AUTOMATED BUILDING DETECTION FROM SATELLITE IMAGES BY USING SHADOW INFORMATION AS AN OBJECT INVARIANT

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BARIŞ YÜKSEL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2012

Approval of the thesis:

AUTOMATED BUILDING DETECTION FROM SATELLITE IMAGES BY USING SHADOW INFORMATION AS AN OBJECT INVARIANT

submitted by **BARIŞ YÜKSEL** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** _____

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering** _____

Prof. Dr. Fatoş Yarman-Vural
Supervisor, **Computer Engineering Dept., METU** _____

Examining Committee Members:

Prof. Dr. Aydın Alatan
Computer Engineering Dept., METU _____

Prof. Dr. Fatoş Yarman-Vural
Computer Engineering Dept., METU _____

Prof. Dr. Göktürk Üçoluk
Electrical and Electronics Engineering Dept., METU _____

Asst. Prof. Dr. Sinan Kalkan
Computer Engineering Dept., METU _____

Asst. Prof. Dr. Ahmet Oğuz Akyüz
Computer Engineering Dept., METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: BARIŞ YÜKSEL

Signature :

ABSTRACT

AUTOMATED BUILDING DETECTION FROM SATELLITE IMAGES BY USING SHADOW INFORMATION AS AN OBJECT INVARIANT

Yüksel, Barış

M.S., Department of Computer Engineering

Supervisor : Prof. Dr. Fatoş Yarman-Vural

September 2012, 75 pages

Apart from classical pattern recognition techniques applied for automated building detection in satellite images, a robust building detection methodology is proposed, where self-supervision data can be automatically extracted from the image by using shadow and its direction as an invariant for building object. In this methodology; first the vegetation, water and shadow regions are detected from a given satellite image and local directional fuzzy landscapes representing the existence of building are generated from the shadow regions using the direction of illumination obtained from image metadata. For each landscape, foreground (building) and background pixels are automatically determined and a bipartitioning is obtained using a graph-based algorithm, Grabcut. Finally, local results are merged to obtain the final building detection result. Considering performance evaluation results, this approach can be seen as a proof of concept that the shadow is an invariant for a building object and promising detection results can be obtained when even a single invariant for an object is used.

Keywords: building detection, markov random fields, graph cut, remote sensing, mathematical morphology

ÖZ

GÖLGE BİLGİSİ KULLANILARAK UYDU GÖRÜNTÜLERİNDEN OTOMATİK BİNA TESPİTİ

Yüksel, Barış
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Fatoş Yarman-Vural

Eylül 2012, 75 sayfa

Uydu görüntülerinden otomatik bina tespiti için literatürde uygulanan klasik örüntü tanıma yöntemlerinin aksine, gölge bilgisinin ve yönünün bina için değişmez olarak kullanılması sayesinde öz denetim verisinin otomatik olarak çıkartılabildiği güçlü bir bina tespiti algoritması önerilmiştir. Bu yöntemde ilk olarak görüntüdeki bitki örtüsü, su ve gölge alanları tespit edilir, ardından binanın varlığını tespit eden yerel yönelimsel bulanık haritalar, gölge alanları ve görüntünün meta-verisinden elde edilen gölge yönü bilgisi kullanılarak yaratılır. Her bulanık harita için, arka plan ve ön plan (bina) pikselleri otomatik olarak belirlenir ve çizge tabanlı Grabcut algoritması kullanılarak ikili bölütleme yapılır. Yerel sonuçların birleştirilmesinin ardından nihai bina tespit algoritması elde edilir. Algoritmanın başarım değerleri göz önüne alındığında, bu yaklaşım gölgenin bina için bir değişmez olduğu, ve nesne tespiti için yalnızca tek bir değişmez kullanıldığında bile tatmin edici sonuçlar elde edilebileceği görülmektedir.

Anahtar Kelimeler: bina tespiti, markov rastgele alanları, çizge kesimi, uzaktan algılama, matematiksel biçimbilim

In memory of Hasan Balıkçı...

ACKNOWLEDGMENTS

I want to thank to Dr. Ali Özgün Ok and Çağlar Şenaras for their valuable comments, supports and research collaboration. I also want to acknowledge fellow Computer Engineering Image Laboratory members for the good old days.

I want to thank to my (small) family for their great support and patience during the stage of writing the thesis.

Finally, I want to thank to my (large) family, that had recreated me with the enlightenment of the **tangible** science inside the **real** life, outside four walls...

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Nowadays; by virtue of the advances in satellite data acquisition, object detection and classification problems in remote sensing, specifically in very high resolution (VHR) satellite images, have been a popular research topic. An important object detection problem in remote sensing is the automated detection of buildings from satellite images.

Up to now, several building detection methodologies have been proposed in the literature. For the cases where multiple sensors are used for data acquisition, the problem becomes easier. Using Light Detection and Ranging (LIDAR) [101] or stereo imaging, it is possible to obtain a reliable height information of the objects in the terrain. This information can be integrated with the multispectral color information obtained from the optical sensor to detect buildings accurately. Moreover, fusion of optical and synthetic aperture radar (SAR) [23] sensors provides more reliable information than utilizing only one of these sensors.

On the other hand, building detection from monocular images is much more difficult. Unfortunately, it is not possible to generalize the shape, size, color of a building. Therefore, building detection methodologies using classical pattern recognition approaches are not guaranteed to work in every case; the performance depends on the statistical correlation between training and test data (for supervised approaches), and distribution of the features extracted from the image (for unsupervised approaches). Introduction of too many parameters and the algorithm's sensitivity to these parameters is another issue.

For a robust and generalizable object detection algorithm; obtaining an *invariant*[1] for the object is essential.

[1] The term *object invariant* is introduced by Prof. Dr. Fatoş Yarman-Vural.

Definition 1.0.1 (Object Invariant) *In an object detection problem; a feature, observation or property of an object which does not change from an image to another is called an invariant of the object.*

To give an example; for detecting balls in images, the circularity can be utilized as an invariant for balls, because every ball in the world has a circular shape. Moreover; in order to detect lemons in an image, yellow color is considered as an invariant for lemon object, since all lemons have a color of yellow. It is worth noting that the object invariance property is not one-to-one and onto. In other words;

- Every object satisfy the constraints related to its invariant (e.g. all lemons in the world are yellow),
- But an arbitrary object satisfying the invariant constraints is not neccessarily the object to be detected (e.g. a yellow object in the image may not be a lemon).

For building detection from monocular satellite images, this invariant can be shadow of a building: All buildings differ from each other in terms of their colors, sizes, shapes, textures; but they all have shadow regions attached to them.

Exploiting the detected shadows and the illumination direction (which are mostly available and will be explained in further chapters) in the image, the fundamental motivation of the proposed approach in this thesis is to develop a robust, accurate, generalizable and efficient building detection framework as a successful proof of concept for utilizing shadow as a building invariant. In this framework; given a satellite image, first the vegetation, water and shadow regions are detected from the image, and local fuzzy landscapes are generated from each shadow region using the direction of illumination obtained from image metadata. Then, considering the fact that there may be other objects with attached shadow regions, the landscapes generated from non-building objects are eliminated. Afterwards, for each fuzzy landscape, seed pixels are determined for foreground and background: The set of pixels near shadow in the direction of illumination are selected as foreground seeds, and non-building regions (vegetation, shadow, water) are selected as background seeds. Using these seed pixels, a bi-partitioning is obtained using a graph-based algorithm called Grabcut [97]. Finally, the local results are merged to obtain the final building detection result. Figure 1.1 shows the steps of the proposed methodology.

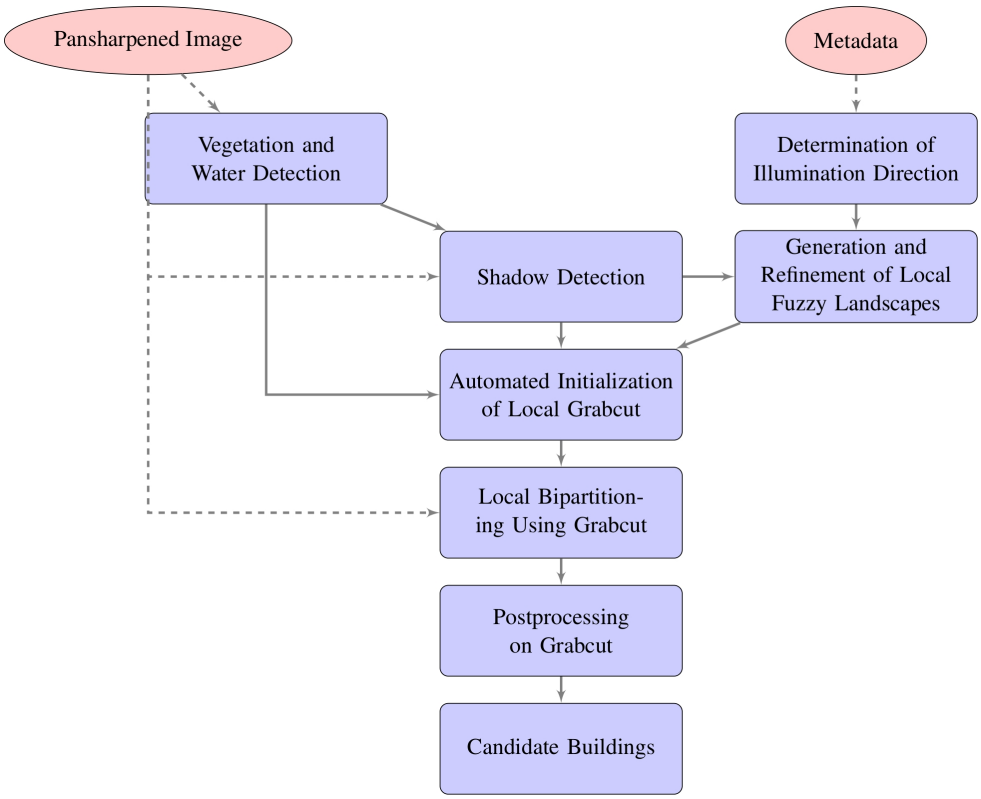


Figure 1.1: Flowchart of the proposed algorithm.

The proposed building detection approach locally extracts learning data from cues of the invariant; by automated selection of foreground and background seeds. This can be considered as the main contribution of the proposed methodology to remotely sensed data analysis field.

In the next chapter, previous works on building detection are mentioned. In Chapter 3, a general information on basic remote sensing concepts is given. Then, in Chapter 4, vegetation, water and shadow detection on satellite images is explained. In Chapter 5, fuzzy landscape generation algorithm is introduced. Afterwards, in Chapter 6, fundamental information on graph cuts is given, Grabcut algorithm is explained and the proposed building detection algorithm is discussed. Experiments and results are given in Chapter 7, and Chapter 8 concludes the thesis with discussing cons and pros of the proposed methodology.

CHAPTER 2

SURVEY ON BUILDING DETECTION

Automated building detection from monocular aerial or satellite images has been an open problem in remote sensing field since late 1980s. As mentioned in the survey articles [79; 119]; considering the initial studies for this problem, the images worked on contain only single panchromatic band, and have a much lower resolution. Therefore, the developed algorithms are limited to low-level vision and extraction of geometric primitives such as edges, lines and corners instead of applying pattern recognition techniques. Then, these primitives are grouped to form higher level features such as rectangle, parallelogram, or more generally, quadrilateral. These are considered as hypotheses for buildings and these hypotheses are passed from a verification process with respect to a set of user-defined geometric rules to determine final building polygons. Huertas and Nevatia [50]; Irvin and McKeown Jr [53]; Lin and Nevatia [76] can be considered as the pioneering building detection algorithms developed during this period.

Until now, many building detection studies have followed this approach: definition of the building with respect to its edges or contours, and rule (or hypothesis) based decision for whether the object is actually a building. Kim and Muller [65] first extract lines from the given panchromatic image, and then generate a line-relation graph based on geometric relations of lines. Building hypotheses are formed with respect to this graph. Yin et al. [131]; Saeedi and Zwick [100]; Song et al. [110]; Guducu and Halici [46] detect lines from the image, and then link or eliminate these lines in order to generate building hypotheses according to a set of geometric constraints (mainly parallellism) and rules dominated by user-defined parameters. For hypotheses, they apply building verification by considering its rectangular, gray, corner and shadow evidences. Peng et al. [89] initially select a set of seed pixels for buildings accord-

ing to user-defined constraints, and apply snake algorithm [61] with a new energy function to detect contours of candidate buildings. Then, they refine the building candidates with shape and shadow based post-processing. Cha et al. [25] develop a probabilistic Hough transform to detect lines in an image and utilize the algorithm on building detection by determining the highest peaks in the Hough transform. Izadi and Saeedi [56] first apply a a set of mean-shift segmentations [28] on the image with different range parameters, and determine the optimal range parameter by a set of geometric constraints related to building rooftop. Then, they obtain candidate rooftops by using a set of shape-based rules and thresholds over the segments, and verify rooftop candidates by shadow evidences. Pakizeh and Palhang [87] first detect candidate building centers by thresholding the image and applying mathematical morphology. Then, for verification, they explore rectangles around the centers by Hough transform.
Differential morphological profiles (DMP), which show morphological characteristics of connected components in an image by morphological reconstruction [90], are widely used in building detection. Jin and Davis [58] detect initial building candidates by using DMP and utilize hypothesis verification by using brightness, shape and shadow information. Sportouche et al. [111] first extract object boundaries using DMP, and then apply Hough transform for rectangular boundary refinement. Aytekin et al. [5] eliminate shadow and vegetation regions from the image and apply smoothing based on mean-shift. Then, they utilize a DMP-based segmentation to detect candidate buildings, and apply shape-based elimination on the building candidates.

Although these methodologies seem reasonable, there exist some issues that cannot be solved. Firstly, there exist many cases where building edges and lines cannot be evidently detected in satellite images. Even $0.5m$ spatial resolution may not be sufficient for a promising detection of lines. Reversely, lots of false edges and lines can possibly occur, generating many wrong hypotheses. A second problem is that the user-defined rules and constraints defined for building hypothesis are mostly threshold dependent, and cannot be generalized for all kind of test sites. Building detecion methodologies based on low-level vision only are not sufficient to detect buildings with arbitrary shapes and therefore, are not generalizable.

Instead of low-level vision base on line and contour detection, a set of studies apply supervised or unsupervised pattern recognition techniques after extracting color features from satellite image. Lee et al. [74] segment the image using ISODATA [7], extract a set of color features from segments and apply multi-class classification using ECHO [64]; whereas Knudsen and

Nielsen [68] segment the image using mean-shift, extract color and texture features from segments and apply multi-class classification using CART [20] to detect buildings. Unsalan and Boyer [120] compute Ω-map from the image, where human activity regions such as rock and stones respond high in Ω-image. Then, Ω-image is bipartitioned using k-means with spatial coherence to detect areas of human activity. Finally, human activity regions are decomposed into building/road by using mathematical morphology and constraints based on road/building characteristics. Bruzzone and Carlin [22] first apply a hierarchical segmentation on the image, and extract different set of features from each segmentation level. Finally, they classify whole segmented image using multi-class SVM [102]. Fauvel et al. [35] first apply a set of different multi-class fuzzy classification schemes independently on the image, and apply decision fusion on the results of fuzzy classifiers. Lari and Ebadi [73] first segment the image using region growing, extract a set of features from each segment and apply neural network-based classification to detect buildings. Sun et al. [112, 113] first apply hierarchical segmentation on the image using normalized cuts [104] and after feature extraction from segments, they apply classification using boosting classifiers [38; 117] to detect buildings. Aytekin et al. [6] segment the image using mean-shift and remove vegetation and shadow regions. Then, they detect the main road in the image using mathematical morphology and later eliminate thin long artifacts using PCA [59]. After eliminating tiny artifacts, the remaining segments are detected as buildings. Sirmacek and Unsalan [108] initially preprocess the image using bilateral filtering [116] and extract SIFT keypoints [77] from the filtered image. Then, they form a graph from the keypoints and apply a subgraph matching algorithm to detect urban area in the image. Finally, they apply graph-cut to detect separate buildings in urban area. Sirmacek and Unsalan [107] calculate Gabor responses [39] of the image over different orientations and determine local feature points from local maximum Gabor response pixels. Then they generate a voting matrix over the feature points considering spatial and spectral proximities of Gabor responses and detect buildings by thresholding over the voting matrix. Sirmacek and Unsalan [109] extract a set of local features (Harris-corner-based [48], GMSR-based [118], Gabor-based and FAST-based [96]) from the image and estimate the probability density function of the features independently by using kernel density estimator [105]. Then, they merge the pdf estimation results by fusion on data level and decision level separately. Yuksel et al. [132] first segment the image using mean shift, and remove vegetation and shadow regions as in [6]. Then, they extract features from the segments and apply fuzzy K-nearest neighbor [63] for binary classification for each feature space independently. Finally, they apply decision

fusion using stacked generalization architecture [130] to detect buildings.

These approaches suffer from generalization issue. Since buildings worldwide vary in terms of shape, color, texture and size, it is not possible to generate a probabilistic model for buildings. Therefore; using a learned or estimated model, considering the lack of generalization of the model, obviously only the buildings which are statistically similar to the model can be detected. Also, there exist invariant cues for buildings such as shadow attached to it and rectangular shape. Ignoring these cues and adhering only to classical pattern recognition techniques results in poor detection performance.

There are also studies combining the pattern recognition techniques and invariant cues mentioned above. Katartzis and Sahli [62] first extract a set of line and contour-based hypotheses, and generate a graph (called hypothesis graph) with hypothesis as nodes and features of hypotheses as edge weights. For hypothesis verification, they solve MRF optimization [66] on the hypothesis graph. Sirmacek and Unsalan [106] first detect buildings with red rooftops and shadows in the image by using invariant color features. Then, they estimate the illumination direction and detect missed buildings by using shadow and its direction. Finally, they apply box fitting considering the edge map, in order to determine building shapes accurately. Ren et al. [94] initially segment the image using multiscale normalized cuts and detect segments consisting of shadows. Then, they roughly detect buildings using initial knowledge of building-shadow configuration, and iteratively refine the building regions by searching segments adjacent to roughly-detected segments and considering chi-square distance between initial and refined building segments.

The proposed algorithm in this thesis is inspired by Akcay and Aksoy [1], where the image is initially oversegmented using watershed [80], and shadow regions in the image are detected. Then, directional fuzzy landscape representing the presence of building is generated for all shadow regions using sun azimuth angle which is assumed to be known. After thresholding the landscapes to obtain high-probability regions, for each landcape a graph is constructed over the remaining segments is constructed and the segments corresponding to building regions are determined by applying minimum spanning tree-based clustering over the graphs. The proposed methodology in this thesis follows a similar path except the nonexistence of segmentation, and detection of buildings by Grabcut, where a seeded binary graph-cut is utilized for detecting buildings.

CHAPTER 3

A BRIEF INTRODUCTION TO REMOTE SENSING

In this chapter, the fundamental remote sensing concepts applied to the system is described briefly. First, the main dissimilarities between the standart RGB images and satellite images are explained and tools for satellite image visualization are mentioned. Then, a specified image enhancement process for satellite images (pansharpening) is described. Finally, usage of metadata for satellite images and the calculation of illumination direction are explained.

3.1 Basics of a Satellite Image

Satellite images are the type of images that are taken by artificial satellites and contain portion of Earth. In that kind of imagery, image data is captured at specific frequencies across the electromagnetic spectrum [128]. Figure 3.1 shows a graph of the electromagnetic spectrum with respect to their wavelength.

In most cases, the spectral range of satellite images is broader than that of the visible light.

In the high resolution (HR) satellite images, the following spectral bands exist with corresponding wavelength intervals [128]:

- Blue: 450 - 515...520 nm
- Green: 515...520 - 590...600 nm
- Red: 600...630 - 680...690 nm
- Near-infrared (NIR): 750 - 900 nm

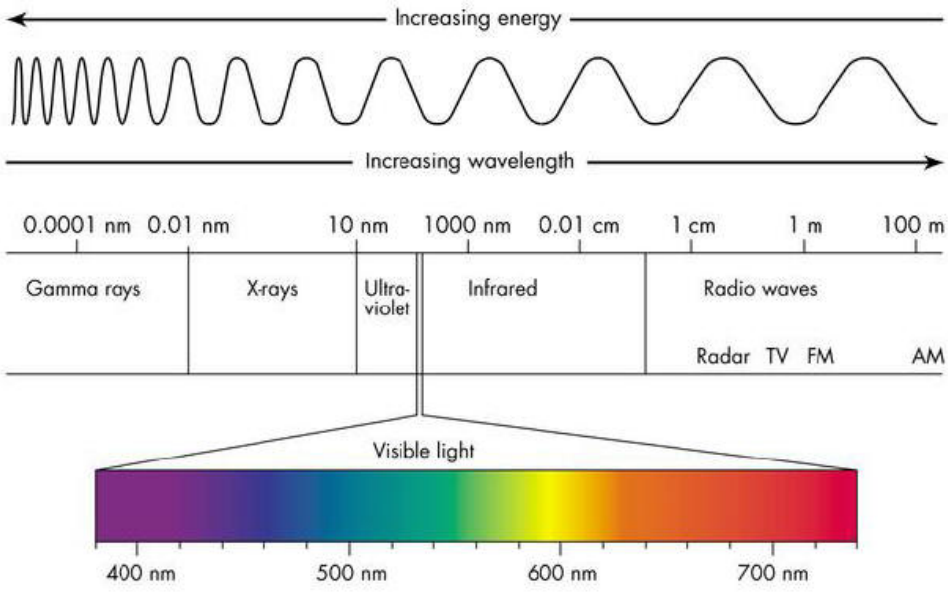


Figure 3.1: EM spectrum.

In the satellite images with lower geographical resolution; bands with longer wavelengths, such as mid-infrared, thermal infrared and radar also exist.

The existence of NIR band provides valuable information for detecting natural regions and shadows in the satellite image. The details of detecting these regions will be explained in the further chapters.

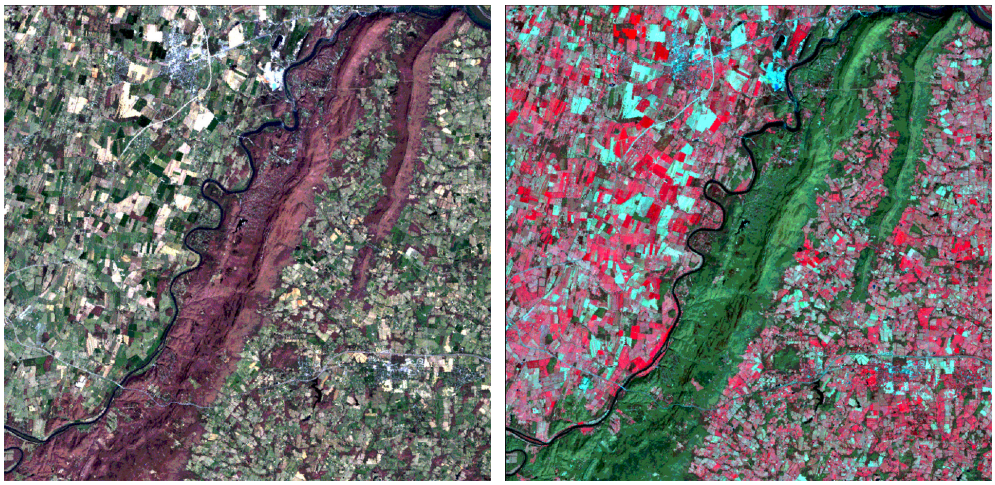
3.1.1 Displaying a Satellite Image

The wavelength intervals of color bands of a satellite image are different than that of an RGB image. Also, the spectral resolution of an RGB image is 8-bit (255 levels) by default; whereas the spectral resolution of a satellite image, in most cases, is different than 8-bit. Considering these discrepancies, the problem of clearly and sharply displaying a satellite image arises.

In order to overcome this, widely known Geographical Information System (GIS) toolboxes such as ERDAS Imagine [51], PCI Geomatica [42], ArcGIS [84] and Orfeo [52] use some image enhancement methods. These can vary from very simple image adjustment and contrast stretching operations to complicated enhancements using closed nonlinear transformations.

The satellite image can be visualized by using red, green, blue channels as its standard order (called true-color), as well as mapping NIR, red, green channels to red, green, blue. This is called false-color. This visualization is much useful for identifying vegetative regions.

Figure 3.2 shows an example satellite image in true-color and false-color respectively.



(a) (b)

Figure 3.2: A satellite image shown in two different visualizations. (a) True color visualization. (b) False color visualization.

3.1.2 Resolution

The spectral (radiometric) resolution of a satellite image; which is also the bit-depth of the sensor or number of gray levels; are most typically:

- 8-bit (0 - 255)
- 11-bit (0 - 2047)
- 12-bit (0 - 4095)
- 16-bit (0 - 65535)

Also, the spatial (geometric) resolution is defined as the pixel size of an image corresponding to the land region with a specific size. For instance, a spectral resolution of 4m means that a

single pixel is mapped with 4m x 4m region in the image. The spatial resolution is determined by instantaneous field of view (IFOV) of the sensor [124].

3.2 Pansharpening

For detecting a specific target such as building, road, vehicle, etc; it is essential to obtain a spatial resolution as high as possible. This can be done by a process called pansharpening.

Along with the multispectral image data, a single grayscale image, whose spectral resolution is higher than the multispectral image, is also acquired by the optical sensor. This grayscale image is called panchromatic image. The pansharpening process fuses the high resolution panchromatic image and low resolution multispectral image together to obtain a single high resolution multispectral image. There are several pansharpening algorithms in the literature with varying methodologies (using FFT [32], Bayesian inference [33], wavelet transformations [67], etc.).

Figure 3.3 shows an example image pansharpened using a commertial GIS tool called Geoimage:

(a) (b) (c)

Figure 3.3: Pansharpening process applied on a satellite image. (a) High-resolution panchromatic image. (b) Low-resolution multispectral image. (c) Resulting high-resolution pansharpened image.

3.3 Metadata

In addition to the image data, the high resolution satellite image products are obtained with a text metadata file. This file covers all relevant information about the satellite image; including geographic coordinates of the image corners, date and time the image had been acquisited, geometric resolution of the image file, the name and type of the optical satellite sensor, the altitude of the sensor and the sun azimuth / zenith angles at the image acquisition time. The direction of illumination, which is simply the opposite of the sun azimuth angle, provides valuable information for detecting potential building regions when combined with the detected shadows. These will be explained in further chapters.

Figure 3.4 shows an illustration of sun azimuth and zenith angles, and the direction of illumination.

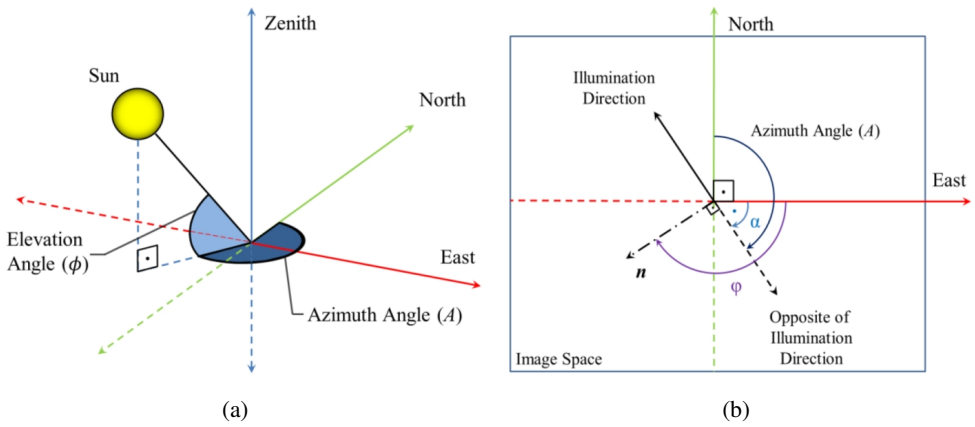


Figure 3.4: Illustration of solar angles in 3-D space and image space [83]. (a) Sun azimuth (A) and zenith (ϕ) angles. (b) Direction of illumination in image space.

CHAPTER 4

DETECTION OF VEGETATION, SHADOW AND WATER

In this chapter, the identification methodologies of shadows and natural regions (specifically vegetation areas and water components) are explained. These processes rely on the reflectance characteristics of the materials constituting the natural areas. The NIR band plays the most important role in detection of these areas. Simple thresholding schemes are applied on the color data to detect these regions properly.

4.1 Detection of Vegetation Areas

The pixels corresponding to the vegetation regions have very high reflectance values in the NIR band; while having low reflectance values in the red band. This is because of the emission / absorption characteristics of healthy vegetation.

Live green plants need a specific range of solar radiation; between 400 nm 700 nm, to carry on the photosynthesis process. This spectral range is called **photosynthetically active radiation** and abbreviated as **PAR**. In Section 3.1, the wavelength of NIR radiation had been mentioned to be longer than 700 nm. Therefore, the radiation with wavelength inside the NIR spectral region is scattered / emitted by the leaf cells; otherwise these rays would overheat and damage the leaves. Hence, healthy vegetation has high reflectance values in NIR spectral region. On the other hand, chlorophyl pigments in leaves absorb the light rays with wavelength equivalent to red, causing red reflectance to have low value [40; 34; 129].

For estimating the degree of vegetation in a given multispectral image, several leaf area indices have been proposed [34; 44]. Among these, the most widely used index is Normalized

Difference Vegetation Index (NDVI) [70]. The formula for calculating NDVI is simply:

$$\rho_{NDVI} = \frac{\rho_{NIR} - \rho_{RED}}{\rho_{NIR} + \rho_{RED}}. \tag{4.1}$$

Where ρ_{NIR} and ρ_{RED} represent the reflectance values for near-infrared and red bands respectively.

For every pixel, the ρ_{NIR} is calculated and an NDVI map is generated for whole image. The decision whether a pixel belongs to a vegetated area or not is made by simply applying Otsu's automatic thresholding method [85]. Figure 4.1 shows an example image of detected vegetation regions in a residential area:

(a) (b)

Figure 4.1: Vegetation detection example. (a) Sample image with large vegetation area. (b) Vegetation regions detected using *NDVI* thresholding.

4.1.1 Detection of Wet Soil

In most of the images, there exist vegetated areas without green vegetation, mostly consisting of wet soil. These areas can be detected by as follows:

1. Transform the false color (Nir-R-G) 8-bit image into opponent color space [121].

2. Convert the transformed image back to false color with utilizing a boosting parameter. b [82]

3. Calculate the boosted-$NDVI$ map as in Equation 4.1.

4. Threshold boosted-$NDVI$ map with the parameter T_{b-NDVI}.

Figure 4.2 shows an example of the detected wet soil regions in the image using this methodology.

(a) (b)

Figure 4.2: Wet soil detection example. (a) Sample image having wet soil regions. (b) Regions detected using boosted-$NDVI$ thresholding ($b = 20$, $T_{b-NDVI} = 0.5$).

4.2 Detection of Water

Another specified reflectance characteristics is that the low reflectance of water in NIR band. This is due to water's increasing absorption of radiation with increasing wavelength [29; 91; 14]. Figure 4.3 shows a water absorption spectrum curve for with respect to wavelength.

In order to detect water regions in the image, a robust histogram thresholding methodology is used. Firstly, a histogram is calculated over the normalized NIR band in order to determine the optimal threshold value. Mutiple histograms with different bins are calculated and binded so that the resulting histogram would not stick to local optima. Therefore, a smoother probability density function is obtained. Afterwards, in order to avoid false water detections in sample

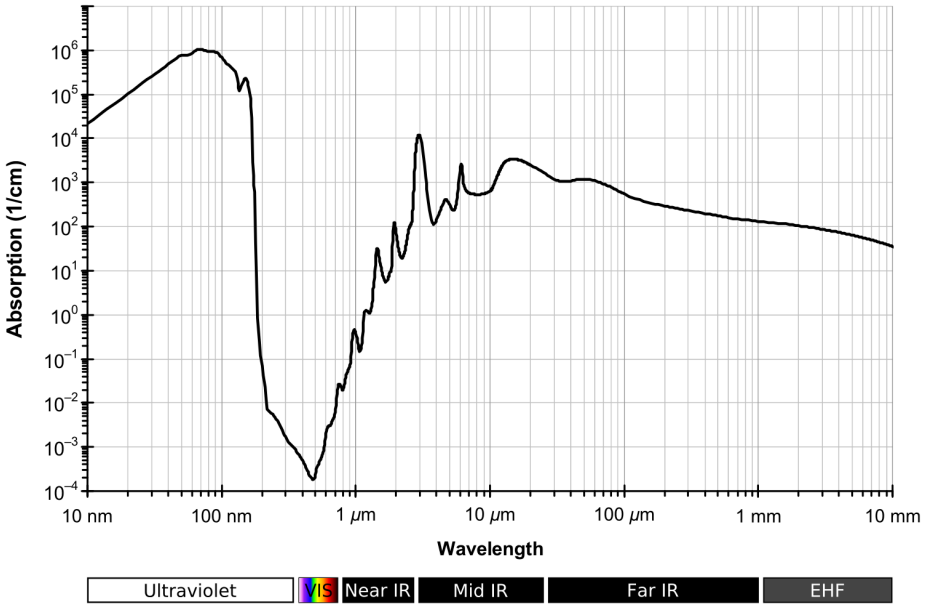


Figure 4.3: Liquid water absorption spectrum across wavelength range. Notice the increase in absorption level inside the near-infrared range.

images with no water, an upper level UL_{NIR} for water is assumed; and local maximum points *Lmax* are detected, considering the histogram bins in the range $[1, UL_{NIR}]$. Then, for each local maxima *Lmax*, the first next local minimum point *Lmin* is detected, and for each pair of local optimum ($Lmax_i$ and $Lmin_i$), a thresholding score $score_i$ is calculated:

$$score_i = Lmax_i / Lmin_i. \tag{4.2}$$

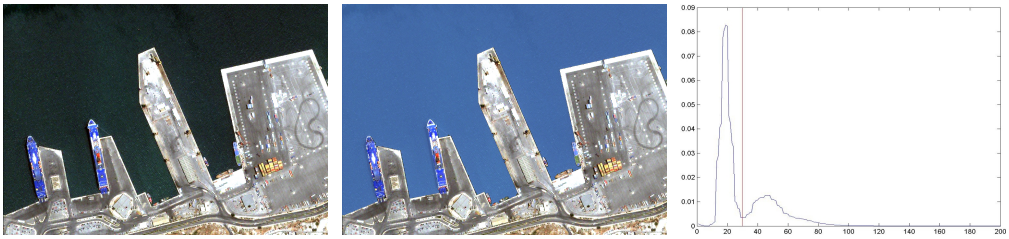
The local minimum point corresponding to the highest thresholding score index is determined to be the optimal threshold. Thus, the sharpest rise and fall is considered in the histogram curve.

In case the determined threshold value is higher than UL_{NIR}, it is assumed that there exists no water in the image. In the experiments, UL_{NIR} value is used as 0.2 for NIR band normalized between $[0, 1]$; and the combined histogram generated by binding 3 histograms with 50, 100 and 200 bins [8].

In this procedure, some shadow regions consisting of a few pixels are occasionally detected as water. In order to overcome this problem, first a morphological closing operation succeeded

by connected component analysis is applied on the water mask, and then blobs with area less than T_{water} are excluded from the map.

Figure 4.4 shows an example of detected water regions.



(a) (b) (c)

Figure 4.4: Water detection example. (a) Sample image with water regions. (b) Water regions detected using *NIR* histogram thresholding. (c) *NIR* histogram for the image and the determined threshold.

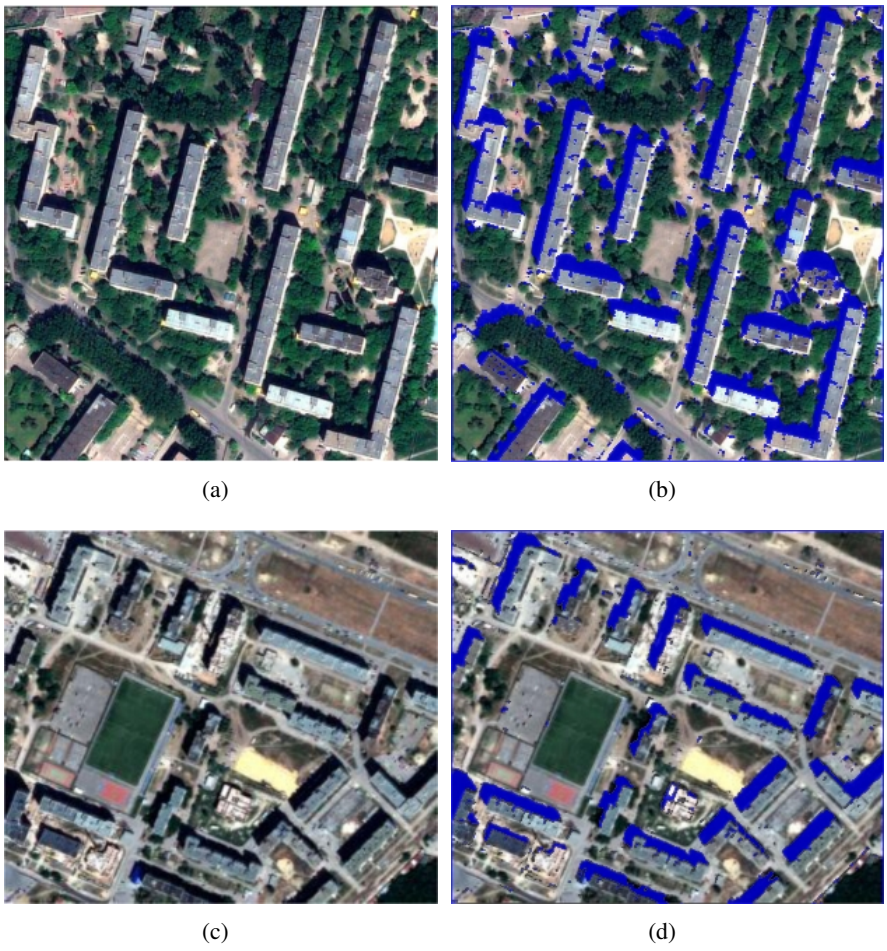
4.3 Detection of Shadows

Recently, Teke et al. [115] presented an original multi-spectral shadow detection approach that utilizes the advantage of the NIR image. The approach generates a false color image in which NIR, red and green bands are employed. The algorithm is simple; first, the false color image is normalized and converted to Hue-Saturation-Intensity (ρ_{HSI}) color space. Then, a ratio map (ρ_{RM}), in which the normalized saturation (ρ_S) and the normalized intensity (ρ_I) values are compared with a ratio, is generated:

$$\rho_{RM} = \frac{\rho_S - \rho_I}{\rho_S + \rho_I}. \tag{4.3}$$

To detect the shadow areas, as utilized in the case of vegetation extraction, Otsu's method is applied to the histogram of the ratio map, ρ_{RM}. Due to the fact that the thresholding scheme detects both shadow and vegetation regions at the same time, the regions that belong to the vegetation are subtracted to obtain a binary shadow mask. This approach provided successful shadow detection results for various satellite images and the major advantage is that it is independent from manual thresholds [115].

Figure 4.5 shows examples of detected shadow regions.



(a) (b)

(c) (d)

Figure 4.5: Shadow detection example. (a), (c) Sample images. (b), (d) Shadow detection results of (a), (c).

CHAPTER 5

GENERATION OF SHADOW PROBABILITY MAP

In this section, modeling of spatial arrangement between building-shadow pairs based on fuzzy relationing, using mathematical morphology, is explained. The key arguments used in this methodology are; the illumination direction determined in Section 3.3, and set of shadow blobs detected in Section 4.3.

5.1 Introduction

As described in Chapter 1, the shadow of a building can be described as an object invariant; since buildings in satellite images can have different colors, shapes, etc, but their common property is that building and its shadow always describe an object pair; i.e. there exists a shadow attached to a building.

After detecting shadows and determining the illumination direction in the image, the next step is to generate fuzzy landscapes for each shadow blob. The main purpose for landscape generation is to provide a set of local maps which will be utilized for determining the automated self-supervision data in the building detection phase described in Chapter 6.2. Also, with the help of the fuzzy landscapes, the potential non-building objects are eliminated by verification with respect to vegetation and shadow length, which are detailly explained in Section 5.5.

5.2 Fuzzy Directional Landscapes

Recently, a model describing directional spatial constraints has been proposed for contextual classification and retrieval problems [27; 26; 3] and succesfully applied in building detection problem [1; 2]. The model is as follows [27]:

Given a reference object O and a direction represented by an angle α, a landscape $\beta_\alpha(O)$ is generated. In $\beta_\alpha(O)$, each pixel x in the image is quantified with a relevance value. This relevance value is defined in terms of the angle between the vector from a point in O to x and α. For a given image pixel x in the image, the smallest such angle is computed by considering all points in O. The value of $\beta_\alpha(O)$ at an image point x is computed in terms of this smallest angle using a decreasing function $h : [0, \pi] \rightarrow [0, 1]$ as below [27]:

$$\beta_\alpha(O)(x) = h\left(\min_{p \in O} \theta_\alpha(x, p)\right). \tag{5.1}$$

where $\theta_\alpha(x, p)$ is the angle between vector $\overrightarrow{px}$ and the unit vector $\overrightarrow{u_\alpha}$, and is computed as [27]:

$$\theta_\alpha(x, p) = \begin{cases} \arccos\left(\frac{\overrightarrow{px} \cdot \overrightarrow{u_\alpha}}{\|\overrightarrow{px}\|}\right) & \text{if } x = p \\ 0 & \text{if } x \neq p. \end{cases} \tag{5.2}$$

In [13], the decreasing function described in Equation 5.1 is used as:

$$h(\theta) = \max\left(0, 1 - \frac{2\theta}{\pi}\right). \tag{5.3}$$

Then, Equation 5.1 is equivalent to the morphological dilation of the reference object O:

$$\beta_\alpha(O)(x) = (O \oplus \nu_\alpha)(x) \cap O^c. \tag{5.4}$$

where ν_α is a non-flat (fuzzy) structuring element:

$$\nu_\alpha = \max\left(0, 1 - \frac{2}{\pi}\theta_\alpha(x, o)\right), \tag{5.5}$$

o is the centroid of the structuring element and O^c is the fuzzy complement of O. The dilation is intersected with O^c since the reference object itself cannot have a relevance value.

However, as discussed in [2], using the linear function in 5.3 often results in a landscape with large spread and unpredictable transitions when the angle moves away from α direction at points distant from O. Therefore, instead of the structuring element in [13], Aksoy and Akçay [2] propose a fuzzy structuring element which they describe as more flexible and intuitive compared to [13], using a nonlinear function (shaped as Bezier curve [10]) corresponding to Equation 5.1:

$$\nu_{\alpha,\lambda,\tau}(x) = g_\lambda\left(\frac{2}{\pi}\theta_\alpha(x,o)\right)\max\left(0, 1 - \frac{\|\overrightarrow{ox}\|}{\tau}\right), \tag{5.6}$$

where g is a one-dimensional function having the shape of Bezier curve with inflection point $\lambda \in (0,1)$, $\|\overrightarrow{ox}\|$ is the distance (in Euclidean metric) between point x and the centroid o of the structuring element, and τ is a distance threshold at which a point loses its visibility from reference object O. The details of Bezier curve generation algorithm, which is actually root determination of third order polynomials with 2 variables, can be followed in [2].

Figure 5.1 shows an example reference object and the fuzzy directional landscape generated by [13] and [2].

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

Figure 5.1: Fuzzy directional landscape generation example. (a) A reference object O. (b) $\beta_\alpha(O)$ for $\alpha = 0$ using [13]. (c) $\nu_{\alpha,\lambda,\tau}$ for $\lambda = 0.3$ and $\tau = 200$. (d) $\beta_\alpha(O)$ for $\alpha = 0$ using [2] and structuring element in (c).

5.3 Further Improvements on Landscape

A major drawback of the fuzzy directional landscape generation methodologies described by Equations 5.4 and 5.6 is exposed in [83], that for the directions α that do not fully coincide with the centroid o of the structuring element $\left(\text{i.e. when } \alpha \notin \left\{\frac{\pi}{4}k \; ; \; k \in \mathbb{Z}\right\}\right)$, the pixels horizontally or vertically adjacent to o may have lower relevance values than some of the pixels far from o, as seen in Figure 5.2. The reason of this issue is as follows.

- Any pixel-based grid $\mathcal{G}$ can be described as a two-dimensional discrete coordinate system, where $\mathcal{G}(x, y)$ is defined if and only if $x, y \in \mathbb{Z}$.
- Any vector $\overrightarrow{v_\alpha}$ having angle α with x-axis and passing through the origin $\mathcal{G}(0, 0)$ can be represented with a line equation $x \cos\alpha + y \sin\alpha = 0$. Since $\mathcal{G}(x, y)$ is defined only when $x, y \in \mathbb{Z}$, the vector $\overrightarrow{v_\alpha}$ on discrete grid $\mathcal{G}$ is exactly represented only for angles α satisfying $\tan\alpha \in -1, 0, 1$ or $\cot\alpha \in -1, 0, 1$. This corresponds to directions $\overrightarrow{(i, j)}$ such that $i, j \in \{-1, 0, 1\}$, or $\alpha \in \left\{\frac{\pi}{4}k \; ; \; k \in \mathbb{Z}\right\}$ in terms of α.
- The structuring elements described by Equations 5.5 and 5.6 calculate fuzziness values for a pixel x with respect to angle differences $\theta_\alpha(x, o)$ between vector $\overrightarrow{ox}$ and the unit vector $\overrightarrow{u_\alpha}$. Therefore; obviously, for angles $\alpha \notin \left\{\frac{\pi}{4}k \; ; \; k \in \mathbb{Z}\right\}$, horizontally or vertically adjacent pixels to o yield larger angular values $\theta_\alpha(x, o)$ than the pixels far from o but having angles close to α. For instance; for $\alpha = 53\,^\circ$, $\mathcal{G}(-1, 0)$ or $\mathcal{G}(0, -1)$ have larger $\theta_\alpha(x, o)$ values compared to $\mathcal{G}(-4, 3)$, since $\overrightarrow{(-4, 3)}$ is directionally closer to $\overrightarrow{u_\alpha}$. For some selections of λ and τ in Equation 5.6, this would cause a lower fuzzy relevance values on the pixels horizontally or vertically adjacent to o compared to the pixels to some of the pixels further from o, but directionally closer to α.

To sum up; the landscape generation methodology described in [2] does not fully consider the discreteness of the lattice. Therefore, utilizing the angular parameter $\theta_\alpha(x, o)$, it is not possible to represent a vector $\overrightarrow{v_\alpha}$ corresponding to a pixel accurately. Instead of $\theta_\alpha(x, o)$, Bresenham's line discretization methodology [21] is proposed for the angular representation. This procedure will be explained in further paragraphs.

Another weak point is that, by using the nonlinear function h with the shape of Bezier curve, a third degree polynomial should be solved for every point in the region of interest [27]. This

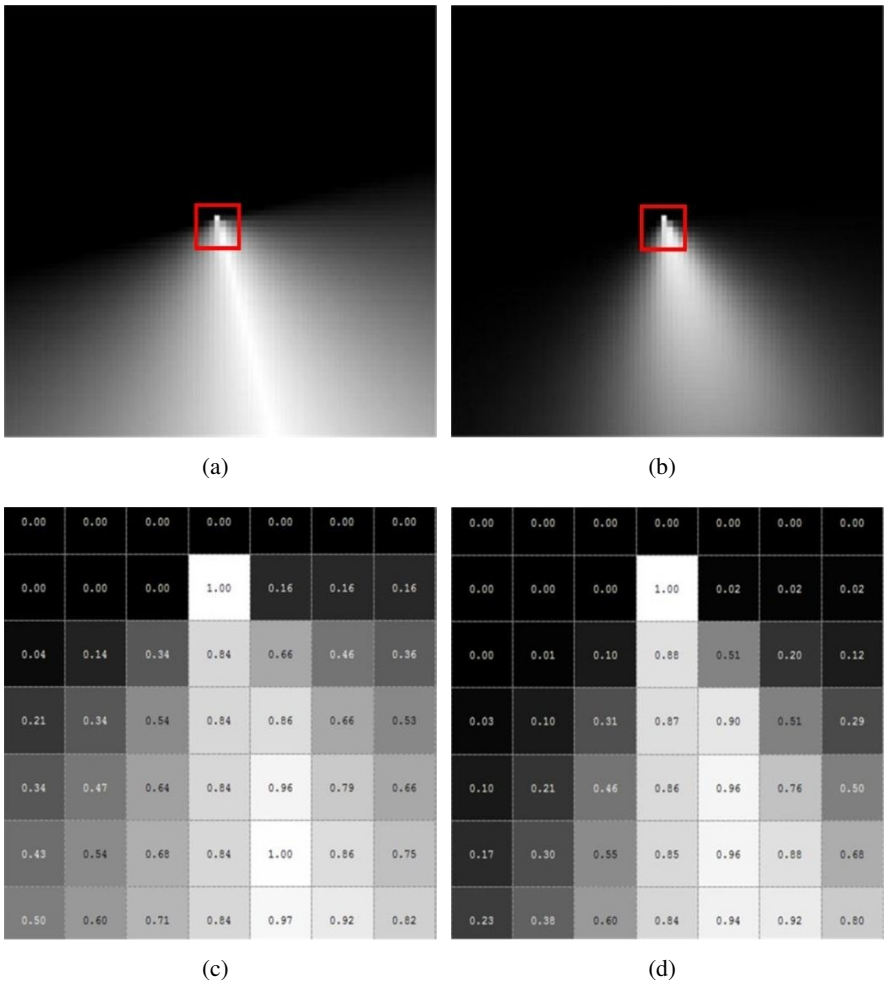


Figure 5.2: Structuring elements generated for $\alpha = 75.6°$. (a) Using the methodology described in [13]. (b) Using the methodology described in [27] (with $\lambda = 0.3$ and $\tau = 100$). (c) and (d) show the relevance values of the pixels inside the ROIs marked by red in (a) and (b) respectively. Notice the problem mentioned above in (c) and (d). [83]

would bring a heavy load on computational time of the algorithm. Finally, even though a small value of λ is used, generating the fuzzy landscape using Equations 5.4 and 5.6 would bring about a significant spread of the relevance values, which would cause the pixels not visible from the reference object in α direction to have nonzero values.

Considering all these deficiencies, a new methodology which is a combination of a fuzzy and a binary structuring element has been proposed in [83]. The fuzzy structuring element $\nu^{(r)}$ is calculated as an exponential function of distances between each image pixel x and centroid o of the structuring element:

$$\nu^{(r)}_{\sigma,\kappa}(x) = e^{\left(\frac{-\|\overrightarrow{ox}\|}{\sigma}\right)} \max\left(0, 1 - \frac{2\|\overrightarrow{ox}\|}{\kappa}\right), \tag{5.7}$$

where σ is the rate parameter of the exponential function and κ is the size of the structuring element. The exponential function is multiplied with the term $\max\left(0, 1 - \frac{2\|\overrightarrow{ox}\|}{\kappa}\right)$ in order to avoid nonzero values at the boundaries.

The line-based binary structuring element $\nu^{(\theta)}$ simply draws a line segment in α direction using Bresenham's line algorithm [21], and is formally defined as:

$$\nu^{(\theta)}_{\alpha,\kappa}(x) = \left\lfloor \left(1.5 - \frac{\theta_\alpha(x,o)}{\pi}\right) \right\rfloor D_\kappa\left(\mathcal{L}_\varphi\right), \tag{5.8}$$

where φ is the angle perpendicular to α, $\mathcal{L}_\varphi$ is a line passing through the origin of the kernel and is defined as

$$\mathcal{L}_\varphi = \left\{(x,y) \in \mathbb{R}^2 \mid x\cos\varphi + y\sin\varphi = 0\right\} \tag{5.9}$$

and D_κ is Bresenham's drawing algorithm procedure for line segment with length κ.

The structuring elements defined in equations 5.7 and 5.8 are combined to yield a single structuring element ν:

$$\nu_{\sigma,\kappa,\alpha} = \nu^{(r)}_{\sigma,\kappa}(x) * \nu^{(\theta)}_{\alpha,\kappa}(x) \tag{5.10}$$

where $*$ is the operator defining per-element multiplication.

The structuring elements generated using algorithm described in [83] are shown in Figure 5.3.

For a given reference object O and direction α, the fuzzy landscape $\beta_\alpha(O)$ is calculated by dilating the boundary of O with $\nu_{\sigma,\kappa,\alpha}$ (instead of dilating every pixel in O, dilating only the boundary is sufficient):

$$\beta_\alpha(O)(x) = (ß(O) \oplus \nu_{\sigma,\kappa,\alpha})(x) \cap O^c \tag{5.11}$$

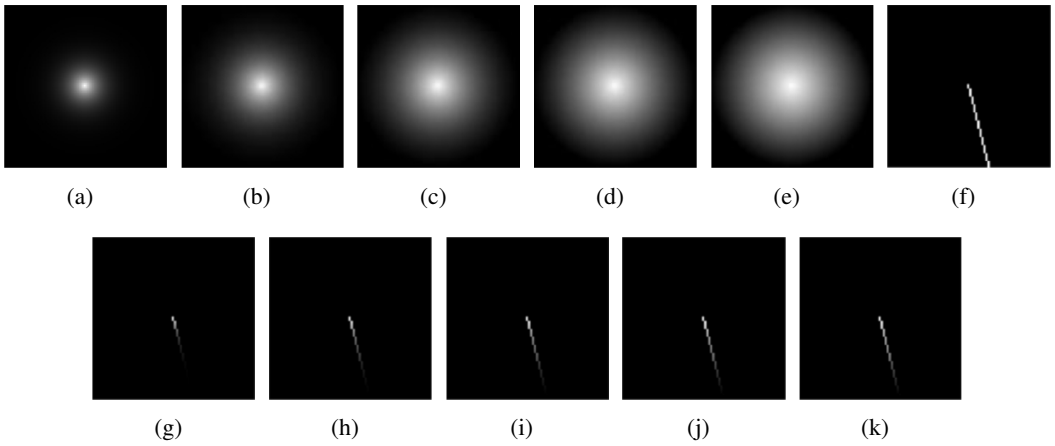


Figure 5.3: (a)-(e) Fuzzy structuring elements using exponentially decreasing function described in Equation 5.7 for $\kappa = 80$ and $\sigma = 10, 25, 50, 100, 250$ respectively. (f) The directional binary structuring element for $\alpha = 75.6$. (g)-(i) Resulting fuzzy structuring elements generated using [83].

where ß(O) is the boundary detection operator for O. ß(O) is calculated as

$$ß(O) = O \cap (O \ominus \epsilon_{3\times3})^c \tag{5.12}$$

where $\epsilon_{3\times3}$ is a 3×3 square structuring element.

5.3.1 Optimization

In [83], the structuring element is calculated in two steps, defined by Equations 5.7 and 5.8 and these steps are processed independently and merged after both structuring elements have been calculated. This methodology would bring a computational complexity of $O(n^2)$ with respect to a kernel size of $n \times n$. However, it is possible to decrease the complexity down to $O(n)$ by first applying Bresenham's line drawing algorithm to obtain the binary line-based structuring element $\nu_{\sigma,\kappa,\alpha}$, and then fuzzifying it by assigning the nonzero elements as in Equation 5.7.

Below is the proposed optimization routine:

1. **(Line drawing initialization)** Starting from structuring element's centroid o, decide which side to draw next pixel by applying quadrant test (i.e. *Quadrant I* if $\alpha = \pi/3$, *Quadrant III* if $\alpha = 5\pi/3$). The implementation is just a line of *if* statement.

2. **(Line drawing iterations)** Draw the next pixel x_i determined from Bresenham's algorithm until $\|\overrightarrow{ox}\|_\infty \leq \kappa$ where $\|\overrightarrow{ox}\|_\infty$ is maximum norm [99] of $\overrightarrow{ox}$ and κ is the length (or width) of the structuring element. Add x_i to the set of kernel points, defined as $\mathcal{P}$, to be utilized.

3. **(Fuzzification of the structuring element)** $\forall x \in \mathcal{P}$, calculate relevance value of x using Equation 5.7.

5.4 Shadow Probability Map for Buildings

After the fuzzy landscape proposed in Section 5.3 has been generated, it can be utilized in the equivalent probability map describing the building-shadow relation.

The relevance values in the landscape correspond to the probability that a pixel x belongs to a building region. Therefore, the pixels adjacent to the shadow boundary in the opposite direction of illumination $\alpha + \pi$ have high values, whereas the pixels further from the boundary in $\alpha + \pi$ direction have relatively smaller values. No matter how distant from the boundary, the pixels not in $\alpha + \pi$ direction have zero relevance values, since they are not visible by the shadow object.

Utilizing the results of shadow detection process and the direction of illumination α, a simple connected component analysis is applied to obtain the set of disconnected shadow blobs, $\mathcal{S}$. Then, for each shadow object $\varsigma_i \in \mathcal{S}$, the local fuzzy landscape $\beta_{\alpha'}(\varsigma_i)$, where $\alpha' = \alpha + \pi$ (since opposite of the illumination direction is considered) is generated independently. The composite landscape formed by union of all local landscapes for shadow blobs are defined as the shadow probability map.

The local fuzzy landscapes for each shadow blob ς_i are used as a cue for candidate building regions. In Chapter 6.2, it will be explained in details.

5.5 Refining the Probability Map

As discussed in [83], buildings are not the only objects to cast shadows in satellite images. All objects with prominent height such as trees, vehicles, bridges, etc. also bring about the

existence of shadows in the image. Since there is no relation between the size or shape of the shadow region and the reference object the shadow is cast by, the the probability map is pruned by checking whether the shadow region is adjacent to vegetation in α' direction in Section 5.5.1, and by exploiting height information in Section 5.5.2.

5.5.1 Eliminating Local Landscapes Due to Vegetation

Although vegetated regions have been detected in Chapter 4.1, there may be shadow regions which are cast by the vegetated areas. Since the local landscapes are generated independently for each shadow blob, the vegetated regions may also have relevance values in the final probability map.

In order to eliminate the local landscapes generated due to vegetation; for each shadow region ς_i, an evidence of vegetation is examined within its neighborhood $\mathcal{N}_i$. Formallly, this neighborhood is represented as:

$$\mathcal{N}_i = \left\{ x \mid T_{low} \leq \beta_{\alpha'}(\varsigma_i)(x) \leq T_{high} \right\}. \tag{5.13}$$

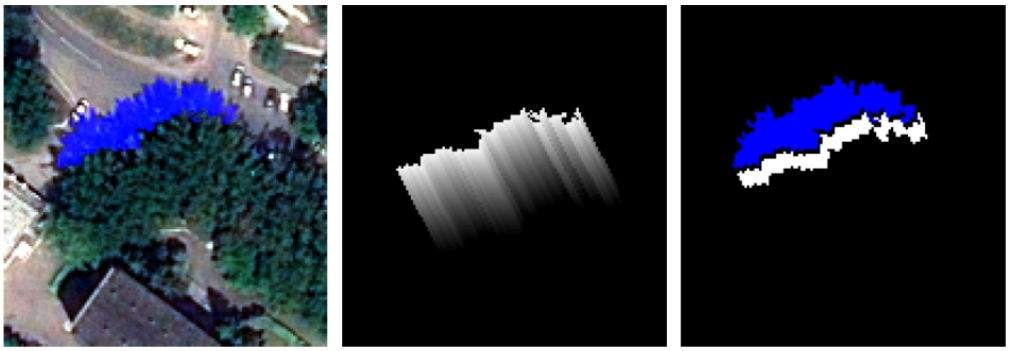
where T_{low} and T_{high} are the lower and upper bounds of the neighborhood interval. If the number of vegetated pixels in $\mathcal{N}_i$ is at least $T_{veg} \cdot |\mathcal{N}_i|$ where $|\mathcal{N}_i|$ represents the area of neighborhood $\mathcal{N}_i$, then its landscape $\beta_{\alpha'}(\varsigma_i)$ is completely eliminated [83].

Figure 5.4 shows a vegetation region casting shadow, and its landscape to be eliminated.

5.5.2 Pruning the Map by Using Height Information

As explained in 3.3, the sun zenith ϕ angle can also be calculated using the image metadata. Using ϕ and assuming that the shadows fall on a flat surface, it is possible to estimate the α-directional shadow length L cast by an object, given the height H of the object, using simple trigonometry [83]:

$$L = H / \tan \phi. \tag{5.14}$$



(a) (b) (c)

Figure 5.4: Elimination of a landscape. (a) Detected shadow of a vegetation region (blue). (b) Landscape to be eliminated, with $\alpha = 62.8^\circ$ and $\kappa = 80$. (c) The examined neighborhood region $\mathcal{N}$ (white). [83]

Assuming that the height of a building cannot exceed H_{min}, the minimum length L_{min} of a shadow can be estimated using 5.14:

$$L_{min} = H_{min} / \tan \phi. \tag{5.15}$$

After L_{min} has been calculated;

1. The binary line-based structuring element $\nu^{(\theta)}_{\alpha,\kappa}(x)$ is calculated as in 5.8, where $\kappa = L_{min}$.
2. For each shadow object ς_i;
   (a) For each boundary pixel $x \in$ ß(ς_i), $\beta_\alpha(\varsigma_i)(x)$ is calculated and the following verification is utilized:

   $$\text{if } \exists x \text{ such that } \beta_\alpha(\varsigma_i)(x) \cap \varsigma_i^c = \emptyset.$$

   In other words, it is examined whether $\beta_\alpha(\varsigma_i)(x)$ is completely within ß(ς_i), which is equivalent to the condition whether α-directional shadow length of ς_i is longer than or equal to L_{min}.
   (b) If the verification fails, ς_i is excluded from the probability map.

Figure 5.5 shows examples of generated shadow probability maps and pruning processes which eliminates most of the non-building regions in the map. As mentioned in Section 5.1,

the local landscapes provide an explicit interface for automated determination of self-supervision data (seeded regions) for the building methodology based on graph-based bipartitioning in Chapter 6.

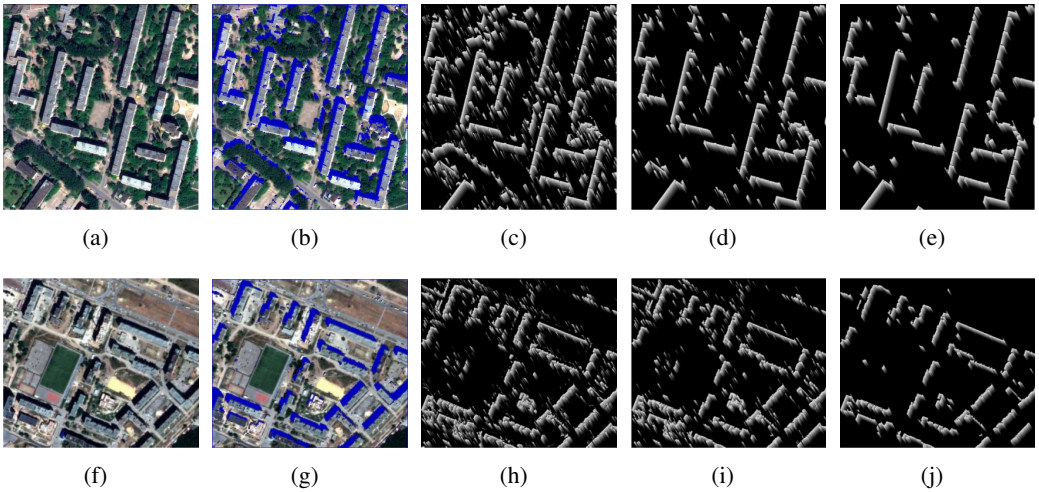


Figure 5.5: (a),(f) Sample images. (b),(g) Shadow detection results of images. (c),(h) Initially generated shadow probability maps using shadow masks in Figure 4.5(b),(d). (d),(i) Refined probability maps after vegetation-based elimination. (e),(j) Final probability maps after height-based elimination, where $H_{min} = 3m$. [83]

On a machine with Intel i5 2.6GHz CPU and 4 GB RAM, implementation in MATLAB, and over 20 images with sizes varying between 300x300 pixels to 1000x1300 pixels, the probability map generation with the proposed methodology takes just 3 seconds in total with an average of 0.15 seconds per image, whereas the it takes 19.66 and 19.92 seconds per image with the approaches described in [13] and [2] respectively. This corresponds to more than 130 times speed-up with the new landscape generation approach.

CHAPTER 6

PROPOSED BUILDING DETECTION SYSTEM USING GRABCUT PARTITIONING

In this chapter, the 2-way partitioning algorithm named Grabcut [97], which is the core part of the proposed algorithm, is explained. First, main concepts in probability theory will be mentioned and Markov random fields will be explained. Then, the utilization of graph cuts in computer vision will be described. Afterwards, the algorithm will be explained in details, starting from the previous methodologies that pave the way for Grabcut algorithm, to the settled algorithm procedure. Finally, the proposed building detection methodology will be explained.

6.1 Background

Definition 6.1.1 (Gaussian Mixture Models) *A Gaussian mixture model (GMM) is a weighted sum of K Gaussian densities, formulated as below [11; 95]:*

$$p(x) = \sum_{k=1}^{K} w_k N(x \mid \mu_k, \Sigma_k) \tag{6.1}$$

where x is the D-dimensional input data to be modeled, w_k is called mixture coefficient satisfying

$$\sum_{k=1}^{K} w_k = 1 \tag{6.2}$$

$$0 \leq w_k \leq 1. \tag{6.3}$$

μ is D-dimensional mean vector, Σ is $D \times D$ covariance matrix and $N(x \mid \mu_k, \Sigma_k)$ is called a

Gaussian mixture component that can be written in the form

$$N(x \mid \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{(D/2)}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} \tag{6.4}$$

where $|\Sigma|$ is the determinant of covariance matrix Σ.

Definition 6.1.2 (Markov Random Field) *Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a set of random variables $X = \{X_i \mid i \in \mathcal{V}\}$ forms a Markov random field with respect to $\mathcal{G}$ if they satisty following Markov properties [127]:*

- ***Local Markov property:*** *A variable is conditionally independent of all other variables given its neighbors:*

  $$X_i \perp\!\!\!\perp X_{\mathcal{V} \cap (\{i\} \cup \mathcal{N}_i)^c} \mid X_{\mathcal{N}_i}$$

  where $\mathcal{N}_i$ is the set of neighbors of vertex i.

- ***Global Markov property:*** *Any two subsets of variables are conditionally independent given a separating subset:*

  $$X_A \perp\!\!\!\perp X_B \mid X_S$$

  where every path between a vertex $i \in A$ and another vertex $j \in B$ passes through a vertex $\sigma \in S$.

- ***Pairwise Markov property:*** *Any two non-adjacent variables are conditionally independent given all other variables:*

  $$\forall \{i, j\} \notin E;\ X_i \perp\!\!\!\perp X_j \mid X_{\mathcal{V} \cap \{i,j\}^c}.$$

Definition 6.1.3 (Gibbs Random Field) *Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a set of random variables $\{X = X_i \mid i \in \mathcal{V}\}$ forms a Gibbs random field (GRF) with respect to $\mathcal{G}$ if and only if its configurations obey a Gibbs distribution, which is defined as [75]:*

$$P(x) = \frac{1}{Z} e^{-\frac{E(x)}{T}} \tag{6.5}$$

where Z is a normalizing constant calculated as

$$Z = \sum_{x \in X} e^{-\frac{E(x)}{T}} \tag{6.6}$$

T is a constant called temperature and $E(x)$ is the energy function which will be described in Definition 6.1.5:

$$U(x) = \sum_{c \in C} V_c(x). \tag{6.7}$$

Theorem 6.1.4 (Hammersley-Clifford Theorem) *Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a set of random variables $X = \{X_i \mid i \in \mathcal{V}\}$ forms an Markov random field with respect to $\mathcal{G}$ if and only if X forms a Gibbs random field with respect to $\mathcal{G}$ [47].*

Definition 6.1.5 (Clique) *A clique C [72] in an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a subset of $\mathcal{V}$, such that the subgraph induced by C is a complete graph (between all pair of vertices, there exists an edge) [125].*

In a Markov random field, the energy function is calculated as the sum of clique potentials over all possible cliques, using Equation 6.7. This can also be formulated as [75]:

$$E(x) = \sum_{i \in C_1} V_1(x_i) + \sum_{i,i' \in C_2} V_2(x_i, x_{i'}) + \sum_{i,i',i'' \in C_3} V_3(x_i, x_{i'}, x_{i''}) + ... \tag{6.8}$$

where C_i denotes the set of cliques having i vertices.

When only 2-vertex cliques are taken into account, the energy function in Equation 6.8 can be written as [75]:

$$E(x) = \sum_{i \in \mathcal{V}} V_1(x_i) + \sum_{i \in \mathcal{V}} \sum_{i' \in \mathcal{N}_i} V_2(x_i, x_{i'}) \tag{6.9}$$

where $\mathcal{N}_i$ is the neighborhood of vertex i. Here, the first term in summation is denoted as E_{data} and the second term is called as $E_{smoothness}$ [114].

The neighborhood relationship satisfies the following properties [75]:

1. *A vertex i is not neighbor of itself: $\forall i \in \mathcal{V}, i \notin \mathcal{N}_i$.*

2. *The neighborhood is mutual: $\forall i, j \in \mathcal{V}, i \in \mathcal{N}_{i'}$ if and only if $j \in \mathcal{N}_i$.*

6.1.1 Graph Cuts in Computer Vision

The graph cuts were first introduced in computer vision in [45], where exact maximum a posteriori (MAP) estimation for MRF models in a binary image restoration problem (MAP-MRF) was obtained by calculating the minimum cut (indeed, by its primal problem maximum flow) over the corresponding image network, introducing the terms *source* and *sink*. Until that time, the aproximate solutions using techniques such as simulated annealing [41] or iterated conditional models [9] had been proposed [126].

In graph cut applications, generally a pairwise Markov random field is used on a regular lattice, where each pixel corresponds to a node and each adjacent pixel pairs form an edge in the associated undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

In the computer vision literature, graph cuts have been used [126]:

- In one-shot [16; 18], where energy function is optimized over the segmentation explicitly
- Iteratively[97; 15], where a simple expectation-maximization algorithm is performed over intensity parameters at first, and the standart graph cut is applied in the second step.

6.1.1.1 Energy Function

For defining Gibbs distribution over the image field, Equation 6.9 is used as the energy function, which is defined as the summation of data term E_{data} and smoothness term $E_{smoothness}$.

The data term E_{data} is the unary term measuring how well a label f_p fits a particular pixel p. It is calculated as the summation of penalty function U over all pixels [114]:

$$E_{data}(f) = \sum_{p \in \mathcal{V}} U(f_p) \tag{6.10}$$

Initially, the image pixels are modeled (using histograms, mixture models, kernel methods, textons, etc.) and probability distribution $P(x \mid \omega_i)$ is obtained over each label. The penalty function U can be calculated using negative log-likelihood, $-\log(P(x \mid \omega_i))$.

The smoothness term $E_{smoothness}$ is the binary term describing the consistency between neighboring pixels. It is calculated as the summation of boundary penalty function P_{ij} over all neighboring pixels [114]:

$$E_{smoothness}(f) = \sum_{\{i,j\}\in\mathcal{E}} P_{ij}(f_i, f_j). \tag{6.11}$$

A well known boundary penalty function is Potts model [92] (named Ising model [54] for binary problems):

$$P_{ij}(f_i, f_j) = \kappa[f_i \neq f_j] \tag{6.12}$$

where the constant κ controls the degree of smoothness and the Iverson bracket $[f_i \neq f_j] = 1$ only when $f_i \neq f_j$.

Definition 6.1.6 (Submodularity) *A binary labeling problem is called submodular if the pairwise costs P_{ij} are in the form [75]:*

$$P_{ij}(1,0) + P_{ij}(0,1) - P_{ij}(0,0) - P_{ij}(1,1) \geq 0 \; \forall\{i,j\} \in \mathcal{E} \; and \; \forall f_i, f_j \in \{0,1\}. \tag{6.13}$$

For multi-label problems, submodularity is defined as [93]:

$\forall\{i,j\} \in \mathcal{E}$ *and* $\forall\alpha,\beta,\gamma,\delta$ *such that* $\beta > \alpha$ *and* $\delta > \gamma$ *where* $\alpha,\beta,\gamma,\delta$ *represent class labels,*

$$P_{ij}(\beta,\gamma) + P_{ij}(\alpha,\delta) - P_{ij}(\beta,\delta) - P_{ij}(\alpha,\gamma) \geq 0. \tag{6.14}$$

6.1.1.2 Exact Inference Using Graph Cut

Definition 6.1.7 (s-t cut) *An s-t cut is of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a partitioning (S, T) of the vertices $v \in \mathcal{V}$ into two subsets S and T such that $s \in S$ and $t \in T$ [88].*

Definition 6.1.8 (Graph-representability) *A function E of n binary variables is called graph-representable if there exists a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with terminal nodes s, t and a set of nodes $\mathcal{V}_0 = \{v_1, ..., v_n\} \subset \mathcal{S} - s, t$ such that given a configuration $(x_1, ..., x_n)$, $E(x_1, ..., x_n)$ is equal to a constant plus the cost of the minimum s-t cut with constraints: $\forall i \in \mathcal{Z} \mid 1 \leq i \leq n$, $v_i \in S$ if $x_i = 0$ and $v_i \in T$ if $x_i = 1$ [69].*

Theorem 6.1.9 *An energy function E of n binary variables that can be written as in Equation 6.9 is graph-representable if and only if its smoothness term $E_{smoothness}$ satisfies the submodularity criterion for all P_{ij} [69].*

Lemma 6.1.10 *Given an energy function E which is graph-representable by an s-t graph $\mathcal{G}$ and a set of nodes $\mathcal{V}_0$, the exact minimum of E can be found in polynomial time by calculating the minimum s-t cut on $\mathcal{G}$ [69].*

6.1.1.2.1 Construction of the s-t graph

Referring to Lemma 6.1.10; after the image has been represented using Markov random fields with a submodular energy function E, exact minimization of E (which corresponds to the maximum a posteriori solution) is possible in polynomial time by calculating the minimum s-t cut on the corresponding graph.

Given a Markov random field with binary variables, energy function E defined as in Equation 6.9 with zero boundary penalty for neighbors with same labels (e.g. $P_{ij}(0,0) = P_{ij}(1,1) = 0$) and symmetric penalty for neighbors with different labels (e.g. $P_{ij}(0,1) = P_{ij}(1,0)$), the corresponding s-t graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with terminal source and sink nodes s, t and MRF nodes $\mathcal{V}_0$ is constructed as follows [93]:

1. (Terminal weights) For each vertex $v \in \mathcal{V}_0$;

   (a) Draw a directed edge from source s to v with cost $U_v(0)$.

   (b) Draw a directed edge from v to sink t with cost $U_v(1)$.

   where $U_v(x)$ is the unary penalty function introduced in Equation 6.10.

2. (Nonterminal weights) For each pair of neighboring vertices i, j; draw directed edges from i to j and vice versa with cost $P_{ij}(0,1)$.

Figure 6.1 shows an graph $\mathcal{G}$ constructed from a binary MRF, and a possible minimum s-t cut calculated from C [93].

6.1.1.2.2 Max-flow / Min-cut

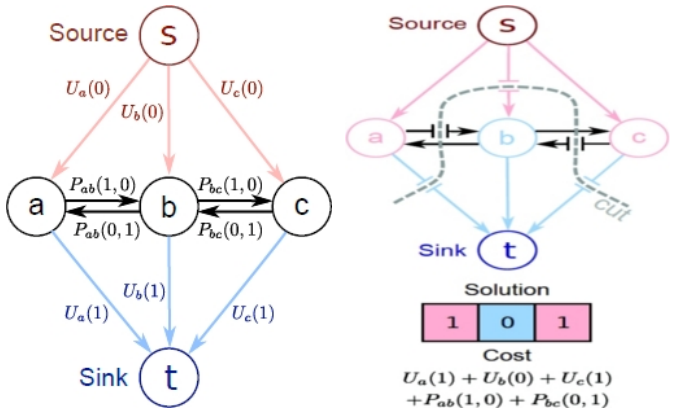


Figure 6.1: s-t graph construction. (a) Constructed graph with directed edge weights shown. (b) A possible minimum s-t cut for (b) with corresponding cost.

Theorem 6.1.11 (Max-flow Min-cut Theorem) *The maximum flow value in a graph is equal to the capacity of the minimum s-t cut with capacity [36; 60]*

$$C(S,T) = \sum_{i \in S, j \in T} w_{ij} \tag{6.15}$$

where w_{ij} is the cost of edge i, j.

Since minimum cut is the dual of maximum flow [88], the minimum s-t cut in a graph can be calculated using a maximum flow algorithm.

The procedure for computing the minimum s-t cut (S,T) in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is below:

1. Construct a residual graph $\mathcal{G}_f = (\mathcal{V}, \mathcal{E}')$ with edge costs w'_{ij} where $w'_{ij} = w_{ji}$.
2. Apply your favourite maximum flow algorithm (e.g. Ford-Fulkerson [36], Edmonds-Karp [31], push-relabel [43], Boykov-Kolmogorov [17], etc.) on $\mathcal{G}_f$.
3. Assign S as the set of vertices in $\mathcal{G}_f$ reachable from source s, and T as the remaining vertices. The set (S,T) is the minimum cut of the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

6.1.2 Grabcut

GrabCut [97] is a semi-automated foreground/background partitioning algorithm. Given a group of pixels interactively labeled as foreground/background by the user, it partitions the

rest of the pixels in an image using a graph-based approach described in Section 6.1.1.

In this section, firstly the founding work on which Grabcut is built [18] will be explained. Then, the Grabcut algorithm will be described in details.

6.1.2.1 Segmentation by Graph Cuts

In [18], a gray-level image is segmented with respect to an initial user defined map (trimap) $\mathcal{T} = \mathcal{T}_F \cup \mathcal{T}_B \cup \mathcal{T}_U$, which contains pixels labeled as foreground in $\mathcal{T}_F$, background in $\mathcal{T}_B$ and unlabeled pixels $\mathcal{T}_U$. A binary MRF is formed over the image pixels, and the energy function $E(x)$ captured by Gibbs distribution is calculated as in Equation 6.9, as the sum of E_{data} (in Equation 6.10) and $E_{smoothness}$ (in Equation 6.11).

Let us denote the foreground and background labels as $\mathcal{F}$ and $\mathcal{B}$ respectively. Using intensity values $\bar{I}_p$ for pixels $p \in \mathcal{T}_F$ and $\bar{I}'_p$ for pixels $p' \in \mathcal{T}_B$, gray level histograms H_F and H_B are computed separately; and unary penalty $U_p(\alpha)$ for a pixel p, where $\alpha \in \{\mathcal{F}, \mathcal{B}\}$ is calculated as log-likelihood of intensity distributions (obtained from H_F and H_B) of foreground and background respectively:

$$U_p(\alpha) = -logP(\bar{I}_p \mid \alpha).$$

The boundary penalty over an edge i, j is calculated as:

$$P_{ij}(\alpha,\beta) = \kappa_1[\alpha \neq \beta] \times \frac{e^{-\kappa_2(I_i - I_j)^2}}{\|\vec{i} - \vec{j}\|}$$

where the Iverson bracket $[\alpha \neq \beta] = 1$ only when $\alpha \neq \beta$; $\|\vec{i} - \vec{j}\|$ is the Euclidean distance between pixels i and j. If $\kappa_2 = 0$, the boundary is equivalent to that of Ising model. In [18], κ_2 is selected to be ≥ 0 to increase penalties where intensity difference is small.

Since $P_{ij}(0,0) = P_{ij}(1,1) = 0$, the energy function $E(x)$ is submodular. Therefore, the global minimum of $E(x)$ can be computed using graph cuts. After the energy function has been defined, the corresponding s-t graph is calculated similar to Section 6.1.1.2.1, with additional terminal weights:

1. For each vertex v corresponding to a pixel $p \in \mathcal{T}_F$;

   (a) Draw a directed edge from source s to v with cost λ.

   (b) Draw a directed edge from v to sink t with cost 0.

2. For each vertex v corresponding to a pixel $p \in \mathcal{T}_B$;

   (a) Draw a directed edge from source s to v with cost 0.

   (b) Draw a directed edge from v to sink t with cost λ.

where

$$\lambda = \max_{p \in \mathcal{V}} \sum_{q:p,q \in \mathcal{E}} \kappa_1 \frac{e^{-\kappa_2 (I_i - I_j)^2}}{\|\vec{p} - \vec{q}\|}.$$

Finally, the inference step is applied as explained in Section 6.1.1.2.2, using Boykov-Kolmogorov maximum flow algorithm [17].

6.1.2.2 Grabcut Algorithm

The procedure applied in Grabcut [97] is very similar to [18]. The minor differences are:

1. The unary penalties are calculated by using Gaussian mixture models for pixel RGB values instead of histograms for gray-level intensities.

2. The energy minimization routine by using graph cuts is applied iteratively.

Using RGB vectors $\bar{I}_p$ for pixels p labeled as α, the background (where $\alpha = \mathcal{B}$) and foreground (where $\alpha = \mathcal{F}$) are modeled using Gaussian mixture models. In order to obtain the mixture parameters w_k, μ and Σ, an initial assignment of pixels p labeled as $\alpha = \mathcal{F}$ or $\alpha = \mathcal{B}$ to corresponding mixture components is utilized by expectation-maximization [30] or k-means [78] algorithm. Afterwards, the parameters for each component k are calculated as:

$$w_k^{(\alpha)} = |GMM_k(\alpha)| / \sum_k |GMM_k(\alpha)| \tag{6.16}$$

$$\mu_k^{(\alpha)} = \sum_{p \in GMM_k(\alpha)} \bar{I}_p / |GMM_k(\alpha)| \tag{6.17}$$

$$\Sigma_k^{(\alpha)} = \frac{1}{|GMM_k(\alpha)|} \sum_{p \in GMM_k(\alpha)} (\bar{I}_p - \mu_k^{(\alpha)})(\bar{I}_p - \mu_k^{(\alpha)})^T. \tag{6.18}$$

where $\alpha \in \{\mathcal{F}, \mathcal{B}\}$, $GMM_k(\alpha)$ is the set of pixels assigned to k-th component of the GMM of α label and $|GMM_k(\alpha)|$ denotes the size of set $GMM_k(\alpha)$.

The unary penalty function $U_p(\alpha)$ for a pixel p, where $\alpha \in \{\mathcal{F}, \mathcal{B}\}$, is calculated as negative log-likelihood of Gaussian mixture models [97]:

$$U_p(\alpha) = \sum_{k=1}^{K} U'_p(\alpha, k) \tag{6.19}$$

where

$$U'_p(\alpha, k) = -log w_k^{(\alpha)} + \frac{1}{2} log|\Sigma_k^{(\alpha)}| + \frac{1}{2}\Big((\bar{I}_p - \mu_k^{(\alpha)})^T (\Sigma_k^{(\alpha)})^{-1} (\bar{I}_p - \mu_k^{(\alpha)})\Big). \tag{6.20}$$

The boundary penalty $P_{ij}(\alpha, \beta)$ on edge i, j is calculated similar to Equation 6.16:

$$P_{ij}(\alpha, \beta) = \kappa_1 [\alpha \neq \beta] \times \frac{e^{-\kappa_2 ||\bar{I}_i - \bar{I}_j||^2}}{||\vec{i} - \vec{j}||}$$

In [97], the number of mixture components K for foreground and background is defined as 5. κ_1 in Equation 6.16 is defined as 50, κ_2 is calculated as

$$\kappa_2 = \frac{|\mathcal{E}|}{\sum_{\{i,j\} \in \mathcal{E}} 2||\bar{I}_i - \bar{I}_j||^2} \tag{6.21}$$

where $|\mathcal{E}|$ is the number of edges in the generated s-t graph, and λ in Section 6.1.2.1 (additional terminal weights) is defined as 9γ.

Below is the Grabcut algorithm step by step [98]:

1. **(Initialization)**

   - **With bounding box:** Set $\mathcal{T}_F$: $\emptyset$, $\mathcal{T}_B$: outside the bounding box, $\mathcal{T}_U$: the bounding box and its interior region
   - **With explicit trimap:** pixels $p \in \mathcal{T}_F, \mathcal{T}_B, \mathcal{T}_U$ are given explicitly with a mask

   with labels $\alpha_p = \mathcal{F}$ if $p \in \mathcal{T}_F \cup \mathcal{T}_U$; $\alpha_p = \mathcal{O}$ if $p \in \mathcal{T}_B$.

2. **(Mixture component assignment)** Assign each pixel p to a component for foreground or background GMM by an initial EM or K-means algorithm.

3. **(Iteration)** Until convergence of GMMs;

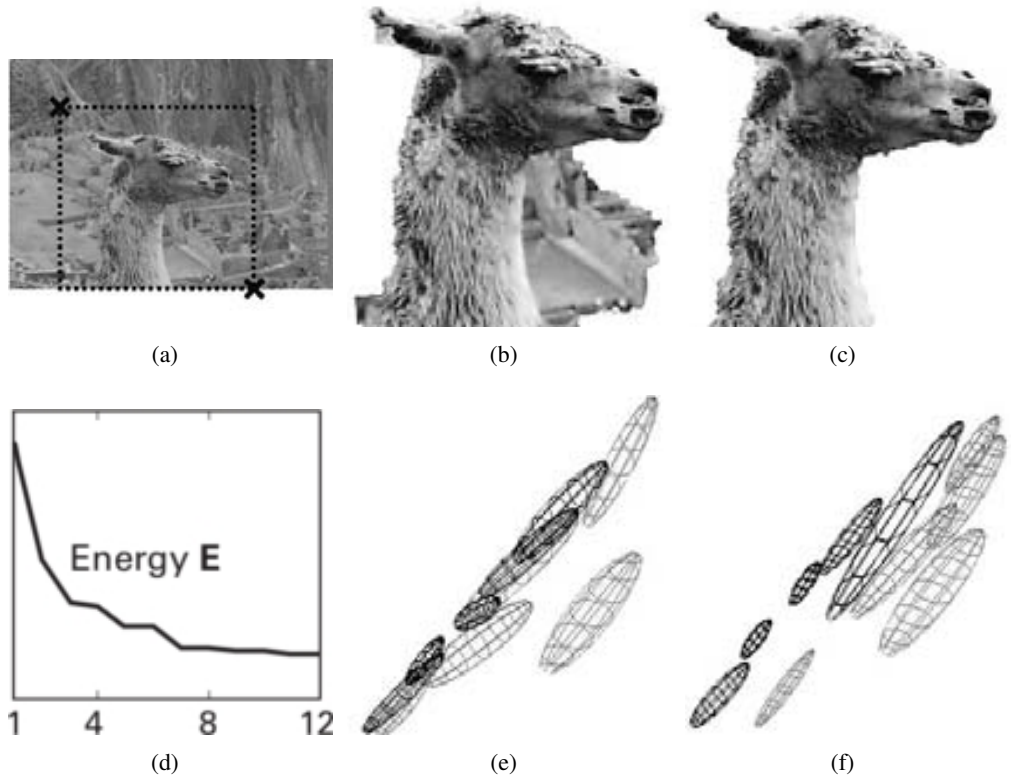
   (a) **(GMM parameter computation)** Calculate mixture parameters for GMMs of $\mathcal{F}$ and $\mathcal{B}$.

   (b) **(Inference by Graph cut bipartitioning)** After defining energy function and unary / boundary penalties, solve the bipartitioning problem over the corresponding s-t graph for the field, generated same as in Section 6.1.2.1; using maximum flow.

   (c) **(Reassignment of mixture components)** Update the GMM component assignments to samples for background and foreground, with respect to the obtained result in inference step.

The complexity of Grabcut algorithm is proportional to complexity of the maximum flow algorithm and number of iterations. For an s-t graph $\mathcal{G} = \mathcal{V}, \mathcal{E}$, the complexity of Grabcut is $O(imn^2|C|)$, where i is the number of iterations, $m = |\mathcal{E}|$, $n = |\mathcal{V}|$ and $|C|$ is cost of the minimum s-t cut [17].

Since the defined energy function $E(x)$ is submodular (as $P_{ij}(0,0) = P_{ij}(1,1) = 0$); at each iteration, the bipartitioning exactly minimizes $E(x)$. Also, for a given initial labeling, GMM estimation converges at least to a local minimum. Therefore, the $E(x)$ is guaranteed to converge. Figure 6.2 shows the minimization of $E(x)$ with respect to iterations [98].



(a) (b) (c)

(d) (e) (f)

Figure 6.2: (a) Grabcut initialization with bounding box, where pixels outside the box are in $\mathcal{T}_B$. (b) Bipartitioning result after first iteration. (c) Bipartitioning result after 12 iterations. (d) Energy minimization with respect to iterations. (e) Background and foreground GMMs (with 5 components, shown in red-green space) at first iteration. Notice the high amount of overlap between GMMs of different labels. (f) Background and foreground GMMs (shown in red-green space) at last iteration, where the models are separated better. [98]

6.2 Automated Building Detection Methodology Using Grabcut

After the non-building regions (vegetation, water, shadow) have been detected in Chapter 4 and the probability map representing the proximity of pixels to the shadow regions in the opposite of illumination direction has been generated in Chapter 5, the remaining problem is to combine these informations in a seeded probabilistic framework to boost the building detection performance. In this work, Grabcut [97] is determined to be a feasible algorithm in which initial foreground and background information can be integrated easily, interaction potentials between pixels with respect to probabilty map can be computed feasibly and the fast, exact bipartitioning can be utilized in polynomial time.

The steps of the proposed building detection algorithm is listed below:

- **(Iteration over shadow components)** For each shadow blob ς_i detected in Section 4.3;

    1. **(Initialization: Local bounding box and trimap generation)** Generate a bounding box BB_i, which defines the boundary of the local image patch I' to be worked on, using the fuzzy landscape generated in Section 5.4. Simultaneously; for each pixel $p \in I'$, determine which pixels are in $\mathcal{T}_F$ and $\mathcal{T}_B$, and define initial value α_p for each pixel p inside BB_i.
    2. **(Assignments to mixture components)** For each pixel $p \in I'$, assign p to a GMM component for either background or foreground (equivalent to the one explained in 6.1.2.2, but with 4-bands).
    3. **(Iterations over energy minimization)** (Same as explained in 6.1.2.2, except that the GMM distributions and parameters are computed for 4-dimensional data, corresponding to 4-band image.)
    4. **(Refinement)** Simple post-processing operations are applied on the partitioning result, using connected component analysis.

- **Unification** Finally, the detected local results are superimposed on the original image by taking the union of all local detection results.

The proposed algorithm runs independent, local 'Grabcut's over each shadow component. This is because of the fact that the most useful information about a building object is obviously obtained within a neighborhood of the object itself. In a satellite image, the assumption that two buildings far from each other share similar color properties can lead to poor results. For instance; there may be a red building covered with a blue environment in a part of an image, whereas a blue building covered with a red environment on another part of the image. Initially modeling parts of both buildings as foreground, and parts of the environment as foreground leads to inconsistency. Dividing the big global problem into many local subproblems does not only improve the detection performance, but also decreases the computation time. Since images with more than 10^6 pixels are utilized, dealing with patches with pixel size 10^4 drastically decreases the node and edge count in the graphical model, and therefore the computation time. As a result; instead of dealing with the problems arising due to globality;

locality, which provides sufficient information for the detection problem, is the one focused on in this approach.

The further subsections explain the steps of the local detection problem.

6.2.1 Initialization: Creation of the Bounding Box and Automatic Generation of Trimap

6.2.1.1 Determining the Foreground Pixels

Considering the local fuzzy landscape $\beta_\alpha(\varsigma_i)$ generated for a shadow component ς_i in Section 5.4, the initial set $\mathcal{T}_F$ of foreground pixels is defined as the ones having relevance values between η_{low} and η_{high}:

$$\mathcal{T}_F = \{p \mid \eta_{low} \leq \beta_\alpha(\varsigma_i)(p) \leq \eta_{high}\}. \tag{6.22}$$

Where η_{low} and η_{high} are the limits of relevance values. Similar to the lower-upper thresholding in 5.5.1, the upper limit η_{high} is defined to be less than 1, since the pixels just adjacent to shadow may not be reliable due to mixed pixels phenomenon [83]. In this step, $\mathcal{T}_F$ may yield a set of multiple separate regions $\mathcal{R}_{ij}$. For these cases, the succeeding steps are applied separately for each $\mathcal{R}_{ij}$ such that $|\mathcal{R}_{ij}| \geq T_{map}$, where $|\mathcal{R}_{ij}|$ denotes the number of pixels in $\mathcal{R}_{ij}$ and T_{map} is a parameter for eliminating tiny artifact-like sub-landscapes.

The initial set of foreground pixels $\mathcal{T}_F$ may contain vegetation and water pixels, and shadow pixels from other objects. Since these are certain non-building pixels, they are removed from $\mathcal{T}_F$. After the removal, tiny artifacts may still remain, and these can be eliminated from $\mathcal{T}_F$ by simple morphological operations or size filtering after connected component analysis.

In [83], another problem for forming the foreground region is discovered: The pixels in the intersecting boundary of shadow and building may also be in $\mathcal{T}_F$. To overcome this problem; first, the one-pixel boundary between shadow blob ς_i and $\mathcal{T}_F$ is detected and this boundary is shrunk from both ends by a shrinking distance d_{shrink}. Afterwards, the cropped boundaries are dilated on both sides, using a line-based structuring element, in the direction of illumination. Finally, the cropped parts are removed from $\mathcal{T}_F$, and the final set of foreground pixels $\mathcal{T}_F$ is obtained [83]. The determination of foreground pixels in a single building detection step is

shown in Figure 6.4.

6.2.1.2 Forming the Bounding Box

To determine the image patch that to which local Grabcut is applied; the bounding box BB_i defining the boundary of the patch is created by considering the set of foreground pixels $\mathcal{T}_F$ determined in Section 6.2.1.1. In order to determine BB_i, a dilation is performed on the region including pixels in $\mathcal{T}_F$, using a line-based binary structuring element $\nu'^{(\theta)}$:

$$\nu'^{(\theta)}_{\alpha,\kappa}(x) = \nu^{(\theta)}_{\alpha,\kappa}(x) + \nu^{(\theta)}_{\alpha+\pi,\kappa}(x) \tag{6.23}$$

where $\nu^{(\theta)}$ is the structuring element defined in Chapter 5. Since the angle parameters of the summed structuring elements $\nu^{(\theta)}$ are directed 180 ° opposite to each other, the resulting structuring element $\nu'^{(\theta)}$ is symmetric with respect to its centroid.

Determining the bounding box BB_i is as follows:

1. **(Dilation)** The initial foreground region denoted as $\mathcal{T}'_F$ and defined as $\{p \mid \eta_{low} \leq \beta_\alpha(\varsigma_i)(p) \leq \eta_{high}\}$ is dilated by $\nu'^{(\theta)}$, with length parameter κ controlled by the parameter d_{Bbox}.

2. **(Cropping the dilation result)** Since $\mathcal{T}'_F$ is dilated on both sides, the dilation result $\beta(\mathcal{T}_F)$ intersects with the shadow component ς_i from which it had been generated, using a relatively large value for d_{Bbox}. For this case; since half section of $\beta(\mathcal{T}_F)$ which on the side of the shadow exceeds the shadow region and lies until regions too far from T_F, it is cut by the shadow component ς_i and the component having intersection with $\mathcal{T}_F$ is taken into account, and denoted as $\beta_{cropped}(\mathcal{T}_F)$.

3. **(Joining with the part of shadow component)** The intersecting pixels IP of the initial dilation result and the shadow component; $IP = \beta(\mathcal{T}_F) \cap \varsigma_i$ is joined with $\beta_{cropped}(\mathcal{T}_F)$. Then, BB_i is formed by computing the bounding box of the resulting region $\mathcal{D}_i$.

Actually, this step could also have been implemented by just dilating the corresponding shadow component ς_i. However, there exist such cases that the shadows of many buildings

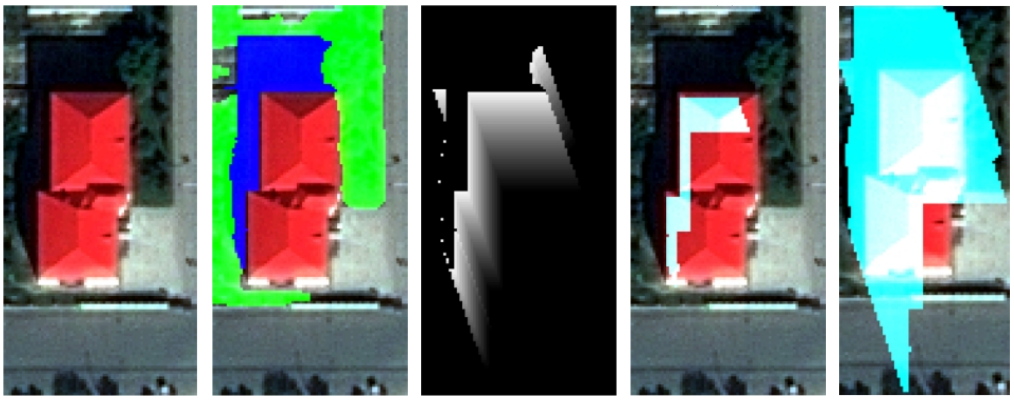
are connected, and several separate sets of foreground pixels $T_F^{(j)}$ are generated. An example of this situation is shown in Figure 6.3, where buildings are too close to each other and their shadows are connected in a web-like structure. In this cases, dilating ς_i would result in a huge bounding box, considering shadows of many unneccessary foreground regions. This would both disrupt the priority of local information described at the beginning of the section, and also bring a large computational load due to large local patch size.

(a) (b)

Figure 6.3: An example of the web-like shadow mentioned in Section 6.2.1.2. (a) Sample image with large shadow areas. (b) Shadow detection result.

6.2.2 Determining Background and Remaining Pixels

After the set of foreground pixels $\mathcal{T}_F$ have been determined and the bounding box BB_i has been generated, the pixels detected as shadow and vegetation which are inside BB_i are determined to be in T_B. Moreover, the remaining unlabeled pixels p with nonzero values in the map $\mathcal{D}_i(p) > 0$ calculated the bounding box determination in Section 6.2.1.2 are defined to be in $\mathcal{T}_U$, and pixels p with zero value $\mathcal{D}_i(p) = 0$ are also determined to be in $\mathcal{T}_B$.



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

Figure 6.4: Determining $\mathcal{T}_F$ for a sample image patch. (a) An image patch with red building. (b) Detected vegetation and shadow regions, represented by green and blue colors respectively. (c) Local fuzzy landscape generated from the shadow object. (d) Selected foreground pixels T_F, represented by cyan color. (e) The selected bounding box with respect to T_F.

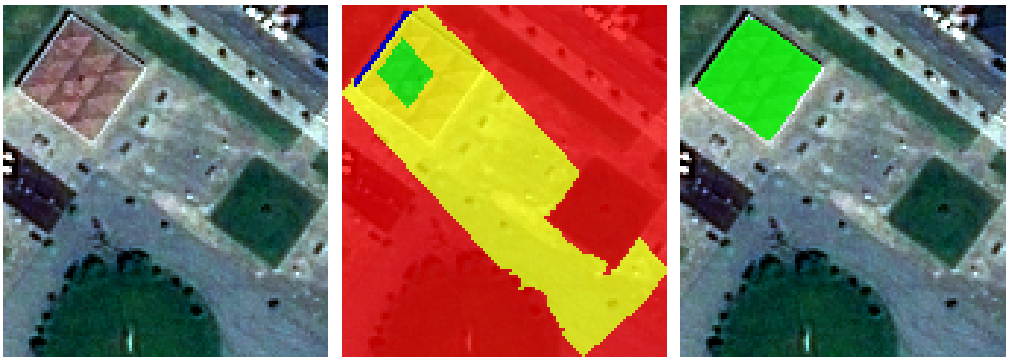
6.2.3 GMM Components Assignment and Iterative Energy Minimization

After the trimap T has been generated, each pixel is assigned to a GMM component value for either background or foreground, as in Section 6.1.2.2, but for 4-dimensional data. In each iteration of the energy minimization, the algorithmic structure is equivalent to [97]: The foreground and background GMMs are computed (4-dimensional instead of 3) with respect to the pixel values in trimap, the unary and boundary penalty functions are defined and graph cut bipartitioning using maximum flow is applied. After convergence, the local result yields the set B_i of candidate building pixels.

6.2.4 Building Detection Refinement

After the bipartitioning has been obtained, a post processing is applied on B_i: Since each local landscape $\beta_\alpha(\varsigma_i)$ is assumed to be associated with a single building; if B_i covers multiple separate regions, only the region intersecting with $\mathcal{T}_F$ is verified as building; and the pixels in remaining regions are removed from B_i. Moreover; after the local Grabcut results have been obtained for all shadow components ς_i, a connected component analysis is applied on the final building detection map M_B in order to denominate separate regions, and the ones with area less than T_{area} are removed from M_B.

The local detection of a single building using Grabcut is shown in Figure 6.5.



(a) (b) (c)

Figure 6.5: Process of a local Grabcut bipartitioning. (a) Image patch in which bipartitioning is utilized. (b) Automatic selection of foreground and background pixels. Blue represents the shadow of building, green represents selected foreground region, red represents selected background region and yellow represents the region to be labeled after Grabcut. (c) Result of bipartitioning, where detected building is shown with green.

CHAPTER 7

PERFORMANCE EVALUATION AND EXPERIMENTS

In this section, the methodologies used in performance evaluation and conducted experiments on building detection are explained. First, pixel-based and object-based measures used for evaluation are described. Then, information about the data set for test images are given. Afterwards, the selected parameters for the algorithm are given. Finally, pixel-based and object-based performance values for building detection are shown and visualized.

7.1 Performance Evaluation

For calculating the detection performance of the proposed methodology, pixel-based and object-based approaches are used. The manually drawn reference ground truth maps for test images are used as in [83].

7.1.1 Pixel-Based Approach

For pixel-based approach, each pixel is associated with one of the four labels:

1. True Positive (TP): Pixels detected as building in the proposed method, and also labeled as building in the ground truth map.
2. True Negative (TN): Pixels detected as not building in the proposed method, and also labeled as not building in the ground truth map.
3. False Positive (FP): Pixels detected as building in the proposed method, but labeled as not building in the ground truth map.

4. False Negative (FN): Pixels detected as not building in the proposed method, but labeled as building in the ground truth map.

Then, precision and recall values are calculated from these labels:

$$Precision = \frac{|TP|}{|TP| + |FP|} \tag{7.1}$$

$$Recall = \frac{|TP|}{|TP| + |FN|} \tag{7.2}$$

where $|\mathcal{L}|$ denotes the number of pixels labeled as $\mathcal{L}$.

7.1.2 Object-Based Approaches

For object-based approaches, several performance measures for detection evaluation have been investigated in [4; 86]. Among these, Hoover-based evaluation [49] and bipartite graph matching [57] are chosen to evaluate the building detection performance.

7.1.2.1 Notations

- The set of building objects in the ground truth is denoted as $GT = \{GT_1, GT_2, ..., GT_{N_{GT}}\}$.
- The set of separately detected building objects in the output map is denoted as $O = \{O_1, O_2, ..., O_{N_O}\}$.
- An object numbered as i in the ground truth is denoted by GT_i.
- An object numbered as i in the output map is denoted by O_i.
- N_{GT} represents the number of objects in the ground truth.
- N_O represents the number of objects in the output map.
- C_{ij} shows the number of overlapping pixels between GT_i and O_j.
- $|O|$ denotes the number of pixels belonging to an object O [86].

7.1.2.2 Hoover's Measure

In Hoover et al. [49], set of ground truth / output object pairs are labeled as correct detection, over-detection, under-detection, false negative or false positive with respect to a threshold [86]. A similar evaluation measure is used in this study, where the building is labeled as true positive if the corresponding output object O_j overlaps with a ground truth object GT_i with a ratio of $T_{overlap}$:

$$C_{ij}/|O_j| \geq T_{overlap}.$$

An output object O_j is labeled as false positive if it does not overlap with any ground truth object $GT_i \in GT$ with ratio of $T_{overlap}$. Furthermore, a ground truth object GT_i is labeled as missed detection (false negative) if no output object O_j overlaps with a ratio of $T_{overlap}$.

After determining the correct, false and missed detections; precision and recall measures are calculated as in Equation 7.1.

7.1.2.3 Bipartite Graph Matching

In bipartite graph matching algorithm, GT and O are represented as a set of nodes $\mathcal{V}$ in a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. A restriction is that, this $\mathcal{G}$ is a bipartite graph; where there can not be an edge between two output nodes, or two ground truth nodes:

$$\forall i, j \{O_i, O_j\} \text{ and } \{GT_i, GT_j\} \notin \mathcal{E}.$$

The edge weights w_{ij} between GT_i and O_j are set to a function of C_{ij}. In this work, w_{ij} is calculated as

$$\forall \{GT_i, O_j\} \in \mathcal{E}\ ,\ w_{ij} = C_{ij}/|GT_i| \cup |O_j|.$$

The aim is to obtain a subgraph $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ with maximum sum of edge weights. This subgraph corresponds to a one-to-one matching between GT and O. Therefore, each node

can cover at most one edge in $\mathcal{G}'$. Computing the one-to-one matching with maximizing sum of edge weights can be solved in polynomial time by using Hungarian (or Munkres) algorithm [71].

The uncovered nodes in O are labeled as false positives, the uncovered nodes in GT are labeled as false negatives and the one-to-one correspondences in the resulting matching algorithm are labeled as true positives. Finally, precision and recall measures are calculated as in Equation 7.1.

7.2 Data Set

The proposed building detection algorithm is tested over 20 pansharpened satellite images, taken from Quickbird and Geoeye satellites. The images from Quickbird have a spatial resolution of 0.6m, whereas the Geoeye images have a spatial resolution of 0.5m. The satellite images contain 4 multispectral bands (red, blue, green, near-infrared) and each band has a spectral resolution of 11 bits per pixel. Most of the images cover residential area, whereas a few hard images contain snow and desert. The sizes of the images in terms of pixels vary from 300x300 to 1000x1300.

7.3 Parameter Values

In this section, the selected parameter values of the proposed algorithm are explained. The parameters having unit of length are set in meters (or m^2) instead of pixels.

The parameters b and T_{b-NDVI} used for wet soil detection in Section 4.1.1 are set to 20 and 0.5 respectively. In Section 5.3, the parameters σ and κ introduced for fuzzy landscape generation are set to 100 and 40m respectively. Also in Section 5.5.1, for eliminating local landscapes due to vegetation, the introduced thresholds T_{low}, T_{high} and T_{veg} are defined as 0.7, 0.9 and 0.7 respectively. Moreover, the height threshold H_{min} introduced in Section 5.5.2 is set to $3m$.

The lower-upper thresholds η_{low} and η_{high} introduced in Section 6.2.1.1 for determining foreground pixels are defined as 0.4 and 0.9 respectively. Also, T_{map} for eliminating artifacs is set to $40m^2$. The shrinking distance d_{shrink} for cropping the both sides of initial foreground region is defined as $2m$. Moreover, the length parameter d_{Bbox} of the structuring element intro-

duced in section 6.2.1.2 is defined as $50m$ and the minimum area parameter T_{area} for building detection refinement is set to $120m^2$.

The overlapping threshold $T_{overlap}$ described in Section 7.1.2.2 is set as 0.6.

Table 7.1 shows all user-defined parameters and their values used in the experiments.

Table 7.1: List of parameters introduced in the algorithm and their defined values.

Parameter name	**Defined value**	**Section**
b	20	4.1.1
T_{b-NDVI}	0.5	4.1.1
σ	100	5.3
κ	$40m$	5.3
T_{low}	0.7	5.5.1
T_{high}	0.9	5.5.1
T_{veg}	0.7	5.5.1
H_{min}	$3m$	5.5.2
η_{low}	0.4	6.2.1.1
η_{high}	0.9	6.2.1.1
T_{map}	$40m^2$	6.2.1.1
d_{shrink}	$2m$	6.2.1.1
d_{Bbox}	$50m$	6.2.1.2
T_{area}	$120m^2$	6.2.4
$T_{overlap}$	0.6	7.1.2

7.4 Results

The experiments are performed on a machine with Intel i5 2.6GHz CPU and 4 GB RAM. The detection stage using Grabcut is implemented in C++ using OpenCV 2.1 [19], and the remaining stages are implemented in MATLAB. The processing time over all 20 images take approximately 7.5 minutes, with an average of 22.5 seconds per image, and Grabcut stage takes approximately 4.5 minutes, with 13.5 seconds per image. Table 7.2 shows the average running times of each step of the proposed algorithm.

Table 7.3 shows the pixel-based and object-based results (using Hoover's method and bipartite graph matching respectively), and Figure 7.1 shows visual results of the proposed building detection methodology.

Table 7.2: Average running time of each step of the algorithm.

	Vegetation & Water & Shadow Detection	Probability Map Generation	Probability Map Refinement	Grabcut Bi-partitioning & Post-processing	Total
Running Time	2.1*sec*	0.15*sec*	6.85*sec*	13.45*sec*	22.55*sec*

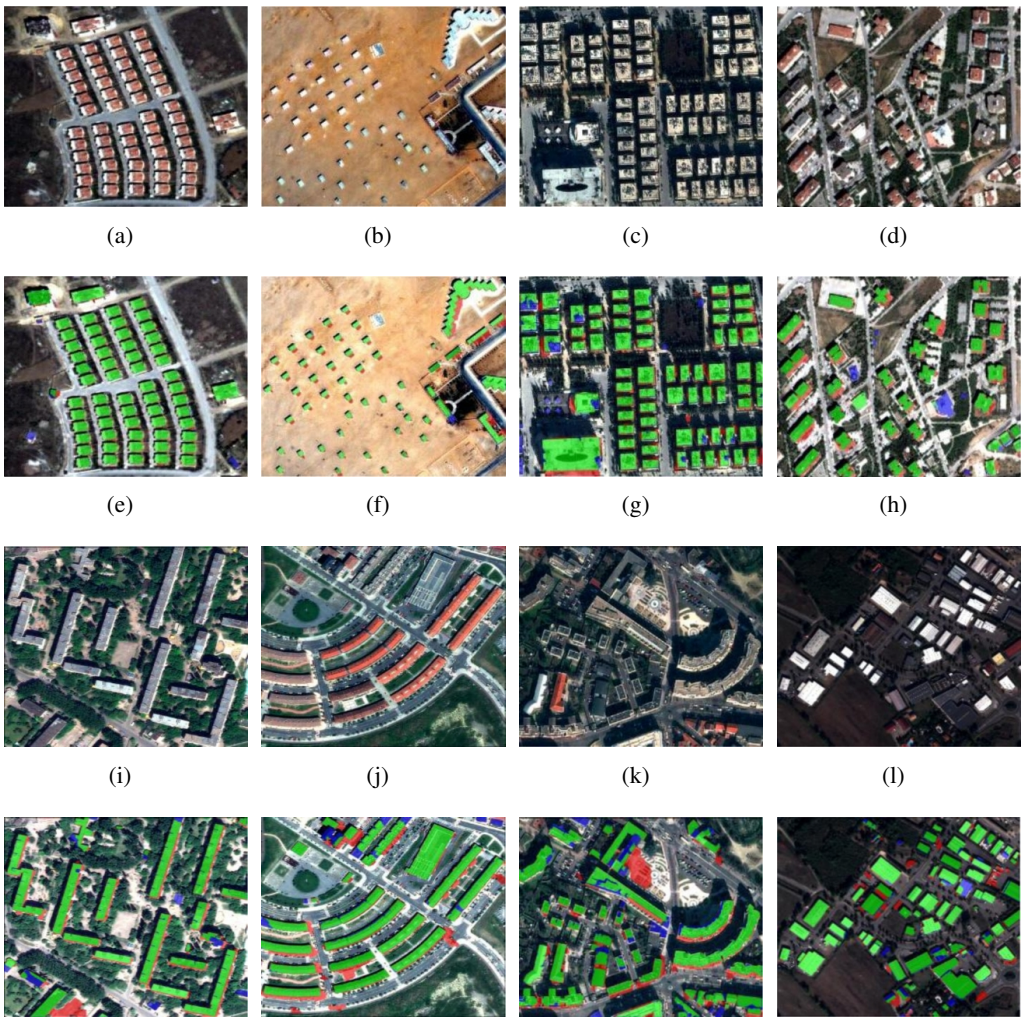


Figure 7.1: Visual detection results with respect to ground truths. The original images are shown in odd rows, the detection results are shown in even rows. Green, blue and red pixels represent detection, miss and false alarm respectively.

Also, Figure 7.2 shows precision and recall values of Hoover-based evaluation with varying values of $T_{overlap}$ parameter, and Figures 7.3, 7.4, 7.5, 7.6 show the pixel-based performance

Table 7.3: Pixel and object-based performance evaluation results for each test image and in overall.

Image ID	Spatial Res.	Pixel-Based Accuracy		Object-Based Accuracy			
				Hoover-based Measure		Bipartite Graph Matching	
		Precision	Recall	Precision	Recall	Precision	Recall
1	0.6m	86.75%	83.32%	100%	81.63%	100%	85.71%
2	0.6m	84.33%	92.50%	92.31%	92.31%	96.15%	96.15%
3	0.6m	89.22%	88.71%	100%	86.49%	100%	94.59%
4	0.6m	89.06%	93.32%	98.39%	93.85%	100%	93.85%
5	0.6m	74.83%	88.45%	100%	82.93%	100%	82.93%
6	0.6m	65.60%	94.97%	73.33%	91.67%	78.57%	91.67%
7	0.5m	43.17%	63.44%	48.28%	60.87%	80.95%	73.91%
8	0.5m	81.71%	80.29%	76.48%	66.67%	72.73%	82.05%
9	0.5m	70.49%	73.41%	75.90%	65.63%	94.83%	57.29%
10	0.5m	60.82%	87.64%	81.97%	96.15%	98.11%	100%
11	0.5m	82.55%	88.86%	95.77%	85.00%	97.18%	86.25%
12	0.5m	74.71%	75.75%	86.75%	74.48%	97.61%	70.34%
13	0.5m	86.64%	82.74%	68.97%	83.33%	61.76%	87.50%
14	0.5m	80.02%	88.16%	94.74%	69.23%	78.57%	84.62%
15	0.5m	78.73%	90.30%	76.00%	82.61%	76.92%	86.96%
16	0.5m	80.79%	80.23%	85.00%	72.86%	87.88%	82.86%
17	0.5m	79.00%	83.21%	84.09%	86.05%	94.87%	86.05%
18	0.5m	87.82%	86.84%	74.32%	79.71%	72.41%	91.30%
19	0.5m	79.17%	57.94%	57.14%	57.14%	60.71%	80.95%
20	0.5m	88.41%	75.81%	85.11%	67.80%	72.31%	79.66%
Average	–	81.01%	81.97%	84.46%	77.82%	86.08%	86.45%
Std.Dev.	–	11.31%	9.66%	14.32%	11.39%	13.47%	9.69%

of the algorithm with respect to different parameter / threshold values.

Moreover, Table 7.4 and Table 7.5 show comparisons in terms of pixel-based and Hoover-based accuracy between the proposed algorithm and the building detection algorithm in [103]. In the ensemble learning-based algorithm [103], first the image is segmented using mean-shift. Then at bottom level, each segment is classified by a set of different and independent binary classifiers using set of different features, and class membership values for each classification are obtained. Afterwards, these membership values are concatenated, and top-level binary decision (building / non-building) is performed by using a linear classifier on the space formed by concatenation of membership values.

At the bottom level, color-based features (segment mean, histogram) and structural / shape-based features (segment area, segment length-to-width ratio, segment rectangularity) are used. The color features are the most promising features for modeling buildings in the experiments in [103], as well as in this study. Also, the structural and shape-based features are used in order contribute the boosting performance of the ensemble classifier as much as possible, by modeling the shape and structure of a single segment.

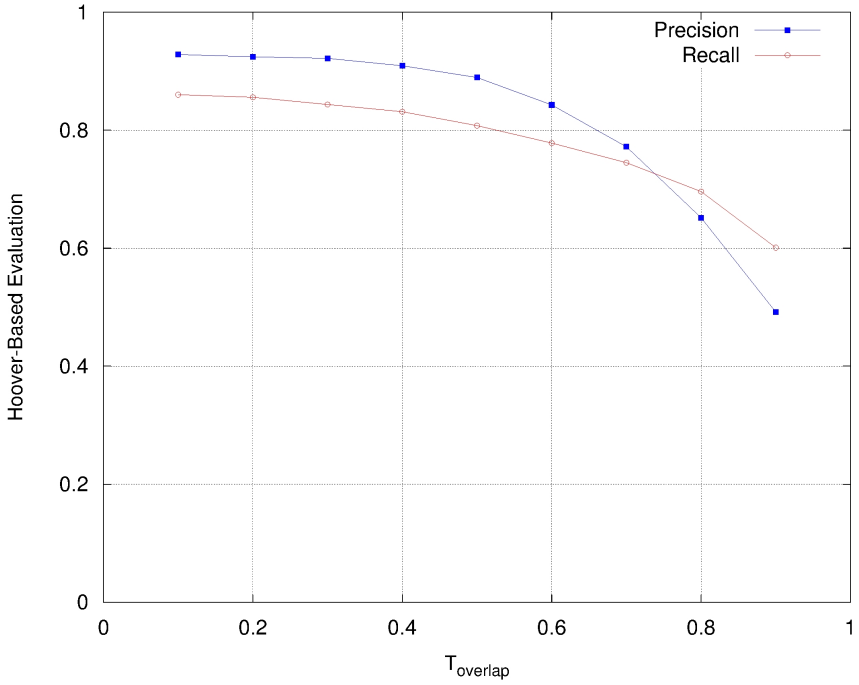


Figure 7.2: Hoover-based evaluation scores with respect to $T_{overlap}$ varying between [0.1, 0.9].

Table 7.4: Comparison of pixel-based scores of the proposed algorithm and the building detection algorithm in [103] over 20 images in the data set.

Image ID	Proposed Algorithm		Algorithm in [103]	
	Pixel-based Precision	Pixel-based Recall	Pixel-based Precision	Pixel-based Recall
1	86.75%	83.32%	95.02%	90.78%
2	84.33%	92.50%	90.43%	89.60%
3	89.22%	88.71%	89.11%	92.98%
4	89.06%	93.32%	87.60%	85.65%
5	74.83%	88.45%	81.85%	77.27%
6	65.60%	94.97%	91.22%	84.54%
7	43.17%	63.44%	81.80%	78.01%
8	81.71%	80.29%	67.86%	57.94%
9	70.49%	73.41%	68.40%	61.85%
10	60.82%	87.64%	67.98%	65.41%
11	82.55%	88.86%	83.33%	59.11%
12	74.71%	75.75%	84.81%	75.99%
13	86.64%	82.74%	89.66%	75.18%
14	80.02%	88.16%	92.86%	85.11%
15	78.73%	90.30%	71.57%	73.55%
16	80.79%	80.23%	71.76%	69.75%
17	79.00%	83.21%	69.47%	53.19%
18	87.82%	86.84%	92.73%	90.76%
19	79.17%	57.94%	59.42%	60.96%
20	88.41%	75.81%	95.91%	88.43%
Average	81.01%	81.97%	81.64%	75.80%
Std.Dev.	11.31%	9.66%	11.15%	12.64%

Table 7.5: Comparison of Hoover-based scores of the proposed algorithm and the building detection algorithm in [103] over 20 images in the data set.

Image ID	Proposed Algorithm		Algorithm in [103]	
	Object-based Precision	Object-based Recall	Object-based Precision	Object-based Recall
1	100%	81.63%	95.83%	93.10%
2	92.31%	92.31%	92.86%	92.86%
3	100%	86.49%	92.59%	96.30%
4	98.39%	93.85%	96.97%	91.43%
5	100%	82.93%	73.33%	85.71%
6	73.33%	91.67%	77.78%	83.75%
7	48.28%	60.87%	90.95%	71.43%
8	76.48%	66.67%	38.71%	56.52%
9	75.90%	65.63%	61.54%	64.58%
10	81.97%	96.15%	68.31%	56.80%
11	95.77%	85.00%	67.86%	53.66%
12	86.75%	74.48%	84.77%	75.78%
13	68.97%	83.33%	75.36%	78.57%
14	94.74%	69.23%	78.57%	90.91%
15	76.00%	82.61%	36.36%	75.71%
16	85.00%	72.86%	72.47%	66.66%
17	84.09%	86.05%	50.53%	55.17%
18	74.32%	79.71%	76.19%	91.30%
19	57.14%	57.14%	33.69%	61.54%
20	85.11%	67.80%	89.57%	92.11%
Average	84.46%	77.82%	72.71%	76.69%
Std.Dev.	14.32%	11.39%	19.77%	14.91%

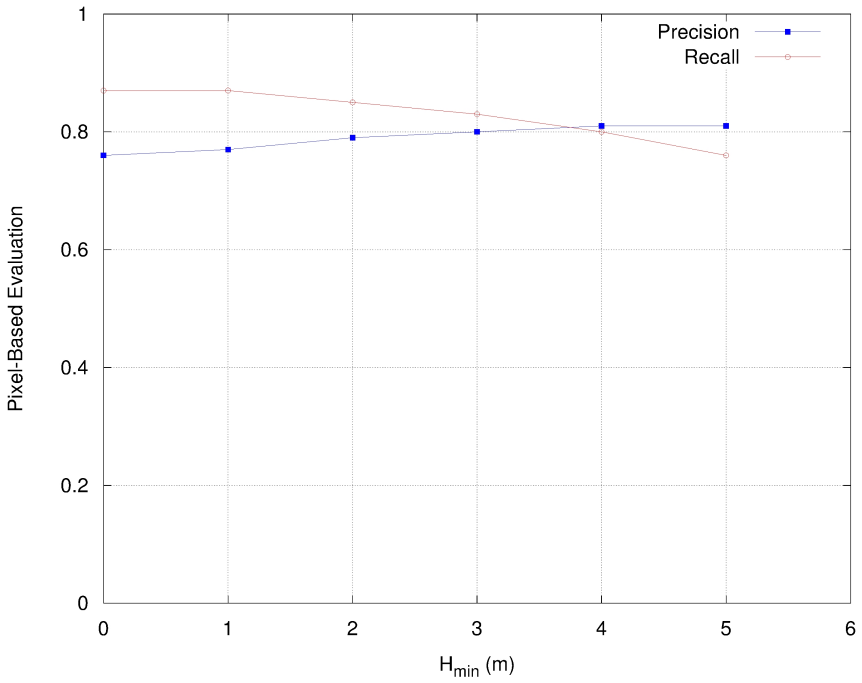


Figure 7.3: Pixel-based performance curve of H_{min} parameter. [83]

The supervision data for the ensemble classification framework is obtained from half-images over all 20 images. In other words, one part of all 20 images are reserved as supervision data, and the algorithm performance is tested on the other halves of the 20 images. Since supervision data is selected from an image patch of test images instead of different, statistically irrelevant images, covariate shift or statistical discordance between training and test data no more becomes a major problem.

Considering the comparison of algorithms; in images where buildings are similar in terms of size, shape and color; the algorithm in [103] performs well, and even outperforms the proposed methodology. However; in harsh images including weather conditons such as snow and desert, or images in which buildings are placed irregularly and differ in terms of shape and color; the supervised algorithm fails; even performs below 50%. This comparison experiment also proves that in these kind of images, shadow and its direction is a powerful cue for detecting buildings.

It should be noted that in these Figures 7.3, 7.4, 7.5, 7.6; the parameters only related to local detection step are analyzed, and the ones with considerable changes in precision & recall performances are shown. The parameters related to landscape generation (σ, κ) are not examined for different values, since this experiment would be equivalent to examining η_{low}. Parameters related to vegetation are also not examined, since these parameter perturbations do not even change vegetation detection elimination results negligibly. Besides, the shrinking distance d_{shrink} is taken to be minimum possible pixel-based integer distance, which corresponds to 2m total from both sides.

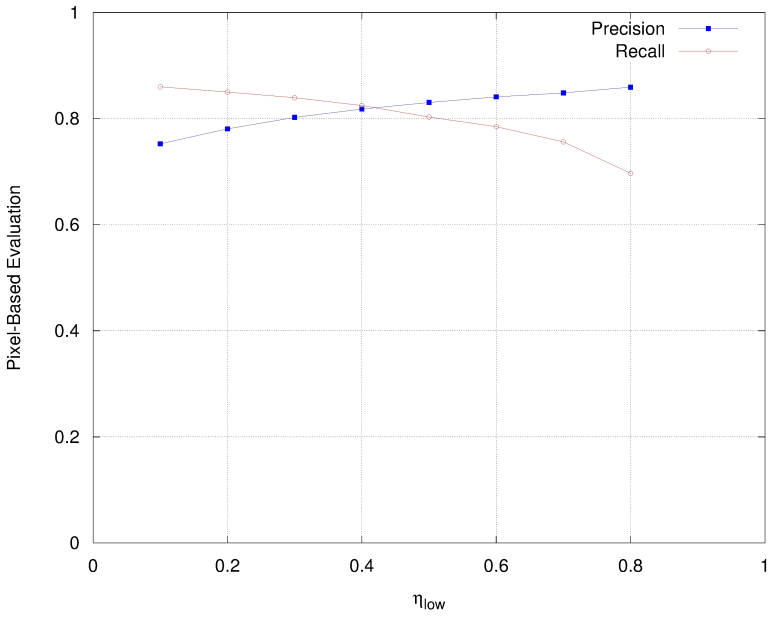


Figure 7.4: Pixel-based performance curve for η_{low} parameter. [83]

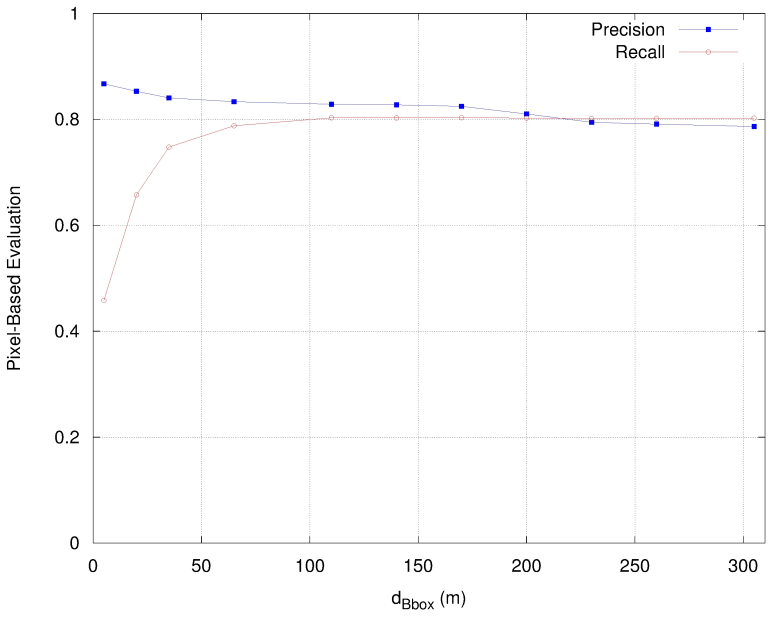


Figure 7.5: Pixel-based performance curve for d_{Bbox} parameter. [83]

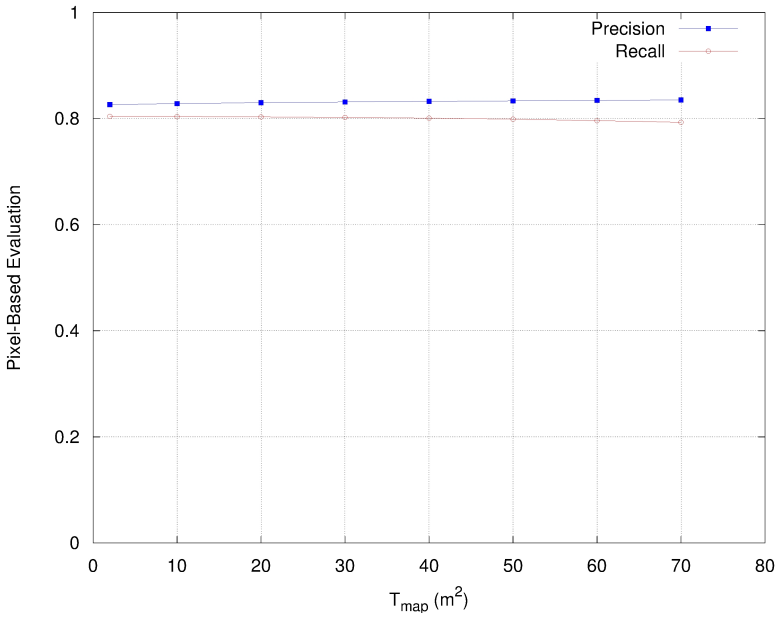


Figure 7.6: Pixel-based performance curve for T_{map} parameter. [83]

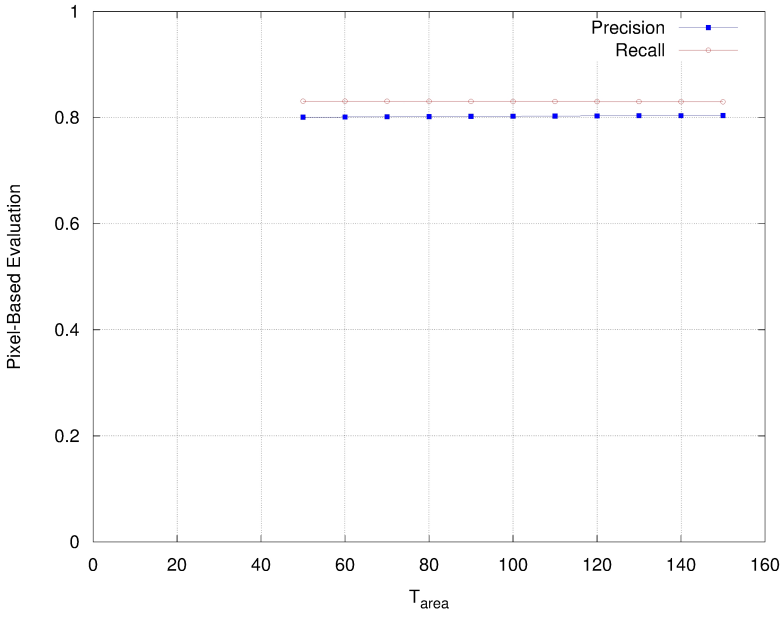


Figure 7.7: Pixel-based performance curve for T_{area} parameter.

As shown both statistically and visually, the proposed building methodology gives fairly

promising results, for varying test regions including complex scenes with buildings with arbitrary shapes or sizes. Also, as shown in Figures 7.3, 7.4, 7.5, 7.6; further experiments [83] with respect to the parameters / thresholds on the proposed methodology prove that it is quite stable and insensitive to the parameters, and fairly strong in terms of algorithmic efficiency.

The errors in the detection results are mostly due to the the nonexisting shadows of the buildings, or the shadow detection error. As expected; for a building with shadow, at least a part of the building is always detected.

The low accuracies in images (such as 7 and 13) are due to the snowy weather conditions reverberated in the image. For these images, detecting buildings even manualy is extremely difficult. On the remaining cases, the pixel and object based accuracies are within a more promising range (70%, 100%).

The recall values in object-based evaluation with Hoover's algorithm slightly decrease compared to pixel-based scores. Failing to detect tiny buildings in an image does not affect pixel-based detection score significantly; however Hoover index evaluates a tiny building and a large building equally. Considering possible misdetections of shadows of small buildings, these errors may dominate the object-based recall evaluation, therefore resulting in a relatively low recall values.

As seen in Table 7.3, bipartite graph matching (BGM) yields higher accuracies compared to Hoover's method. The reason is that unlike the overlapping thresholds defined in Hoover-based accuracy, BGM determines a one-to-one correspondence between output and ground truth objects. A one-to-one assignment between an output object O_i and a ground truth object GT_j is possible even if small overlapping region C_{ij} exist between them.

Using cues related to shadow in a robust manner, in most cases the potential building regions are successfully detected, and with an effective post-processing on local Grabcut results most of the false positives are eliminated. Thus, a high precision and recall values are obtained for BGM.

Occasionally, over-growth (the detected region can exceed the extents of real building) and under-growth (the building region cannot be fully detected) cases may occur in local Grabcut results. These issues and possible improvements are explained in Chapter 8.

CHAPTER 8

CONCLUSION

In this work, a new automated building detection methodology for satellite images is proposed. In this methodology; first the vegetation, water and shadow regions are detected from a given satellite image, and local fuzzy landscapes are generated from the shadow regions using the direction of illumination obtained from image metadata. Afterwards, for each fuzzy landscape, foreground and background pixels are automatically determined and a bipartitioning is obtained using a graph-based algorithm called Grabcut. Finally, the local results are merged to obtain the final building detection result. Considering the performance evaluation results obtained in Chapter 7.4, this approach can be seen as a proof of concept that the shadow is an invariant for a building object and promising detection results can be obtained even when only a single invariant for an object is used.

Considering the ”pro”s of the algorithm; firstly it is computationally efficient. A running time of less than 25 seconds per image can be seen as the proof of concept of the time-efficiency. Also, no manual supervision of user interaction is required. Instead; the supervision data is automatically generated inside the algorithm, using shadow cues. Moreover, the proposed algorithm gives successful evaluation performance even in complex environments including wilderness and snow, and also in environments including different types of buildings in terms of size, shape, color and texture.

The ”con”s of the proposed algorithm are as follows. At first, the algorithm is eminently dependent to shadow detection. If the shadow of a building cannot be detected, then the detection of that building using the proposed methodology is not possible. However; considering the shadow detection performances over varying environments, this is an infrequent issue. Also; the algorithm cannot efficiently detect buildings in regions where the shadow of

a building falls onto another building, or where self-occlusion occurs for building rooftops. Moreover; in some test cases, there may exist non-building objects such as road segments and bridges that cannot be eliminated using height verification and mistakenly detected as buildings. The final limitation of the algorithm is that in some residential areas where buildings are located in a dense and crowded manner, multiple separate buildings can be under-detected as a single building.

Comparing the proposed building detection methodology with the other approaches in the literature; this approach can be considered as one of recherche algorithms that automatically generate supervision data for object detection. Also, the global ”building detection” problem is divided into many local bipartitioning problems in this methodology, and this way of approaching is unique in the literature. Unlike the PR-based approaches described in Chapter 2, this methodology is generalizable over different environmental conditions, and unlike the line and contour-based algorithms, the detection is not limited to only rectangular buildings; complex and non-rectangular shaped buildings can also be detected.

For detecting buildings in satellite images, more object invariants can be integrated into the architecture. One of these invariants can be considered as the star shape prior, where star shape is described as *”for any point p inside the object, all points on the straight line between the center c and p also lie inside the object”* [122]. This shape prior can easily be integrated by changing the boundary penalty term of the energy function of MRF. More generally; given a shape prior whose defining curve can be parameterized, this shape prior can be added to the graph cut partitioning framework by only changing the energy function [37].

Another improvement may be obtained by estimating the initial contour of the building candidate. For the estimation; while the approaches like snakes [61] or level sets [24] have a problem of sticking to the local minimum, the active segmentation algorithm [81] first computes the polar transformation of the image, and then computes the globally optimum contour by utilizing graph cut on the polar edge map. This approach may be useful for a better determination of trimap.

The detection results of other objects (roads, vehicles, etc.) outside the proposed framework can be used to boost the building detection results using a global ”between” relationship as described in [2]. This relationship information would also overcome the lack of using global information in the proposed algorithm.

Apart from those mentioned above, more improvements can be listed below:

- **(Estimating initial foreground GMM from only $\mathcal{T}_F$)** In Grabcut, the pixels initially labeled as unknown are included in the foreground GMM calculation. Instead, for the initial step, the foreground GMM can be estimated by using only the pixels in $\mathcal{T}_F$.

- **(Elegant fuzzy landscape utilization)** In the proposed approach, the locally generated fuzzy landscapes are used only for selecting the initial foreground pixels. However, they can be directly integrated into the energy function of MRF as a coefficient for unary potentials, or a directional constraint for boundary potentials.

- **(Flow reuse)** For iterative energy minimization in Grabcut, a fresh s-t graph is initialized at every step. Instead, the result of previous flow capacity can be utilized in successive steps, only increasing capacities of the nodes, whose labels change, is sufficient [18]

- **(Generation of the graphical model over segmentation result / Using higher order cliques)** Forming the MRF architecture over segments instead of pixels would seem a more elegant solution. However, this would make the inference problem much harder. Since the graph induced by segments are irregular, the energy terms of higher order cliques rather than simple unary + binary potentials would dominate the true energy function. This would cause the problem to be *NP*-hard, and approximate polynomial-time solutions would mostly stick in a local minimum [123].

- **(Grabcut parameter estimation for satellite images)** The internal parameters of Grabcut are estimated by using a learning scheme explained in [12]. Since the images used in [12] are 3-band and 8-bit; a similar scheme can be utilized for 4-band, 11-bit satellite images.

REFERENCES

[1] H. Akcay and S. Aksoy. Building detection using directional spatial constraints. In *IEEE International Geoscience and Remote Sensing Symposium*, 2010.

[2] S. Aksoy and H. Akçay. *Image Classification and Object Detection Using Spatial Contextual Constraints*. CRC Press, 2012. ISBN 978-1-43985-596-6.

[3] S. Aksoy and R.G. Cinbis. Image mining using directional spatial constraints. *Geoscience and Remote Sensing Letters, IEEE*, 7(1):33–37, 2010.

[4] S. Aksoy, B. Ozdemir, S. Eckert, F. Kayitakire, M. Pesarasi, O. Aytekin, C.C. Borel, J. Cech, E. Christophe, S. Duzgun, et al. Performance evaluation of building detection and digital surface model extraction algorithms: Outcomes of the prrs 2008 algorithm performance contest. In *Pattern Recognition in Remote Sensing (PRRS 2008), 2008 IAPR Workshop on*, pages 1–12. IEEE, 2008.

[5] O. Aytekin, I. Ulusoy, EZ Abacioglu, and E. Gokcay. Building detection in high resolution remotely sensed images based on morphological operators. In *Recent Advances in Space Technologies, 2009. RAST'09. 4th International Conference on*, pages 376–379, 2009.

[6] O. Aytekin, I. Ulusoy, A. Erener, and HSB Duzgun. Automatic and unsupervised building extraction in complex urban environments from multi spectral satellite imagery. In *Recent Advances in Space Technologies, 2009. RAST'09. 4th International Conference on*, pages 287–291, 2009.

[7] G.H. Ball and D.J. Hall. Isodata, a novel method of data analysis and pattern classification. Technical report, DTIC Document, 1965.

[8] U. Bayram, G. Can, B. Yuksel, S. Duzgun, and N. Yalabik. Unsupervised land use-land cover classification for multispectral images. In *Signal Processing and Communications Applications (SIU), 2011 IEEE 19th Conference on*, pages 574–577. IEEE, 2011.

[9] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 259–302, 1986.

[10] P. Bezier. Mathematical and practical possibilities of unisurf. *Computer Aided Geometric Design*, 1(1), 1974.

[11] C.M. Bishop and SpringerLink (Service en ligne). *Pattern recognition and machine learning*, volume 4. springer New York, 2006.

[12] A. Blake, C. Rother, M. Brown, P. Pérez, and P. Torr. Interactive image segmentation using an adaptive gmmrf model. *Computer Vision-ECCV 2004*, pages 428–441, 2004.

[13] I. Bloch. Fuzzy relative position between objects in image processing: a morphological approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(7): 657–664, 1999.

[14] D.E. Bowker, R.E. Davis, D.L. Myrick, K. Stacy, and W.T. Jones. Spectral reflectances of natural targets for use in remote sensing studies. 1985.

[15] Y. Boykov and G. Funka-Lea. Graph cuts and efficient nd image segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.

[16] Y. Boykov and M.P. Jolly. Interactive organ segmentation using graph cuts. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2000*, pages 147–175. Springer, 2000.

[17] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, 2004.

[18] Y.Y. Boykov and M.P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112. IEEE, 2001.

[19] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, 2008.

[20] L. Breiman. *Classification and regression trees*. Chapman & Hall/CRC, 1984.

[21] J.E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems journal*, 4(1):25–30, 1965.

[22] L. Bruzzone and L. Carlin. A multilevel context-based system for classification of very high spatial resolution images. *Geoscience and Remote Sensing, IEEE Transactions on*, 44(9):2587–2600, 2006.

[23] W.G. Carrara, R.S. Goodman, and R.M. Majewski. Spotlight synthetic aperture radar-signal processing algorithms(book). *Norwood, MA: Artech House, 1995.*, 1995.

[24] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International journal of computer vision*, 22(1):61–79, 1997.

[25] J. Cha, RH Cofer, and SP Kozaitis. Extended hough transform for linear feature detection. *Pattern Recognition*, 39(6):1034–1043, 2006.

[26] R.G. Cinbis and S. Aksoy. Relative position-based spatial relationships using mathematical morphology. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 2, pages II–97. IEEE, 2007.

[27] R.G. Cinbis and S. Aksoy. Modeling spatial relationships in images. Technical report, Department of Computer Engineering, Bilkent University, 2007.

[28] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[29] J.A. Curcio and C.C. Petty. The near infrared absorption spectrum of liquid water. *JOSA*, 41(5):302–302, 1951.

[30] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[31] J. Edmonds and R.M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (JACM)*, 19(2):248–264, 1972.

[32] M. Ehlers. Beyond pansharpening: advances in data fusion for very high resolution remote sensing data. In *Proceedings, ISPRS Workshop High-Resolution Earth Imaging for Geospatial Information*, pages 17–20, 2005.

[33] D. Fasbender, J. Radoux, and P. Bogaert. Bayesian data fusion for adaptable image pansharpening. *Geoscience and Remote Sensing, IEEE Transactions on*, 46(6):1847–1857, 2008.

[34] K.S. Fassnacht, S.T. Gower, M.D. MacKenzie, E.V. Nordheim, and T.M. Lillesand. Estimating the leaf area index of north central wisconsin forests using the landsat thematic mapper. *Remote Sensing of Environment*, 61(2):229–245, 1997.

[35] M. Fauvel, J. Chanussot, and J.A. Benediktsson. Decision fusion for the classification of urban remote sensing images. *Geoscience and Remote Sensing, IEEE Transactions on*, 44(10):2828–2838, 2006.

[36] DR Ford and D.R. Fulkerson. *Flows in networks*. Princeton university press, 2010.

[37] D. Freedman and T. Zhang. Interactive graph cut based segmentation with shape priors. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 755–762. Ieee, 2005.

[38] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting, 1995.

[39] D. Gabor. Theory of communication. part 1: The analysis of information. *Electrical Engineers-Part III: Radio and Communication Engineering, Journal of the Institution of*, 93(26):429–441, 1946.

[40] D.M. Gates. *Biophysical ecology*. Dover Pubns, 2003.

[41] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):721–741, 1984.

[42] PCI Geomatics. Geomatica, version 10.1. 3. *PCI Geomatics*, 50, 2008.

[43] A.V. Goldberg and R.E. Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM (JACM)*, 35(4):921–940, 1988.

[44] P. Gong, R. Pu, G.S. Biging, and M.R. Larrieu. Estimation of forest leaf area index using vegetation indices derived from hyperion hyperspectral data. *Geoscience and Remote Sensing, IEEE Transactions on*, 41(6):1355–1362, 2003.

[45] DM Greig, BT Porteous, and A.H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 271–279, 1989.

[46] V. Guducu and U. Halici. Hypothesis based detection of building with rectilinear projection in satellite images using shade and color information. In *Signal Processing and Communications Applications Conference (SIU), 2010 IEEE 18th*, pages 680–683, 2010.

[47] J.M. Hammersley and P. Clifford. Markov fields on finite graphs and lattices. 1968.

[48] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.

[49] A. Hoover, G. Jean-Baptiste, X. Jiang, P.J. Flynn, H. Bunke, D.B. Goldgof, K. Bowyer, D.W. Eggert, A. Fitzgibbon, and R.B. Fisher. An experimental comparison of range image segmentation algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(7):673–689, 1996.

[50] A. Huertas and R. Nevatia. Detecting buildings in aerial images. *Computer Vision, Graphics, and Image Processing*, 41(2):131–152, 1988.

[51] E. Imagine. Erdas imagine tour guides. *Leica Geosystems*, page 730, 2006.

[52] J. Inglada and E. Christophe. The orfeo toolbox remote sensing image processing software. In *Geoscience and Remote Sensing Symposium, 2009 IEEE International, IGARSS 2009*, volume 4, pages IV–733. IEEE, 2009.

[53] R.B. Irvin and D.M. McKeown Jr. Methods for exploiting the relationship between buildings and their shadows in aerial imagery. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(6):1564–1575, 1989.

[54] E. Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik A Hadrons and Nuclei*, 31(1):253–258, 1925.

[55] M. Izadi and P. Saeedi. Three-dimensional polygonal building model estimation from single satellite images. *Geoscience and Remote Sensing, IEEE Transactions on*, (99): 1–19.

[56] M. Izadi and P. Saeedi. Automatic building detection in aerial images using a hierarchical feature based image segmentation. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 472–475, 2010.

[57] X. Jiang, C. Marti, C. Irniger, and H. Bunke. Distance measures for image segmentation evaluation. *EURASIP Journal on Applied Signal Processing*, 2006:209–209, 2006.

[58] X. Jin and C.H. Davis. Automated building extraction from high-resolution satellite imagery in urban areas using structural, contextual, and spectral information. *EURASIP Journal on Applied Signal Processing*, 2005(14):2196–2206, 2005.

[59] I.T. Jolliffe and MyiLibrary. *Principal component analysis*, volume 2. Wiley Online Library, 2002.

[60] S. Joseph. The max-flow min-cut theorem, 2007.

[61] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.

[62] A. Katartzis and H. Sahli. A stochastic framework for the identification of building rooftops using a single remote sensing image. *Geoscience and Remote Sensing, IEEE Transactions on*, 46(1):259–271, 2008.

[63] J.M. Keller. A fuzzyˆ-nearest neighbor algorithm james m. keller, michael r. gray, and james a. givens, jr. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(4):581, 1985.

[64] RL Kettig and DA Landgrebe. Classification of multispectral image data by extraction and classification of homogeneous objects. *Geoscience Electronics, IEEE Transactions on*, 14(1):19–26, 1976.

[65] T. Kim and J.P. Muller. Development of a graph-based approach for building detection. *Image and Vision Computing*, 17(1):3–14, 1999.

[66] R. Kindermann, J.L. Snell, and American Mathematical Society. *Markov random fields and their applications*. American Mathematical Society Providence, RI, 1980.

[67] R.L. King and J. Wang. A wavelet based algorithm for pan sharpening landsat 7 imagery. In *Geoscience and Remote Sensing Symposium, 2001. IGARSS'01. IEEE 2001 International*, volume 2, pages 849–851. IEEE, 2001.

[68] T. Knudsen and A.A. Nielsen. Detection of buildings through multivariate analysis of spectral, textural, and shape based features. In *Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International*, volume 5, pages 2830–2833, 2004.

[69] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):147–159, 2004.

[70] FJ Kriegler, WA Malila, RF Nalepka, and W. Richardson. Preprocessing transformations and their effects on multispectral recognition. In *Remote Sensing of Environment, VI*, volume 1, page 97, 1969.

[71] H.W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[72] K. Kuratowski. Sur le probleme des courbes gauches en topologie. *Fund. Math*, 15 (271-283):79, 1930.

[73] Z. Lari and H. Ebadi. *Automated Building Extraction from High-Resolution Satellite Imagery Using Spectral and Structural Information Based on Artificial Neural Networks*. Hannover: ISPRS Commission III, WGIII/4.(available at: http://www. ipi. uni-hannover. de/fileadmin/institut/pdf/Lari_Ebadi. pdf), 2007.

[74] D.S. Lee, J. Shan, and J.S. Bethel. Class-guided building extraction from ikonos imagery. *Photogrammetric Engineering and Remote Sensing*, 69(2):143–150, 2003.

[75] S.Z. Li. *Markov random field modeling in image analysis*. Springer-Verlag New York Inc, 2009.

[76] C. Lin and R. Nevatia. Building detection and description from a single intensity image. *Computer vision and image understanding*, 72(2):101–121, 1998.

[77] D.G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision,*

1999. The Proceedings of the Seventh IEEE International Conference on, volume 2, pages 1150–1157. Ieee, 1999.

[78] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, page 14. California, USA, 1967.

[79] H. Mayer. Automatic object extraction from aerial imagery-a survey focusing on buildings. *Computer vision and image understanding*, 74(2):138–149, 1999.

[80] F. Meyer and S. Beucher. Morphological segmentation. *Journal of visual communication and image representation*, 1(1):21–46, 1990.

[81] A. Mishra, Y. Aloimonos, and C.L. Fah. Active segmentation with fixation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 468–475. IEEE, 2009.

[82] A.O. Ok. *Matching and reconstruction of line features from ultra-high resolution stereo aerial imagery*. PhD thesis, Middle East Technical University, 2012.

[83] A.O. Ok, C. Senaras, and B. Yuksel. Automated detection of arbitrarily-shaped buildings in complex environments from monocular vhr optical satellite imagery. *Accepted for publication in Transactions on Geoscience and Remote Sensing Review*, 2012.

[84] T. Ormsby, E. Napoleon, and R. Burke. *Getting to Know ArcGIS Desktop: The Basics of ArcView, ArcEditor, and ArcInfo Updated for ArcGIS 9*. Esri Press, 2004.

[85] N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11: 285–296, 1975.

[86] B.I. Özdemir, S. Aksoy, S. Eckert, M. Pesaresi, and D. Ehrlich. Performance measures for object detection evaluation. *Pattern Recognition Letters*, 31(10):1128–1137, 2010.

[87] E. Pakizeh and M. Palhang. Building detection from aerial images using hough transform and intensity information. In *Electrical Engineering (ICEE), 2010 18th Iranian Conference on*, pages 532–537, 2010.

[88] C.H. Papadimitriou and K. Steiglitz. *Combinatorial optimization: algorithms and complexity*. Dover Pubns, 1998.

[89] J. Peng, D. Zhang, and Y. Liu. An improved snake model for building detection from urban aerial images. *Pattern Recognition Letters*, 26(5):587–595, 2005.

[90] M. Pesaresi and J.A. Benediktsson. A new approach for the morphological segmentation of high-resolution satellite imagery. *Geoscience and Remote Sensing, IEEE Transactions on*, 39(2):309–320, 2001.

[91] R.M. Pope and E.S. Fry. Absorption spectrum (380–700 nm) of pure water. ii. integrating cavity measurements. *Applied optics*, 36(33):8710–8723, 1997.

[92] R.B. Potts. Some generalized order-disorder transformations. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 48, pages 106–109. Cambridge Univ Press, 1952.

[93] S.J.D. Prince. Computer vision: models, learning, and inference. 2011.

[94] K. Ren, H. Sun, Q. Jia, and J. Shi. Building recognition from aerial images combining segmentation and shadow. In *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on*, volume 4, pages 578–582, 2009.

[95] D. Reynolds. Gaussian mixture models. *Encyclopedia of Biometric Recognition*, 2008.

[96] E. Rosten, R. Porter, and T. Drummond. Faster and better: a machine learning approach to corner detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):105–119, 2010.

[97] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 309–314. ACM, 2004.

[98] C. Rother, V. Kolmogorov, Y. Boykov, and A. Blake. Interactive foreground extraction using graph cut. *Advances in Markov Random Fields for Vision and Image Processing*, 2011.

[99] W. Rudin. *Principles of mathematical analysis*, volume 3. McGraw-Hill New York, 1976.

[100] P. Saeedi and H. Zwick. Automatic building detection in aerial and satellite images. In *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, pages 623–629, 2008.

[101] S. SANGAM. Light detection and ranging.

[102] B. Schölkopf and A.J. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. the MIT Press, 2002.

[103] C. Senaras, B. Yuksel, M. Ozay, and F.Y. Vural. Automatic building detection with feature space fusion using ensemble learning. *IEEE IGARSS 2012*, 2012.

[104] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, 2000.

[105] B.W. Silverman. *Density estimation for statistics and data analysis*, volume 26. Chapman & Hall/CRC, 1986.

[106] B. Sirmacek and C. Unsalan. Building detection from aerial images using invariant color features and shadow information. In *Computer and Information Sciences, 2008. ISCIS'08. 23rd International Symposium on*, pages 1–5, 2008.

[107] B. Sirmacek and C. Unsalan. Building detection using local gabor features in very high resolution satellite images. In *Recent Advances in Space Technologies, 2009. RAST'09. 4th International Conference on*, pages 283–286, 2009.

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

[109] B. Sirmacek and C. Unsalan. A probabilistic framework to detect buildings in aerial and satellite images. *Geoscience and Remote Sensing, IEEE Transactions on*, 49(1): 211–221, 2011.

[110] Z. Song, C. Pan, Q. Yang, F. Li, and W. Li. Building roof detection from a single high-resolution satellite image in dense urban area. In *Proc. ISPRS Congress*, pages 271–277, 2008.

[111] H. Sportouche, F. Tupin, and L. Denise. Building detection by fusion of optical and SAR features in metric resolution data. In *Geoscience and Remote Sensing Symposium, 2009 IEEE International, IGARSS 2009*, volume 4, pages IV–769, 2009.

[112] X. Sun, K. Fu, H. Long, Y. Hu, L. Cai, and H. Wang. Contextual models for automatic building extraction in high resolution remote sensing image using Object-Based

boosting method. In *Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International*, volume 2, pages II–437, 2008.

[113] X. Sun, H. Long, and H. Wang. An automatic interpretation approach for high resolution urban remote sensing image using objects-based boosting model. In *Urban Remote Sensing Event, 2009 Joint*, pages 1–5, 2009.

[114] T. Svoboda. Image as markov random field and applications.

[115] M. Teke, E. Başeski, A. Ok, B. Yüksel, and Ç. Şenaras. Multi-spectral false color shadow detection. *Photogrammetric Image Analysis*, pages 109–119, 2011.

[116] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998.

[117] A. Torralba, K.P. Murphy, and W.T. Freeman. Sharing visual features for multiclass and multiview object detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(5):854–869, 2007.

[118] C. Unsalan. Gradient-magnitude-based support regions in structural land use classification. *Geoscience and Remote Sensing Letters, IEEE*, 3(4):546–550, 2006.

[119] C. Unsalan and K.L. Adviser-Boyer. Multispectral satellite image understanding. 2003.

[120] C. Unsalan and K.L. Boyer. A system to detect houses and residential street networks in multispectral satellite images. *Computer Vision and Image Understanding*, 98(3): 423–461, 2005.

[121] J. Van De Weijer, T. Gevers, and A.D. Bagdanov. Boosting color saliency in image feature detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(1):150–156, 2006.

[122] O. Veksler. Star shape prior for graph-cut image segmentation. *Computer Vision–ECCV 2008*, pages 454–467, 2008.

[123] S. Vicente, V. Kolmogorov, and C. Rother. Joint optimization of segmentation and appearance models. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 755–762. Ieee, 2009.

[124] Wikipedia. Satellite imagery — wikipedia, the free encyclopedia, 2012. from http://en.wikipedia.org/w/index.php?title=Satellite_imagery&oldid=493809994. [Online; accessed 27-June-2012].

[125] Wikipedia. Clique (graph theory) — wikipedia, the free encyclopedia, 2012. from http://en.wikipedia.org/w/index.php?title=Clique_(graph_theory)&oldid=499682777. [Online; accessed 2-July-2012].

[126] Wikipedia. Graph cuts in computer vision — wikipedia, the free encyclopedia, 2012. from http://en.wikipedia.org/w/index.php?title=Graph_cuts_in_computer_vision&oldid=499577835. [Online; accessed 2-July-2012].

[127] Wikipedia. Markov random field — wikipedia, the free encyclopedia, 2012. from http://en.wikipedia.org/w/index.php?title=Markov_random_field&oldid=497114678. [Online; accessed 2-July-2012].

[128] Wikipedia. Multispectral image — wikipedia, the free encyclopedia, 2012. from http://en.wikipedia.org/w/index.php?title=Multispectral_image&oldid=481601855. [Online; accessed 27-June-2012].

[129] Wikipedia. Normalized difference vegetation index — wikipedia, the free encyclopedia, 2012. from http://en.wikipedia.org/w/index.php?title=Normalized_Difference_Vegetation_Index&oldid=497563081. [Online; accessed 28-June-2012].

[130] D.H. Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[131] D. Yin, L. Hou, and X. You. A new method of automatic building detection based on multi-characteristic fusion from remote sensing images. In *Proceedings of SPIE*, volume 6044, page 60440G, 2005.

[132] B. Yuksel, C. Senaras, M. Ozay, and F.Y. Vural. Automatic building detection from satellite images using stacked generalization architecture. In *Signal Processing and Communications Applications (SIU), 2011 IEEE 19th Conference on*, pages 694–697, 2011.