley images and Figure [6.2](#) shows all the manually marked damages areas colored in red.

The simplest class of algorithms just thresholds the difference between two images. That is, the change mask $B(x)$ is generated according to the following decision rule:

$$B(x) = 1 \quad if \mid D(x) \mid > T, \quad 0 \quad otherwise$$

(6.1)

$D(x)$ is the difference image and $T$ is an empirically-chosen threshold such that $0 \leq T \leq 1.0$. This process is known as simple differencing.

Another class of change detection algorithms uses a statistical hypothesis test to make a decision. The decision as to whether or not a change has occurred at a given pixel $x$ corresponds to choosing one of two competing hypotheses: the null
Figure 6.1. Pre- and post-storm images (grayscale). QuickBird imagery from DigitalGlobe, Inc.

hypothesis $H_0$ or the alternative hypothesis $H_1$, corresponding to no-change and change decisions, respectively. The image pair $(I_1(x), I_2(x))$ is viewed as a random vector. Knowledge of the conditional joint probability density functions (pdfs) $p(I_1(x), I_2(x)|H_0)$ and $p(I_1(x), I_2(x)|H_1)$ allows us to choose the hypothesis that best describes the intensity change at $x$ using the classical framework of hypothesis testing. In our evaluation we use a statistical change detection algorithm that uses a Gaussian model for the noise. Under the null hypothesis, unchanged regions in the difference image are modeled as zero-mean Gaussian and the noise variance is estimated from the images. Characterizing the alternative (change) hypothesis $H_1$ is more challenging, since the observations consist of change components that are not known a priori or cannot easily be described by parametric distributions. When both conditional pdfs are known, a likelihood ratio can be formed as:

$$L(x) = \frac{p(D(x)|H_1)}{p(D(x)|H_0)}$$  \hfill (6.2)
This ratio is compared to a threshold defined as:

$$ T = \frac{P(H_0)(C_{10} - C_{00})}{P(H_1)(C_{01} - C_{11})} $$  \hspace{1cm} (6.3) $$

where $P(H_i)$ is the a priori probability of hypothesis $H_i$, and $C_{ij}$ is the cost associated with making a decision in favor of hypothesis $H_i$ when $H_j$ is true. In particular, $C_{10}$ is the cost associated with false alarms and $C_{01}$ is the cost associated with misses. This algorithm is referred to as likelihood ratio test.

Several change detection techniques are based on the shading model for the intensity at a pixel. Such algorithms generally compare the ratio of image intensities $R(x) = I_0(x)/I_1(x)$. This uses a null hypothesis that assumes linear dependence between vectors of corresponding pixel intensities, giving rise to the test statistic $R(x)$. In our evaluation we use the linear dependence algorithm suggested by the
work in Skifstad and Jain (1989) to assess the linear dependence between two blocks of pixel values.

More sophisticated change detection algorithms result from exploiting the relationships between nearby pixels. These algorithms are classified under predictive models. In this approach to change detection we fit the intensity values of each block to a polynomial function of the pixel coordinates. Hsu et al. (1984) discussed generalized likelihood ratio tests using a constant, linear, or quadratic model for image blocks. The null hypothesis in the test is that corresponding blocks in the two images are best fit by the same polynomial coefficients, whereas the alternative hypothesis is that the corresponding blocks are best fit by different polynomial coefficients. In each case, the various model parameters are obtained by a least-squares fit to the intensity values in one or both corresponding image blocks. An extension of this algorithm uses mean derivatives of the quadratic models at each pixel in the block.

6.1.1 Comparison of Different Techniques

Damage areas that are manually marked in our ground truth change masks include rooftops, missing trees, missing pools, and debris. Changes such as shadows due to buildings are not marked, as these changes could not represent storm damage. The goal is to compare detected change with ground truth. This comparison requires an algorithm that tells us about the accuracy of the detected change in an objective, quantitative manner. We propose to use the measures suggested in Hoover et al. (1996). For the measures to be computed, the change mask and ground truth should be separated into different regions. For example, damage
to building 1 would be a region, damage to building 2 would be another region etc. In order to classify a change mask into these regions, we use a connected components algorithm. Then the quantitative measures are obtained from the comparison Dillencourt et al. (1992).

The comparison process is summarized as follows. One, apply a connected components algorithm to label each damage region as a separate segment. Two, the labeled images (Ground Truth and Change Detection) are passed to the comparison algorithm. Three, the comparison algorithm produces a set of quantitative measures. Four, steps 2 and 3 are repeated for different threshold values in change detection until the best possible match is found.

Firstly the connected components labeling scans an image and groups its pixels into components based on connectivity. Pixels in a connected component share similar pixel intensity values and are spatially connected with each other. In our evaluation each connected component can be considered to be a separate damage region. Each damage region is assigned a unique number. This process is shown in Figure 6.3. The labeled regions in the ground truth (GT) and the change detected (CD) images are then classified as in Hoover et al. (1996).

We consider five types of result for the comparison of detected damage regions to ground-truth damage regions: correct detection, over-segmentation, under-segmentation, missed, and noise. The formulas for deciding classifications are based upon threshold $T$, for the mutual overlap between two regions, where $0.5 < T \leq 1.0$. The value of $T$ can be set to reflect the strictness of definition.
Figure 6.3. The pictures show the comparison process for (a) ground truth (GT) and (b) change detection (CD). The different connected components in the change masks are computed and labeled by using different pixel intensities. Then the connected components are numbered and compared to obtain the quantitative measures.

desired. The following metrics define each classification (adapted from Hoover et al. (1996)):

1. An instance of correct detection: A pair of regions $R_n$ in the GT image and $R_m$ in the CD image are classified as an instance of correct detection if a) At least $T$ percent of the pixels in region $R_m$ in the CD image are marked as pixels in region $R_n$ in the GT image, and b) At least $T$ percent of the pixels in region $R_n$ in the GT image are marked as pixels in region $R_m$ in the CD image.

2. An instance of over-segmentation: A region $R_n$ in the GT image and a set of regions in the CD image are classified as an instance of over-segmentation if a) At least $T$ percent of the pixels in each region in the CD image are marked as pixels in region $R_n$ in the GT image, and b) At least $T$ percent of the pixels in region $R_n$ in the GT image are marked as pixels in the union of regions in the CD image.
3. An instance of under-segmentation: A set of regions in the GT image and a region \( R_m \) in the CD image are classified as an instance of under-segmentation if a) At least \( T \) percent of the pixels in region \( R_m \) in the CD image are marked as pixels in the union of regions in the GT image, and b) At least \( T \) percent of the pixels in each region in the GT image are marked as pixels in region \( R_m \) in the CD image.

4. An instance of missed classification: A region \( R_m \) in the GT image that does not participate in any instance of correct detection, over-segmentation or under-segmentation is classified as missed.

5. An instance of a noise classification: A region \( R_m \) in the CD image that does not participate in any instance of correct detection, over-segmentation or under-segmentation is classified as noise.

Although these definitions result in a classification for every region in the GT and CD images, they are not unique for \( T < 1.0 \). However, for \( 0.5 < T < 1.0 \) any region can contribute to at most three classifications, one each of correct detection, over-segmentation and under-segmentation.

6.1.2 Evaluation

We compared the change detection algorithms discussed above using six pairs of pre- and post-storm images. We computed the region classifications for (1) simple differencing algorithm, (2) statistical hypotheses test using a Gaussian noise model, (3) linear dependence algorithm, and (4) polynomial algorithms- constant, quadratic and mean derivative. The change-detected regions were classified as correct, under-segmented, over-segmented, missed and noise. For each algorithm the change masks were computed for 40 detection threshold values ranging from 0 to 1, and for each algorithm the value giving the best result for that algorithm was kept. This is to ensure that each algorithm is well tuned. The change mask corresponding to the maximum number of correct classifications was chosen for
calculating the result. Table 6.1 summarizes the results by averaging the region classifications for the 6 pairs of images.

**TABLE 6.1**

AVERAGE REGION CLASSIFICATIONS FOR 6 PAIRS OF BEFORE-AND-AFTER IMAGES, THE AVERAGE OF REGIONS CLASSIFIED AS CORRECT, UNDER-SEGMENTED, OVER-SEGMENTED, MISSED AND NOISE WERE COMPUTED FOR EACH CHANGE DETECTION ALGORITHM EVALUATED

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Correct</th>
<th>Under-segmented</th>
<th>Over-segmented</th>
<th>Missed</th>
<th>Noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Diff.</td>
<td>7.67</td>
<td>0.34</td>
<td>0.16</td>
<td>16.00</td>
<td>224.00</td>
</tr>
<tr>
<td>Stat Hyp. Test</td>
<td>7.83</td>
<td>0.16</td>
<td>0.00</td>
<td>16.16</td>
<td>65.34</td>
</tr>
<tr>
<td>Linear Dep.</td>
<td>0.50</td>
<td>0.00</td>
<td>0.00</td>
<td>23.67</td>
<td>74.16</td>
</tr>
<tr>
<td>Polynomial-const.</td>
<td>1.00</td>
<td>0.00</td>
<td>0.00</td>
<td>23.17</td>
<td>98.50</td>
</tr>
<tr>
<td>Polynomial-quad.</td>
<td>2.16</td>
<td>0.00</td>
<td>0.00</td>
<td>22.00</td>
<td>102.50</td>
</tr>
<tr>
<td>Mean Derivative</td>
<td>2.34</td>
<td>0.16</td>
<td>0.16</td>
<td>21.50</td>
<td>129.00</td>
</tr>
</tbody>
</table>

From Table 6.1 it is evident that the simple differencing algorithm and the statistical hypothesis test algorithm outperformed the other algorithms. The statistical hypothesis test algorithm found a slightly larger number of correct classifications in comparison to the simple differencing algorithm. However we can conclude that the frequency of noise regions is the highest for the simple differ-
encing algorithm. Among polynomial algorithms, the quadratic model performed better than the constant model. The extension of the quadratic model performed better than other two models. The constant model had the least number of noise classifications among the polynomial models. The shading model approach found the least number of correct classifications, showing that measure based on linear dependence is least desirable.

Using best performance result for each algorithm, most algorithms seem to have relatively fewer regions classified as over- or under-segmented. The largest number of under-segmented regions is found in simple differencing. Statistical hypothesis test approach finds the highest number of correct classifications for almost all image pairs. As it uses a Gaussian model for noise variance, statistical hypothesis test also finds the least number of noise classifications.

Studying the correlation between the quantitative measures obtained from change detection and the level of damage actually present is important for using them in classifying damage in an automated system. For qualitative characterization of building damage, Womble suggested a classification based on roof structure damage \cite{Womble2005}. He identified certain visual signatures of wind damage that could be used to categorize damage into four categories: RS-A (no damage), RS-B (tiles removed), RS-C (Decking removed) and RS-D (roof structure collapsed).

Figure 6.4 show plots of windstorm damage profile categories as suggested by Womble, versus damage metric values from the simple differencing and the statisti-
cal testing change detection, respectively. For these plots, 16 residential buildings were selected from our image dataset, with Remote-Sensing Damage Scale values distributed as: RS-A (4), RS-B (4), RS-C (4), and RS-D (4). The change detection measure used here is expressed as a percentage of pixels that have been detected as changed in the pre- and post-storm roof top images. These damage profiles show a general correlation of the change detection metrics with damage states. The damage profile for simple differencing exhibits a stronger correlation as compared to statistical testing. While there is a clear distinction between no damage and other damage states, distinction between mid-level damage states (RS-B or RS-C) is difficult. However, the data contains too much scatter to accurately assign one of the four damage states based on a given value of damage metric. It may be possible to achieve greater accuracy by taking into consideration other factors such as the changes detected due to debris.

6.2 Object-based Change Detection

To overcome the shortcomings of the pixel based methods, we consider an object-based change detection approach. In this case the objects are buildings. The change measures are considered to reflect changes due to building damages. These are more intuitive in understanding damages that happen due to a disaster. Our approach assumes three types of damage. Minor damage includes removal of tiles, slight irregularities in edges etc. A moderate version would include holes in the roof, dislodged decking and partial change in elevation of the roof. The third kind is when the rooftop collapses or is missing. The change measures extracted from the images should be able to discriminate well between the severity of all three kinds. A data transformation process that includes conversion of the images
Figure 6.4. A windstorm damage profile for damage metrics obtained from (a) simple differencing and (b) for statistical hypothesis testing. The damage metrics are based on our own assigned ratings for the Remote-Sensing Damage Scale suggested in Womble (2005) based on pre- and post-storm comparison of 16 buildings in our image dataset.

from one color model to another is particularly useful in extracting meaningful change measures. Each change measure is not calculated per building. Instead, we divide each building into a grid of cells, extract the features and calculate the damage measure for each cell. See Figure 6.5 for an example. This is done in order to capture localization and spatial proximity.

6.2.1 Edge Density

When a roof collapses, the image shows an increase in the number of new edges and non-linearity of existing edges. An edge detection performed on the difference of before-and-after images can capture this change in appearance of edges. We
Figure 6.5. A grid is placed over (a) before event building image and (b) after disaster building. (c) Features are extracted for the building. (d) Damage measure is calculated for each cell of the grid.

propose edge density as a measure of number of new or changed edges that appear per unit area. First before-and-after images are converted into grayscale. Then the absolute difference of the before-and-after images is computed. A binary thresholding of this difference image will set all pixels above a certain value to 255 and 0 otherwise. This thresholding will eliminate all non-relevant or minor changes in edges. Next, a Canny edge detection [Canny (1986)] is performed on the binary image. An example of before and after disaster images is shown in Figure 6.6 and Figure 6.7. See Figure 6.8 for absolute differencing and edge detection result. Finally, the number of pixels classified as edges in a rectangular cell is divided it by the size of the cell. This will give an edge density value per each cell of the grid. If this measure is high, then it is more likely that the roof has collapsed.
6.2.2 HSV Measures

As discussed earlier, conversion into HSV color model helps identify features that intuitively have more discriminating capacity. The RGB to HSV conversion is done as follows:

\[
\begin{align*}
    h &= \begin{cases} 
        0, & \text{if } max = min \\
        (60^\circ \times \frac{g-b}{max-min} + 0^\circ) \text{mod}360^\circ, & \text{if } max = r \\
        60^\circ \times \frac{b-r}{max-min} + 120^\circ, & \text{if } max = g \\
        60^\circ \times \frac{r-g}{max-min} + 240^\circ, & \text{if } max = r 
    \end{cases} \\
    s &= \begin{cases} 
        0, & \text{if } max = 0 \\
        \frac{max-min}{max} = 1 - \frac{min}{max}, & \text{otherwise} 
    \end{cases}
\end{align*}
\]

(6.4) (6.5)
Figure 6.7. Buildings extracted from (a) before and (b) after 2009 Haiti earthquake images shown in Figure 6.6.

\[ v = \max \]  \hspace{1cm} (6.6) \]

The \( \max, \min \) denotes the maximum and minimum of RGB values respectively. The V (value) component can be used to discriminate roofs with cracks, openings or holes. This happens when the roof structure is partially or completely destroyed, leaving a huge cavity. Consider the V component of pixels and for each cell create a histogram with \( n \) bins of V values. A Bhattacharya distance of histograms \( H_1 \) and \( H_2 \) of respective before/after cells respectively is computed as:

\[
d(H_1, H_2) = \sqrt{1 - \sum_I \frac{\sqrt{H_1(I)H_2(I)}}{\sqrt{\sum_I H_1(I)} \sum_I H_2(I)}} \] \hspace{1cm} (6.7) \]

The distance \( d(H_1, H_2) \) is calculated for each cell. If this measure is high it is more likely that the cell corresponds to a cavity or a hole developed after the
Figure 6.8. (a) Absolute difference of the before and after images. (b) Result of Canny edge detection.

We call this measure $V$ histogram.

The Hue component corresponds to the color value of a pixel. The variation in color that occurs in the case of milder damage like removed tiles, exposed decking etc are noticeable in the hue spectrum. The hue value varies from $0^\circ$ to $360^\circ$. The distance between two hue angles can tell us how different the colors are. However notice that the distance should always correspond to the smaller angle between the two hue values. Hence we compute another measure, the hue means, to reflect the change in hue spectrum. We define Hue means as the distance between means of hue components for each cell in before-and-after images.

6.3 Damage Classification

The different edge and color measures proposed can be used to classify buildings in an image into categories ranging from no damage to severely damaged. The
damage classification process proposed is a supervised learning approach that uses the object-based change measures as features for classifying damage into different qualitative states.

6.3.1 Learning Algorithms

Supervised machine learning is the search for algorithms that reason from externally supplied instances to produce general hypotheses, which then make predictions about future instances (quoted from Kotsiantis [2007]). In other words, the goal of supervised learning is to build a general hypothesis that models the distribution of class labels in terms of predictor features. The resulting classifier is then used to assign class labels to the testing instances where the values of the predictor features are known, but the value of the class label is unknown. In this evaluation we experimented with many supervised learning algorithms for predicting the damage state of buildings. Supervised learning algorithms include decision trees, rule learners, perception based techniques, Bayesian Networks, Support vector machines etc.

In addition to these classifiers, we experimented with ensemble of classifiers. An ensemble is an aggregation of predictions of multiple classifiers with the goal of improving accuracy. Evaluating the prediction of an ensemble typically requires more computation than evaluating the prediction of a single model, so ensembles may be thought of as a way to compensate for poor learning algorithms by performing a lot of extra computation. Common types of ensembles include bagging, boosting and bucket of models.
Bootstrap aggregating, often abbreviated as bagging, involves having each model in the ensemble vote with equal weight. In order to promote model variance, bagging trains each model in the ensemble using a randomly-drawn subset of the training set. As an example, the random forest algorithm combines random decision trees with bagging to achieve very high classification accuracy \cite{Breiman1996}. Boosting involves incrementally building an ensemble by training each new model instance to emphasize the training instances that previous models misclassified. In some cases, boosting has been shown to yield better accuracy than bagging, but it also tends to be more likely to over-fit the training data. By far, the most common implementation of Boosting is Adaboost. A bucket of models is an ensemble in which a model selection algorithm is used to choose the best model for each problem. When tested with only one problem, a bucket of models can produce no better results than the best model in the set, but when evaluated across many problems, it will typically produce much better results, on average, than any model in the set \cite{Zenko2004}.

6.3.2 Localization and Spatial Proximity

Now we have three features for each cell of a grid; the edge density, V histogram and H means (See Figure 6.9). A grid is placed over each building such that it is a rectangle of minimum area that entirely circumscribes the building. We could average the values over all the cells for each building, but that would fail the purpose for which we use a grid. The grid is used with the intention of finding damages which are partial and localized, such as a partial collapse, removed tiles etc. The values for such partial damage will be averaged over the larger area which is not collapsed unless we use a grid of cells. But we do not want to average values
of cells either. Instead, the proposed solution is as follows. Firstly, all features are normalized to lie between 0 and 1. Discretize value of each feature for each cell into $k$ intervals. For each feature, see which interval the cell belongs to and total the number of cells which belong to each of the $k$ intervals. For each feature, we have thus $k$ totals corresponding to the $k$ bins for each building. Discretize the totals to a value between 0 and 1. Now we have a total of $k \times 3$ features per building.

![Figure 6.9](image)

Figure 6.9. The first row shows a collapsed building and the corresponding false color images for edge density, V Histogram and H means. Notice that the edge density values are significantly higher in this case. The second row corresponds to a partially damaged building with a cavity in the rooftop. V histogram indicates a significant change in the cavity area and H means shows a minor damage on the roof.

This process captures localization; i.e., small areas of damage, but not neces-
sarily the spatial proximity of damage. Cells of a bin which corresponds to partial collapsed damage are more likely to be present together than to be distributed. This property can be captured by computing the proximity measures for each cell. Proximity of a cell is defined as the number of neighboring cells in the 8-connected neighbors which lie in the same bin/interval as the cell. For each of the $k$ bins, find the totals of proximity measures for cells that belong to each bin. Thus we have another $k \times 3$ features for each building making a total of $k \times 6$ features. The higher the proximity total of a bin is, the more likely the cells belonging to that bin are indeed reflecting the damage state of that bin.

6.3.3 Evaluation

The NOAA images from Florida used for this evaluation are not geo-referenced and ground truth determination for these images needs to be done manually. The building extraction process described in previous section aids this partially. However the automatic extraction process may contain noise/ errors which need to be corrected manually. The ground truth preparation is further complicated by the ability to visually inspect the damage states and make a correct judgment of the appropriate damage state. The labeling is done into one of the various classes based on the damage metric scale used. Table 6.2 shows a simple 3-scale metric. However a more expanded and fine-grained 4-scale damage metric similar to the one used in [Womble (2005)] is shown below in Table 6.3

A total of 150 buildings have been labeled into scales mentioned in Table 6.2 and Table 6.3. 75 buildings were used in the training set and another 75 in the testing set. We experimented with decision trees, rule learners, perceptron based
techniques, Bayesian Networks, and ensembles of classifiers and report only the top two performers here. An ensemble of J48 trees with AdaBoosting performed the best for the 4-scale damage metric while Random Forrest performed better for the 3-scale damage metric. The best corresponding accuracies were 72% and 80% respectively. The confusion matrix in Table 6.4 corresponds to the Table 6.4 that was obtained through AdaBoosting. It is evident that the predicted values for all three classes were mostly correct. However for the 4-scale damage metric in Table 6.5 we see that the predictions of intermediate classes (B and C) is affected adversely. Performance of Random Forests with varying number of trees is shown in Figure 6.10. The performance is shown in terms of the overall accuracy and the average of TP (true positives) of all the classes. We observe that for 3-scale metric we find a consistent 80% accuracy when the number of trees is between 6 and 10. Performance of AdaBoosting with varying number of iterations is shown in Figure 6.11. In this case, the peak of accuracy for 4-scale is achieved at 3 iterations and for 3-scale at 7 iterations. In both classifiers, the larger gap between Average TP and Accuracy for 4-scale as compared to 3-scale indicates that the predictions for intermediate classes are more adversely affected for the 4-scale metric.
TABLE 6.3

A 4-SCALE DAMAGE METRIC

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>No damage</td>
</tr>
<tr>
<td>B</td>
<td>Partial 'minor' damage. This indicates shingles/tiles exposed, non-linear edges appear</td>
</tr>
<tr>
<td>C</td>
<td>Partial 'major' damage. This indicates decking removed leaving roof structure exposed, holes in roof, change in elevation and/or partial collapse</td>
</tr>
<tr>
<td>D</td>
<td>Total collapse, Missing rooftop/building</td>
</tr>
</tbody>
</table>

TABLE 6.4

A CONFUSION MATRIX FOR 3-SCALE METRIC WITH RANDFOREST, NUMBER OF TREES = 7

<table>
<thead>
<tr>
<th>Predicted</th>
<th>Actual</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>A</td>
<td>35</td>
<td>3</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
TABLE 6.5

A CONFUSION MATRIX FOR 4-SCALE METRIC WITH ADABOOST, NUMBER OF ITERATIONS = 3

<table>
<thead>
<tr>
<th>Predicted</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>32</td>
<td>5</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>15</td>
</tr>
</tbody>
</table>
Figure 6.10. Performance of RandomForests with different number of trees for (a) 3-scale damage metric and (b) 4-scale damage metric.
Figure 6.11. Performance of AdaBoost with different number of iterations for (a) 3-scale damage metric and (b) 4-scale damage metric.
CHAPTER 7

CONCLUSION

This thesis presented a first-of-its-kind automatic system for the assessment of damage from high resolution imagery. Towards that end, we have proposed and tested techniques for image registration, building extraction, change detection and damage classification; each of which require little or no manual supervision. While the accuracy in previous approaches were limited by the number of control points in manual registration [Bitelli et al.] (2004), our application of SURF-based feature detection was found to produce near perfect registration. We proposed a novel segmentation-based building detection algorithm. Unlike previous approaches [Sir-macek and Unsalan] (2009), [Noronha and Nevatia] (2001), [Song et al.] (2006) we do not make the assumption that buildings are composed of homogenous segments or linear edges. Our algorithm was able to accurately extract the boundary contours of buildings in a reasonable amount of time. We proposed object-based change detection measures that reflect the kind of damages that occur after a windstorm. We used a combination of edge-based and color-based measures to classify damage into qualitative states, whereas previous methods [Adams et al.] (2004), [Matsuoka et al.] (2004), [Womble et al.] (2007) do not combine the two. The final results in classification were promising; 80% accuracy for a 3-scale damage metric and 72% accuracy for a fine grained 4-scale damage metric.
While the proposed system attempts at automating every aspect of damage assessment, there are several areas which require manual supervision. Further, there are areas which require improvement in terms of computational time and robustness. All these provide new challenges that must be addressed in future work. In image registration a major area that requires improvement is the interest point matching process. We could speed up the neighborhood matching algorithm by using efficient data structures and geometric properties of points. For the building detection process the performance is heavily dependent on the quality of segmentation. This can be achieved by either increasing the training data for our maximum-likelihood with clustering scheme or adopting a different segmentation technique. Also, the segment combination process is computationally intensive. Using additional features to identify possible locations for buildings could help reduce search space. Lastly, for damage classification, we could use additional features such as boundary compactness, perceptual feature organization etc. to improve the performance. We also need to evaluate the quality of ground truth and its impact on performance by using multiple volunteers to label the data-set. Expanding our existing ground truth data-set is also an important initiative that could benefit future experiments.


