COMPUTER VISION TECHNIQUES FOR DAMAGE ASSESSMENT FROM HIGH RESOLUTION REMOTE SENSING IMAGERY

A Dissertation

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

Doctor of Philosophy

by

Jim Thomas

______________________________
Kevin Bowyer, Co-Director

______________________________
Ahsan Kareem, Co-Director

Graduate Program in Computer Science and Engineering
Notre Dame, Indiana
December 2012
COMPUTER VISION TECHNIQUES FOR DAMAGE ASSESSMENT FROM HIGH RESOLUTION REMOTE SENSING IMAGERY

Abstract

by

Jim Thomas

Techniques in post-disaster assessment from remote sensing imagery have been studied by different research communities in the past decade. Such an assessment benefits everybody from government organizations and insurance agencies to individual home owners. This work explores the application of existing and novel computer vision algorithms for an automated damage assessment caused by windstorm. The various subproblems studied include geometric and photometric correction, rooftop recognition and change classification based on textural differences. Past work done in this area by remote sensing, geoscience, civil engineering and image processing communities had established that the problems addressed in these areas were challenging and largely unsolved. The solutions proposed in this work are strongly motivated towards building a system capable of fast, robust and fine-grained damage analysis from aerial or satellite imagery. The algorithms introduced are thoroughly evaluated and compared with previous works. The results demonstrate that this work promises higher leaps in the field of automated damage classification and provides insights into the reliability of such analysis in real world scenarios.
FIGURES

4.1 An unregistered after and before storm image pair (above) is downsampled (below)   25
4.2 The image pair in Figure 4.1 after application of hanning window   25
4.3 The cross power spectrum of the image pair in Figure 4.2   26
4.4 Features extracted from (a) before and (b) after disaster images shown in red circles   30
4.5 Matching between 10 features in (a) before and (b) after disaster shown in red lines   38
4.6 Registered (a) before and (b) after disaster images. Non-overlap area is filled with black pixels   40
4.7 Time taken for matching points at different resolutions using best bin first (BBF) and Naive NN search   42
4.8 Time taken for matching points at different resolutions using randomized kd-trees, heirarchial k-means trees and Naive NN search   43
4.9 The approximately registered image pair from Figure 4.1 (above) is divided into 100 × 100 grid cells and features are matched across these cells (below)   45
4.10 (Above) Notice the perspective effects on the grids after transformation. (Below) The final registered images are shown   46
4.11 (above) Before damage Google Earth Image from Joplin, Missouri. (below) After damage NOAA image from Joplin tornado path   51
4.12 The image pair shown in Figure 4.11 registered using our proposed approach   52
4.13 (above) Before storm USGS image from Galveston, Texas. (below) NOAA image from Galveston, Texas after Hurricane Ike   53
4.14 The image pair shown in Figure 4.13 registered using our proposed approach   54
5.1 (Top left) Source image (after storm). (Top right) Target image (before storm). (Bottom) New target image after applying color balancing using the proposed algorithm   57
5.2 \textit{(Top left)} Source image. \textit{(Top right)} Target image. \textit{(Bottom left)} 11X11 windows, artifacts visible. \textit{(Bottom right)} 101X101 windows, no artifacts visible. ................................................................. 61
5.3 Color similarity measures for different algorithms. .................... 67
5.4 SSIM for different color balancing algorithms. ......................... 68
5.5 \textit{(1st column)} Source images. \textit{(2nd column)} Target images. \textit{(3rd column)} New target image transformed using gain compensation. \textit{(4th column)} New target image transformed using the proposed adaptive windowing based local color transfer. ......................... 69
6.1 A 4077 \times 4092 NOAA image from Pensacola Florida. ............ 81
6.2 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% ....................... 82
6.3 \textit{(a)} A parking lot \textit{(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. .......................................................... 86
6.4 \textit{(a)} A building \textit{(b)} The labeled numbers indicate the MEPS value for the segment classified as the building. Note that MEPS = 0.96 for the building. ................................. 87
6.5 \textit{(a)} A few buildings \textit{(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. .................. 88
6.6 A before storm NOAA image from Pensacola, Florida ............. 91
6.7 Buildings extracted from Figure 6.6 is shown in white. \(C_{th} = 0.4, \ M_{th} = 0.4\) and \(A_{th} = 100\) pixels ................................. 92
6.8 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 . 93
6.9 Buildings extracted from Figure 6.8 is shown in white. \(C_{th} = 0.3, \ M_{th} = 0.3\) and \(A_{th} = 100\) pixels ................................. 94
6.10 \textit{(a)} Positive images and \textit{(b)} Negative images used in training set. 97
6.11 Buildings extracted from Figure 6.6 using HOG-based detection are marked with rectangles. .................................................. 98
6.12 Buildings extracted from Figure 6.6 using the probabilistic framework algorithm are marked with yellow and blue labels. .................. 100
6.13 Buildings extracted from Figure 6.7 using the probabilistic framework algorithm are marked with yellow and blue labels. .......... 101
6.14 An aerial image from Pensacola, Florida with distinctly shaped rooftops. .................................................. 104
6.15 The meanshift filtered version of Figure 6.14 ................................................................. 105
6.16 The meanshift segmented version of Figure 6.14. The different regions are shown in different colors. .................................................. 106
6.17 The shadows extracted from 6.14 using our proposed technique is shown as white pixels. .................................................. 108
6.18 The shadow confidence map extracted from 6.14 using MEPS values of individual segments. .................................................. 110
6.19 The shape confidence map extracted from 6.14 using compactness values of individual segments. .................................................. 111
6.20 Probable building segments extracted from 6.14 using shadow and shape confidence maps. .................................................. 112
6.21 Probable non-building segments extracted from 6.14 using shadow and shape confidence maps. .................................................. 113
6.22 A false color image representing hue, saturation and edges extracted from Figure 6.14 as RGB channels. .................................................. 114
6.23 Examples of labels assigned using EM algorithm trained with 5 gaussians, 5 features on the generated samples from 6.14. The 3 labels are shown in red, green and blue. .................................................. 117
6.24 The derivative in x direction computed from 6.14 and used as a smoothing cost for graph cut optimization. .................................................. 119
6.25 The derivative in y direction computed from 6.14 and used as a smoothing cost for graph cut optimization. .................................................. 120
6.26 Rooftops extracted using our proposed algorithm. .................................................. 121
6.27 (1st column) Rooftop images from Pensacola, Florida, New Orleans, Louisiana and Haiti. (2nd column) The blue and yellow markers denote the rooftops detected using the probabilistic framework approach. (3rd column) Rooftops extracted using our proposed algorithm. Complex shapes of varying texture and color are extracted with precision. .................................................. 123
6.28 (1st column) Image with rooftops of multiple colors from Pensacola, Florida. (2nd column) The blue and yellow markers denote the rooftops detected using the probabilistic framework algorithm. (3rd column) Rooftops extracted using our proposed algorithm. .................................................. 124
6.29 (1st column) Rooftop image of an urban location in Florida (2nd column) Rooftops extracted using our proposed algorithm. (3rd column) The blue and yellow markers denote the rooftops detected using the probabilistic framework algorithm. .................................................. 124
7.1 A grid is placed over (a) before event building image and (b) after disaster building. (c) Features are extracted for each cell. (d) Each feature calculate per cell is quantized into bins as indicated by the intensity of the heat map. A histogram of these cells is then typically computed.

7.2 A heat map displaying severity of damage found by variance and uniformity features, with more severe damages shown closer to red. (First column) Before storm image, (Second column) after storm image (Third Column) Variance, and (Fourth Column) Uniformity.

7.3 A heat map displaying severity of damage found by PCC and LBP features, with more severe damages shown closer to red. (First column) Before storm image, (Second column) after storm image (Third Column) PCC, and (Fourth Column) LBP.

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

7.5 Buildings extracted from (a) before and (b) after 2009 Haiti earthquake images shown in Figure 7.4.

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

7.7 Before and after storm images from (top) Florida and (bottom) Texas are shown. The gradient magnitude images extracted around each rooftop shows how strong gradients in certain directions change significantly if the rooftop is missing.

7.8 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.

7.9 RS Damage scale for hurricane damages described in Womble (2005).

7.10 The first row shows before storm images. The second row shows after storm images. The first column shows an image pair where the exact state of the rooftop is ambiguous and hard to discern. The tiles are removed but it is unclear whether there are holes or cavities. The second column is an example where the entire rooftop structure is missing and yet the building is standing as evidenced by the shadows. The column shows either a collapsed or missing rooftop. The exact nature of damage is unclear though it is evident that it is serious.
7.11 A J48 tree trained on the 635 rooftops dataset to predict RS damage scale. The leaves represent A to D categories in the RS damage state and nodes represent features. These features are building missing ($F_0$), cavity presence ($F_1$), missing tiles ($F_2$) and collapse state ($F_3$).

7.12 The actually and predicted damage categories for some of the images in the dataset are shown here. All the misclassifications are marked in red.

7.13 Effect of color balancing on Hue Means feature is shown in this image. After the color correction, only exact places where tiles are removed are shown as severely damaged.

7.14 A left-right rectified stereo image pair of an urban scene.

7.15 Disparities computed using (a) Graph Cut, (b) Hirschmuller algorithm and (c) line growing algorithm.

7.16 A left (above) and right (below) rectified stereo image pair from Joplin, Missouri.

7.17 Disparities computed from Figure 7.16 using (top) Graph Cut, (middle) Hirschmuller algorithm and (bottom) line growing algorithm.

7.18 A meanshift segmented version of the urban area images.
TABLES

2.1 SUMMARY OF RELATED WORK. .............................................. 16
3.1 SPECIFICATIONS OF THE DATA-SET. ................................. 18
4.1 A COMPARISON OF PERFORMANCE OF VARIOUS COLOR AND GRAYSCALE DESCRIPTORS ........................................... 35
4.2 REGISTRATION SUCCESS AT VARYING IMAGE RESOLUTIONS ................................................................. 48
4.3 AVERAGE REGISTRATION ERROR AND EXECUTION TIME AT 50CM RESOLUTION ...................................................... 49
5.1 LIST OF ALGORITHMS USED IN OUR STUDY. ..................... 66
6.1 PERFORMANCE OF BUILDING DETECTION ........................... 95
6.2 PERFORMANCE OF HOG-BASED BUILDING DETECTION .......... 98
6.3 PERFORMANCE OF PROBABILISTIC FRAMEWORK FOR BUILDING DETECTION ...................................................... 100
7.1 APPEARANCE BASED QUALITATIVE CATEGORIES FOR DAMAGE CLASSIFICATION .................................................. 149
7.2 AGREEMENT% OF VISUAL INTERPRETATION BY TWO GROUPS OF VOLUNTEERS. ......................................................... 151
7.3 CLASS DISTRIBUTION OF APPEARANCE BASED DAMAGE STATES IN THE 635 ROOFTOPS DATASET. .............................. 155
7.4 BEST CLASSIFICATION PERFORMANCE FOR DIFFERENT FEATURE SELECTION SCHEMES. THE MEANS OF CLASS ACCURACY, PRECISION AND RECALL FOR EACH CATEGORY ARE REPORTED. ................ 158
7.5 BEST CLASSIFICATION PERFORMANCE FOR DIFFERENT PREPROCESSING SCHEMES. THE VARIOUS RESULTS REPORTED INCLUDE NO PREPROCESSING (noprepro), AFTER COLOR BALANCING (cb), AND AFTER COLOR BALANCING PLUS ROOFTOP REGISTRATION (cb + rr). MEAN ACCURACY, F-MEASURE AND ROC AREA FOR EACH CLASS ARE REPORTED. ................................................. 159

7.6 A CONFUSION MATRIX FOR PREDICTING RS-SCALE FOR WINDSTORM DAMAGE ................................. 161
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. Patrick Flynn, Dr. Aaron Striegel and Dr. Scott Emrich, 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 would like to thank Dr. A. Womble for providing some of the sample images used in his work and for sharing hurricane images. 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 the One who makes all things possible!
CHAPTER 1

INTRODUCTION

1.1 Motivation

Damage assessment following natural disasters is a compelling application for computer vision technology. In recent times, images of affected areas are easily obtained through satellite or aerial sensors. There are many parties who are interested in damage assessment immediately following a disaster. This includes individual homeowners, local authorities, Federal Emergency Management Agency and insurance companies. This 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. Assessment from such images can also assist in providing rapid loss estimates.

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 Atkinson (2010). Prior to 1999, earth-observing satellites acquired 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 these were neither able to resolve components of individual buildings nor to discern temporal changes (damages) in individual buildings.

Since 1999, a new generation of commercial imaging satellites with resolutions of 1 m or high has enabled the rapid and automated detection of physical changes in individual buildings by comparing pre-event and post-event images 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. Work done in the past decade Yamazaki (2001), Adams et al. (2004), Matsuoka 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 opportunities for using this technology were with Hurricanes Charley and Ivan in 2004 Womble (2005).

As reported by the New York Times (Sept. 5, 2005), a widespread use of aerial images for rapid damage assessment was conducted in the aftermath of Hurricane Katrina. Hundreds of displaced residents and their relatives turned to the Internet for information about a home feared damaged or destroyed. Many used Google Earth, a program available at the Google web site that lets users zoom in on any address for an aerial view drawn from a database of satellite photos. A grass-roots effort had identified scores of post hurricane images, determined the
geographical coordinates and visual landmarks to enable their integration into
the Google Earth program, and posted them to a Google Earth bulletin board.
Most of the images originated with the Remote Sensing Division of the National
Oceanic and Atmospheric Administration (NOAA), which had been posting them
to its web site. Taking inspiration from the online volunteers, Google, NASA and
Carnegie Mellon University made the effort more formal, incorporating thousands
of post-hurricane images into the Google Earth database for public use.

NOAA remote sensing continued to acquire post-hurricane images including
images along with images from other sources such as Digital Globe satellite images,
are made available for public use immediately after a hurricane. This is also true
for other disasters such as earthquakes and tornadoes. However, much of the work
done in assessment is still manual and strongly motivates the need for a robust,
fast and efficient system that can do all the processing automatically.

1.2 Problem Statement

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 im-
ages. Certain visual signatures of damage could be used to categorize damage
into categories such as no damage, removed tiles, removed decking, partially col-
lapsed, 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 propose the use of existing and new computer vision algorithms for pre-processing the images, extracting buildings, and the identifying level of damages. As the first step in pre-processing the before-and-after storm images, we propose to use phase and feature-based image registration. This research explores fast registration of images with low overlap percentage and large amount of noise. As a second step in pre-processing, we apply color balancing to correct color differences between two images of the same scene. A local color transfer based algorithm is proposed. For building extraction, existing and new building detection algorithms are explored. A novel supervised segment grouping-based algorithm that extracts rooftops of a broad range of variations in shape, texture or color is described and evaluated. To compensate for the deficiencies of this approach, another unsupervised algorithm based on Graph Cut optimization is also proposed. Lastly, this work evaluates the reliability of aerial images in damage assessment and identifies discriminatory features that are useful for classifying rooftops into qualitative damage states.

We make minimal assumptions about the quality of available data-set. 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 aerial or satellite images. These images are taken under different lighting conditions and seasons of the year. We assume all images to have 70cm or finer resolution. Further, we assume that for registration a perspective transform model is sufficient and the effect of tall buildings can be
1.3 Outline

First, we present a broad literature review in Chapter 2. The previous work that has been done in the fields of earthquake and hurricane damage assessments are discussed. Note that we postpone the discussion of work directly related to the various subproblems until their respective chapters. Chapter 3 elaborates on the sources and specifications of data used in this work. Chapters 4 and 5 focus on preprocessing steps that are important in getting the dataset ready for damage assessment. The proposed coarse and fine registration algorithms for geometric correction of images are discussed in Chapter 4. Next, the application of a new color transfer algorithm that corrects the photometric differences between image pairs are presented in Chapter 5. The shortcomings of previous approaches and two new building detection algorithms are presented in Chapter 6. In Chapter 7, we describe a study that was conducted on estimating the reliability of remote sensed images in fine grained damage analysis and compare the performance of various existing and novel features for classifying the damage state. Additionally, we briefly explore the possibility of using stereoscopic Aerial imagery for disparity estimation and damage detection. Finally, Chapter 8 summarizes the main contributions of this work and discusses directions for further exploration.
CHAPTER 2
RELATED WORK

Previous works in damage assessment from images vary in the nature of the disasters analyzed (earthquakes, hurricanes, tornadoes, landslides etc), approaches to classification (pixel-based or object-based), type of images used (Aerial TV, Optical, SAR etc.) and scales of damage identified (no damage/damaged, RS scale etc). Ghosh et al. (2011) describes how the Global Earth Observation Catastrophe Assessment Network (GEO-CAN) was formed to facilitate a rapid damage assessment after the 12 January 2010 Haiti earthquake. GEO-CAN uses crowd sourcing for remote sensing-based damage interpretation and represents a new paradigm in post-disaster damage assessment. The GEO-CAN community, working with the World Bank (WB), the United Nation Institute for Training and Research (UNITAR) Operational Satellite Applications Programme (UNOSAT) and the European Commissions Joint Research Center (JRC) led the way for a rapid Post Disaster Needs Assessment (PDNA) utilizing remote-sensing based analysis as the primary source of information for building damage.

The work done by GEO-CAN involved crowd sourcing of damage classification to expert volunteers. The volunteers were asked to grade the level of damage in high-resolution and very high resolution aerial imagery using a web portal. They also made use of pictometry or oblique images to confirm the state of level of...
damage. After comparison with actual data collected by JRC, this study was found to produce nearly 78% total accuracy. This recent interest in damage assessment through crowdsourcing is a good indication of the difficulty in automatic assessment and the insufficiency of existing research. The rest of this section highlights many of the prominent studies done in earthquake and windstorm damage assessment in the past decade.

2.1 Earthquake Damage

Some of the earlier work takes a pixel-oriented approach where the damage map is created by a purely pixel-based analysis between before and after damage images. This is useful for a high-level analysis of various geographic areas. The use of optical images for the pixel-oriented approach was discussed in [Yamazaki 2001]. The pixel-based approach uses pixel values of each band in the multi-spectral imagery. Because the pixel values in the satellite images vary depending on the observation and surface conditions, they suggested the use of a normalization step. This was done in order to eliminate the seasonal difference in lighting conditions. The normalization involved band ratioing, which is defined as the ratio of the pixel value of each band to the value of a reference band. 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 pixel values, clouds vegetation etc were removed. Slight damage and no damage of buildings were selected from the images to characterize the pixel value 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)](#), [Matsuoka 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 [Adams et al. (2004)](#).

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 e-Cognition 3.0 (a software by Definiens Imaging). The first operation performed in e-Cognition 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. The experiments involved classifying built-up area into 3 levels of damage using algorithms in e-Cognition and ERDAS software. Based on accuracy in classifying the built-up area of an image 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 as well as correlation analysis displays the best agreement with a manually prepared damage map Chen and Hutchinson (2005).

The case of the earthquake of Bam is studied in Chesnel et al. (2007). It uses two very high resolution images and focuses on the footprints of the buildings. The need for an accurate registration of the buildings is demonstrated; a
registration method that improved the damage assessment is proposed. Using an object based assessment and correlation coefficients as features, it achieved a classification performance of the buildings among four damage grades up to 69%. The impact of a lower accuracy of the buildings roofs segmentation was evaluated. The study showed that it mainly leads to a decrease of the capacity to identify the partial damage on buildings.

Ehrlich et al. (2009) studied the effectiveness of SAR and VHR imagery in identifying damages after earthquakes. Aerial photography with spatial resolution of 0.5 m or better was shown to be the more effective tool for damage assessment. If acquisition is possible it is likely to remain the data source of choice for damage assessment. This study also showed that even on the highest resolution SAR image products available, damage assessment accuracy was low. Li et al. (2010) trained an SVM with cross-correlation based features to classify damaged vs undamaged areas in VHR images of earthquake areas and reported 82% overall accuracy.

2.2 Windstorm Damage

Sampath (2004) presented one of the earliest works in remote sensing for windstorm damage assessment. Registration was done manually at a coarser scale and using cross-correlation at a finer scale. Segmentation was used to divide the image into areas of interest and simple grayscale statistical measures were used to detect change.

Womble et al. (2006) classified building damage resulting from Hurricanes Charley and Ivan into four qualitatively categories according to their roofing sys-
tem (type of roof construction). They used roofing system 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.

[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 [Womble et al. (2007)] 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. Comparison of before-and-after object level statistics (such as by differencing or ratioing) resulted in damage metrics, which numerically described 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)]. [Womble et al. (2008a)] described challenges in fully automated damage assessment using temporal remote-sensing image sequences. In [Womble et al. (2008b)], storm-surge actions on residential buildings along the Mississippi Coast were studied, and the analysis of remote-sensing imagery along with related ground surveys helped to
identify visual signatures of storm-surge damage to residences, thereby bolstering the library of visual damage signatures required for automation of damage assessment.

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 was 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.

A unique and different approach to damage classification 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 showed 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.

More recently, Vijayaraj et al. (2008) used a pixel-based approach rather than an object-based one. The study used local binary pattern (LBP), local edge pattern (LEP) and Gabor texture features computed over pixels and blocks of pixels. The features were compared for changes by comparing the angle between the principal components of the feature vector. This was used to identify no damage and significantly damaged areas. In Benedek and Szirnyi (2010) a new probabilistic method which integrates building extraction with change detection in remotely sensed image pairs was proposed. A Marked Point Process framework was proposed for building extraction in remotely sensed image pairs taken with significant time differences. The method incorporates object detection and low level change information in a joint probabilistic approach.

In Bishop (2010) the utility of damage detection products, derived using object-based image analysis, for use in preliminary damage assessment for hurricane disasters, the degree to which these products detect damage to residential structures, and the viability of a point-polygon accuracy assessment as an alternative to a traditional pixel-based accuracy assessment were discussed. The approach classified damage into not damaged and damaged states. In Myint et al. (2008) a pixel based approach employed several geospatial approaches, specifically the Getis index, Geary C, and two lacunarity approaches to categorize damage characteristics according to the original Fujita tornado damage scale (F-scale).
results indicated strong relationships between spatial indices computed within a local window and tornado F-scale damage categories identified through the ground survey.

Szantoia et al. (2011) developed a tool to detect downed trees and debris volume to better aid disaster response efforts and tree debris removal. The tool estimates downed tree debris volume in hurricane affected urban areas using a Leica Airborne Digital Sensor (ADS40) and very high resolution digital images. The tool employs a Sobel edge detection algorithm combined with spectral information based on color filtering using 15 different statistical combinations of spectral bands. The algorithm identified downed tree edges based on contrasts between tree stems, grass, and asphalt and color filtering was then used to establish threshold values.

Several recent work used only post-disaster images. Sirmacek and Unsalan (2009a) used color-invariants to segment buildings. Estimating illumination direction using centers of buildings and shadows, this algorithm found other buildings using estimated shadows. Ratio of rooftop area to shadow region area was used as a damage metric. 14 buildings were used in the study to classify buildings as damaged or not damaged. Radhika et al. (2010) used wavelet feature extraction to enhance features suggested by Womble et al. (2007). Wavelet feature extraction of the standard deviation and maximum value of RGB pixel values, and also of edge intensities were found to produce better discrimination between classes. However the study conducted building extraction manually. The features were trained using ANN classifier and results on 11 buildings were reported. Radhika
studied tornado damage to building structures. Tornado path was found using texture-wavelet analysis. Standard deviation, entropy and peak value of wavelet features were used to train ANN into 4 damage scales to create the damage map. Buildings were extracted using color invariant features. However no evaluation of building extraction or damage classification was reported. See Table 2.1 for a summary of damage levels, accuracy and total images used in studies mentioned in this Chapter. Note that many of the studies did not report accuracies or total number of rooftops used in their dataset. It can be concluded that most previous studies were not comprehensive and that automated damage classification is largely an unsolved and less studied problem.
<table>
<thead>
<tr>
<th>Paper</th>
<th>Damage Levels</th>
<th>Accuracy</th>
<th>Total buildings, images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yamazaki (2001)</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Matsuoka et al. (2004)</td>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Adams et al. (2004)</td>
<td>3</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Bitelli et al. (2004)</td>
<td>3</td>
<td>60%</td>
<td>-</td>
</tr>
<tr>
<td>Chen and Hutchinson (2005)</td>
<td>3</td>
<td>74%</td>
<td>190 buildings, 1 image-pair</td>
</tr>
<tr>
<td>Chesnel et al. (2007)</td>
<td>4</td>
<td>69%</td>
<td>3 image-pairs</td>
</tr>
<tr>
<td>Womble et al. (2006)</td>
<td>4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Friedland (2009)</td>
<td>4</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Vijayaraj et al. (2008)</td>
<td>2</td>
<td>-</td>
<td>2 image-pairs</td>
</tr>
<tr>
<td>Myint et al. (2008)</td>
<td>2</td>
<td>-</td>
<td>1 image-pair</td>
</tr>
<tr>
<td>Sirmacek and Unsalan (2009a)</td>
<td>2</td>
<td>80%</td>
<td>15 buildings, 2 image-pairs</td>
</tr>
<tr>
<td>Radhika et al. (2010)</td>
<td>4</td>
<td>92%</td>
<td>11 buildings, 1 image-pair</td>
</tr>
<tr>
<td>Radhika et al. (2011)</td>
<td>4</td>
<td>-</td>
<td>1 image-pair</td>
</tr>
</tbody>
</table>
CHAPTER 3

DATA

The aerial imagery used for this research was acquired from the NOAA Aerial storm imagery and USGS Orthoimagery databases. NOAA 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) or coarser. Image file size is between 2 MB and 3 MB. Each image varies from 4077 pixels to 8000 pixels for height/width dimensions. The High Resolution Orthoimagery collection has been acquired by the USGS through contracts, partnerships with other Federal, state, tribal, or regional agencies, and direct purchases from private industry vendors. Since data comes from a variety of sources, the resolution, area of coverage, file size, and projection varies by dataset. The USGS EROS Center manages and distributes this orthoimagery, which includes black-and-white, natural color, color infrared, and color near-infrared. However all Orthoimagery images used in this study were between 60cm and 1m resolution. The images used were from Joplin tornado (2011), Hurricane Ike (2008), Hurricane Katrina (2005), Hurricane Dennis (2005) and Hurricane Ivan (2004) images of Joplin (Missouri), Galveston (Texas), Pensacola (Florida) and New Orleans (Louisiana).

In addition, 2 Google Earth images of the 2010 Haiti Earthquake and 3 QuickBird satellite image pairs of Punta Gorda, Florida from Digital Globe were also
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. In all, an assorted collection of large image pairs were used in this work. A summary of details about the dataset we compiled is presented in Table 3.1.

### TABLE 3.1

**SPECIFICATIONS OF THE DATA-SET.**

<table>
<thead>
<tr>
<th>Image Provider</th>
<th>Resolution, Size</th>
<th>Number of Image-pairs</th>
<th>Event, Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOAA Remote Sensing Division, USGS Orthoimagery</td>
<td>37cm-1m, approx 4000 px to 8000 px per dimension</td>
<td>23</td>
<td>Hurricane Katrina (2005), Hurricane Dennis (2005) and Hurricane Ivan (2004), Pensacola, Florida and New Orleans, Louisiana</td>
</tr>
<tr>
<td>Google Earth</td>
<td>50cm</td>
<td>1</td>
<td>Haiti</td>
</tr>
<tr>
<td>Digital Globe</td>
<td>60cm, 8000 X 10000 px</td>
<td>3</td>
<td>Punta Gorda, Florida</td>
</tr>
</tbody>
</table>
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. 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. The satellite or aerial images from before and after a disaster may have been taken by different cameras at different altitudes, angles and positions. By registering two images, such geometric differences can be corrected.

While image providers may provide registered or georeferenced images, often this is not the case. Further, as noted in Ghosh et al. (2011), even georeferenced images can have large registration errors which can significantly impact change detection. This section focuses on automatic registration of remote sensing imagery, particularly that of before and after storm images. Registering such images is challenging due varying overlap percentage, changes due to passage of time and storm impact. In this study we ignore the effects of tall buildings and specifically address the problem of finding the perspective transform required for fine image
registration. In general, we assume that the perspective transform model finds
better pixel to pixel correspondence for large high resolution aerial or satellite
imagery than simpler similarity transforms. In other words, the goal is to find a
transform that maps one arbitrary 2D quadrilateral into another. Mathematically,
this can be expressed as

\[
X = \frac{ax + by + c}{gx + hy + 1} \tag{4.1}
\]

\[
Y = \frac{dx + ey + f}{gx + hy + 1} \tag{4.2}
\]

where \((X, Y)\) represents the coordinates of the transformed quadrilateral and
\((x, y)\) represent coordinates of the target quadrilateral.

Thus the problem is reduced to finding the best 8 parameters \((a, b, c, d, e, f, g, h)\)
for calculating perspective transform. While there is no universal solution for find-
ing this model, this work proposes on combining phase correlation with feature-
based matching as a viable solution for change-detection applications. The basic
idea is to first register the images approximately before performing a fine registra-
tion. This approach works better than using pyramidal schemes for registering at
different resolutions as matching features at lower resolutions is shown to be less
robust. We first present a modified version of the the phase correlation to find
similarity transforms. This step introduces the robustness into our approach and
ensures that all images are atleast coarsely registered easily. Next, the problem of
feature-point matching is studied and novel modifications to standard matching
process ensures efficient fast matching. This step improves the accuracy for fine
registration. Thus the combined phase-feature algorithmic fusion is evaluated and
found to produce robust as well as accurate results.

4.1 Related Work

The most widespread approach for registering remote sensing imagery in the absence of georeferenced data is manual registration. Control points are manually picked from the images and a geometric model is computed. However, previous approaches in damage detection from remote sensing imagery were limited by the number of control points in manual registration [Bitelli et al. (2004)]. Even most recent works [Radhika et al. (2011), Womble et al. (2008b), Li et al. (2010)] do not propose a viable solution for automatic accurate registration. Some attempts made in the past have tried to used general registration techniques. [Sampath (2004)] used manual means for coarse registration and an algorithm that maximized cross-correlation for finer registration. In [Chesnel et al. (2007)] the registration method was based on estimates of the maximum a posteriori of the correlation computed on the group of pixels held in the roof footprints in both images.

A comprehensive survey of general image registration methods was published in 1992 by [Brown (1992)]. A more recent survey was published in 2003 by [Zitova and Flusser (2003)]. Various area-based and feature-based registration algorithms were described and compared in this work. A comparison of image registration algorithms purely for remote sensing imagery was presented in [Moigne et al. (1998)]. They evaluated spatial correlation, phase correlation, iterative edge matching and wavelet maxima matching techniques for registration.
4.2 Phase Correlation for Coarse Registration

For coarse registration, we propose that perspective effects can be approximated using similarity transformation at a coarser scale. Thus, approximate values for scale $s$, rotation $\theta$ and translation $(t_x, t_y)$ can be estimated by applying phase correlation after smoothing and subsampling the images. Phase correlation relies on the translation property of the Fourier transform, which is referred to as the Fourier shift theorem. Let $f_1$ and $f_2$ be the two images that differ only by a displacement $(t_x, t_y)$ i.e.,

$$f_2(x, y) = f_1(x - t_x, y - t_y) \quad (4.3)$$

Their corresponding Fourier transforms $F_1$ and $F_2$ will be related by

$$F_2(\xi, \eta) = e^{-j2\pi(\xi t_x + \eta t_y)} * F_1(\xi, \eta) \quad (4.4)$$

The cross-power spectrum of two images $f_1$ and $f_2$ is defined as

$$\frac{F_1(\xi, \eta)F_2^*(\xi, \eta)}{|F_1(\xi, \eta)F_2^*(\xi, \eta)|} = e^{j2\pi(\xi t_x + \eta t_y)} \quad (4.5)$$

where $F_2^*$ is the complex conjugate of $F_2$. The Fourier shift theorem guarantees that the phase of the cross-power spectrum is equivalent to the phase difference between the images. By taking inverse Fourier transform of the representation in the frequency domain, we will have a function that is an impulse; that is, it is approximately zero everywhere except at the displacement $(t_x, t_y)$ that is needed to optimally register the two images. The method can be extended to determine rotation and scaling differences between two images by performing phase correlation on the log polar images of the magnitudes of Fourier transforms [Reddy and]
If $f_1$ is a translated, rotated (by $\theta_0$) and scaled (by $a$) replica of $f_2$ their Fourier magnitude spectra in polar representation are related by

$$M_1(\rho, \theta) = M_2(\rho a, \theta - \theta_0) \quad (4.6)$$

where $M_1$ and $M_2$ are magnitudes of $F_1$ and $F_2$ respectively. Taking the log transform of $\rho$ in the above equation,

$$M_1(log\rho, \theta) = M_2(log\rho - loga, \theta - \theta_0) \quad (4.7)$$

i.e,

$$M_1(\xi, \theta) = M_2(\xi - d, \theta - \theta_0) \quad (4.8)$$

where

$$\xi = logp \quad (4.9)$$

$$d = loga \quad (4.10)$$

Using the above expression and standard phase correlation technique, scale $a$ and angle $\theta_0$ can be found out. Once these values are found, translation can be found from the rotated and scaled images. However in practice, the determination of rotation and scaling was found to be unreliable and not robust to the changes between before- and after-storm images. Instead, we devise an algorithm that finds the scale and rotation which maximizes the peak of cross power spectrum.
The proposed algorithm is detailed below.

1. **Downsample**: The images are smoothed and downsampled to $256 \times 256$. This step is useful in reducing time required per each phase correlation step while ensuring a good approximation in computing similarity transformation (See Figure [4.1]).

2. **Preprocessing**: In practice, it is more likely that $f_2$ will be a linear shift of $f_1$, rather than a circular shift. In addition, noise due to changes in the scene might be present in $f_2$. In such cases, inverse Fourier transform of the cross power spectrum will not be a simple delta function, which will reduce the performance of the method. To overcome this, a hanning window function is applied to the images so that the edge effects can be ignored (See Figure [4.2]).

3. **Phase Correlation**: For $s_{\text{min}} \leq s \leq s_{\text{max}}$ and $\theta_{\text{min}} \leq \theta \leq \theta_{\text{max}}$, update $s = s + \delta_s$, $\theta = \theta + \delta_\theta$.

   Scale and rotate $f_2$ by $s$, $\theta$ to get $f_2'$. Perform phase correlation on $f_2'$ and $f_1$, obtain the peak value $r$. Instead of looking for an interpolated peak, $r$ is stored as the center of mass of the peak of the inverse Fourier transform of the cross power spectrum (See Figure [4.3]).

4. **Resampling**: The $s$, $\theta$, $t_x$, $t_y$ corresponding to the maximum value of $r$ is used to compute a transformation function. $f_2$ is then resampled with this function resulting in a coarsely registered image pair.
Figure 4.1. An unregistered after and before storm image pair (above) is downsampled (below).

Figure 4.2. The image pair in Figure 4.1 after application of hanning window.
4.3 Feature-based Fine Registration

For fine registration, we adopt the use of feature-based matching and use interest point detectors which are popular in object recognition. Our motivation for choosing this approach is that it is more accurate than correlation techniques and works well even with low overlap percentage. In this approach, features are matched and an appropriate transformation model is calculated based on the matching. This transformation can be applied to post or pre-storm image to register it with respect to the other image. The general steps in feature based matching are:

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.

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. **Outlier Detection**: The correspondences found in the previous step need not be true matches. These outliers can be detected by finding the largest possible subset that fits to a given transform model. RANSAC (RANdom SAmple Consensus) and LMedS (Least Median of Squares) are two popular outlier detection algorithms.

5. **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.

6. **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.

### 4.3.1 Feature Detectors

For finding good interest points, we consider 2 feature detectors which are commonly used in computer vision problems; SIFT feature detector introduced by [Lowe and G (2004)](URL) and SURF feature detector [Bay et al. (2006)](URL).

#### 4.3.1.1 SIFT Detector

Typically, interest point detectors find structures like corners and blobs in the images. Extrema in the Laplacian of Gaussian (LoG) function have traditionally
been used as scale invariant interest point locations. SIFT detector \cite{Lowe2004} works by computing a set of Difference of Gaussian filters, looking for space and scale maxima in the resulting structure, and then computing a sub-pixel space and scale location using a quadratic fit. The number of sub-octave levels in the set of DoG filters was determined empirically. The SIFT detector is implemented by first computing the local Hessian of the difference image $D$,

$$
H = \begin{bmatrix}
D_{xx} & D_{xy} \\
D_{xy} & D_{yy}
\end{bmatrix}
$$

(4.11)

where $D_{xx}$, $D_{xy}$, $D_{xy}$, $D_{yy}$ represent second order gradients of the difference image and then rejecting points based on

$$
\frac{\text{Tr}(H)^2}{\text{Det}(H)} \geq \text{Thresh}
$$

(4.12)

where $\text{Tr}(H)$ is the trace of Hessian, $\text{Det}(H)$ is the determinant and $\text{Thresh}$ is a threshold value determined experimentally.

4.3.1.2 SURF Detector

The SURF-based approach for interest point detection is an inspired faster version of SIFT and uses a very basic Hessian matrix approximation \cite{Herbert2008}. 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 $X$.

$$
I_\Sigma(X) = \sum_{i=0}^{i\leq x} \sum_{j=0}^{j\leq y} I(i, j)
$$

(4.13)
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

$$H(X, \sigma) = \begin{bmatrix} L_{xx}(X, \sigma) & L_{xy}(X, \sigma) \\ L_{xy}(X, \sigma) & L_{yy}(X, \sigma) \end{bmatrix} \quad (4.14)$$

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

$$\det(H_{\text{approx}}) = D_{xx}D_{yy} - (wD_{xy})^2 \quad (4.15)$$

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 4.4 shows an example of the detected interest points using the Fast-Hessian detector.

Figure 4.4. Features extracted from (a) before and (b) after disaster images shown in red circles.

4.3.1.3 Discussion

One of the most important features of interest point detectors is that they should find repeatable points in both images. To evaluate the repeatability of SURF and SIFT interest point detectors, we compared their pixel locations with
manually prepared ground truth. First, ground truth for registration was prepared by marking 20 corresponding points per image pair and calculating the best perspective transform model. Then, the locations of detected points in the registered pairs were compared and if the distance was less than 3 pixels apart the points were classified as ‘repeated’. We evaluated both detectors on our dataset of 23 image-pairs. The average repeatability of SIFT detector was found to be 38.8% and for SURF it was 32%. However SURF detector was found to be atleast 2x faster than the SIFT counterpart and hence is used for the rest of this work.

4.3.2 Descriptor Evaluation

Good descriptors are important for retaining higher inlier % after matching point-pairs. Recently, color feature descriptors with higher discriminating power than their grayscale based counterparts have been gaining popularity in the research community. Vandesande et al. (2010) evaluated various illumination invariant descriptors and compared them with traditional grayscale based ones. In this section, we describe and evaluate some of these color and grayscale-based descriptors on our dataset.

4.3.2.1 SIFT Descriptors

SIFT features are formed by computing the gradient at each pixel in a 16×16 window around the detected keypoint and using the appropriate level of the Gaussian pyramid at which the keypoint was detected. The gradient magnitudes are downweighted by a Gaussian fall-off function in order to reduce the influence of gradients far from the center. In each 4 × 4 quadrant, a gradient orientation histogram is formed by adding the weighted gradient value to one of eight orientation
histogram bins. To reduce the effects of location and dominant orientation misme-

timation, each of the original 256 weighted gradient magnitudes is added to $2 \times 2$

$\times 2$ histogram bins using trilinear interpolation. The resulting 128 non-negative

values form a raw version of the SIFT descriptor vector. To reduce the effects

of contrast or gain (additive variations are already removed by the gradient), the

128-D vector is normalized to unit length. To further make the descriptor robust

to other photometric variations, values are clipped to 0.2 and the resulting vector

is once again renormalized to unit length.

4.3.2.2 SURF Descriptors

For SURF, we build on the distribution of first order Haar wavelet responses

in x and y direction rather than the gradient that SIFT uses, 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 4×4 square sub-regions. This preserves important spatial information. For each sub-region, we compute Haar wavelet responses at 5×5 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 4×4 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.
4.3.2.3 Color Descriptors

Descriptors can be extracted for individual channels in different color spaces and concatenated to increase their discriminating power. In addition to RGB color space, we also evaluate descriptors computed in C-invariant and Opponent color spaces. This choice is inspired from a previous evaluation [Vandesande et al. (2010)] which reported that OpponentSIFT, RGBSIFT and CSIFT performed the best on their images. Opponent color space can be computed from RGB as follows:

\[
\begin{bmatrix}
o_1 \\
o_2 \\
o_3
\end{bmatrix} = \begin{bmatrix}
\frac{R-G}{\sqrt{2}} \\
\frac{R+G-2B}{\sqrt{6}} \\
\frac{R+G+B}{\sqrt{3}}
\end{bmatrix}
\]

(C.16)

C-invariant space can then be represented as the normalized opponent color space \(o_1, o_3\) and \(o_2, o_3\).

4.3.2.4 Evaluation

An evaluation showing performance of descriptors described previously on the dataset of 23 image pairs is shown in Table 4.1. Match % is the percentage of number of detected interest points in the reference image that matched with points in the target image. Inlier % is the number of matched points that were found to be correct on comparison with groundtruth. Registration success is the percentage of images that registered correctly with feature matching alone and less than 5 pixel error.
TABLE 4.1

A COMPARISON OF PERFORMANCE OF VARIOUS COLOR AND GRAYSCALE DESCRIPTORS

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>Matches</th>
<th>Inliers</th>
<th>Registration Success</th>
</tr>
</thead>
<tbody>
<tr>
<td>SURF</td>
<td>3.2%</td>
<td>31.2%</td>
<td>74%</td>
</tr>
<tr>
<td>SIFT</td>
<td>4.2%</td>
<td>34%</td>
<td>74%</td>
</tr>
<tr>
<td>rgbSIFT</td>
<td>3.6%</td>
<td>40%</td>
<td>82%</td>
</tr>
<tr>
<td>cSIFT</td>
<td>1.2%</td>
<td>53.7%</td>
<td>82%</td>
</tr>
<tr>
<td>opponentSIFT</td>
<td>1.8%</td>
<td>46.07%</td>
<td>91%</td>
</tr>
<tr>
<td>rgbSURF</td>
<td>1.9%</td>
<td>15.9%</td>
<td>86%</td>
</tr>
</tbody>
</table>

Ideally, both match % and inlier % should be as high as possible. But as the discriminating power of the descriptor increases, the match % is likely to decrease. However, inlier % should be high enough so that a good subset of points is used for model estimation in the final step. opponentSIFT was found to perform the best, followed by rgbSURF, cSIFT and rgbSIFT. It should be noted that despite the good performance, color descriptors require high computation time and memory requirements making them undesirable for practical applications. For the rest of this chapter, we choose SURF as the choice of feature because it is computationally least expensive and despite having the weakest discriminating power, works well with our matching scheme.
4.4 Interest Point Matching

Once the interest points are detected and described, we need to find a one-to-one correspondence between the points detected from the two images. This is further complicated by the fact that there may be 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. As described earlier, we consider a perspective transform model. In our proposed approach, we first extract SURF features after dividing the images into $k \times k$ grids and match the features across the grids instead of the whole image. For the rest of this section however, we describe and evaluate the general feature matching problem.

A 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 \cite{Krystian and Cordelia (2005)}, we use it on the SURF descriptors described earlier. 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. 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. A matching pair is detected if its distance is closer than the threshold ($t = 0.7$) times the distance of the second nearest neighbor. 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 \cite{Herbert et al. (2008)}. Figure 4.5 shows an example of matched feature points using the nearest neighborhood algorithm.
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 model as mentioned earlier. This model exactly describes the deformation of a flat scene photographed by a pin-hole camera the optical axis of which is not perpendicular to the scene. Slight violations of these assumptions may lead to the use of the second or the third-order polynomial models. Higher order polynomials usually are not used in practical applications because they may unnecessarily warp the sensed image in areas away from the detected points when aligning with the ref-
Given a $3 \times 3$ perspective transformation matrix $H$ with elements $h_{ij}$ we determine the value of the matrix by minimizing the back projection error:

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

This is done by using a RANSAC algorithm Zhang (1999) to try many different random subsets of 4 matched pairs each. For each subset, the backprojection error is calculated and the subset that produces the least error after a certain number of iterations is considered the correct one. The final perspective parameters are further refined with the Levenberg-Marquardt Algorithm.

Next, the perspective transformation is applied to warp 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 4.6 shows images from Figure 4.5 registered and cropped to fit only the overlapping region.
4.4.1 Faster Robust Matching and Outlier Detection

The nearest neighbor (NN) search algorithm described in previous section is a slow $O(n^2)$ algorithm. Further, the ability of RANSAC algorithm to find a good subset depends on the number of inliers present in the matched point-pairs and termination criteria. A larger number of iterations in RANSAC can increase robustness at the cost of speed. To deal with the speed issues in the brute force NN matching, we tried various approximate NN matching algorithms.

Figure 4.6. Registered (a) before and (b) after disaster images. Non-overlap area is filled with black pixels.
Best bin first Beis and Lowe (1997) is an approximate NN search algorithm that is based on a variant of kd-tree. As opposed to kd-tree search, backtracking is done according to a priority queue based on closeness and search is limited to a fixed number of nearest candidates. In general, the algorithm returns the nearest neighbor for a large fraction of queries and a very close neighbor otherwise. For evaluation on our database, we used an algorithm that built 2 kd-trees (for positive and negative laplacian) and performed a Best bin first search on both the trees. The results obtained were merged and the matched point-pairs were processed as described earlier. The times taken for matching in this experiment are shown in Figure 4.7. This approach gained a speed-up of nearly 50x in comparison to the brute-force naive neighborhood matching at higher resolutions. However, the number of inliers significantly reduced to between 0% to 2% of the matched point pairs.
Figure 4.7. Time taken for matching points at different resolutions using best bin first (BBF) and Naive NN search.

We also compared two NN algorithms introduced in Muja and Lowe (2009), namely randomized kd-trees and hierarchical k-means trees. These algorithms were found to work best in terms of speed and accuracy for nearest neighbor search. A system for automatically choosing the best algorithm and optimum parameters depending on the dataset was also described in Muja and Lowe (2009). We individually evaluated these approaches and their performance is reported in Figure 4.8. A speed-up of 30x was found at higher resolutions and the approach consistently found at least 5% inliers. To deal with the problem of reduced inliers,
we proposed a modified form of RANSAC similar to CONSAC described in Guo et al. (2008).

Figure 4.8. Time taken for matching points at different resolutions using randomized kd-trees, heirarchial k-means trees and Naive NN search.

This constrained form of RANSAC was different in that a) when subsets of 4 matched pairs were chosen, these 4 pairs had to follow certain geometric constraints; namely, cyclic or anti-cyclic order had to be conserved and collinearity (of
any 3 points) had to be conserved. b) the upper limit on the number of iterations of RANSAC was determined by a greedy method which counts the iterations since last best subset was found. We call this new algorithm modified CONSAC: These simple constraints not only increase the probability of finding good subsets, but also speed-up the process.

As mentioned earlier, in the proposed algorithm feature matching process was done across $k \times k$ grids because the images are already approximately registered (See Figure 4.9). This introduces a spatial constraint which increases the overall robustness of the approach. Since the features are matched only locally, the approximate NN matching algorithm provide very good results fast. The modified CONSAC algorithm is then used to find the best set of inlier matches and compute a perspective transform model as described previously (see Figure 4.10).
Figure 4.9. The approximately registered image pair from Figure 4.1 (above) is divided into 100 x 100 grid cells and features are matched across these cells (below).
Figure 4.10. (Above) Notice the perspective effects on the grids after transformation. (Below) The final registered images are shown.

4.5 Results

To study the effect of image resolution on the proposed registration process, we used images from publicly available NOAA and USGS aerial imagery dataset. This included 15 image pairs from different locations in coastal Florida before- and after- Hurricane Dennis. Additionally, we used 8 image pairs from Texas before and after Hurricane Ike. These images were $4077 \times 4092$ and approximately 50cm resolution originally. They were downsampled to smaller versions with 1m,
2m, and 4m resolutions. The image pairs vary in overlap, lighting conditions and viewpoint. Ground truth for registration was prepared by manually marking 25 corresponding landmark points for each image in a pair. Registration error is calculated by taking the average distance between the marked points and the transformed points.

We evaluated phase-correlation, feature-driven registration, and hybrid phase-feature registration. Proposed methods of evaluation include registration success, execution time and registration error. Our metric for successful registration is the percentage of images that are deemed to be registered with less than a 6-pixel error in the finest resolution. The effect of registering at various resolutions is shown in Table 4.2. The average registration errors and run times are shown in Table 6.2. The average error figures presented represent only the image pairs that registered with less than 50px error.
### Table 4.2

**Registration Success at Varying Image Resolutions**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>400cm</th>
<th>200cm</th>
<th>100cm</th>
<th>50cm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive NN and RANSAC</td>
<td>22%</td>
<td>35%</td>
<td>74%</td>
<td>74%</td>
</tr>
<tr>
<td>Approximate NN and modified CONSAC</td>
<td>65%</td>
<td>78%</td>
<td>78%</td>
<td>78%</td>
</tr>
<tr>
<td>Phase Correlation</td>
<td>22%</td>
<td>22%</td>
<td>22%</td>
<td>22%</td>
</tr>
<tr>
<td>Phase Correlation, Approximate NN on grids and modified CONSAC</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>
Among feature-driven methods, we evaluated 2 matching schemes. They included Naive NN with RANSAC and approximate NN with the proposed modified CONSAC. The phase-correlation algorithm presented in Section 2 was evaluated with $s_{min} = 0.7$, $s_{max} = 1.2$, $\theta_{min} = 0$, $\theta_{max} = 360$, $\delta_s = 0.1$ and $\delta_\theta = 1$. All the images were registered with about 18px error and 22% of the image-pairs with less than 6px error with phase-correlation. In Table 4.2, while approximate NN with modified CONSAC performed better than the naive approach, registration using phase correlation had a lower success rate indicating its suitability only for a coarse registration.

Our proposed hybrid phase-feature method combines phase-correlation for coarse registration with feature-based matching for fine registration. The new
algorithm performed with 100% registration success on our dataset. Also, the average registration error and run time both turn out best for our proposed approach. Using this scheme, images are registered with 3 pixel error or less and 4x total speed up in comparison to feature-based matching without a phase-correlation step.

4.6 Conclusion

This work presented an automatic registration scheme which is fast and robust enough to work well with real world remote sensing imagery. From the study described in the previous section, it is clear that a combination of coarse registration using phase correlation and fine registration with an approximate nearest neighbor search algorithm combined with a constrained RANSAC algorithm for point-pairs subset selection registers the images with 4x speed-up and least registration error. This is done without compromising robustness and is hence recommended for change detection and disaster response applications. Even though SURF descriptors had the least discriminatory power, the overall robustness of our approach ensures that even weaker features can be used for getting good performance. Our proposed algorithm is highly suited for scenes where a lot has changed due to illumination differences, passage of time or damages (See Figure 4.11, Figure 4.12, Figure 4.13 and Figure 4.14).
Figure 4.11. (above) Before damage Google Earth Image from Joplin, Missouri. (below) After damage NOAA image from Joplin tornado path.
Figure 4.12. The image pair shown in Figure 4.11 registered using our proposed approach.
Figure 4.13. (above) Before storm USGS image from Galveston, Texas. (below) NOAA image from Galveston, Texas after Hurricane Ike.
Figure 4.14. The image pair shown in Figure 4.13 registered using our proposed approach.
CHAPTER 5

COLOR BALANCING

Color transfer or color balancing is the process of transferring the color characteristics of a source image to a target image. Depending on the application, the color differences between source and target images arise due to different reasons. In aesthetic applications, the source and target images may belong to completely different scenes. When both the images roughly correspond to different views of the same scene, the goal becomes to correct the color differences of the overlapping regions. Several factors affect the intensities recorded by cameras. They include: exposure variances, white balancing, gamma correction, vignetting and digitizer parameters. Photographed images generally exhibit a radial falloff of intensity from the center of the image. This effect is known as vignetting. Although lens manufacturers attempt to design their lenses so as to minimize the effects of vignetting, it is still present to some degree in all lenses, and can be quite severe for some aperture and focal length settings. Exposure variances are caused by differences in shutter duration. White balancing removes unrealistic color casts, so that objects which appear white in reality are rendered white in images. Different automatic white balancing techniques used in cameras can have different effects.
While most previous work in color balancing has been done for aesthetic purposes or for image stitching, to the best of our knowledge no effort has been put into color correction for change detection. For change detection applications, we ignore non-overlapped areas. As the overlapped area may contain changed regions, the ideal algorithm should transfer local characteristics of the unchanged regions while transferring global characteristics for changed regions. While transferring local and global characteristics, the transition should also be smooth so that no artifacts are produced.

We are particularly interested in the problem of color balancing for images taken at different times; such as aerial imagery taken before and after a hurricane or other disaster. A damage assessment from such before- and after-storm images require color balancing in the preprocessing stage. In addition to previously mentioned factors, the color differences in this case arise due to different camera parameters, local reflective properties of the object on the ground, changes due to storm damages and changes that occur over time. Previous research Womble et al. (2007) in damage assessment has ignored color balancing and hence the outcome of change detection was affected by lighting differences. In this work we focus on a solution to correct images where natural or man-made changes occur to landscape (see Figure 5.1).
5.1 Related Work

Color balancing algorithms can be classified into model-based, parametric approaches and model-less, non-parametric approaches. Parametric approaches use statistical models to transfer characteristics to a target image. Such approaches may use a global model that is applied to one or all three channels [Reinhard et al. (2001)]. Global modeling usually provides only a rough mapping between the color of two images. A local color transfer scheme based on probabilistic im-

Figure 5.1. (Top left) Source image (after storm). (Top right) Target image (before storm). (Bottom) New target image after applying color balancing using the proposed algorithm.
age segmentation, region mapping using Gaussian mixture models (GMM) and the EM algorithm was proposed by Tai et al. Tai et al. (2005). Non-parametric methods Jia and Tang (2005), Kim and Pollefeys (2008), Pitie et al. (2005), Fecker et al. (2008) assume no particular parametric format of the color mapping function and most of them use a look-up table to directly record the mapping of the full range of color/intensity levels.

A recent survey of color correction algorithms has compared nine color correction methods Xu and Mulligan (2010). The study found that parametric approaches outperform their non-parametric counterparts. Gain compensation Brown and Lowe (2007) and EM segmentation-based local color transfer algorithm Tai et al. (2005) outperformed all others and are recommended as the first options to try for a general image and video stitching application. Very recently, a mean-shift segmentation based local color transfer was proposed in Oliveira et al. (2011). This approach was found to outperform Tai et al. (2005) in terms of color similarity but structural similarity was not taken into account in the evaluation.

Using a segmentation-based method greatly constrains the granularity of local mapping. For instance, reflective properties of a small rooftop may cause it to vary in how the color changes from source to target, from all the other rooftops of the same color but a different material. Further, segmentation-based methods require a matching of segments which gets harder when parts of a region have changed significantly. The main contributions of this work can be summarized as: a) For a fine-grained local mapping, we use moving windows which adapt in size to transfer local statistics. b) The proposed method ensures that the structural integrity of the target image is preserved. c) We use integral images to make computations faster.
5.2 Local Color Transfer

As discussed in previous section, color balancing algorithms require a matching of local regions. The images are registered as described in Chapter 4 and cropped to neglect the non-overlapping areas. Consider the source image \( s(i, j) \), target image \( t(i, j) \) and new target image \( t_{\text{new}}(i, j) \) formed after color transfer. Global color transfer as proposed by Reinhard et al. (2001) first converts the RGB color space into \( l\alpha\beta \) color space. Once the channels have thus been decorrelated, the statistics are transferred by the following equations:

\[
t_{\text{new}}(i, j) = \mu_s + \frac{\sigma_s}{\sigma_t}(t(i, j) - \mu_t) \tag{5.1}
\]

where \( \mu_s \) and \( \mu_t \) are means of the source and target image respectively. Similarly, \( \sigma_s \) and \( \sigma_t \) are the standard deviations of the source and target images respectively.

We propose to extend the above method to a local approach where statistics are calculated for \( k \times k \) windows over the image. The choice of using Reinhard et al. (2001) as a starting point is due to the fact that the computations are simple, no training is required and the algorithm makes no strong assumptions such as constant illumination. The above equation now becomes:

\[
t_{\text{new}}(i, j) = \mu_{k_s(i,j)}^k + \frac{\sigma_{s(i,j)}^k}{\sigma_{t(i,j)}^k}(t(i, j) - \mu_{t(i,j)}^k) \tag{5.2}
\]

\[
\mu_{k_s(i,j)}^k = \frac{1}{k^2} \sum_{l=-\frac{k}{2}}^{i+\frac{k}{2}} \sum_{m=-\frac{k}{2}}^{j+\frac{k}{2}} s(l, m) \tag{5.3}
\]
\[
\sigma_{k(i,j)}^k = \frac{1}{k} \sqrt{\sum_{l=i-\frac{k}{2}}^{i+\frac{k}{2}} \sum_{m=j-\frac{k}{2}}^{j+\frac{k}{2}} (s(l,m) - \mu_{k(i,j)}^k)^2}
\]  \hspace{1cm} (5.4)

The means are now indicated by \( \mu_{s(i,j)}^k \) and \( \mu_{t(i,j)}^k \), where \( k \) denotes the length of the window used for transferring the statistics around the pixel \((i,j)\). Similarly, the standard deviations are indicated by \( \sigma_{s(i,j)}^k \) and \( \sigma_{t(i,j)}^k \). However, in this approach, \( k \) remains a constant for the transfer functions corresponding to all pixels. The outcome of the transfer thus depends on the value of \( k \). If \( k \) is too small, it causes artifacts to appear as the transferred statistics distort the structural components of the target image (See Figure \[5.2\]). This can be solved by choosing a sufficiently large size for \( k \).
5.2.1 Adaptive Windowing

In order to characterize local differences between the images which may occur due to lighting variations, damages, shadows or other structural variations, we propose to compute normalized cross correlations (NCC) between the corresponding windows in source and target images. $NCC$ corresponding to a pixel...
\((i, j)\) and window length \(k\) can be computed as:

\[
NCC(i, j) = \frac{\sigma_k^{s(i,j)} t(i,j)}{\sigma_k^{s(i,j)} \sigma_k^{t(i,j)}} \tag{5.5}
\]

\(\sigma_k^{s(i,j)}\) and \(\sigma_k^{t(i,j)}\) are local standard deviations of source and target images. 
\(\sigma_k^{s(i,j)} t(i,j)\) is the cross-covariance between corresponding windows, calculated as:

\[
\sigma_k^{s(i,j)} t(i,j) = \sum_{i=-\frac{k}{2}}^{i+\frac{k}{2}} \sum_{m=-\frac{k}{2}}^{j+\frac{k}{2}} \frac{(s(l,m) - \mu_k^{s(i,j)})(t(l,m) - \mu_k^{t(i,j)})}{k^2} \tag{5.6}
\]

The window length \(k\) for each pixel can be fixed by calculating the value of \(NCC(i, j)\) for a range of window sizes and choosing the smallest window size that gives a sufficiently high NCC value. Thus, the proposed method has three parameters, \(k_{\text{min}}\) and \(k_{\text{max}}\) which control the range of window sizes to be considered and \(NCC_{\text{min}}\) which decides the minimum value of NCC that is required to fix a window size. Our approach can be summarized as the following: For each pixel \((i, j)\) compute the final values using equation (2) where \(k\) is minimized such that \(k_{\text{min}} \leq k \leq k_{\text{max}}\) and 
\(NCC(i, j) \geq NCC_{\text{min}}\).

5.2.2 Computation using Integral Images

Since computing the means, standard deviation and cross-covariance for multiple large window sizes could be expensive, we used integral images to speed up the process. Once calculated for an image, an integral image can be used to perform summation over any rectangular area in the image in constant time. An integral image \(I\) of an input source image \(s\) is defined as the image in which the intensity at a pixel position is equal to the sum of the intensities of all the pixels above and to the left of that position in the original image. So the intensity at position \((i, j)\)
can be written as:

\[ I_s(i, j) = \sum_{l=0}^{i} \sum_{m=0}^{j} s(l, m) \]  

(5.7)

The integral image of any greyscale image can be efficiently computed in a single pass. Once we have the integral image, the local mean for any window size can be computed simply by using two addition and two subtraction operations instead of the summation over all pixel values within that window: \( \mu_{s(i,j)}^k = (I(i + k/2, j + k/2) + I(i - k/2, j - k/2) - I(i + k/2, j - k/2) + I(i - k/2, j + k/2))/k^2 \)

Similarly, if we consider the computations of the local standard deviation and local cross-covariance

\[ \sigma_{s(i,j)}^k = \frac{1}{k^2} \sqrt{\sum_{l=i-k/2}^{i+k/2} \sum_{m=j-k/2}^{j+k/2} s^2(l, m) - \left( \mu_{s(i,j)}^k \right)^2} \]  

(5.8)

\[ \sigma_{s(i,j)t(i,j)}^k = \sum_{l=i-k/2}^{i+k/2} \sum_{m=j-k/2}^{j+k/2} \frac{s(l, m)t(l, m)}{k^2} - \mu_{s(i,j)}^k \mu_{t(i,j)}^k \]  

(5.9)

The first term in Equations 5.8, 5.9 can be computed in a similar way by using integral images of the squared pixel intensities and integral images of pixel products respectively. Once the integral image is calculated, local means, standard deviations and NCC can be computed in constant time and independent of the local window size.

5.3 Proposed Algorithm

1. All the integral images \( I_s(i, j), I_t(i, j), I_{st}(i, j), I_{s2}(i, j) \) and \( I_{t2}(i, j) \) are computed for the three channels using the procedure mentioned earlier.
2. Convert \(s(i, j)\) and \(t(i, j)\) to grayscale and initialize \(k = k_{\text{min}}\). Record values of \(k\) for each pixel \((i, j)\) in a window index map \(WIM(i, j)\).

3. For each pixel \((i, j)\), compute \(NCC(i, j)\) using integral images. If \(NCC(i, j) < NCC_{\text{min}}\) and \(WIM(i, j) < k_{\text{max}}\), increment \(WIM(i, j)\) by a small value \(\delta k\).

4. Repeat Step 3 until there is no more change for any pixel \((i, j)\) in \(WIM(i, j)\).

5. Smooth \(WIM(i, j)\) using a Gaussian blur to allow smooth transition of window sizes and avoid artifacts at boundaries.

6. Using the value of \(k\) recorded in \(WIM(i, j)\), apply Equation 5 for all pixels \((i, j)\) using integral images.

The use of integral images brings down the time complexity of the algorithm to \(O(rN)\) where \(N\) is the number of pixels in the image and \(r = k_{\text{max}} - k_{\text{min}}\).

5.4 Results

For our evaluation, we used 15 pairs of images, each of size 4077 \(\times\) 4092 pixels before registration. The images were obtained from a mix of different sources. Most of the aerial imagery was acquired by the NOAA Remote Sensing Division. The images were approximately 50 cm resolution. The NOAA images used were from Hurricane Dennis (2005) and Hurricane Ivan (2004) images of Pensacola, Florida. Additionally, images were downloaded from USGS High resolution Orthoimagery database which were of 60 cm resolution.

We use the two measures of performance that were used in [Xu and Mulligan (2010)]; namely, color similarity and structural similarity. While color similarity is of primary importance in evaluating the performance, structural similarity is essential to ensure that the color transfer does not lead to destroying the structure of the target image. For instance, if the algorithm equates the values of each
pixel in the source to the target, we have an optimal color similarity measure but poor structural similarity. Let \( r \) be the transformed image, \( s \) is the source image and \( t \) be the target image. Color similarity (CS) measure \( CS(r, s) \) is defined as \( CS(r, s) = PSNR(r, s) \) where \( PSNR = 20 \times \log_{10}(L/RMS) \) is the peak signal to noise ratio. \( L \) is the largest possible value in the dynamic range of an image, and \( RMS \) is the root mean square difference between two images. The higher the value of \( CS(r, s) \) the more similar the color between the two images \( r \) and \( s \). The other measure used is structural similarity index measure or \( SSIM(r, t) \), which is described in Wang et al. (2004). \( SSIM \) is as a combination of luminance, contrast and structure components and is computed over local windows. The higher \( SSIM \) is, the higher the similarity between the structure of \( r \) and \( t \), and \( SSIM(r, t) = 1 \) if there is no structure difference.

We compared our algorithm with 7 other algorithms that were presented in Xu and Mulligan (2010). The algorithms used are listed in Table 5.1. The results of running these algorithms on our image pairs are shown in Figure 5.3 and Figure 5.4. Note that our proposed algorithm, Algorithm 8 in the Table, has the highest color similarity measures, outperforming the other algorithms. From Figure 5.3 it is clear that the superior performance in transferring local color characteristics is done without sacrificing the structural integrity of the target image. The slightly lower values of SSIM index as compared to some of the other algorithms is an expected result as color balancing could lead to changes in illumination. Similar to what was observed in Xu and Mulligan (2010), gain compensation and local color transfer are the next best performers. The output of our algorithm and the output of gain compensation are shown in Figure 5.5 for some of the images in
our database.

TABLE 5.1

LIST OF ALGORITHMS USED IN OUR STUDY.

<table>
<thead>
<tr>
<th>#</th>
<th>Approach</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>brightness transfer function</td>
<td>local</td>
</tr>
<tr>
<td>2</td>
<td>cumulative histogram mapping</td>
<td>local</td>
</tr>
<tr>
<td>3</td>
<td>gain compensation</td>
<td>local</td>
</tr>
<tr>
<td>4</td>
<td>global color transfer</td>
<td>global</td>
</tr>
<tr>
<td>5</td>
<td>local color transfer</td>
<td>local</td>
</tr>
<tr>
<td>6</td>
<td>iterative color distribution transfer</td>
<td>global</td>
</tr>
<tr>
<td>7</td>
<td>principle regions mapping</td>
<td>local</td>
</tr>
</tbody>
</table>
Figure 5.3. Color similarity measures for different algorithms.
Figure 5.4. SSIM for different color balancing algorithms.
Figure 5.5. (1st column) Source images. (2nd column) Target images. (3rd column) New target image transformed using gain compensation. (4th column) New target image transformed using the proposed adaptive windowing based local color transfer.
5.5 Conclusion

The work done so far presented a parametric local color balancing approach that uses adaptive windowing. The proposed algorithm is shown to outperform other state of the art algorithms for multitemporal aerial images. The simplicity of approach and application of integral images makes it easy to implement and use. A limitation to our proposed algorithm is that the images need to be registered with reasonable accuracy. As a part of future work, the method should be evaluated for approximately registered images. The effects of color balancing on change classification is described in Chapter 7.
Many image repositories do not contain metadata, therefore we require automatic building detection from post-disaster images. Building detection and rooftop extraction are often thought of as separate problems. While it can be argued whether detection leads to extraction or vice versa, for this work we consider building detection to include finding the spatial locations of buildings in aerial images as well as extracting the exact rooftop contour. Numerous methods have addressed building extraction in the past. 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 most aerial and satellite image repositories lack stereo or multi-sensor information.

Thus, building identification and rooftop extraction becomes here a challenging monocular object recognition task based on purely optical data. This case is the focus of our research and is addressed in this chapter. We present two approaches towards building detection. One is a slow, supervised combinatorial
process that produces impressive results in challenging conditions. The other is a fast unsupervised algorithm based on clustering and graph cut optimization, but less robust. Both approaches are described in detail and compared with a state-of-the-art building detection algorithm.

6.1 Related Work

Previous approaches in damage detection (Chen and Hutchinson 2005; Matsuo et al. 2004; Womble et al. 2007; Vijayaraj et al. 2008; Radhika et al. 2011) either do not extract buildings automatically or provide only approximate estimates. A comprehensive survey of building detection methods was published in 1999 by Mayer (1999). Since then, hundreds of papers have been published in building detection. However the development of a general building extraction algorithm that can work with any aerial or satellite image remains an open research problem. The remainder of this section describes some of the work done in the past decade and their limitations.

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 (2009b) 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. There-
after 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 over-segmented 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.

More recently, Sirmacek and Unsalan (2011) described a probabilistic approach to detect buildings using local feature vectors. Extracted local feature vectors are considered to serve as observations of the probability density function (pdf) to be estimated. Using a variable-kernel density estimation method, they estimate the corresponding pdf. The building locations are represented as joint random variables in the image as and their pdf is estimated. Using the modes of the estimated density, as well as other probabilistic properties, they detect building locations. Data and decision fusion methods based on the probabilistic framework were used to do this in an unsupervised manner. They report 83% true detect rate and 39% false detect rate on aerial images. As this approach is recent, unsupervised and considered to be the state-of-the-art, we use this for our evaluation.

6.2 Supervised Approach based on Segment Grouping

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.

6.2.1 Segmentation

The first step in our proposed building detection algorithm is segmentation. The segmentation step groups together pixels of similar spectral content. We use maximum likelihood classification to predict the spectral class of each pixel. 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 among the most common supervised classification methods used with remote sensing image data. Let the spectral
classes for an image be represented by

$$\omega_i, i = 1, \ldots M$$  \hspace{1cm} (6.1)

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

$$p(\omega_i|x), i = 1, \ldots M$$  \hspace{1cm} (6.2)

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

$$x \in \omega_i, \text{ if } p(\omega_i|x) > p(\omega_j|x) \text{ for all } j \neq i$$  \hspace{1cm} (6.3)

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 (6.3) and the available $$p(x|\omega_i)$$, estimated from training data, are related by Bayes theorem:

$$p(\omega_i|x) = p(x|\omega_i)p(\omega_i)/p(x)$$  \hspace{1cm} (6.4)

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

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

(6.5)

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 (6.4) it can be seen that the classification rule of (6.3) is:

\[ x \in \omega_i \text{ if } p(x|\omega_i)p(\omega_i) > p(x|\omega_j)p(\omega_j) \text{ for all } j \neq i \]  

(6.6)

where \( p(x) \) has been removed as a common factor. The rule of (6.6) is more acceptable than that of (6.3) 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 (6.6) the definition

\[ g_i(x) = \ln p(x|\omega_i)p(\omega_i) = \ln p(x|\omega_i) + \ln p(\omega_i) \]  

(6.7)

is used, where \( \ln \) is the natural logarithm, so that (6.6) is restated as

\[ x \in \omega_i \text{ if } g_i(x) > g_j(x) \text{ for all } j \neq i \]  

(6.8)

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 (6.7) therefore, it is now assumed for $N$ bands that

$$p(x|\omega_i) = (2\pi)^{-N/2}|\Sigma_i|^{-1/2} e^{\exp \left\{ -1/2 (x - m_i)^t \Sigma_i^{-1} (x - m_i) \right\}}$$

(6.9)

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 statistics, is:

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

(6.10)

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 (6.10) 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:

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

(6.11)

Implementation of the maximum likelihood decision rule involves using (6.11) in (6.8). 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 each 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 and shadows. The maximum-likelihood classification results in
salt and pepper noise as the class conditional mean vector and covariance matrix
is calculated only from a fraction of the total pixels. To reduce the noise, misclas-
sification, over-segmentation and under-segmentation, we require more training
pixels.

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 covariance of the resultant clusters
are then used with the maximum-likelihood classifier. 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

$$\hat{m}_i, i = 1, \ldots, M.$$  \hspace{1cm} (6.12)

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 produced in Step 2 are computed. Let these be denoted as

\[ \tilde{m}_i, i = 1, \ldots, M. \]  

(6.13)

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 6.2. 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.

The resultant segmentation (see Figure 6.1 and Figure 6.2) can now be used to identify buildings by applying certain rules to combine segments.
Figure 6.1. A $4077 \times 4092$ NOAA image from Pensacola Florida.
Figure 6.2. 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%

6.2.2 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), 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 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:

\[ Perimeter \approx \sum N_8 + \pi \frac{\text{area}}{0.900} \]  

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 [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:

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

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 de-
crease this measure to lower values. Shape measures are pose-invariant: change of orientation, location and size leave a measure unchanged. 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 Serra (1983).

6.2.3 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 Benedek and Szirnyi (2010) and has its own literature for remote sensing applications Tsai (2006). In approach we detect the shadow by the segmentation process described previously. We detect shadows using the $N$ dimensional multi-spectral space of pixels (in aerial images these correspond to the color channels) rather than simply using the brightness values of a single channel. 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 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. Higher MEPS values are expected for elevated surfaces like rooftops and lower values for flat areas like roads, pavements and parking lots. The whole 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 [Douglas and Peucker (1973)]. Find polygons with 4 sides and 90° angles between the sides, i.e. find buildings that are rectangles.

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,

$$\vec{v}_{sh} = \vec{v}_i$$  \hspace*{1cm} (6.16)

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

$$\vec{v}_{sh} = \vec{v}_i + \vec{v}_j$$  \hspace*{1cm} (6.17)

3. Once $\vec{v}_{sh}$ is found, the perimeter along which a shadow is predicted to be cast is the convex perimeter that is “visible” to a line perpendicular to $\vec{v}_{sh}$. 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.

$$\theta_i = \arccos \left( \frac{\vec{v}_{sh},\vec{y}_i}{|\vec{v}_{sh}| |\vec{y}_i|} \right) \times \frac{180}{\pi}$$  \hspace*{1cm} (6.18)

If $0° \leq \theta_i \leq 90°$, then we predict that a shadow pixel will be found for $x_i$. 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

$$MEPS_x = \frac{N_{sh}}{N_x}$$  \hspace*{1cm} (6.19)
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 $MEPS_x$. The MEPS values for a parking lot and a building are shown in Figure 6.3 and Figure 6.4 respectively. Higher MEPS values are expected for elevated surfaces like rooftops and lower values for flat areas like roads, pavements and parking lots.

Figure 6.3. (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.
6.2.4 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 fulfill 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, buildings are usually composed of multiple segments (see Figure 6.5). This results in a poorly performing extraction process if we consider buildings as single segments.
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 to 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 1. 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 1 \textit{segmentCombination}

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

6.2.5 Results

Building detection experiments were performed on 3 images of coastal Florida before Hurricane Dennis. Figure 6.6 and Figure 6.7 show an example with 50 buildings in the Ground Truth. The ground truth was prepared manually by an expert volunteer. Another image with 35 buildings in Ground Truth is shown in Figure 6.8 and Figure 6.9. 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

\begin{equation}
DP = \frac{TP}{AP}
\end{equation}
\[ \text{Accuracy} = \frac{TP}{TP + FP} \] (6.21)

Figure 6.6. A before storm NOAA image from Pensacola, Florida
Figure 6.7. Buildings extracted from Figure 6.6 is shown in white. $C_{th} = 0.4$, $M_{th} = 0.4$ and $A_{th} = 100$ pixels.
Figure 6.8. 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 6.9. Buildings extracted from Figure 6.8 is shown in white. $C_{th} = 0.3$, $M_{th} = 0.3$ and $A_{th} = 100$ pixels

Table 6.1 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).
TABLE 6.1
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>

6.2.5.1 Comparison with Building Detection using HOG

We also tried building detection using a generic object recognition technique described in [Dalal and Triggs (2005)](https://www.cs.cmu.edu/~cmgm/talks/hog05_talk.pdf). Histogram of Oriented Gradients (HOG) has been found to work well in detecting humans and objects like vehicles. The essential thought behind the Histogram of Oriented Gradient descriptors is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. The image is divided into small connected regions, called cells, and for each cell a histogram of gradient directions or edge orientations for the pixels within the cell is computed. The combination of these histograms represents the descriptor. For improved accuracy, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to changes in illumination or shadowing [Dalal and Triggs (2005)](https://www.cs.cmu.edu/~cmgm/talks/hog05_talk.pdf).
The HOG descriptor maintains a few key advantages over other descriptor methods. One is the invariance to geometric and photometric transformations, with the exception of object orientation. Such changes would only appear in larger spatial regions. For detecting human pedestrians, it was found that coarse spatial sampling, fine orientation sampling, and strong local photometric normalization permits the individual body movement to be ignored so long as they maintain a roughly upright position. Hence, HOG descriptors are thus better suited over other descriptors for capturing the shape variations in buildings.

The final step in object recognition using Histogram of Oriented Gradient descriptors is to feed the descriptors into some recognition system based on supervised learning. The Support Vector Machine classifier is a binary classifier which looks for an optimal hyperplane as a decision function. Once trained on images containing buildings, the SVM classifier can make decisions regarding the presence of buildings in additional test images.

We evaluated HOG based detection for building extraction using a database of 185 images. We used a dataset of 95 positives and 90 negatives to train a SVM classifier. The positive images contained one building each and were of different shapes and colors as shown in Figure 6.10 (a). The negative images were of parking lots, vegetation, grass, roads, pavements and other non-building areas (see Figure 6.10 (b)). The images evaluated in Section were used as testing set. The results of detection on image Figure is shown in Figure 6.11. The results of this classification are shown in Table 6.2. In general HOG based detection is shown to be much slower than our algorithm. While the low accuracy due to false
positives can be eliminated by lower thresholds, false negatives appear to happen when buildings are clustered together closely.

Figure 6.10. (a) Positive images and (b) Negative images used in training set.
Figure 6.11. Buildings extracted from Figure 6.6 using HOG-based detection are marked with rectangles.

### TABLE 6.2

PERFORMANCE OF HOG-BASED 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>$785 \times 719$</td>
<td>50</td>
<td>58%</td>
<td>38%</td>
<td>70</td>
</tr>
<tr>
<td>2</td>
<td>$2056 \times 1320$</td>
<td>250</td>
<td>52%</td>
<td>42%</td>
<td>125</td>
</tr>
<tr>
<td>3</td>
<td>$659 \times 496$</td>
<td>35</td>
<td>60%</td>
<td>40%</td>
<td>45</td>
</tr>
</tbody>
</table>
6.2.5.2 Comparison with Building Detection using Probabilistic Framework

As described earlier, Sirmacek and Unsalan (2011) described a probabilistic approach to detect buildings using local feature vectors. We evaluated their algorithm on our dataset and used gabor filters in 10 different orientations to compute the local feature vectors. Gabor filters are used in their algorithm to find building edges and corners. Then a kernel density estimation method is used to determine probability density function for the local features. As buildings may consist of many edges and corners, they adjust the effect of local feature vectors w.r.t. their orientation and weight. In doing so, for bright building corners and edges, gradient directions are toward the building center and vice versa for dark buildings. The results generated using this method are reported in Table 6.3. See Figure 6.12 and Figure 6.13 for buildings detected using this algorithm. While the probabilistic framework has a low computational cost and runs very fast, the detection percentage on our images was significantly lesser in comparison to our algorithm. Further, this approach also cannot extract exact rooftop contours.
TABLE 6.3

PERFORMANCE OF PROBABILISTIC FRAMEWORK FOR 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>54%</td>
<td>75%</td>
<td>0.01</td>
</tr>
<tr>
<td>2</td>
<td>2056 × 1320</td>
<td>250</td>
<td>60%</td>
<td>72%</td>
<td>0.2</td>
</tr>
<tr>
<td>3</td>
<td>659 × 496</td>
<td>35</td>
<td>52%</td>
<td>68%</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Figure 6.12. Buildings detected from Figure 6.6 using the probabilistic framework algorithm are marked with yellow and blue labels.
6.3 Unsupervised Approach for Rooftop Extraction

The accuracy of our algorithm described in previous section depends on how well the segmentation is done. The outer contours of the extracted objects are essentially the boundaries of the segments. To extract the exact boundaries of the objects in the image, other features such as gradients needs to be taken into account. Further, the quality of segmentation is also determined by the number of pixels labeled in the training phase before clustering. Also, the combinatorial segment grouping process makes this algorithm computationally intensive. To overcome these issues we propose an unsupervised algorithm in this section. First, object level features are extracted from the segmented image and used to identify possible candidates as rooftops/non-rooftops. These labels are then used to cluster the image into various labellings such as foreground pixels (rooftops or other

Figure 6.13. Buildings extracted from Figure 6.7 using the probabilistic framework algorithm are marked with yellow and blue labels.
structures) and background pixels. More precise rooftop boundaries can then be extracted by ensuring that these pixel labels vary smoothly while preserving the discontinuities at object boundaries. This is formulated as an energy minimization problem and solved by using a Graph Cut algorithm as described later in this section.

6.3.1 Meanshift Segmentation

One of the most common and powerful image segmentation techniques used for vision problems is meanshift segmentation. It is based on a non-parametric feature space analysis that was widely introduced into computer vision problems in Comaniciu and Meer (2002). Let \( f(x) \) be the (unknown) probability density function underlying a N-dimensional feature space, and \( x \), the available data points in this space. Under its simplest formulation, the mean shift property can be written as

\[
\hat{\Delta} f(x) \approx \left( \text{avg}_{x_i \in S_{h,x}} [x_i - x] \right)
\]  

(6.22)

where \( S_{h,x} \) is the N-dimensional hypersphere with radius \( h \) centered on \( x \). The above equation states that the estimate of the density gradient at location \( x \) is proportional to the offset of the mean vector computed in a window, from the center of that window. Recursive application of the mean shift property yields a simple maxima detection procedure. The local maxima of the density can be found by moving at each iteration the window \( S_{h,x} \) by the mean shift vector, until the magnitude of the shifts becomes less than a threshold. See Comaniciu and Meer (2002) for more detailed presentation.
In meanshift based color image segmentation algorithms, a five-dimensional feature space is usually used. The LUV color space was employed in Comaniciu and Meer (2002) since its metric is a satisfactory approximation to Euclidean, thus allowing the use of spherical windows. The remaining two dimensions were the spatial coordinates. The quality of segmentation is controlled by the spatial $h_y$, and the color $h_r$, parameters defining the radii of the windows in the respective domains.

The segmentation algorithm has two major steps. First, the image is filtered using mean shift in 5D, replacing the value of each pixel with the 3D (color) component of the 5D mode it is associated to. Note that the filtering is discontinuity preserving. In the second step, groups of pixels located within $h_{r/2}$ in the color space are recursively fused until convergence. The resulting large delineated regions have their values set to their average. Mean shift based color image segmentation is already popular in the computer vision community and several implementations exist. We used the implementation described in Christoudias et al. (2002). An image with before-storm rooftops is shown in 6.14. The filtered image after meanshift is used in Figure 6.15. The final segmented output is shown in Figure 6.16.
Figure 6.14. An aerial image from Pensacola, Florida with distinctly shaped rooftops.
Figure 6.15. The meanshift filtered version of Figure 6.14.
6.3.2 Shadow Extraction

Shadow detection and extraction from aerial images is a widely studied research area. In the algorithm described in the previous section, this was done by using supervised classification. For this work, we used an unsupervised method based on invariant color models and similar to the one proposed in Tsai (2006). First,
the images are converted into HSI color mode.

\[
\begin{bmatrix}
I \\
V_1 \\
V_2
\end{bmatrix} = \begin{bmatrix}
\frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\
\frac{\sqrt{6}}{6} & \frac{\sqrt{6}}{6} & \frac{\sqrt{6}}{3} \\
\frac{1}{\sqrt{6}} & -\frac{2}{\sqrt{6}} & 0
\end{bmatrix} \begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
\]  
(6.23)

\[S = \sqrt{V_1^2 + V_2^2} \]  
(6.24)

\[H = \sqrt{\tan^{-1}\left(\frac{V_1}{V_2}\right)} \]  
(6.25)

if \(V_1 \neq 0\) otherwise, \(H\) is undefined.

The method in [Tsai (2006)] uses \(H\) and \(I\), both scaled to \([0, 1]\). Spectral ratioing technique is applied to obtain the \(\frac{(H+1)}{(I+1)}\) ratio image, which is scaled to have pixels’ values in the \([0, 255]\). The ratio image is considered to enhance the increased hue property of shadows with low luminance, i.e., pixels in shadowed regions will have higher values in the ratio image than those pixels in nonshadowed regions. The Otsu’s method is then applied over the histogram of the ratio image to automatically determine the threshold for segmenting the regions in shadow into a logical shadow mask for those pixels with value greater than the threshold. The Otsu’s method finds an optimal threshold \(T\), which maximizes

\[V(T) = \frac{(\bar{w}(T) - \mu(T))^2}{w(T).\mu(T)} \]  
(6.26)

where \(w(T) = \sum_{i=0}^{T} p_i\), \(\mu(T) = \sum_{i=T+1}^{255} p_i\), \(\bar{w} = \sum_{i=0}^{255} i p_i\), and \(p_i\) is the probability of pixels with grey level \(i\) in the image. However in practice, we found that using saturation instead of hue in calculating the spectral image \(\frac{(S+1)}{(I+1)}\) is more robust to variations in image lighting conditions. The extracted shadows are shown
6.3.3 Object-level features for Generating Foreground Candidates

Once the shadows are extracted and the segmentation is done, the direction vector $\vec{v}_{sh}$ can be computed using the voting procedure described in the previous section. While spectral information can be used to calculate the posterior proba-
bilities of pixel labeling, object level features such as shadow strength $MEPS_x$ and compactness can provide additional evidence of height and man-made structures. Hence for each region $x$ in the segmentation image, we compute shadow confidence map and shape confidence map by calculating the $MEPS$ and compactness values. The purposes in computing these maps include: a) they can be used to find candidate foreground/background objects b) they can be used to label and assign probabilities to pixels before graph cut optimization. Examples of shadow and compactness confidence maps are shown in Figure 6.18 and Figure 6.19 respectively.
Figure 6.18. The shadow confidence map extracted from 6.14 using MEPS values of individual segments.
Figure 6.19. The shape confidence map extracted from 6.14 using compactness values of individual segments.

The generated maps are then used to extract candidate foreground objects by thresholding their values. We compute $MEP_{th}$ and $Shape_{th}$ using Otsu’s thresholding method to find objects that are highly likely to be rooftop structures and some that are highly unlikely to be rooftops. Those segments with values higher than these thresholds exhibit evidence of both height and symmetry. These segments may correspond to parts of man-made structures or the rooftop itself. When the values are lower, structures like roads, vegetation, parking lots etc
can be extracted. See Figure 6.20 and Figure 6.21 for examples on rooftop and non-rooftop candidates extracted using this method.

Figure 6.20. Probable building segments extracted from 6.14 using shadow and shape confidence maps.
6.3.4 Illumination Invariant Features

While shapes and shadows provide good basis for determining the object-level characteristics of the region to which each pixel belongs to, spectral features computed at pixel levels provide additional clues to what labels could be assigned for each pixel. We extract spectral features using hue $H$, saturation $S$ components of the $HSV$ color model. This model is similar to the $HSI$ color model described previously and decorrelates luminance and chromaticity components. Addition-
ally, we use improved Sobel edge derivatives described in Scharr (2000) as a third feature. A false color map generated using this proposed HSE model is shown in Figure 6.22.

Figure 6.22. A false color image representing hue, saturation and edges extracted from Figure 6.14 as RGB channels.
6.3.5 Clustering using Expectation Maximization

The object level features (shape and shadow) and spectral features (hue, saturation, edges) can be used to cluster and assign labels to each pixel in the image. The generated rooftop/non-rooftop candidates are used as samples used for predicting the labels and posterior probabilities for each pixel. First, we use Expectation Maximization to estimate the parameters of a multivariate probability density function in the form of a Gaussian mixture distribution with a specified number of mixtures. Next, we trained 3 models with k- gaussians using the generated rooftop samples, generated non-rooftop samples and probable background pixels (entire image). The posterior probabilities are then used to label the pixels and define a data cost term for energy minimization formulation described in next section. In the rest of this section, we briefly describe Expectation Maximization algorithm.

The Expectation Maximization(EM) algorithm estimates the parameters of the multivariate probability density function in the form of a Gaussian mixture distribution with a specified number of mixtures. Consider the set of the \( N \) feature vectors \( x_1, x_2, ..., x_N \) from a \( d \)-dimensional Euclidean space drawn from a Gaussian mixture:

\[
p(x; a_k, S_k, \pi_k) = \sum_{k=1}^{m} \pi_k p_k(x), \pi_k \geq 0, \sum_{k=1}^{m} \pi = 1 \quad (6.27)
\]

where \( m \) is the number of mixtures, \( p_k \) is the normal distribution density with the mean \( a_k \), covariance matrix \( S_k \), \( \pi_k \) is the weight of the \( k \)th mixture. Given the number of mixtures and a set of samples, the algorithm finds the maximum likelihood estimate for all mixture parameters, that is, \( a_k, S_k \) and \( \pi_k \). The EM
algorithm is an iterative procedure. Each iteration includes two steps. At the first step (Expectation step), we find a probability $p_{i,k}$ of sample $i$ to belong to mixture $k$ using the currently available mixture parameter estimates. At the second step (Maximization step), the mixture parameter estimates are refined using the computed probabilities. See Bilmes (1998) for a more detailed explanation. Since we trained the mixtures on 3 separate models, the maximum value of $p_{i,k}$ among the models was used to determine the label among the models. Examples of labels assigned using EM algorithm trained on the generated samples are shown in Figure 6.23.
Figure 6.23. Examples of labels assigned using EM algorithm trained with 5 gaussians, 5 features on the generated samples from 6.14. The 3 labels are shown in red, green and blue.

6.3.6 Multi-label Graphcut Optimization

Many problems require estimating spatially varying quantity (such as regions) from noisy measurements such that they vary smoothly on the surface of an object, but change dramatically at object boundaries. These can be formulated as energy minimizations and solved using graph cut algorithms. We consider pixels of the image to be nodes of a graph and edges between the nodes indicate pixel
connectivity. In our problem, every pixel \( p \in P \) of a grid graph must be labeled in a finite set of labels \( L \). The goal is to find a labeling \( f \) that assigns each pixel \( p \in P \) a label \( f_p \in L \), such that \( f \) is both piecewise smooth and consistent with observed data. Thus, the goal is to minimize the energy \( E(f) \) such that

\[
E(f) = E_{\text{smooth}}(f) + E_{\text{data}}(f)
\]  

(6.28)

where \( E_{\text{smooth}} \) measures the extent to which \( f \) is not piecewise smooth, while \( E_{\text{data}} \) measures the disagreement between \( f \) and observed data. The form of \( E_{\text{data}} \) is typically

\[
E_{\text{data}}(f) = \sum_{p \in P} D_p(f_p)
\]  

(6.29)

where \( D_p \) measures how well the label \( f_p \) fits the pixel \( p \) given the observed data. In our algorithm we set the data cost \( D_p \) to \( P(f_p) \), the posterior probability computed for pixel \( p \) belonging to the label \( f_p \). This is obtained from \( \max_{i,k} p_{i,k} \) of EM algorithm as described in the previous section. The smoothing cost \( E_{\text{smooth}} \) is of form

\[
E_{\text{data}} = \sum_{p,q \in N} V_{p,q}(f_p, f_q)
\]  

(6.30)

where \( N \) consists of adjacent interacting pixels. In our algorithm, we propose to compute the penalty \( V_{p,q} \) based on the gradient between the pixels \( p \) and \( q \). In other words, the penalty should be higher when there are strong edges in the original image.

The choice of algorithm in minimizing the energy \( E(f) \) depends on the number of labels. If there are only 2 labels, the standard max-flow min-cut based
algorithms such as Ford Fulkerson. Instead we focus on multi-label optimization which is inherently a NP hard problem. We use a good approximate solution that’s described in Boykov et al. (2001) for training 3 different models mentioned previously. Figure 6.24 and Figure 6.25 show the $E_{smooth}$ function expressed as gradients in x and y directions. Figure 6.26 shows the final extracted rooftops by using the proposed energy minimization approach.

Figure 6.24. The derivative in x direction computed from 6.14 and used as a smoothing cost for graph cut optimization.
Figure 6.25. The derivative in y direction computed from 6.14 and used as a smoothing cost for graph cut optimization.
6.3.7 Results

We evaluated the proposed EM and Graph Cut based algorithm on a subset of images in our dataset. The results are shown in Figure 6.27, Figure 6.28 and Figure 6.29. As seen in Figure 6.27, the proposed algorithm can extract rooftops with varying shapes, texture and heterogeneous regions. Unlike the probabilistic framework algorithm, it does not pick up many false detect samples. Figure 6.28 shows a scene where rooftops of multiple colors are present. In this case, the proposed
algorithm can have false rejects if all the colors of the rooftop were not generated in the foreground modeling step or if the color of a rooftop is too similar to the background. Finally, Figure 6.29 shows a small urban area with many buildings that are close to each other and varying in shape and colors. Our algorithm picked up most of these rooftops successfully while the probabilistic framework algorithm had many false rejects and false accepts.
Figure 6.27. (1st column) Rooftop images from Pensacola, Florida, New Orleans, Louisiana and Haiti. (2nd column) The blue and yellow markers denote the rooftops detected using the probabilistic framework approach. (3rd column) Rooftops extracted using our proposed algorithm. Complex shapes of varying texture and color are extracted with precision.
Figure 6.28. * (1st column) Image with rooftops of multiple colors from Pensacola, Florida. (2nd column) The blue and yellow markers denote the rooftops detected using the probabilistic framework algorithm. (3rd column) Rooftops extracted using our proposed algorithm.

Figure 6.29. * (1st column) Rooftop image of an urban location in Florida (2nd column) Rooftops extracted using our proposed algorithm. (3rd column) The blue and yellow markers denote the rooftops detected using the probabilistic framework algorithm.
6.4 Conclusion

This section provided two unique approaches to building detection and extraction. Both methods have their strengths and deficiencies. In the supervised approach, the number of training samples can significantly affect the segmentation quality. It maybe required to train a few samples per image to obtain satisfactory results. Further, this algorithm is slow and computationally intensive. As a result, a faster and unsupervised approach is also described. This algorithm can adapt itself to each new image by generating samples based on shadow and shape evidence. Graph Cut based energy minimization is found to be useful in extracting the precise outer contours of the rooftops. This approach is also found to outperform a state-of-the-art building detection algorithm. However, further investigation needs to be done to make this algorithm more robust to shape, texture and coloring variations.
CHAPTER 7

DAMAGE ESTIMATION

Damage estimation can be considered from two aspects—change detection and change 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 correlates with the extent of change in the roof structure seen from an aerial view [Womble (2005)]. Further, the damage states observed at ground-level have been found to correlate well those observed from space using remote-sensing data [Brown et al. (2011)].

In this chapter, we explore change classification as opposed to just change detection. The objective of this work is to determine the reliability of high resolution images for a detailed fine-grained damage analysis and to match the performance of human volunteers who label the images in various damage categories. Our work is unique from previous studies in that a) we study the reliability of using 37cm to 1m aerial images for not just finding out whether the building has been damaged but also the type and severity of damage, b) the effect of pre-processing steps such as correction of geometric and photometric differences is explored, c) we propose new color, edge and intensity-based features, for damage classification.
and compare them with previous works. d) For resolving ground truth ambiguity, we ask expert and non-expert volunteers to identify different damage states using visual inspection and compare the predictions provided by the algorithms with their interpretation.

7.1 Related Work

Generally, damage features can be classified as edge-based or color-based features. Yamazaki (2001) discusses the use of color indices and edge elements in damage classification. Adams et al. (2004), Matsuoka et al. (2004) involve the use of edge-based measures to analyze textural dissimilarity. Simple grayscale statistics were calculated in Sampath (2004). Womble et al. (2007), Womble et al. (2006) computed statistics for each band, including standard deviation, variance, average deviation, skewness, uniformity, and entropy. Radhika et al. (2010) used wavelet feature extraction to enhance features suggested by Womble et al. (2007). More recently other texture and statistical measures have been used for damage detection. Chen and Hutchinson (2005) used correlation analysis, principal component analysis and boundary compactness index of extracted rooftops. Vijayaraj et al. (2008) used Local binary pattern (LBP), local edge pattern (LEP) and Gabor texture features computed over pixels and blocks of pixels. Sirmacek and Unsalan (2009a) used shadow length differences as a hint of damage and ratio of rooftop area to shadow region was used as a damage metric. In contrast to all these efforts, we address the ground truth ambiguity problem and present the evaluation of our features on the largest known collection of rooftop images used in any study.
7.2 Preprocessing

7.2.1 Rooftop Registration

First, the two step process described in Chapter 4 is used to achieve fast and robust registration of before- after-disaster aerial image pairs. The images are coarsely registered using a phase-correlation based algorithm. For fine image registration, we adopted the use of SURF-feature based matching. The coarsely registered images are divided into grids and features are matched across corresponding grids using an approximate nearest neighbor search algorithm combined with a constrained RANSAC algorithm for point-pairs subset selection.

Second, to overcome the effect of heights, each extracted rooftop may require a separate registration stage. It should be noted that a previous work [Chesnel et al. (2007)] used a cross correlation based technique for rooftop registration and observed improvements in classification rates. We propose to improve rooftop registration accuracy by maximizing and thresholding the Fourier cross power spectrum which is known to be more robust than cross-correlation. The bounding rectangle for each extracted rooftop is calculated by fitting a rectangle circumscribing the boundary contour of the rooftop. Additionally, the dimensions of the rectangle are expanded to account for possible registration error. Each before-storm rooftop defined by its bounded rectangle is then compared with the corresponding area in the after-storm image. Let \( f_1 \) and \( f_2 \) be the two rooftop images that differ only by a displacement \((t_x, t_y)\) i.e.,

\[
f_2(x, y) = f_1(x - t_x, y - t_y)
\]  

(7.1)
The cross-power spectrum \( ps \) of two images \( f_1 \) and \( f_2 \) is defined as

\[
ps = \text{IFT}_{\text{maxpeak}} \left( \frac{F_1(\xi, \eta)F_2^*(\xi, \eta)}{|F_1(\xi, \eta)F_2^*(\xi, \eta)|} \right) = \text{IFT}_{\text{maxpeak}} \left( e^{j2\pi(\xi t_x + \eta t_y)} \right) \tag{7.2}
\]

where \( F_2^* \) is the complex conjugate of \( F_2 \). By taking inverse Fourier transform of the representation in the frequency domain, we will have a function that is an impulse; that is, it is approximately zero everywhere except at the displacement \((t_x, t_y)\) that is needed to optimally register the two images. Slight rotation and scaling differences between two images are found by computing the scale and rotation which maximizes \( ps \) as described in Chapter 4. In addition, we thresholded the value of the peak of phase correlation \( ps \leq p_{\text{thresh}} \) to filter rooftop images which do not register correctly. The effect of rooftop registration on classification are described in the evaluation section.

### 7.2.2 Color Correction

Color correction is done by transferring the color characteristics of the before-storm rooftop image to the after-storm rooftop image. The color balancing algorithm described in Chapter 5 is used. We briefly discuss the main steps in this process here. Consider the before-storm rooftop image \( s(i, j) \), after-storm image \( t(i, j) \) and new after-storm image \( t^{\text{new}}(i, j) \). Color transfer proposed previously first converts the \( RGB \) color space into \( l\alpha\beta \) color space. Once the channels have thus been decorrelated, the statistics are transfered by the following equations:

\[
t^{\text{new}}(i, j) = \mu_{s(i, j)}^k + \frac{\sigma_{s(i, j)}^k}{\sigma_{t(i, j)}^k} (t(i, j) - \mu_{t(i, j)}^k) \tag{7.3}
\]
\[
\mu_{k(s(i,j))}^k = \frac{1}{k^2} \sum_{l=i-k}^{i+k} \sum_{m=j-k}^{j+k} s(l,m)
\]  
\[ (7.4) \]

\[
\sigma_{k(s(i,j))}^k = \frac{1}{k} \sqrt{\sum_{l=i-k}^{i+k} \sum_{m=j-k}^{j+k} (s(l,m) - \mu_{k(s(i,j))}^k)^2}
\]  
\[ (7.5) \]

The means are now indicated by \( \mu_{k(s(i,j))}^k \) and \( \mu_{k(t(i,j))}^k \), where \( k \) denotes the length of the window used for transferring the statistics around the pixel \((i, j)\). Similarly, the standard deviations are indicated by \( \sigma_{k(s(i,j))}^k \) and \( \sigma_{k(t(i,j))}^k \).

The window length \( k \) for each pixel can be fixed by calculating the value of \( \text{NCC}(i, j) \) for a range of window sizes and choosing the smallest window size that gives a sufficiently high NCC value.

7.3 Features for Damage Classification

Due to their ease of implementation and efficacy in identifying damaged rooftops, we implemented various features used in previous works. These features were identified as the top performers in their respective works and provide simple measures to quantify textural change. These include standard deviation, uniformity \( \text{Womble et al. (2007)} \), correlation analysis \( \text{Chen and Hutchinson (2005)} \), local binary pattern (LBP) and local edge pattern (LEP) \( \text{Vijayaraj et al. (2008)} \). In addition, we propose new features which include gradient magnitude bins, features based on an invariant color model, and edge density.

Two approaches can be used to calculate each feature; extraction per rooftop and extraction over grid of cells. However, due to poor performance of features computed over entire rooftops, we do not report those results in this work. For
the rest of this work (with the exception of Gradient Magnitude Bins), all features are calculated after dividing each rooftop into a grid of cells. In this proposed approach, each rooftop is divided using a grid, the features are extracted for each cell in the grid, quantized into bins and described by histograms of cells. See Figure 7.1 for an example. This is done in order to capture the localization of damage and improve classification. Thus a feature vector is extracted for each rooftop by concatenating all features as: \( v = \{v_{stddev}, v_{uniform}, \ldots \text{etc.} \} \), where \( v_{stddev}, v_{uniform}, \ldots \text{etc} \) correspond to various features that are calculated over grid of cells as described in this section.

![Figure 7.1](image.png)

Figure 7.1. A grid is placed over (a) before event building image and (b) after disaster building. (c) Features are extracted for each cell. (d) Each feature calculate per cell is quantized into bins as indicated by the intensity of the heat map. A histogram of these cells is then typically computed.
7.3.1 Variance and Uniformity

The variance is commonly used to describe the variability of a given value from its mean. It can be used to describe relative smoothness of the texture in grayscale images. $Var_{i,j}$ of cell $(i,j)$ is calculated using the pixel values $I_{i,j}(x,y)$ of the cell as

$$Var_{i,j} = \sigma^2 = \frac{\sum_x \sum_y (I_{i,j}(x,y) - \mu_{i,j})^2}{N}$$  \hspace{1cm} (7.6)

where $N$ is the total number of pixels in the cell and $\mu_{i,j}$ denotes the mean of the cell.

Uniformity is an illumination invariant textural measure used to describe the coarseness of an object based on its histogram \cite{Womble2007}. For a particular RGB channel or band, the frequency of a pixel represents the number of occurrences of a particular pixel value within the object. The occurrence probability of a pixel value, $p(x)$, is simply the frequency of pixel value $x$ divided by the number of pixels in the cell, and ranges in value from 0 to 1. The sum of the probabilities for all pixel values is exactly 1. The uniformity for each cell is given by

$$Unif_{i,j} = \sum p_{i,j}^2(x)$$  \hspace{1cm} (7.7)

In our implementation, we calculated the variances and uniformities for each 10x10 pixels cell in both the before and after-storm images. The absolute differences of these values for each cell were then quantized into 10 bins. Histogram of bin values were then computed to have two histograms $v_{stddev}$ and $v_{uniform}$ each
of size 10. Figure 7.2 shows a heat map for both these features. As evident from the figure, uniformity is usually very sensitive to slight textural differences and hence less reliable than variance.

Figure 7.2. A heat map displaying severity of damage found by variance and uniformity features, with more severe damages shown closer to red. (First column) Before storm image, (Second column) after storm image (Third Column) Variance, and (Fourth Column) Uniformity.

7.3.2 Correlation Analysis

The Pearson correlation coefficient is described as a measure of the linear relationship between two random variables. It is calculated between cells $(i,j)$
corresponding to the before and after-storm images as:

$$PCC(i, j) = \frac{\sigma_{s_{i,j}t_{i,j}}}{\sigma_{s_{i,j}}\sigma_{t_{i,j}}}$$ (7.8)

$\sigma_{s_{i,j}}$ and $\sigma_{t_{i,j}}$ are variances of before- and after-storm image cells. $\sigma_{s_{i,j}t_{i,j}}$ is the cross-covariance between the two cells.

Physically, a higher correlation coefficient indicates less change occurred between a pair of images. Correlation coefficients can be calculated over a pair of cells. Alternatively, a pixelwise calculation within a sliding fixed-size window can first be calculated throughout the before and the after images, and a representative correlation coefficient can use the average value of all the calculated correlation coefficients confined by the boundary of the cell. Previous work [Chen and Hutchinson (2005)] found that the latter method better characterizes structural damage and hence was implemented for this work. In our implementation, the cell sizes were 10X10 pixels and correlation coefficient values for each cell were then quantized into 10 bins. The feature extracted $v_{CC}$ is thus a histogram of bin values and is of length 10.

7.3.3 Local Binary and Edge Pattern

LBP (Local Binary Pattern) based features have been used in various applications like face detection, image analysis and image retrieval because of its better tolerance of illumination changes. [Vijayaraj et al. (2008)] used LBP for damage detection of rooftops. The LBP is computed by using a moving window operator and producing the binary pattern by thresholding the window elements by the center pixel. The binary pattern is assigned to the center pixel. The histogram of the binary patterns in an image is computed and compared. The LBP values
encode different patterns like line, edges, spots and corner to their corresponding patterns under varying illuminations. LEP (Local Edge Pattern) is similar to LBP, except that it is extracted from edge maps rather than pixel intensity values. For more details on LBPs the reader is advised to refer to [Ojala et al. (1996)]. In our implementation, LBP/LEP histogram of size 256 were calculated for each cell in the before and after image. The cell sizes were 10X10 pixels and a chi square distance of corresponding histograms were calculated as follows:

$$\chi^2(i, j) = \frac{1}{2} \sum_{k=1}^{256} \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}$$

(7.9)

These distance values for each cell were then quantized into 10 bins and features extracted $v_{LBP}, v_{LEP}$ are thus histograms of bin values and each is of length 10. Figure 7.3 shows a heat map for LBP features. In general, while LBP is usually able to distinguish no damage and damage states, they do not appear to capture the severity of the damage.
Figure 7.3. A heat map displaying severity of damage found by PCC and LBP features, with more severe damages shown closer to red. (First column) Before storm image, (Second column) after storm image, (Third Column) PCC, and (Fourth Column) LBP.

7.3.4 Edge Density

Some previous works [Adams et al. (2004), Radhika et al. (2010)] used edge based features. We propose the use of edge density to identify and categorize more fatal structural damage. When a roof collapses partially or fully, 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 define 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 7.4 and Figure 7.5. See Figure 7.6 for absolute differencing and edge detection result. Finally, the number of pixels classified as edges in each 10X10 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 very high, then it is usually more likely that the roof has collapsed. These density values for each cell were then quantized into 10 bins and features extracted $v_{ED}$ are thus histograms of bin values and each is of length 10.

![Figure 7.4. Google Earth images of National Palace, Haiti from (a) before and (b) after 2009 Haiti earthquake.](image-url)
Figure 7.5. Buildings extracted from (a) before and (b) after 2009 Haiti earthquake images shown in Figure 7.4.

Figure 7.6. (a) Absolute difference of the before and after images. (b) Result of Canny edge detection.
7.3.5 Gradient Magnitude Bins

Rooftops and other man-made structures are typically characterized by strong gradients towards certain directions in the image. For example, a rectangular building would show strong gradients in 4 orientations. When rooftops are missing or totally destroyed, these gradients become weak in the after-storm image and the magnitude of gradients are more evenly distributed among all possible orientations (See Figure 7.7). To capture this, we propose a new feature called gradient orientation bins and which is computed over the entire rooftop (unlike other features described in this section) as follows. First, we calculate smoothed (using a Gaussian function) gradients in the x- and y-directions in the grayscale image $I(x, y)$.

$$g_x(x, y) = -\frac{x}{2\pi T_g^4} \exp\left(-\frac{x^2 + y^2}{2T_g^2}\right)$$ (7.10)

$$g_y(x, y) = -\frac{y}{2\pi T_g^4} \exp\left(-\frac{x^2 + y^2}{2T_g^2}\right)$$ (7.11)

where $T_g$ is the smoothing parameter. We calculate the smoothed gradients for the image $I(x, y)$ as

$$I_x == \frac{d}{dx} I(x, y) = g_x(x, y) \ast I(x, y)$$ (7.12)

$$I_y == \frac{d}{dy} I(x, y) = g_y(x, y) \ast I(x, y)$$ (7.13)

where $\ast$ stands for the 2-D convolution operation. The orientation and mag-
The magnitude of gradients can be calculated as

\[ I_\theta(x, y) = \tan^{-1}\left( \frac{I_y}{I_x} \right) \]  

(7.14)

\[ I_{mag}(x, y) = \sqrt{(I_x)^2 + (I_y)^2} \]  

(7.15)

Gradient magnitude bins \( hist_{I_\theta} \) can then be computed for all pixels \((x, y)\) as:

\[ hist_{I_\theta}(\frac{I_\theta(x, y)}{N}) = hist_{I_\theta}(\frac{I_\theta(x, y)}{N}) + I_{mag}(x, y) \]  

(7.16)

where \(0 \leq I_\theta(x, y) \leq 360\) and \(N\) denotes the bin size. The size of \(N\) is typically larger than 1 to reduce the space required to store the features as well as make it more invariant to noise. In our implementation \(N = 5\) gave good performance. The feature vector \(v_{GMB}\) is obtained by simply differencing the corresponding gradient magnitude bins for before and after rooftops and is of length 72.
7.3.6 Features from invariant color model

Most previous works extracted features from grayscale-based or RGB images. In contrast, we propose new features that are based on the HSV color model. Previous studies [Tsai (2006)] have established the use of such color models in problems such as shadow compensation. This color space is more invariant to photometric differences as it decouples luminance and chromaticity. Color tone is a powerful descriptor and conversion into HSV color model helps identify features that intuitively have more discriminating capacity. In this section we introduce two features based on HSV model called Hue Means and V Difference. 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 RGB to HSV conversion is done as follows:

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

(7.17)

\[
    s = \begin{cases} 
        0, & \text{if } \text{max} = 0 \\
        \frac{\text{max-min}}{\text{max}} = 1 - \frac{\text{min}}{\text{max}}, & \text{otherwise} 
    \end{cases} 
\] 

(7.18)

\[
    v = \text{max} 
\] 

(7.19)

The \text{max}, \text{min} denotes the maximum and minimum of RGB values respectively.

7.3.6.1 V Difference

When the roof structure is partially or completely destroyed, it can be seen as from aerial view dark cavity. The change in V (value) components can be used to discriminate roofs with such cracks, openings or holes as it represents a decrease in illumination in the corresponding regions. Hence, to compute V Difference, the values are subtracted and the difference is summed up. A larger sum usually indicates the presences of cavities. Consider the V components of corresponding cells of before and after images to be \text{V}_1 and \text{V}_2 respectively. The proposed feature
(7.20)

\[ V_{diff}(V_1, V_2) = \sum_x \sum_y (V_1(x, y) - V_2(x, y)) \]

\[ V_{diff} \] is the quantized into 10 bins and and a histogram \( v_{VDiff} \) is calculated.

7.3.6.2 Hue Means Histogram

The Hue component corresponds to the color value of a pixel. Since it is independent of luminance and saturation (richness of the color), it is assumed to be invariant to photometric differences. 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° to 360°. 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.

Further, damages like removed tiles show distinct patterns in the corresponding after-storm hue histograms. However those patterns can be trusted only when the difference between the hue values of before and after storm is high enough.

Hence we compute another feature, a 2D hue means histogram, to reflect the hue spectrum of after-storm image as well as the degree of change corresponding to each spectrum. More formally, consider the hue components of before and after image cells to be \( H_1 \) and \( H_2 \) respectively and their corresponding means are \( \overline{H_1} \) and \( \overline{H_2} \).

\[ H_{diff} = |(\overline{H_1} - \overline{H_2})| \]  

(7.21)

We can ensure that the smaller angle is chosen by computing \( 360 - H_{diff} \), if
$H_{\text{diff}} > 180$. A 2D Histogram of cells is computed by considering $H_{\text{diff}}$ as the first dimension and $T_2$ as the second dimension. In order to reduce the space required to store the values, $H_{\text{diff}}$ and $T_2$ were quantized to 3 bins and 90 bins respectively. The feature vector $v_{HMHist}$ is obtained by concatenating the rows of the 2D histogram to form a single linear vector of length 270. See Figure 7.8 for examples that demonstrate how edge density, V difference and Hue Means could be used to capture different types of damage.

Figure 7.8. 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.
7.4 Supervised Learning for Damage Classification

The different edge, intensity and color based features proposed in the previous section can be used to classify buildings in an image into categories ranging from no damage to severely damaged. The most popular damage scale used for hurricane damages is the RS scale which was proposed by Womble (2005) and categorizes rooftops varying from no-damage to collapsed/missing. A description of the RS scale is shown in 7.9. The damage classification process proposed is a supervised learning approach that uses the features for classifying damage into these qualitative states described by RS scale.
<table>
<thead>
<tr>
<th>Damage Rating</th>
<th>Most Severe Physical Damage</th>
<th>Remote-Sensing Appearance</th>
</tr>
</thead>
</table>
| RS-A          | No Apparent Damage         | • No significant change in texture, color, or edges.  
|               |                            | • Edges are well-defined and linear.  
|               |                            | • Roof texture is uniform.  
|               |                            | • Larger area of roof and more external edges may be visible than in pre-storm imagery if overhanging vegetation is removed.  
|               |                            | • No change in roof-surface elevation. |
| RS-B          | Shingles/tiles removed, leaving decking exposed | • Nonlinear, internal edges appear (new material boundary with difference in spectral or textural measures).  
|               |                            | • Newly visible material (decking) gives strong spectral return.  
|               |                            | • Original outside roof edges are still intact.  
|               |                            | • No change in roof-surface elevation. |
| RS-C          | Decking removed, leaving roof structure exposed | • Nonlinear, internal edges appear (new material boundaries with difference in spectral or textural measures)  
|               |                            | • Holes in roof (roof cavity) may not give strong spectral return.  
|               |                            | • Original outside edges usually intact.  
|               |                            | • Change in roof-surface elevation.  
|               |                            | • Debris typically present nearby. |
| RS-D          | Roof structure collapsed or removed. Walls may have collapsed. | • Original roof edges are not intact.  
|               |                            | • Texture & uniformity may/may not experience significant changes.  
|               |                            | • Change in roof-surface elevation.  
|               |                            | • Debris typically present nearby. |

Figure 7.9. RS Damage scale for hurricane damages described in [Womble (2005)]

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 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 Breiman (1996). 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 Zenko (2004).

7.5 Reliability of ground-truth preparation

For ground truth preparation, we used before and after hurricane images from publicly available NOAA and USGS aerial imagery dataset described in Chapter 3. First, an image dataset of 635 rooftops was created by manually outlining individual rooftops in the before-storm images. Subsequently, the mask images produced contain white pixels for every rooftop pixel and black pixel for every non-rooftop image. Multiple volunteers ensured that the masks produced were accurate and reliable. Next, an expert volunteer labelled all 635 buildings into the 4 RS damage scales described in the previous section. As this damage scale can be ambiguous, various appearance-based qualitative categories described in Table 7.1 were also labeled for each individual rooftop. These simple categories are more intuitive and thought of as less ambiguous for visual interpretation.
<table>
<thead>
<tr>
<th>Category</th>
<th>Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is the building missing in the after-storm image?</td>
<td>Yes, No</td>
</tr>
<tr>
<td>Is there a lot of debris lying in close proximity to the building?</td>
<td>Yes, No</td>
</tr>
<tr>
<td>Are there one or more dark cavities/holes in the rooftop caused by the storm?</td>
<td>Yes, No</td>
</tr>
<tr>
<td>What percentage of the total area of the rooftop has missing/displaced tiles after the storm?</td>
<td>&lt; 5%, 5-25%, 26-70%, &gt;70%</td>
</tr>
<tr>
<td>What percentage of the total area of the rooftop has collapsed after the storm?</td>
<td>&lt; 5%, 5-25%, 26-70%, &gt;70%</td>
</tr>
</tbody>
</table>

Ambiguity in the definition of RS scale classes that may have affect the manual ground truth preparation (See Figure 7.10). Previous studies have mostly ignored this problem or used a very small dataset. For estimating the level of ambiguity while preparing ground truth, a study of manual classification by multiple expert and non-expert volunteers was proposed. Each volunteer was presented with before- and after-storm rooftop images and asked to choose one of the four damage states. In addition, various appearance categories present in Table 7.1 were also labeled as a part of the questionaire. At the end of the experiment, the classifications from all the volunteers was compiled to produce agreement percent-
age levels for damage state of rooftops. 9 volunteers worked on a set of 50 rooftop image pairs. Another 13 volunteers worked on a separate set of 65 rooftop image pairs. Both the results are shown in Table 7.2.
TABLE 7.2

AGREEMENT% OF VISUAL INTERPRETATION BY TWO
GROUPS OF VOLUNTEERS.

<table>
<thead>
<tr>
<th>Category</th>
<th>Classes</th>
<th>Agreement % on 50 set</th>
<th>Agreement % on 65 set</th>
<th>Mean Agreement %</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS Damage scale</td>
<td>A, B, C, D</td>
<td>70%</td>
<td>68%</td>
<td>69%</td>
</tr>
<tr>
<td>Building missing</td>
<td>Y, N</td>
<td>98%</td>
<td>97%</td>
<td>98%</td>
</tr>
<tr>
<td>Debris presence</td>
<td>Y, N</td>
<td>78%</td>
<td>74%</td>
<td>76%</td>
</tr>
<tr>
<td>Cavity presence</td>
<td>Y, N</td>
<td>76%</td>
<td>81%</td>
<td>79%</td>
</tr>
<tr>
<td>Missing tiles percent</td>
<td>&lt; 5%, 5-25%, 26-70%, &gt;70%</td>
<td>66%</td>
<td>70%</td>
<td>68%</td>
</tr>
<tr>
<td>Collapse percent</td>
<td>&lt; 5%, 5-25%, 26-70%, &lt;70%</td>
<td>70%</td>
<td>71%</td>
<td>71%</td>
</tr>
<tr>
<td>Missing tiles</td>
<td>≤ 5%, &gt;5%</td>
<td>82%</td>
<td>84%</td>
<td>83%</td>
</tr>
<tr>
<td>Any collapse</td>
<td>≤ 5%, &gt;5%</td>
<td>79%</td>
<td>74%</td>
<td>77%</td>
</tr>
</tbody>
</table>
Figure 7.10. The first row shows before storm images. The second row shows after storm images. The first column shows an image pair where the exact state of the rooftop is ambiguous and hard to discern. The tiles are removed but it is unclear whether there are holes or cavities. The second column is an example where the entire rooftop structure is missing and yet the building is standing as evidenced by the shadows. The column shows either a collapsed or missing rooftop. The exact nature of damage is unclear though it is evident that it is serious.

For the group of 9 volunteers, a class label was considered agreed upon if at least 6 volunteers marked a rooftop into the same label. For the group of 13, this threshold was 9 volunteers. Both groups of volunteers appear to have made similar level of agreements. Since the percentage of missing tiles or collapse area show higher levels of interpretation ambiguity, we simplify these categories and add 2 new ones that could have two labels $leq 5\%$ or $>5\%$. The overall agreement % shown helps redefine accuracy expectations of supervised classification for dam-
age interpretation. It is proven that damage classification from high resolution aerial imagery is an inherently ambiguous problem and the estimates provided by classifiers should not be expected to exceed the agreement % shown in the above table. As the agreement % of RS damage scale is the lowest, we focus on classifier performance in identifying appearance based damage categories. These included 5 categories; missing building, collapse state, debris presence, cavity presence and state of tiles.

To prove that these classes are highly correlated with RS damage scale we train a J48 decision tree. The classifier was trained on the 635 rooftops dataset using the marked classes as features to predict the RS damage scale. The resultant decision tree is shown in Figure 7.11 and predicts RS scale with 94% accuracy. Since debris presence was not selected as a feature by J48 algorithm and is the category that is the hardest to predict (demonstrated by the lower agreement %), we focus on predicting other 4 classes for the rest of this work.
Figure 7.11. A J48 tree trained on the 635 rooftops dataset to predict RS damage scale. The leaves represent A to D categories in the RS damage state and nodes represent features. These features are building missing ($F_0$), cavity presence ($F_1$), missing tiles ($F_2$) and collapse state ($F_3$).

7.6 Evaluation

For evaluation, we used the dataset of 635 rooftop image pairs with the manually prepared groundtruth described in the previous section. The class distribution for various damage appearance states is shown in Table 7.3. It can be observed that with the exception of missing tiles state, the dataset has an imbalanced class distribution. This can be problem while learning models from the dataset as the classifier can be inclined favor of the majority class. To overcome this, we used cost sensitive classification. Each class label is weighted with a pretermined cost. Reweighting the training instances according to the total cost assigned to each class is a common cost sensitive classification approach and is adopted in this work. To establish the statistical validity of classification results, all results
reported in this section were generated using k-fold cross validation. In k-fold cross-validation, the original dataset is randomly partitioned into k subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k - 1 subsamples are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds are then averaged to produce a single estimate. We used 10-fold cross-validation for generating all our results.

### Table 7.3

<table>
<thead>
<tr>
<th>Class</th>
<th>Building missing</th>
<th>Cavity presence</th>
<th>Missing tiles</th>
<th>Collapse state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>23</td>
<td>32</td>
<td>240</td>
<td>51</td>
</tr>
<tr>
<td>No</td>
<td>612</td>
<td>603</td>
<td>395</td>
<td>584</td>
</tr>
</tbody>
</table>

7.6.1 Selected Features and Classification Performance

We considered using all features described in previous section for the various classification experiments. After being extracted from the images all features were L2 normalized. We experimented with a variety of different classifiers mentioned
in Section 4 and chose random forests based on best performance. For the rest of this chapter, all results reported were generated by various iterations of random forests. Additionally, we experimented with each feature separately, all features together and with features selected using a best first subset evaluation algorithm. We report only the best results and their corresponding features.

The results are reported in Table 7.4. The metrics used for evaluation are accuracy, precision and recall. If $TP$ is the true positive rate, $FP$ is the false positive rate, $TN$ the true negative rate and $FN$ the false negative rate

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{7.22}
\]

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{7.23}
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{7.24}
\]

It can be observed from Table 7.4 that missing buildings are best identified using Gradient Magnitude Bins. It also had the highest accuracy among all other predictions. This result correlates well with the ground truth agreement percentages reported previously. Feature selection with best bin first was the best strategy for cavity presence and collapse state. The missing tiles state had the worst performance in terms of precision and recall. This is because of the high number of false positives and false negatives in identifying missing tiles. Figure 7.12 shows some of rooftops and their classifications.
<table>
<thead>
<tr>
<th>Before</th>
<th>After</th>
<th>Actual Missing tiles</th>
<th>Actual Cavity presence</th>
<th>Actual Collapse state</th>
<th>Predicted Missing tiles</th>
<th>Predicted Cavity presence</th>
<th>Predicted Collapse state</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.jpg" alt="Image 1" /></td>
<td><img src="image2.jpg" alt="Image 2" /></td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td><img src="image3.jpg" alt="Image 3" /></td>
<td><img src="image4.jpg" alt="Image 4" /></td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td><img src="image5.jpg" alt="Image 5" /></td>
<td><img src="image6.jpg" alt="Image 6" /></td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td><img src="image7.jpg" alt="Image 7" /></td>
<td><img src="image8.jpg" alt="Image 8" /></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>V</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td><img src="image9.jpg" alt="Image 9" /></td>
<td><img src="image10.jpg" alt="Image 10" /></td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
</tbody>
</table>

Figure 7.12. The actually and predicted damage categories for some of the images in the dataset are shown here. All the misclassifications are marked in red.
TABLE 7.4

BEST CLASSIFICATION PERFORMANCE FOR DIFFERENT FEATURE SELECTION SCHEMES. THE MEANS OF CLASS ACCURACY, PRECISION AND RECALL FOR EACH CATEGORY ARE REPORTED.

<table>
<thead>
<tr>
<th>Category</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Building missing</td>
<td>93.7%</td>
<td>0.956</td>
<td>0.945</td>
<td>$v_{GMB}$</td>
</tr>
<tr>
<td>Cavity presence</td>
<td>77.5%</td>
<td>0.926</td>
<td>0.827</td>
<td>selected with best first</td>
</tr>
<tr>
<td>Missing tiles</td>
<td>70.7%</td>
<td>0.704</td>
<td>0.707</td>
<td>$v$ (all)</td>
</tr>
<tr>
<td>Any collapse</td>
<td>72.3%</td>
<td>0.877</td>
<td>0.737</td>
<td>selected with best first</td>
</tr>
</tbody>
</table>

7.6.2 Effect of preprocessing

The preprocessing steps described in Section 7.2 have varying effects on the final accuracy. The effect of color balancing and rooftop registration on preprocessing is reported in Table 7.5. The measures used for evaluation include accuracy, F-measure and ROC area. If $TP$ is the true positive rate, $FP$ is the false positive rate, $TN$ the true negative rate and $FN$ the false negative rate,

$$F_{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times 100$$

(7.25)

The ROC area represents the area under the curve of a receiver operating characteristic (ROC) plot created by plotting the true positive rate vs. the false
positive rate, at various threshold settings. Values closer to 1 are more desirable for ROC area.

TABLE 7.5

BEST CLASSIFICATION PERFORMANCE FOR DIFFERENT PREPROCESSING SCHEMES. THE VARIOUS RESULTS REPORTED INCLUDE NO PREPROCESSING (noprepro), AFTER COLOR BALANCING (cb), AND AFTER COLOR BALANCING PLUS ROOFTOP REGISTRATION (cb + rr). MEAN ACCURACY, F-MEASURE AND ROC AREA FOR EACH CLASS ARE REPORTED.

<table>
<thead>
<tr>
<th>Class</th>
<th>Accuracy%</th>
<th>F-measure</th>
<th>ROC area</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>no pre-pro</td>
<td>cb</td>
<td>cb + rr</td>
</tr>
<tr>
<td>Building</td>
<td>93.7</td>
<td>95.6</td>
<td>94.5</td>
</tr>
<tr>
<td>missing</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cavity</td>
<td>77.5</td>
<td>83.7</td>
<td>82.7</td>
</tr>
<tr>
<td>presence</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Missing</td>
<td>70.7</td>
<td>77.9</td>
<td>76.7</td>
</tr>
<tr>
<td>tiles</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Any collapse</td>
<td>72.3</td>
<td>73.5</td>
<td>73.7</td>
</tr>
</tbody>
</table>
Overall, images after color balancing alone perform best in terms of accuracy, F-measure and ROC area. The highest improvement in accuracy is seen for missing tiles category. This maybe because the correction of hue spectrum reduced false positives and increases true negatives (See Figure 7.13). Collapse state appears to be hardly affected by any preprocessing while cavity presence shows significant improvement due to color balancing. Detecting missing buildings however shows only marginal improvement in accuracy. There appears to be little or no advantage in rooftop registration. Although this appears to be counter-intuitive, the inefficacy maybe partly attributed to the fact that most rooftops are already registered correctly and rooftop registration may in fact be counterproductive by introducing minor registration error.

Figure 7.13. Effect of color balancing on Hue Means feature is shown in this image. After the color correction, only exact places where tiles are removed are shown as severely damaged.
7.6.3 Estimating RS Damage Scale

To evaluate the performance of predicting RS damage scale, we first trained a classifier to separate RS-A from rest of the classes. Then, the instances categorized as non-RS-A were classified using appearance based damage states. The predictions of these damage states were then used to predict the RS damage scale using the simple decision tree described in Section 4. The final results are shown in Table 7.6. The overall accuracy in classifying into the 4-categories is 67.7%. Many of RS-B categorized buildings are confused with RS-A and RS-C. RS-C and RS-D labelled buildings had the worst performance with a lot of the buildings being confused with each other.

TABLE 7.6

A CONFUSION MATRIX FOR PREDICTING RS-SCALE FOR WINDSTORM DAMAGE

<table>
<thead>
<tr>
<th>Actual</th>
<th>Predicted</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>291</td>
<td>53</td>
<td>25</td>
</tr>
<tr>
<td>B</td>
<td>30</td>
<td>108</td>
<td>30</td>
</tr>
<tr>
<td>C</td>
<td>7</td>
<td>5</td>
<td>18</td>
</tr>
<tr>
<td>D</td>
<td>4</td>
<td>1</td>
<td>15</td>
</tr>
</tbody>
</table>
7.7 Conclusion

A common outcome of all previous research including ours which was described in this chapter, is that predicting multiple levels of damage categories is an inherently hard problem. Through our experiments we prove that this is because of the ambiguity in classifying a rooftop into any category. We estimated expected accuracies by establishing correlation with the performance of human volunteers. Further, we identified more intuitive appearance-based damage categories and proved that predicting them could decrease the ambiguity in identifying RS-damage scale as well. Towards this end, we evaluated novel as well as existing textural features for damage recognition and classification. Additionally, a color balancing preprocessing step is shown to be useful in improving performance.

Final evaluation indicates that our techniques match the performance of human visual interpretation for most of the damage categories identified. Missing buildings can be detected with 95.6% accuracy, presence of cavities with 83.7%, missing tiles with 77.9% and collapse state with 73.5%. Finally RS scale can be predicted using our techniques with 67%, that compares well the agreement % of volunteers which was nearly 70%. While the reliability of high resolution aerial imagery has shown to be limited in fine-grained damage analysis this work has pushed the boundaries of what is possible. Future work can concentrate on other textural measures, redefining damage states for a more meaningful damage analysis and analysis of very high resolution imagery.
7.8 Damage Estimation from Stereo Disparity

Height estimation from monocular aerial images is a hard problem. The images used to generate results in the previous section are inherently incapable of providing evidence of height except through the shadows that are cast by the buildings. If stereoscopic image pairs were available, 3D reconstruction or disparity estimation could provide useful information about variations in elevation of the rooftop before- and after a storm or earthquake. In this section, we explore the feasibility of generating disparity maps for damage detection and compare some of the existing algorithms.

Stereo vision relies to triangulation based on epipolar geometry to compute disparity maps and estimate depth from a pair of images of the same scene. Between two cameras there is a problem of finding a corresponding point viewed by one camera in the image of the other camera (known as the correspondence problem). In most camera configurations, finding correspondences requires a search in two-dimensions. However, if the two cameras are aligned to be coplanar, the search is simplified to one dimension - a horizontal line parallel to the line between the cameras (also known as epipolar lines). Furthermore, if the location of a point in the left image is known, it can be searched for in the right image by searching left of this location along the line. The 3D depths \( Z \) and disparities \( d \) is related by

\[
d = f \frac{B}{Z} \tag{7.26}
\]

163
where $f$ is the focal length, and $B$ is the baseline. Also,

$$x' = x + d(x, y), y' = y$$ (7.27)

describes the relationships between the corresponding pixels in left and right images.

Image rectification is an alternative to perfect camera alignment. The basic idea is that the images can be warped for they are co-planar and epipolar lines are horizontal. Once the corresponding horizontal scanlines are epipolar lines [Hartley and Zisserman (2004)], it is possible to match horizontal scanlines independently or to shift images horizontally while computing matching scores.

7.8.1 Comparison of existing Correspondence algorithms

Once the images are rectified, correspondences between pixels in left and right images need to be found to compute the disparity map $d(x, y)$. This is an area of ongoing research in computer vision community and there are no universal solutions for finding stereo correspondences that would work in all conditions. There is a wide range of stereo correspondence algorithms with different properties. Local methods, which are based on correlation are the most popular and are found to be suitable for real time applications. However, these methods assume constant disparities within a correlation window, which can lead to blurred object boundaries. Pixelwise matching [Birchfield and Tomasi (1998)] avoids this problem, but requires other constraints for unambiguous matching (e.g. piecewise smoothness). Dynamic Programming techniques can enforce these constraints efficiently, but only within individual scanline. Global approaches like Graph Cuts
and Belief Propagation enforce the matching constraints in two dimensions but are considered to be slower. We selected 3 algorithms based on their popularity and uniqueness.

### 7.8.1.1 Semi-global matching

Hirschmuller (2005) proposed SemiGlobal Matching method that performs pixelwise matching based on Mutual Information and the approximation of a global smoothness constraint. The algorithm was found to run very fast and with occlusions found with subpixel accuracy. This approach used a hierarchical calculation of Mutual Information based matching. An approximation of a global cost calculation is proposed that it can be performed in a time that is linear to the number of pixels and disparities.

### 7.8.1.2 Graph-cut matching

Kolmogorov and Zabih (2001) formulated stereo disparity matching as an energy minimization function. They presented two new methods which addressed finding occlusions with high accuracy, while preserving the advantages of graph cut algorithms. Their results showed that the proposed method works well for both detecting occlusions and computing disparities for stereo as well as motion. This method is generally regarded as robust and more accurate in comparison to other stereo algorithms. But Graph Cut optimization can be a time consuming step.
7.8.1.3 Line growing based matching

Alagoz (2008) described a line growing based stereo matching algorithm. The line-growing mechanism in two phases operation. In the first phase, the root point to grow lines is selected. This is known as the root selection process. In the second phase, the lines are gown using a rule. The rule for associating a point to root point in the growing process is to have lower error energy than a predetermined threshold of error energy. Being associated to a root points means to have the same disparity by root point. Thereby, the lines emerged from all associated points has a disparity value.

7.8.1.4 Evaluation

We evaluated the selected algorithms on 2 left right aerial image pairs (See Figure 7.14, Figure 7.16). The generated disparity maps with brighter pixels representing higher disparities are shown in Figure 7.15 and Figure 7.17. In Figure 7.15 it can be seen all 3 algorithms introduces noise and certain distortions along rooftop boundaries. The Graph Cut solution appears to be the better performer among the three while the line based algorithm performs the worst. However disparities generated and shown in Figure 7.17 appear to be very noisy and entirely unreliable. The resolution and lower quality of the aerial image pair are suspected to be the probable causes. In general, the evaluated algorithms seem insufficient for disparity estimation from aerial image pairs. This maybe because the underlying primitives such as pixels, blocks of pixels and lines are hard to match between such images. In this case, domain specific constraints such as searching only over a set width along horizontal scanlines and the fact that rooftop are usually planar regions of homogeneous color may help. Figure 7.18
shows a meanshift segmented version of Figure 7.14. Notice, how segmentation is consistent across the left right image pairs. Matching segments as opposed to pixels are hence recommended as future directions for stereo correspondence in aerial images.

Figure 7.14. A left-right rectified stereo image pair of an urban scene.
Figure 7.15. Disparities computed using (a) Graph Cut, (b) Hirschmuller algorithm and (c) line growing algorithm.
Figure 7.16. A left (above) and right (below) rectified stereo image pair from Joplin, Missouri.
Figure 7.17. Disparities computed from Figure 7.16 using (top) GraphCut, (middle) Hirschmuller algorithm and (bottom) line growing algorithm.
Figure 7.18. A meanshift segmented version of the urban area images.
8.1 Contributions

- A novel algorithmic fusion of phase and feature-based matching for registration of remote sensing imagery for change detection is presented [Thomas et al. (2012b)]. In comparison to conventional matching algorithm, and individual phase or feature based registration, our approach was more robust and had a speed-up of 30x at higher resolutions. The registration errors are consistently less than 3px even at finer resolutions and this work presents a breakthrough in registration for change detection applications.

- A novel color balancing algorithm with a 10% improvement in color similarity measures over other state-of-the-art algorithms was proposed. This work was published in [Thomas et al. (2012a)]. Also, a study on the effectiveness of color balancing as a preprocessing step shows that this power local color correction scheme improves classification rates.

- The first application of multi-label Graph Cut optimization for rooftop extraction is presented. Additionally, a novel and robust segment combination based algorithm that detected rooftops with a detection percentage of 88% is discussed. This work was published in [Thomas et al. (2011)]. Our evaluation also shows that the proposed algorithms outperform an existing state-of-the-art building detection algorithm.

- The first known study on the reliability of aerial images in damage classification was presented. We proposed a novel fusion of color and edge based features to classify rooftops into various damage categories ([Thomas et al. (2009a), Thomas et al. (2009b), Thomas et al. (2011)]). This is the first ever attempt to find the specific nature of damage to individual rooftops from aerial images. The evaluation was performed on the largest collection of pre and post-hurricane rooftops images known to have been studied to date. The proposed scheme was found to perform closely in comparison to human volunteers.
8.2 Summary

This research presented an automatic system for the assessment of damage from high resolution imagery. Towards that end, we have proposed and tested techniques for image registration, color balancing, building extraction, and damage classification; each of which requires little or no manual supervision. While the accuracy in previous approaches was limited by the number of control points in manual registration [Bitelli et al. (2004)], our application of phase correlation, feature detection along with robust matching schemes was found to produce near perfect registration on all images in our dataset. We proposed two novel segmentation-based building detection algorithms. Unlike previous approaches [Sirmacek and Unsalan (2009b), Noronha and Nevatia (2001), Song et al. (2006)] we do not make the assumption that buildings are composed of homogeneous segments or linear edges. Our algorithms were able to accurately extract the boundary contours of buildings in a reasonable amount of time.

Our study on the reliability of aerial images in fine-grained estimation of rooftop damage with multiple volunteers show that RS scale can only be estimated with 69% percent confidence. However, missing tiles, absence of rooftop, collapsed state and prescence of cavities can be predicted with 83%, 98%, 77% and 79% confidences. We proposed new features 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 this automated classification were promising and comparable to performance by human volunteers.
8.3 Future Work

While the proposed system attempts at automating every aspect of damage assessment, there are several areas which may require some 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, major areas that requires improvement are the speed of the interest point matching process. Currently the system can register very large images in a few minutes on a Desktop machine. For the building detection process the performance is heavily dependent on the quality of segmentation. Also, the segment combination process is computationally intensive. Tuning parameters maybe required for the algorithms to run correctly with different types of image datasets. More work needs to be done to make the algorithm less sensitive to parameters. Lastly, for damage classification, we could use additional features such as boundary compactness, perceptual feature organization etc. to improve the performance even further. However the performance is not expected to improve dramatically for the image resolution ranges used in this work. Future work must address the reliability and estimation of damage from VHR (very high resolution) aerial imagery. A study on calculating disparity maps from VHR imagery would also be useful in incorporating 3D evidence in the damage estimation scheme. In closing, it is envisaged that the robust automated tools for damage assessment from high resolution images presented in this work will play a significant role in rapid post-hurricane damage assessment and in helping rescue and recovery missions.
BIBLIOGRAPHY


