AUTOMATIC DETECTION OF COMPOUND STRUCTURES BY JOINT SELECTION OF REGION GROUPS FROM MULTIPLE HIERARCHICAL SEGMENTATIONS

A DISSERTATION SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

COMPUTER ENGINEERING

By

Hüseyin Gökhan Akçay

September 2016

AUTOMATIC DETECTION OF COMPOUND STRUCTURES BY JOINT SELECTION OF REGION GROUPS FROM MULTIPLE HIERARCHICAL SEGMENTATIONS

By Hüseyin Gökhan Akçay

September 2016

We certify that we have read this dissertation and that in our opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Selim Aksoy(Advisor)

Abdullah Aydın Alatan

Çiğdem Gündüz Demir

Pınar Duygulu Şahin

Selen Pehlivan

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

AUTOMATIC DETECTION OF COMPOUND STRUCTURES BY JOINT SELECTION OF REGION GROUPS FROM MULTIPLE HIERARCHICAL SEGMENTATIONS

Hüseyin Gökhan Akçay
Ph.D. in Computer Engineering
Advisor: Selim Aksoy
September 2016

A challenging problem in remote sensing image interpretation is the detection of heterogeneous compound structures such as different types of residential, industrial, and agricultural areas that are comprised of spatial arrangements of simple primitive objects such as buildings and trees. We describe a generic method for the modeling and detection of compound structures that involve arrangements of unknown number of primitives appearing in different primitive object layers in large scenes. The modeling process starts with example structures, considers the primitive objects as random variables, builds a contextual model of their arrangements using a Markov random field, and learns the parameters of this model via sampling from the corresponding maximum entropy distribution. The detection task is reduced to the selection of multiple subsets of candidate regions from multiple hierarchical segmentations corresponding to different primitive object layers where each set of selected regions constitutes an instance of the example compound structures. The combinatorial selection problem is solved by joint sampling of groups of regions by maximizing the likelihood of their individual appearances and relative spatial arrangements under the model learned from the example structures of interest. Moreover, we incorporate linear equality and inequality constraints on the candidate regions to prevent the co-selection of redundant overlapping regions and to enforce a particular spatial layout that must be respected by the selected regions. The constrained selection problem is formulated as a linearly constrained quadratic program that is solved via a variant of the primal-dual algorithm called the Difference of Convex algorithm by rewriting the non-convex program as the difference of two convex programs. Extensive experiments using very high spatial resolution images show that the proposed method can provide good localization of unknown number of instances of different compound structures that cannot be detected by using spectral and

shape features alone.

Keywords: Object detection, spatial relationships, context modeling, Markov random field, maximum entropy distribution, Gibbs sampling, Swendsen-Wang sampling, quadratic programming, primal-dual algorithm.

ÖZET

BİLEŞİK YAPILARIN ÇOKLU SIRADÜZENSEL BÖLÜTLEMELERDEN BÖLGE GRUPLARININ ORTAKLAŞA SEÇİLMESİYLE OTOMATİK SEZİMİ

Hüseyin Gökhan Akçay
Bilgisayar Mühendisliği, Doktora
Tez Danışmanı: Selim Aksoy
Eylül 2016

Yüksek çözünürlüklü uzaktan algılama görüntülerinin analizinde önemli bir problem kendi içinde heterojen yapıların sezilmesidir. Bina, yol ve ağaç gibi *temel nesnelerin* uzamsal yerleşimlerinden oluşan farklı türlerdeki yerleşim alanları, tarım alanları, ticari ve endüstriyel alanlar *bileşik yapılar* olarak da adlandırılan bu yapılara örnek olarak verilebilir. Farklı temel nesne katmanlarından gelen bilinmeyen sayıda temel nesnelerin yerlesimlerini içeren bileşik yapıların modellenmesi ve büyük görüntülerde otomatik sezimi için genel bir yöntem sunmaktayız. Verilen örnek bileşik yapılardaki bölgeler olasılıksal değişkenler olarak temsil edilmekte, bir Markov rasgele alanı ile bu bölgelerin uzamsal yerleşim modeli oluşturulmakta ve bu modele ait parametre kümesi ilgili maksimum entropi olasılık dağılımından örneklenerek öğrenilmektedir. Benzer bölge gruplarının sezimi, farklı temel nesne katmanlarını temsil eden çoklu bölge sıradüzenlerinden gelen çoklu aday nesneler arasından sorgun modelini enyükselten altkümelerin seçilmesine indirgenmektedir. Kombinatoryal seçme problemi, bölge gruplarının ilgi duyulan yapı örneklerinden öğrenilen model altında bireysel görünüşleri ve bağıl uzamsal yerleşimleri enyükseltilerek ortaklaşa örneklenmesi ile çözülmektedir. Bundan başka, birbiriyle örtüşen fazlalık bölgelerin beraberce seçilmesini engellemek ve seçilen bölgeleri belirli bir uzamsal yerleşimin sağlanmasına zorlamak amacıyla aday bölgeler üzerine içbükey eşitlik ve eşitsizlik kısıtları koymaktayız. Kısıtlı seçme problemi, lineer kısıtlı karesel program olarak formule edilmekte ve dışbükey olmayan bu program iki dışbükey programın farkı olarak yeniden yazılarak ilkin-eşlek (primal-dual) algoritmasının bir çeşidi olan Dışbükeylerin Farkı algoritması ile çözülmektedir. Çok yüksek çözünürlükteki görüntüler kullanılarak yapılan kapsamlı deneyler, görüntülerdeki sadece spektral ve şekilsel öznitelikler kullanılarak elde edilemeyen bilinmeyen

sayıdaki farklı bileşik yapıların yerlerinin önerilen yöntemle doğru bir şekilde saptandığını göstermektedir.

Anahtar sözcükler: Nesne sezimi, uzamsal yerleşimler, içerik modelleme, Markov rasgele alanı, maksimum entropi dağılımı, Gibbs örnekleme, Swendsen-Wang örnekleme, karesel programlama, ilkin-eşsel algoritması.

Acknowledgement

First and foremost, I would like to record my sincere gratitude to my supervisor, Assoc. Prof. Dr. Selim Aksoy for his invaluable guidance, motivation and support from the very early stage of this study.

I would also like to thank Prof. Dr. A. Aydın Alatan, Assoc. Prof. Dr. Çiğdem Gündüz Demir, Assoc. Prof. Dr. Pınar Duygulu Şahin and Assist. Prof. Dr. Selen Pehlivan for their insightful comments and encouragement.

My friends Mesut Göksu, Çağlar Arı, Murat Ak, Emre Varol, Sermetcan Baysal, Burak Tokcan, Sıtar Kortik, Çağrı Ergül, Kamil Kahveci, Aykut Kocaoğlu, Aykut Kuşdemir, Kıvanç Yaman and Alp Yüce helped me whenever I stucked with their warm welcome.

Last but not the least, I am grateful in my very heart to my beloved family, my merciful mother, decent father and unique sister for being always with me throughout my life.

This work was supported in part by TÜBİTAK (The Scientific and Technological Research Council of Turkey) Grant 109E193 and the GEBİP Award from the Turkish Academy of Sciences.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Problem Definition 2

1.3 Contributions 4

1.4 Outline 5

2 Literature Review **6**

3 Compound Structure Model **10**

3.1 Primitive Representation 10

3.2 Spatial Arrangement Model 11

3.3 Probabilistic Region Processes 14

3.4 Dynamic Topology of Probabilistic Region Processes 17

4 Learning **19**

4.1 Maximum Likelihood Estimation 19

4.2 Sampling Region Processes 20

5 Inference and Region Selection **26**

5.1 Hierarchical Region Extraction 26

5.2 Inference without Constraints 27

5.2.1 Bayesian Formulation 28

5.2.2 CRF Formulation 30

5.2.3 CRF Inference 31

5.3 Inference with Constraints 37

5.3.1 Quadratic Programming Formulation 38

5.3.2 Difference of Convex Programming Formulation 39
5.3.3 Difference of Convex Programming Inference 40

6 Experiments **43**
6.1 Evaluation Criteria 43
6.2 Unconstrained Single-layer Experiments 44
6.2.1 Data Set 45
6.2.2 Experimental Protocol 48
6.2.3 Baselines for Comparison 51
6.2.4 Results 53
6.3 Unconstrained Multi-layer Experiments 73
6.4 Constrained Single-layer Experiments 83
6.4.1 Problem Definition 83
6.4.2 Results 85
6.5 Constrained Multi-layer Experiments 95
6.5.1 Problem Definition 95
6.5.2 Results 97
6.6 Discussion 104

7 Summary and Conclusions **107**

List of Figures

1.1 Examples of compound structures in WorldView-2 images. Each 150×150 pixel window includes one or more examples for residential, industrial, and agricultural structures composed of various spatial arrangements of primitives (buildings and trees) with different color and shape characteristics. 3

1.2 Object/process diagram of the proposed approach. Rectangles represent objects and rounded rectangles represent processes. The details of all steps are presented in the following chapters. 5

3.1 Neighborhood graph for a single primitive layer. (a) RGB image. (b) Primitive objects (blue ellipses) and the edges (green lines) representing the neighbors of one primitive. (c) The graph for all primitives. 12

3.2 Neighborhood graph for multiple primitive layers. (a) RGB image. (b) Primitive objects from three different layers: buildings (red), vegetation (green), pool (blue). (c) Graph vertices (blue ellipses) and the edges that connect the primitives in the same layer (red edges for buildings and green edges for vegetation) and between different layers (yellow edges). 12

3.3 Pairwise feature examples. $\phi^1, \phi^2, \phi^3, \phi^4$ are described in the text. 14

3.4 Example histograms for the building layers of four different types of compound structures. Two examples are shown for each compound structure. Each histogram is obtained through the concatenation of the marginal histograms for the features ϕ_6, ϕ_7, ϕ_8, ϕ_1, ϕ_2, ϕ_3, ϕ_4, ϕ_5 in that order. These feature histograms are separated by dashed vertical green lines. 15

4.1 An example iteration for updating β corresponding to the relative orientation histogram bins. (1) We begin with an average histogram of the observed region processes and a β of all 0's. The histogram shows two peaks in orientations close to 0 and 90 degrees which means that the primitives are mostly parallel or perpendicular to each other. (2) Sample primitives are drawn from the current model (i.e., uniform distribution). (3) Average relative orientation histograms are computed from these samples. The sampled histogram is shown in dark gray on top of the observed histogram to show the difference between them. (4) β is re-weighted according to this difference. (5) Newly sampled primitives according to this β appear oriented more consistently with the initial observed histogram. 21

4.2 Illustration of the Gibbs sampler in Algorithm 2 for a single primitive layer. (a) The compound structure V given as input to stochastic gradient ascent in Algorithm 1. (b)-(f) Samples $\tilde{V}^{(t)}$ at iterations $t = 0, 50, 200, 500, 1000$ in Algorithm 2. 23

4.3 Illustration of the Gibbs sampler in Algorithm 2 for two primitive layers. (a) The compound structure V given as input to stochastic gradient ascent in Algorithm 1. Different primitive layers are colored with blue and red, respectively. (b)-(f) Samples $\tilde{V}^{(t)}$ at iterations $t = 0, 50, 250, 600, 1000$ in Algorithm 2. 25

5.1 Graph construction for a single primitive layer. The hierarchical candidate regions (V) at three levels are shown in gray. (a) The edges that represent parent-child relationship are shown in red. (b) The edges E that represent the within-level and between-level neighbor relationship are shown in blue. For clarity, we do not show the edges between two levels that are not consecutive even though there are edges between all level pairs. 28

5.2 Graph construction for two primitive layers (i.e., building and pool). The hierarchical candidate regions at three and two levels for these layers are shown in red and light blue, respectively. (a) The edges that represent parent-child relationship for both layers are shown. (b) The edges that represent the within- and between-level neighbor relationship within the same layer are shown. (c) The edges that represent the neighbor relationship between the layers are shown. For better visualization of edges, only 20 percent of all between-layer edges are shown. 29

5.3 Illustration of the Swendsen-Wang procedure in Algorithm 3. In each figure, the labels of the primitives are shown in red for selected ($z_i = +1$) and blue for not selected ($z_i = -1$). (a) The labels at the beginning of a particular sampling iteration. The Voronoi edges (E) are shown in green. (b) The edges with positive bond probabilities as candidates for forming connected components of their corresponding vertices. (c) The sampled edges that form connected components of vertices bonded together. (d) The result of randomly flipping the labels of the primitives in some of these components. A single scale is shown for simplicity even though the algorithm normally runs on the graph for the whole candidate region hierarchy for all layers. 36

6.1 Ankara data set used for quantitative evaluation. 45

6.2 (a) Manually delineated polygons reflecting compound structures of interest. (b) Manually delineated buildings inside these polygons. These buildings are used as the primitives in the validation data. The colors of the polygons and buildings correspond to the scenarios given in Table 6.1. 47
6.3 Kusadasi data set used for qualitative evaluation. 49
6.4 Darfur data set used for qualitative evaluation. 50
6.5 Candidate regions obtained by the morphological profile hierarchy. Regions appearing in different levels of the hierarchy are shown with different pseudocolors. 52
6.6 Marginal probabilities for the selected regions for the first and second scenarios. Brighter values indicate higher probabilities. The example primitives are also shown. 55
6.7 Marginal probabilities for the selected regions for the third and fourth scenarios. Brighter values indicate higher probabilities. The example primitives are also shown. 56
6.8 Marginal probabilities for the selected regions for the fifth and sixth scenarios. Brighter values indicate higher probabilities. The example primitives are also shown. 57
6.9 Precision-recall curves for the first three scenarios. The columns correspond to the scenarios one to three from left to right. The top row corresponds to the pixel-based evaluation and the bottom row is for the object-based evaluation. The solid red curves correspond to the proposed approach, dashed green ones are for the first baseline (shape and arrangement without selection), and dashed blue ones are for the second baseline (color and shape-only selection with no arrangement). The solid cyan curves will be explained in Section 6.4. 58

6.10 Precision-recall curves for the last three scenarios. The columns correspond to the scenarios four to six from left to right. The top row corresponds to the pixel-based evaluation and the bottom row is for the object-based evaluation. The solid red curves correspond to the proposed approach, dashed green ones are for the first baseline (shape and arrangement without selection), and dashed blue ones are for the second baseline (color and shape-only selection with no arrangement). The solid cyan curves will be explained in Section 6.4. 59

6.11 Zoomed detection examples. The first column shows the 500×500 sub-scenes. The second column shows the two-level hierarchical candidate regions for the first five rows and six-level hierarchy from left to right and bottom to top for the last row. The selected regions are colored with red. The third column shows the marginal probabilities at the end of selection. The fourth column shows the thresholded detections overlayed as red and the validation polygons overlayed with the corresponding colors in Table 6.1. 61

6.12 Zoomed detection examples. The image pairs show the marginal probabilities and the overlayed detection results. The first and last pairs in each row show the results of our algorithm and the second baseline, respectively. 62

6.13 Zoomed detection examples. The image pairs show the marginal probabilities and the overlayed detection results. The first and last pairs in each row show the results of our algorithm and the second baseline, respectively. 63

6.14 Zoomed detection examples. The image pairs show the marginal probabilities and the overlayed detection results of our algorithm. Each row corresponds to a particular scenario. 64

6.15 (a) Candidate regions for orchard detection obtained by the morphological profile hierarchy. (b) Marginal probabilities for the selected regions. The example primitive mask is also shown. 67

6.16 Example results for the detection of orchards as agricultural structures in three Kusadasi subimages on the left column. The right column shows the corresponding marginal probabilities of the selected regions (emphasized with the copper colormap) as well as the discarded input candidate regions (shown in white). 68
6.17 Example results for the detection of orchards. The left column shows the marginal probabilities at the end of selection. The right column shows the thresholded detections overlaid as red. We used a 21×21 pixel Gaussian smoothing filter to enhance the binary detection results before overlaying. 69
6.18 Example results for the detection of orchards. The left column shows the marginal probabilities at the end of selection. The right column shows the thresholded detections overlayed as red. We used a 21×21 pixel Gaussian smoothing filter to enhance the binary detection results before overlaying. 70
6.19 Example results for the detection of refugee camps as rural structures. The top image shows the marginal probabilities as well as the example primitives used for learning on the bottom left corner. The bottom image shows the thresholded detections overlayed as red. We used dilation with a disk with radius of 3 pixels to enhance the line segments for display. 72
6.20 (a) Candidate regions for the detection of housing estates. (b) Marginal probabilities for the regions selected from the building and pool layers. The marginals for the selected pool regions are depicted with a blue colormap. The example primitive mask is also shown. 76
6.21 (a) Test site. (b) Marginal probabilities of the selected buildings and pools. (c) Manually delineated polygons reflecting compound structures of interest. (d) Manually delineated buildings inside these polygons. These buildings are used as the primitives in the validation data. 77
6.22 Examples of local details of red building rooftops. 78

6.23 Precision-recall curves for housing estate detection. The left plot corresponds to the pixel-based evaluation and the right plot is for the object-based evaluation. The solid red curves correspond to the multi-layer approach, and dashed green ones are for the single-layer approach. 78
6.24 Samples obtained by the Swendsen-Wang procedure ran on single and multiple layers. (a) RGB image containing a housing estate with a pool. (b) and (c) show the sampled edges and primitive labels at iterations $1, 10$ and 20 for a single layer (i.e., building) and two layers (i.e., building and pool), respectively. The labels of the primitives are shown in red for selected and blue for not selected. Notice that in the bottom middle sample, a single edge sampled from the pool candidate to a building candidate was able to turn the color of the corresponding connected component including many buildings to red. 80
6.25 Example results for the detection of housing estates. (a) RGB image. (b) The thresholded selected regions, using only the building layer, overlayed as yellow. (c) The marginal probabilities obtained by selection with building and pool layers. (d) The selected regions after thresholding. A newly detected housing estate that was missed with single layer selection is enclosed by a red convex hull. 81
6.26 Example results for the detection of housing estates. (a) RGB image. (b) The thresholded selected regions, using only the building layer, overlayed as yellow. (c) The marginal probabilities obtained by selection with building and pool layers. (d) The selected regions after thresholding. A newly detected housing estate that was missed with single layer selection is enclosed by a red convex hull. 82
6.27 Zoomed detection examples. The image pairs show the marginal probabilities and the overlayed detections after thresholding these probabilities. 84
6.28 An example hierarchy 85

6.29 Conditional probabilities for the selected regions for the first and second scenarios. Brighter values indicate higher probabilities. The example primitives are also shown. 87
6.30 Conditional probabilities for the selected regions for the third and fourth scenarios. Brighter values indicate higher probabilities. The example primitives are also shown. 88
6.31 Conditional probabilities for the selected regions for the fifth and sixth scenarios. Brighter values indicate higher probabilities. The example primitives are also shown. 89
6.32 Zoomed detection examples. The selected regions are colored with red in (d). For three selected regions, all the paths involving them are drawn. 90
6.33 Zoomed detection examples. The selected regions are colored with red in (d). For three selected regions, all the paths involving them are drawn. 91
6.34 Zoomed detection examples. The selected regions are colored with red in (d). For three selected regions, all the paths involving them are drawn. 92
6.35 Zoomed detection examples. The selected regions are colored with red in (d). For three selected regions, all the paths involving them are drawn. 93
6.36 Zoomed detection examples. The selected regions are colored with red in (d). For three selected regions, all the paths involving them are drawn. 94
6.37 Zoomed detection examples. The selected regions are colored with red in (d). For two selected regions, all the paths involving them are drawn. 95
6.38 Selected regions for the green areas surrounded by building sites scenario. Brighter values indicate higher objective values. The example primitives are also shown on the top left corner. 98

6.39 A zoomed detection example for a 300×300 sub-scene in (a). k_1 is set to 4. (b) shows the two-level hierarchy of building candidate regions. (c) shows single-level green area candidate regions. The selected regions are colored with red. (d) shows the X^* values at the end of selection ($fVal = -2.97$). Selected buildings and green area are colored by the copper and the pink colormap, respectively. Brighter colors indicate higher values. (e) shows the thresholded detections overlayed as yellow (buildings) and green (green area). 99

6.40 A zoomed detection example for a 300×300 sub-scene in (a). k_1 is set to 6. (b) shows the two-level hierarchy of building candidate regions. (c) shows single-level green area candidate regions. The selected regions are colored with red. (d) shows the X^* values at the end of selection ($fVal = -3.01$). Selected buildings and green area are colored by the copper and the pink colormap, respectively. Brighter colors indicate higher values. (e) shows the thresholded detections overlayed as yellow (buildings) and green (green area). 100

6.41 A zoomed detection example for a 300×300 sub-scene in (a). k_1 is set to 6. (b) shows the two-level hierarchy of building candidate regions. (c) shows single-level green area candidate regions. The selected regions are colored with red. (d) shows the X^* values at the end of selection ($fVal = -3.05$). Selected buildings and green area are colored by the copper and the pink colormap, respectively. Brighter colors indicate higher values. (e) shows the thresholded detections overlayed as yellow (buildings) and green (green area). 101

6.42 Zoomed detection examples for different values of $k_1 = 4, 6, 8$. . . 102

6.43 Zoomed detection examples for different values of $k_1 = 4, 6, 8$. . . 103

6.44 Zoomed detection examples for different values of $k_1 = 4, 6, 8$. . . 104

List of Tables

6.1 Detection scenarios for the experiments. Example primitives used for learning the compound structure model for each scenario are shown in a different color. The number of polygons and buildings in the validation data are also given. 46
6.2 The number of candidate and detected regions for single and multi-layer selection scenarios. 75

Chapter 1

Introduction

1.1 Motivation

Remote sensing imagery provide large scale global content about the Earth as well as small local details upto a 30 cm resolution. Nearly 1.6 terabytes of data are being sent to the Earth every day by ESA's multi-spectral high-resolution imaging satellite for land monitoring. The WorldView-2 satellite as the mostly used image source in our experiments is capable of collecting up to 975,000 square kilometers of imagery per day. This amount of available geospatial data sets have the potential to serve as data sources for many applications involving topics such as environment, forestry, agriculture, crisis management, security, industry, energy, natural resource management, urban planning and mapping. Furthermore, many scientific remote sensing applications are becoming more feasible by the increasing power of the Internet, computing skills and technical infrastructures such as storage systems. This made remotely sensed imagery an interesting tool for scientists, governmental agencies and the general public to understand the world and its surrounding environment in a wider perspective. Advances in satellite technology also increased the amount of generated data which also brought the need for its effective processing and storage. The urgent need for the extraction of useful information from these raw data necessitated automatic or semi-automatic

analysis of such images. Hence, automatic content extraction and efficient access to this content have become very important for developing intelligent systems processing these images.

The improving spatial and spectral resolution of the images acquired from new-generation satellites have increased the capability to capture additional details about the objects of interest, and have permitted new applications that rely on effective identification of these objects. A common approach to object-based image classification and object recognition is to assume the existence of homogeneous regions that can be modeled with spectral or shape features alone. However, as the spatial resolution increases, such homogeneous regions often correspond to very small details. Consequently, a new requirement for semantic image understanding has become the modeling and identification of image regions that are intrinsically heterogeneous. Examples of such regions, also called *compound structures*, include different types of residential, industrial, and agricultural areas that are comprised of spatial arrangements of simple primitive objects such as buildings and trees [1–3] as shown in Figure 1.1. However, detection of these structures is a challenging problem because there is no single color, shape, or texture feature that can effectively model their appearances. Mostly, the assumptions and models for objects (e.g., sideways, indoor furnitures, animals) in the computer vision literature are not well suited to high resolution remote sensing images. Because, these images contain a large amount of information due to thousands of individual objects that mostly do not have distinctive features and can arrange in many different combinations in the overhead view. Hence, developing new representations, learning these representations, and making inferences for compound structure detection in the new images are still open problems.

1.2 Problem Definition

In this dissertation, we propose a generic method for the modeling and detection of compound structures that can involve the arrangements of an unknown

Figure 1.1: Examples of compound structures in WorldView-2 images. Each 150×150 pixel window includes one or more examples for residential, industrial, and agricultural structures composed of various spatial arrangements of primitives (buildings and trees) with different color and shape characteristics.

number of primitive objects coming from different primitive layers. The procedure starts with example compound structures each containing primitive objects that are used to estimate a probabilistic appearance and arrangement model. The modeling process considers the primitive objects as random variables in a Markov random field (MRF) where potentially related objects are connected. MRFs have been used in the literature to model contextual information in neighborhoods of pixels [4] or regions [5, 6]. Our aim is to learn a flexible arrangement model with a small number of examples that can distinguish between different types of compound structures inside a large scene instead of dedicating the MRF to model the whole scene with only a limited set of relationships. The parameters of the proposed MRF model are learned via sampling from the corresponding maximum entropy distribution.

The detection task is formulated as the selection of multiple coherent subsets of candidate regions obtained from hierarchical segmentations where each set of selected regions, when grouped together, constitutes an instance of the example compound structure. This differs from the earlier work [3] that did not need an initial segmentation of the primitives but required that their number is given a

priori. The proposed selection algorithm models the spatial relationships among the candidate regions by using the multi-scale neighborhood graph created by connecting different multi-scale graphs. The first version of our algorithm uses a sampling procedure to maximize the likelihood of groups of regions where the decision of selecting or not selecting regions is done jointly as groups instead of individual decisions. The second version solves a constrained quadratic programming problem which enables adding multi-variable convex constraints to the unconstrained formulation. These constraints handle redundant detections caused by multi-scale overlapping regions and enforce a particular spatial configuration defined by multiple regions. In most cases, our algorithm does not have any a priori knowledge of the number of regions to be selected. It also enables the detection of regions that cannot be detected by using spectral and shape features alone, thanks to the contextual information that the model captures.

1.3 Contributions

Our major contributions are fourfold.

- First, we describe a model for the individual appearance properties of primitive objects of different types as well as their spatial arrangements within compound structures.
- Second, we propose a solution to the combinatorial region selection problem for detecting and localizing an unknown number of instances of a given compound structure in a large scene.
- Third, to avoid over- or under-segmentation of candidate regions, we seamlessly integrate multi-scale information, and search for the most meaningful regions appearing at different scales of multiple hierarchical segmentations.
- Fourth, we propose a constrained region selection model framework which allows us to specify global constraints on the selected regions.

  Parts of this dissertation was published as [7].

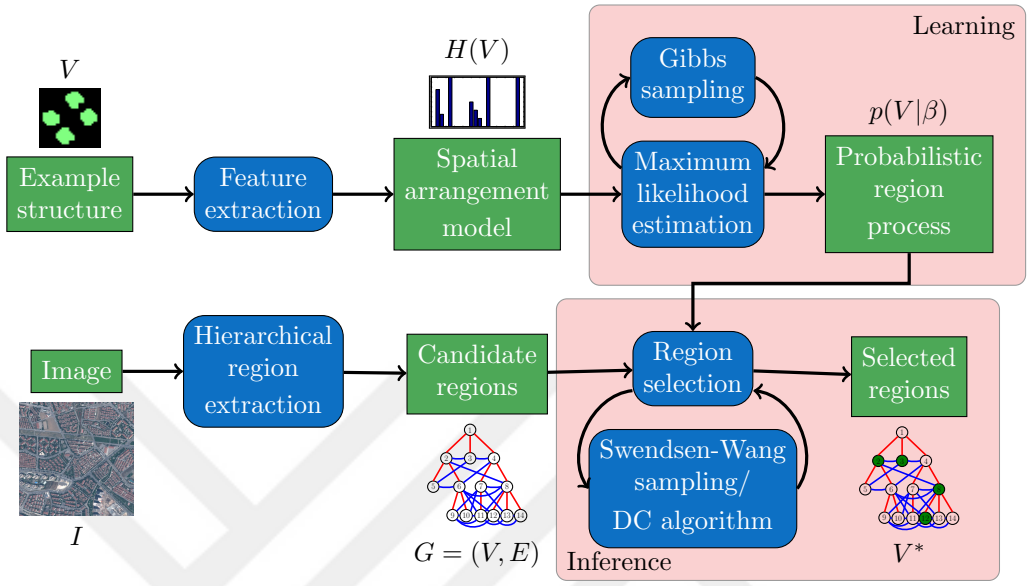


Figure 1.2: Object/process diagram of the proposed approach. Rectangles represent objects and rounded rectangles represent processes. The details of all steps are presented in the following chapters.

1.4 Outline

An overview of the proposed approach is shown in Figure 1.2. The rest of the dissertation is organized as follows. Chapter 3 introduces the representation for primitive objects and the probabilistic model for their spatial arrangement and shape characteristics. Chapter 4 describes the learning algorithm for the estimation of the parameters in the proposed model. Chapter 5 describes the selection algorithms for finding the structures with similar arrangements among a set of candidate regions. Chapter 6 presents experimental results, followed by conclusions in Section 7.

Chapter 2

Literature Review

State-of-the-art algorithms in remote sensing have been highly interested in identifying various geospatial objects such as residential buildings [8], factory storage buildings [8], highways [8], local roads [8, 9], runways [9], vehicles [8, 10], airplanes [8, 10, 11] and boats [11]. However, these algorithms did not explicitly model spatial arrangements of simpler primitive objects but rather the corresponding target objects can be used as primitives in more complex compound structures.

One of the most common alternatives for detecting compound structures is to use a window-based approach where the image is divided into tiles and these tiles are classified according to their features. The bag-of-words (BoW) model has been popular in recent years for modeling the tile content. First, visual words are formed by quantizing local features. Then, each tile is described by the frequency of these words, and is classified [12–15] or retrieved [10,16–18]. The main problem in the BoW representation is that it does not consider spatial arrangements that can be very crucial for many types of compound structures. In other words, BoW is a first-order-model that primitives contribute independently of their position and independently from each other.

In an attempt to exploit spatial information, Vaduva *et al.* [19] modeled relative positions between objects by extracting object pair signatures as words that characterize the tiles. However, the tile-based modeling still enforces artificial boundaries on the image. Segmentation algorithms can produce flexible boundaries and promise to be adaptive to the image content. For example, Kurtz *et al.* [20] extracted heterogeneous objects in multiple levels of details where the segmentation in the high resolution image was provided by clustering the segmentation in a lower resolution image. Gaetano *et al.* [21] performed hierarchical texture segmentation by iteratively merging neighboring homogeneous regions that had frequently co-occurring region types. In both approaches, segmentation still relies on interactions between neighboring pixels and may get stuck at boundaries corresponding to local details that do not necessarily correspond to compound structures. Certain segments in certain scales may correspond to compound structures, but the grouping criteria still do not involve spatial arrangements, and hence, may fail in detecting and delineating many other structures.

Another problem with tile-based modeling is the assumption that the whole window corresponds to a compound structure where feature extraction is performed holistically. To identify structure-sensitive neighborhoods, Vanegas *et al.* [22] proposed a graph-based method to determine aligned groups of objects from a given segmentation. However, this method was designed for specific arrangements such as alignment and parallelism. It also worked in a single scale and was sensitive to segmentation errors. The use of multiple partitionings of the image via segmentation hierarchies has been identified as an important problem in remote sensing. However, it is mainly addressed as the problem of selecting individual regions from a set of candidates [23–28] with no consideration of the contextual interactions between neighboring regions.

Graphs have been a popular representation for representing high-level structures in images. In a very recent study [29], relationships between the adjacent regions on the same multi-scale segmentation level as well as the regions across different levels were utilized in a normalized cut framework aiming at a globally optimized segmentation of the whole image. However, the similarity measures characterizing these relationships still depend on low-level spectral, texture, and

SIFT features.

Inglada and Michel [30] build a graph-based description of the relationships between multi-scale regions that are assumed to correspond to the parts of a complex object (e.g. plane) and then, use graph-matching to detect the instances an object of interest. However, the relationships are more appropriate for finding complex objects rather than compound structures, i.e., do not model complex arrangements such as proximity, orientation, alignment. Also, the structural graph matching is limited to the number of parts composing the complex object of interest preventing the detection of relatively more complex combination of objects such as urban areas composed of many primitive objects with a particular spatial arrangement distribution.

Recently, Vanegas *et al.* [31] proposed conceptual graphs for encoding the spatial relationships, such as near, far, between, parallel and surrounds, between objects or between object groups. Detection of complex objects in the images are formalized as a graph homomorphism problem between a model graph representing the fuzzy object relationships and the graph of candidate regions. The graph isomorphism was carried out by solving a fuzzy constraint satisfaction problem in which the retrieved group of candidate regions is strictly connected. The results on airports or harbors did not show any case where two separate complex objects exist at the same time in an image. This model also requires a detailed manually constructed model graph describing which relations should be present between which objects. Finally, constraints that are not satisfied by two objects inhibit their concurrent detections even though they are perfectly laid out with the other objects. In our approach, such repelling objects can be detected together since we look at the general distribution of the primitives' arrangements instead of looking at each constraint separately.

Arı and Aksoy [3] presented a Gaussian mixture model that assumes the individual Gaussian components as primitive objects of a compound structure, and detects these objects by an expectation-maximization algorithm that can consider spectral and spatial constraints. This method requires a one-to-one correspondence between the model Gaussian components and the detected Gaussian

components. So, the number of primitive objects to be detected should be given apriori. The authors find multiple instances of a compound structure by initializing the algorithm multiple times at different locations of the image.

Chapter 3

Compound Structure Model

Compound structures arise from local interactions between primitive objects as well as their individual properties. The set of factors that make the individual primitives members of a compound structure can be motivated by the Gestalt rules that attempt to model the perceptual grouping process in the human vision system. In the following, we present the representation for the primitives, propose a generic spatial arrangement model for grouping these primitives according to semantic cues such as proximity, continuity, parallelism, alignment, etc., and describe a statistical model that encodes the spatial arrangement properties of these groupings into a probabilistic region process.

3.1 Primitive Representation

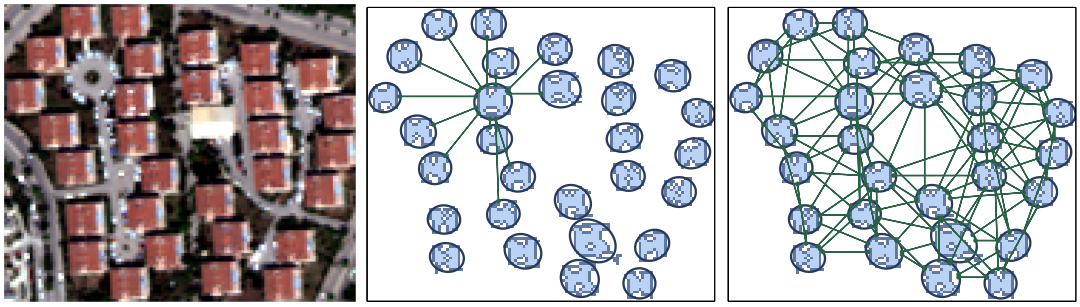
In this dissertation, compound structures are defined as high-level heterogeneous objects that are composed of spatial arrangements of multiple, relatively homogeneous, and compact primitive objects. The set of primitives includes objects that can be relatively easily extracted using low-level operations that exploit spectral, textural, or morphological information. These objects, such as buildings and trees, can be used as building blocks of more complex structures. In

particular, we assume that a compound structure V consists of R layers of primitive object maps $V = \bigcup_{r=1,\ldots,R} V^r$. Each primitive object $v_i \in V$ is represented by an ellipse $v_i = (l_i, s_i, \theta_i)$ where $l_i = (l_i^x, l_i^y) \in [0, X_{max} - 1] \times [0, Y_{max} - 1]$ represents the ellipse's center location, $s_i = (s_i^h, s_i^w) \in [s_{min}^h, s_{max}^h] \times [s_{min}^w, s_{max}^w]$ contains the ellipse's major and minor axis lengths, respectively, and $\theta_i \in [0, \pi)$ is the orientation measured as the angle between the major axis of the ellipse and the horizontal image axis. Here, X_{max} and Y_{max} are the width and height of the image, respectively, and (s_{min}^h, s_{max}^h) and (s_{min}^w, s_{max}^w) are the minimum and maximum major and minor axis lengths, respectively.

Ellipses have often been used as the image primitives in perceptual organization [32] and object recognition [33] tasks in the computer vision literature, and the underlying assumption that the primitives have relatively compact shapes also holds for many objects of interest in remotely sensed scenes. Ellipses provide simple but sufficiently flexible approximations that can model the most fundamental object characteristics like location, scale, and orientation, and can generalize to other shapes such as circles, rectangles, and line segments with additional constraints on specific parameters. The following sections show that they also enable effective and efficient feature extraction and model estimation steps.

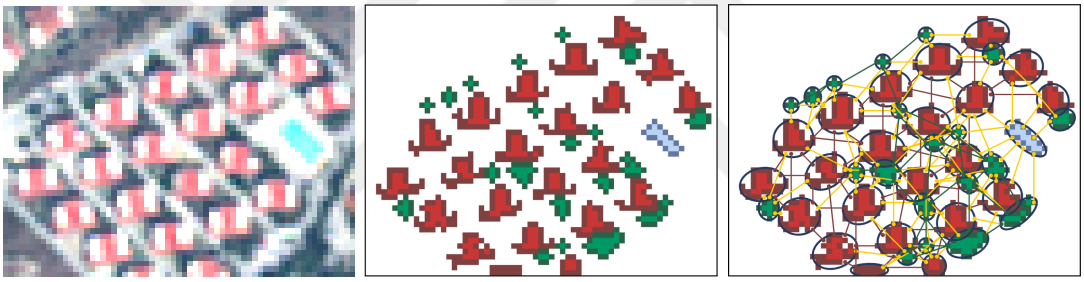
3.2 Spatial Arrangement Model

For a given compound structure consisting of N primitive objects, we construct a neighborhood graph $G = (V, E)$ where the vertices $V = \{v_1, \ldots, v_N\}$ correspond to the individual primitive objects, and the edges $E = \bigcup_{r_1, r_2 = 1,\ldots,R} E^{r_1 r_2}$ model their spatial relationships where $E^{r_1 r_2}$ denotes the edges between the vertices at layers V^{r_1} and V^{r_2} (Figures 3.1 and 3.2). Note that, when $r_1 = r_2$, $E^{r_1 r_2}$ represents the edges between the vertices at the same layer and $E^{r_1 r_2} = E^{r_2 r_1}$. The neighborhood information is obtained by proximity analysis where a threshold on the distance between the closest pixels of each object pair is used to determine the neighbors. In particular, let P_i denote the set of pixels inside the ellipse v_i. Then, $(v_i, v_j) \in E$ if and only if the distance between the closest pixels of v_i and



(a) (b) (c)

Figure 3.1: Neighborhood graph for a single primitive layer. (a) RGB image. (b) Primitive objects (blue ellipses) and the edges (green lines) representing the neighbors of one primitive. (c) The graph for all primitives.



(a) (b) (c)

Figure 3.2: Neighborhood graph for multiple primitive layers. (a) RGB image. (b) Primitive objects from three different layers: buildings (red), vegetation (green), pool (blue). (c) Graph vertices (blue ellipses) and the edges that connect the primitives in the same layer (red edges for buildings and green edges for vegetation) and between different layers (yellow edges).

v_j is less than a proximity threshold δ, i.e., $E = \{(v_i, v_j) \in V \times V : \exists (p_i, p_j) \in P_i \times P_j$ such that $\forall (p'_i, p'_j) \in P_i \times P_j, d(p_i, p_j) \leq d(p'_i, p'_j)$ and $d(p_i, p_j) \leq \delta\}$ where $d(p_i, p_j)$ denotes the Euclidean distance between two pixels p_i and p_j.

For each neighboring primitive object pair $(v_i, v_j) \in E$, we compute the following five features (Figure 3.3):

- Distance between the closest pixels, $\phi_1(v_i, v_j) = \min_{p_i \in P_i, p_j \in P_j} d(p_i, p_j)$,
- Relative orientation, $\phi_2(v_i, v_j) = \min\{|\theta_i - \theta_j|, 180 - |\theta_i - \theta_j|\}$ where $|\cdot|$ denotes the absolute value operator,

- Angle between the line joining the centroids of the two objects and the major axis of v_i as the reference object, $\phi_3(v_i, v_j) = \min\{|\alpha_{ij} - \theta_i|, 180 - |\alpha_{ij} - \theta_i|\}$ where α_{ij} is the angle of the line segment connecting the centroids of v_i and v_j,
- Distance between the closest antipodal pixels that lie on the major axes, $\phi_4(v_i, v_j) = \min_{p_i \in P_i^a, p_j \in P_j^a} d(p_i, p_j)$ where P_i^a denotes the two antipodal pixels on the major axis of v_i,
- Relative size, $\phi_5(v_i, v_j) = \frac{\min\{|P_i|, |P_j|\}}{\max\{|P_i|, |P_j|\}}$ where $|P_i|$ denotes the number of pixels inside P_i.

These features capture various Gestalt properties such as proximity, parallelism, directional continuity, proximal continuity, and similarity, respectively. Furthermore, ϕ_2 and ϕ_3 together measure how much the two objects are aligned. In addition to the pairwise features, we also compute the following four individual features for each primitive object v_i:

- Area, $\phi_6(v_i) = \pi(s_i^h/2)(s_i^w/2)$,
- Eccentricity, $\phi_7(v_i) = \sqrt{1 - (s_i^w/s_i^h)^2}$,
- Solidity, $\phi_8(v_i) = \frac{|\tilde{P}_i|}{|P_i|}$ where $\tilde{P}_i$ denotes the pixels inside the real region represented by the ellipse v_i, and $|\tilde{P}_i|$ denotes the number of pixels inside $\tilde{P}_i$,
- Regularity, $\phi_9(v_i) = \Vert_{r_2=1}^{R} h_9^{r_2}(v_i)$ where $\Vert$ is the concatenation operator, and $h_9^{r_2}(v_i)$ is the histogram of relative angles, i.e., $\min\{|\alpha_{ij} - \alpha_{ik}|, 180 - |\alpha_{ij} - \alpha_{ik}|\}$, between all pairs of edges $(v_i, v_j) \in E^{r_1 r_2}$ and $(v_i, v_k) \in E^{r_1 r_2}$ where r_1 is the index of the layer that v_i belongs to.

Let $H_k^{r_1 r_2}$ be a function that constructs a histogram of its input values whose bins are designed according to the feature ϕ_k and the pair of layer indices r_1 and r_2. Then, given the set of primitives V and the corresponding features,

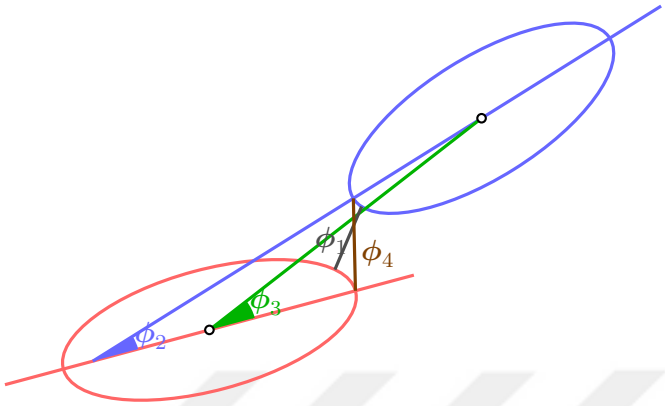


Figure 3.3: Pairwise feature examples. $\phi^1, \phi^2, \phi^3, \phi^4$ are described in the text.

a one-dimensional marginal histogram $H_k^{r_1 r_2}(E^{r_1 r_2})$ is constructed for each feature $\phi_k, k = 1, \ldots, 5$, computed over all edges for each pair of layers V^{r_1} and V^{r_2} where $E^{r_1 r_2}$ is assumed to be deterministically computed from V^{r_1} and V^{r_2}. Also, a one-dimensional marginal histogram $H_k^r(V^r)$ is constructed for each feature $\phi_k, k = 6, \ldots, 8$, computed over all vertices at each layer V^r. Since ϕ_9 is already a histogram (i.e., a concatenation of histograms) for an input vertex, $H_9^r(V^r)$ is the vector addition of histograms $\phi_9(v_i), v_i \in V^r$. Thus, a total of $5 \times \left(\binom{R}{2} + R\right)$ pairwise feature histograms and $4 \times R$ individual feature histograms are computed. The number of bins for each single marginal histogram is determined by the corresponding feature and the layers involved. The concatenation $H(V)$ of all marginal histograms $H_k^{r_1 r_2}(E^{r_1 r_2}), k = 1, \ldots, 5, r_1, r_2 = 1, \ldots, R$, and $H_k^r(V^r), k = 6, \ldots, 9, r = 1, \ldots, R$, is used as a non-parametric approximation to the distribution of the feature values of the primitive objects in the compound structure. The vector length $|H(V)|$ is the total number of bins in all marginal histograms. Visual inspection of the marginal histograms in Figure 3.4 reveals a clear discrimination of each type of compound structure from each other.

3.3 Probabilistic Region Processes

The diversity of the patterns in different scenes and the richness of details in each scene entails the use of statistical approaches. In our model, each primitive object

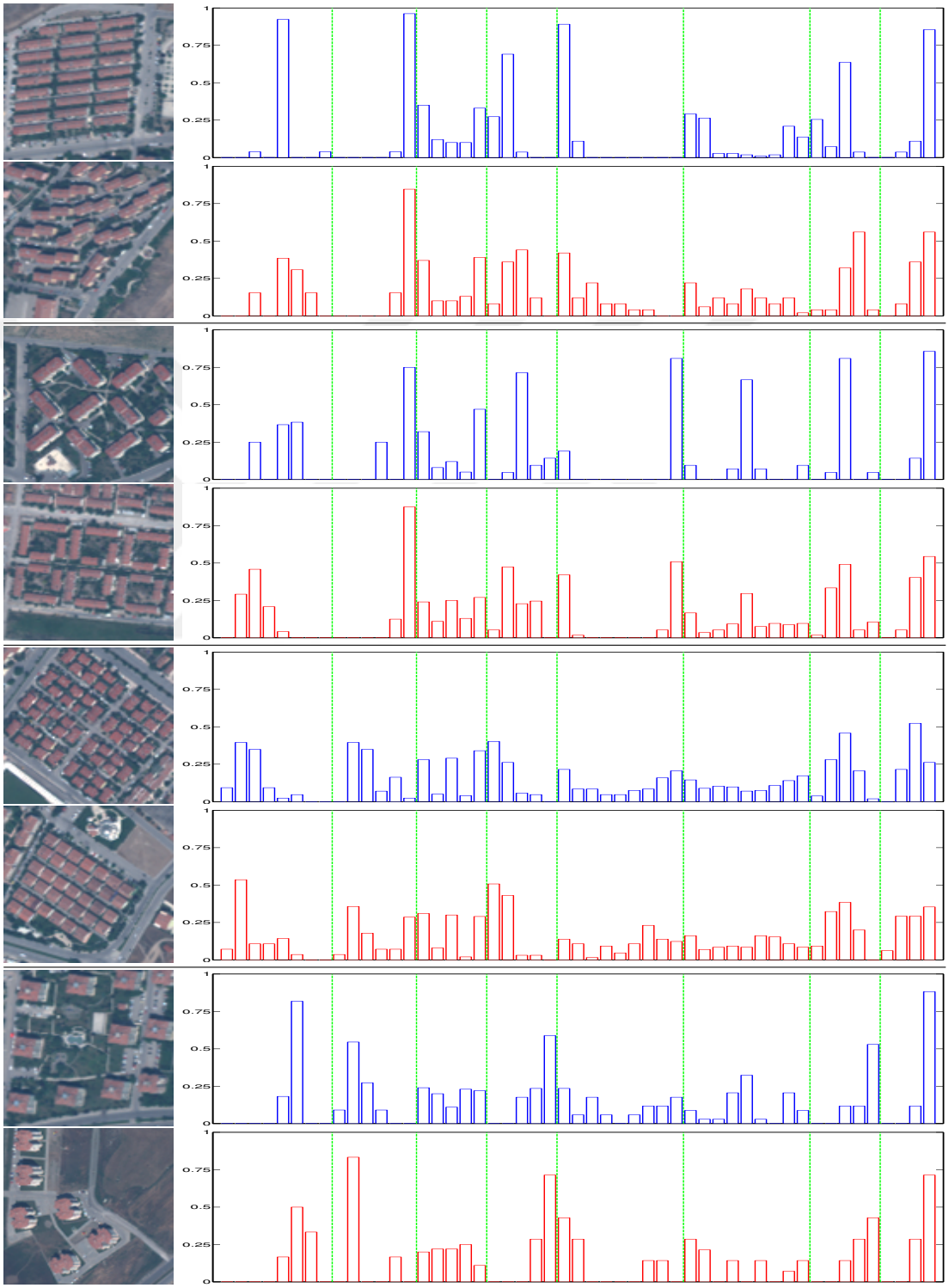


Figure 3.4: Example histograms for the building layers of four different types of compound structures. Two examples are shown for each compound structure. Each histogram is obtained through the concatenation of the marginal histograms for the features ϕ_6, ϕ_7, ϕ_8, ϕ_1, ϕ_2, ϕ_3, ϕ_4, ϕ_5 in that order. These feature histograms are separated by dashed vertical green lines.

v_i (i.e., the ellipse parameters) is considered a vector-valued random variable. Hence, a compound structure is represented by a set of random variables that leads to a region process that follows some true unknown distribution.

The informativeness of a distribution can be measured using the information entropy where the uniform distribution achieves the maximum possible value. When there is incomplete information about a probability distribution, it is desired to use the least informative distribution that makes the fewest number of assumptions. The principle of maximum entropy states that the desired distribution is the one that has the largest possible entropy while still being consistent with the information available in the data [34]. Given M independent and identically distributed observations $\mathcal{V} = \{V_1, \ldots, V_M\}$ and their histogram-based representations $H(V_m), m = 1, \ldots, M$, as described in the previous section, the information in the training data can be summarized using the empirical expectation

$$\mathbb{E}_{\mathcal{V}}[H(V)] = \frac{1}{N} \sum_{m=M}^{N} H(V_m). \tag{3.1}$$

The consistency of the desired model with the evidence in the training data can be enforced by equating the expectation

$$\mathbb{E}_p[H(V)] = \int_V H(V)p(V)dV \tag{3.2}$$

with respect to the model distribution $p(V)$ to the empirical expectation in (3.1). Then, given $\mathcal{P}$ as the set of all probability distributions on the random variable V, the maximum entropy distribution is obtained as the solution to the constrained optimization problem

$$\begin{aligned} p^* = \arg\max_{p \in \mathcal{P}} & -\int_V p(V) \log p(V) dV \\ \text{subject to } & \mathbb{E}_p[H(V)] = \mathbb{E}_{\mathcal{V}}[H(V)]. \end{aligned} \tag{3.3}$$

The region process is governed by the optimal solution p^*, that is also known as the Gibbs distribution, and by calculus of variations, takes the form

$$p(V|\beta) = \frac{1}{Z_v} \exp\Big\{\beta^T H(V)\Big\} \tag{3.4}$$

where β is the parameter vector controlling each histogram bin, and Z_v is the partition function [35]. A region process is equivalent to a Markov random field (MRF) according to the following proposition.

Proposition 1. *Let G define an MRF. p in* (3.4) *satisfies the conditional independence properties of G.*

Proof. We show that p can be represented as a product of factors, one per maximal clique in the graph. Note that we can restrict the parametrization to the edges and vertices of the graph, rather than the maximal cliques. Let $p(V|\beta) = \frac{1}{Z_v} \prod_{r_1=1}^{R} \prod_{r_2=1}^{R} \prod_{e \in E^{r_1 r_2}} \varphi_1^{r_1 r_2}(e) \varphi_2^{r_1 r_2}(e) \varphi_3^{r_1 r_2}(e) \varphi_4^{r_1 r_2}(e) \varphi_5^{r_1 r_2}(e) \prod_{r=1}^{R} \prod_{v \in V^r} \varphi_6^r(v)$ $\varphi_7^r(v) \varphi_8^r(v) \varphi_9^r(v)$ where Z_v is the partition function. We define the edge and vertex factors as $\varphi_k^{r_1 r_2}(e) = \exp\{(\beta_k^{r_1 r_2})^T H_k^{r_1 r_2}(e)\}, k = 1, \ldots, 5$, and $\varphi_k^r(v) = \exp\{(\beta_k^r)^T H_k^r(v)\}, k = 6, \ldots, 9$, where $\beta_k^{r_1 r_2}$, $k = 1, \ldots, 5$, and β_k^r, $k = 6, \ldots, 9$, are the parameter vectors controlling the one-dimensional marginal histograms $H_k^{r_1 r_2}$, $k = 1, \ldots, 5$, and H_k^r, $k = 6, \ldots, 9$, respectively. The proof is complete by the Hammersley-Clifford theorem [35]. □

3.4 Dynamic Topology of Probabilistic Region Processes

Unlike the traditional MRFs, the neighborhood structure of a region process in our model is not determined a priori. The topology of the underlying graph depends on the values of the variables in the process. Assigning a new value to a primitive object (e.g., moving, scaling, or rotating the corresponding ellipse) may change its set of neighbors, i.e., produce new neighbors and remove existing ones. An important observation is that using neighborhood structures based on Voronoi tessellations or k-nearest neighbors may cause changes in the neighborhood relations of other variables whenever a variable is modified. Conversely, determining the neighborhood structure using proximity makes the neighborhood relations between the other variables remain unchanged. Using the above property and Proposition 1, we derive the corollary below that helps the estimation procedure in the following chapter.

Corollary 1. *The conditional distribution for each individual variable v_i depends*

only on its neighbors given a realization of the process $V = \{v_1, \ldots, v_N\}$ *as*

$$\begin{aligned} p(v_i|V\backslash v_i) &= \frac{p(V)}{\sum_{v_i'} p(v_i' \cup V\backslash v_i)} \\ &= \frac{\prod_{c_{v_i} \in C(G)} \varphi(c_{v_i}) \prod_{c_{\backslash v_i} \in C(G)} \varphi(c_{\backslash v_i})}{\sum_{v_i'} \prod_{c_{v_i'} \in C(G')} \varphi(c_{v_i'}) \prod_{c_{\backslash v_i'} \in C(G')} \varphi(c_{\backslash v_i'})} \\ &= p(v_i|nb(v_i)) \end{aligned} \tag{3.5}$$

where $C(G)$ *represents the cliques of graph* G, c_{v_i} *and* $c_{\backslash v_i}$ *represent each clique that involves and does not involve* v_i, *respectively,* $nb(v_i)$ *denotes the neighbors of* v_i, *and* G' *in the denominator represents the graph that is formed for the current value of* v_i'.

The equality in (3.5) follows from the observation that all terms that do not involve v_i cancel out between the numerator and denominator, so only the products of cliques that contain v_i are left. However, if we use Voronoi tessellations or k-nearest neighbors, the cancellations would not occur because the $c_{\backslash v_i'}$ would be different for every assignment of v_i' in the summation.

Chapter 4

Learning

In order to use the probability model $p(V|\beta)$ to perform retrieval or detection, it is first necessary to learn the parameter set β for a given compound structure. Below, we first explain how to estimate β and then, show how to obtain samples for a given β that are also used in the estimation process.

4.1 Maximum Likelihood Estimation

Suppose that we observe a set of region processes $\mathcal{V} = \{V_1, \ldots, V_M\}$ that are assumed to be independent and identically distributed realizations of the same compound structure. Note that a region process is the set of all primitives from all layers. These observations can be manually marked on an image or drawn by a human analyst. We can estimate a compound structure model via maximum likelihood estimation (MLE) of the unknown parameter vector β by maximizing the log-likelihood of the data

$$\ell(\beta|\mathcal{V}) = \sum_{m=1}^{M} \log p(V_m|\beta). \tag{4.1}$$

The gradient of the log-likelihood is given by

$$\frac{d\ell(\beta|\mathcal{V})}{d\beta} = \mathbb{E}_p[H(V)] - \frac{1}{M}\sum_{m=1}^{M} H(V_m). \tag{4.2}$$

Since the MLE problem is differentiable and jointly convex in the vector β, gradient ascent algorithms are guaranteed to converge to the global optimum. We use the stochastic gradient ascent algorithm where the expectation $\mathbb{E}_p[H(V)]$ in (4.2) is approximated by a finite sum of histograms of samples $V_{(s)}, s = 1, \ldots, S$, drawn independently from the distribution $p(V|\beta)$, as

$$\hat{\mathbb{E}}_p[H(V)] = \frac{1}{S}\sum_{s=1}^{S} H(V_{(s)}). \tag{4.3}$$

Figure 4.1 shows an example run of the β learning procedure. Initially, we are given an example histogram for relative orientations whose distribution we wish to learn. We initialize β with all 0's which corresponds to a uniform distribution. Hence, the primitives sampled from this β are nearly randomized. Then, we calculate the average histogram for relative orientations of the sampled primitives. Since β was all 0's, this histogram also looks uniform as expected. We update β proportional to the difference between each bin in the observed and sampled histograms as given in (4.2). The samples generated by the new β exhibit more intuitive relative orientations. These samples are used to update β again. This procedure of sampling from the current β and obtaining the new β from these samples continues until the log-likelihood in (4.1) does not change.

The pseudocode for the resulting method is shown in Algorithm 1. In the next section, we describe a Markov chain Monte Carlo-based (MCMC) method for generating each sample $V_{(s)}$ in line 5 of the algorithm.

4.2 Sampling Region Processes

We use a Gibbs sampler that samples a variable conditioned on the values of all the other variables in the distribution parameterized by β in a particular

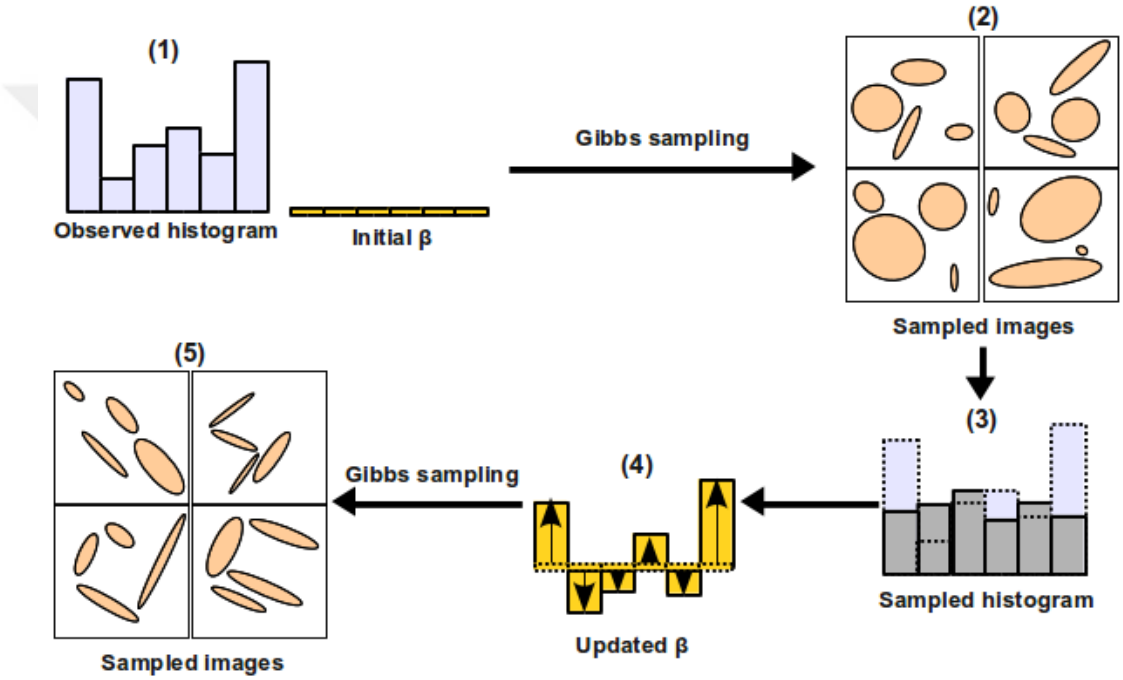


Figure 4.1: An example iteration for updating β corresponding to the relative orientation histogram bins. (1) We begin with an average histogram of the observed region processes and a β of all 0's. The histogram shows two peaks in orientations close to 0 and 90 degrees which means that the primitives are mostly parallel or perpendicular to each other. (2) Sample primitives are drawn from the current model (i.e., uniform distribution). (3) Average relative orientation histograms are computed from these samples. The sampled histogram is shown in dark gray on top of the observed histogram to show the difference between them. (4) β is re-weighted according to this difference. (5) Newly sampled primitives according to this β appear oriented more consistently with the initial observed histogram.

Algorithm 1 Stochastic gradient ascent for maximum likelihood estimation of β.

Input: $\mathcal{V} = \{V_1, \ldots, V_M\}$
Output: β

Initialize weights β randomly
$\eta \leftarrow 1$
repeat
    for $s \leftarrow 1$ to S **do**
        Sample $V_{(s)} \sim p(V|\beta)$
    end for
    $\hat{\mathbb{E}}_p[H(V)] \leftarrow \frac{1}{S}\sum_{s=1}^{S} H(V_{(s)})$
    $\beta \leftarrow \beta + \eta\Big(\hat{\mathbb{E}}_p[H(V)] - \frac{1}{M}\sum_{m=1}^{M} H(V_m)\Big)$
    Decrease step size η by a factor of 0.5
until log-likelihood in (4.1) unchanged

iteration of the stochastic gradient ascent procedure. Given a joint sample $\tilde{V}^{(t)} = \{v_1^{(t)}, \ldots, v_N^{(t)}\}$ of N variables at the t'th sampling iteration, the next step involves replacing the value of a particular variable $v_i^{(t)}$ by a new value $v_i^{(t+1)}$ drawn from the full conditional distribution $p(v_i|\tilde{V}^{(t)} \backslash v_i^{(t)}, \beta)$. We move from $v_i^{(t)}$ to $v_i^{(t+1)}$ by sampling only one ellipse component (i.e., either one of l_i, s_i, or θ_i) at a time. That is, we choose either one of l_i, s_i, or θ_i to be updated at random, with equal probability, and then a candidate value is randomly generated for that component from a uniform proposal distribution over the object parameter space defined in Section 3.1. This corresponds to randomly translating, scaling, or rotating an ellipse at each sampling iteration. The new value of the selected component together with the old values of the remaining components produce a candidate sample v_i^*. Since the proposal distribution is symmetric, the acceptance probability [36] of the candidate sample is obtained as

$$\alpha = \min\left(1, \frac{p(v_i^*|\tilde{V}^{(t)} \backslash v_i^{(t)}, \beta)}{p(v_i^{(t)}|\tilde{V}^{(t)} \backslash v_i^{(t)}, \beta)}\right). \tag{4.4}$$

If the proposal is accepted, $v_i^{(t+1)}$ is set to v_i^*; otherwise, $v_i^{(t+1)}$ stays the same as $v_i^{(t)}$. All the other variables remain unchanged, i.e., $v_j^{(t+1)} = v_j^{(t)}$ for $j \neq i$ and $j = 1, \ldots, N$.

By Corollary 1, to sample a variable, we only need to know the values of its

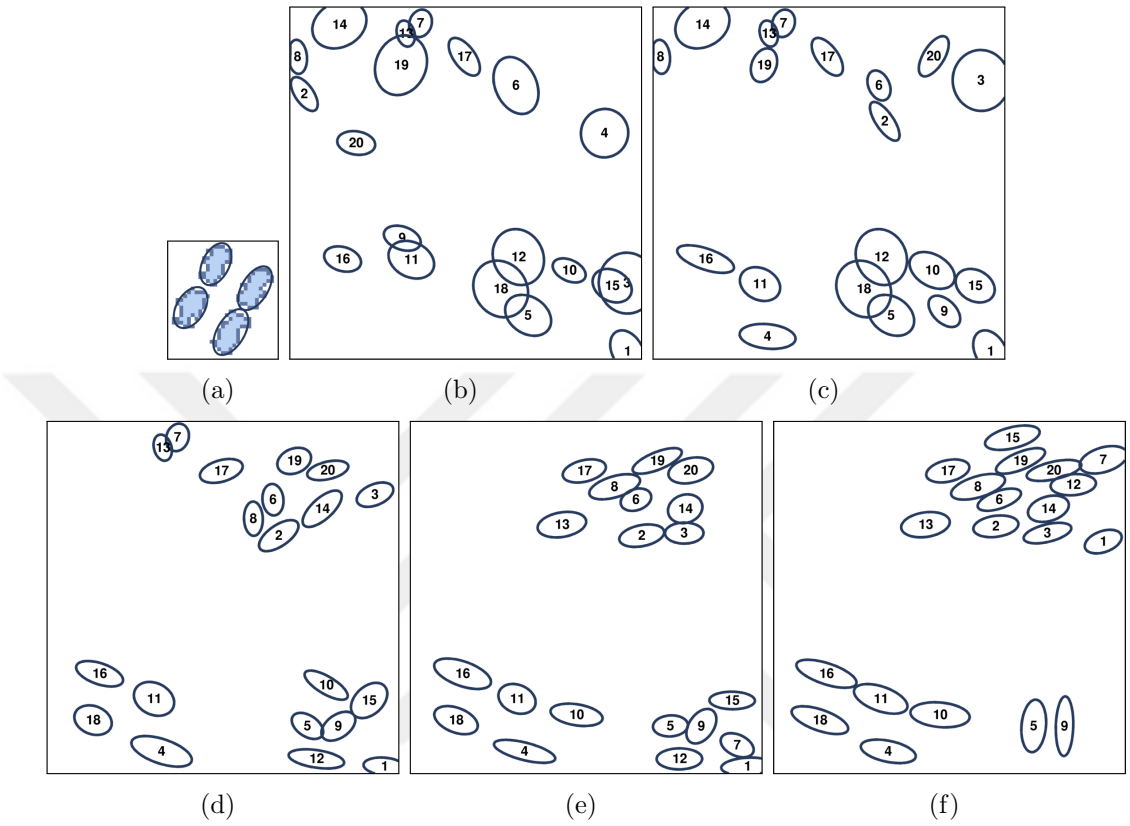


Figure 4.2: Illustration of the Gibbs sampler in Algorithm 2 for a single primitive layer. (a) The compound structure V given as input to stochastic gradient ascent in Algorithm 1. (b)-(f) Samples $\tilde{V}^{(t)}$ at iterations $t = 0, 50, 200, 500, 1000$ in Algorithm 2.

neighbors before and after the proposal. Thus, the acceptance probability reduces to $\alpha = \min\left(1, \frac{p(v_i^*|nb(v_i^*),\beta)}{p(v_i^{(t)}|nb(v_i^{(t)}),\beta)}\right)$. Since p can be represented as a product of potentials over vertices and edges, it can be further shown that $p(v_i|nb(v_i),\beta) = \frac{1}{Z_v}\exp\left\{\beta^T H(v_i \cup nb(v_i))\right\}$, and we can write $\alpha = \min\left(1, \frac{\exp\{\beta^T H(v_i^* \cup nb(v_i^*))\}}{\exp\{\beta^T H(v_i^{(t)} \cup nb(v_i^{(t)}))\}}\right)$. As a result, when evaluating α, we do not need to calculate the normalization constant Z_v. The sampling procedure is summarized in Algorithm 2 and is illustrated in Figures 4.2 and 4.3.

Algorithm 2 Gibbs sampler for producing a particular $V_{(s)}$.

Input: β
Output: $V_{(s)}$

Initialize $\tilde{V}^0 = \{v_1^{(0)}, \dots\ v_N^{(0)}\}$
for $t \leftarrow 0, 1, 2, \dots, T-1$ **do**
    Choose one v_i at random, with equal probability
    Choose l_i, s_i, or θ_i at random, with equal probability
    if l_i is chosen **then**
        Sample $l_i^* \sim U([0, X_{max} - 1] \times [0, Y_{max} - 1])$
        $v_i^* \leftarrow (l_i^*, s_i^{(t)}, \theta_i^{(t)})$
    end if
    if s_i is chosen **then**
        Sample $s_i^* \sim U([s_{min}^h, s_{max}^h] \times [s_{min}^w, s_{max}^w])$
        $v_i^* \leftarrow (l_i^{(t)}, s_i^*, \theta_i^{(t)})$
    end if
    if θ_i is chosen **then**
        Sample $\theta_i^* \sim U([0, \pi))$
        $v_i^* \leftarrow (l_i^{(t)}, s_i^{(t)}, \theta_i^*)$
    end if
    $v_i^{(t+1)} \leftarrow$ UpdatePrimitive$(v_i^*, \tilde{V}^{(t)}, \beta)$
    $v_j^{(t+1)} \leftarrow v_j^{(t)}$ for $j \neq i$ and $j = 1, \dots, N$
end for
$V_{(s)} \leftarrow \tilde{V}^{(t)}$
procedure UpdatePrimitive$(v_i^*, \tilde{V}, \beta)$
    Compute $nb(v_i) \in \tilde{V} \backslash v_i$ and $nb(v_i^*) \in \tilde{V} \backslash v_i$
    Compute acceptance probability α
    Sample $q \sim U(0, 1)$
    if $q < \alpha$ **then**
        return v_i^*
    else
        return v_i
    end if
end procedure

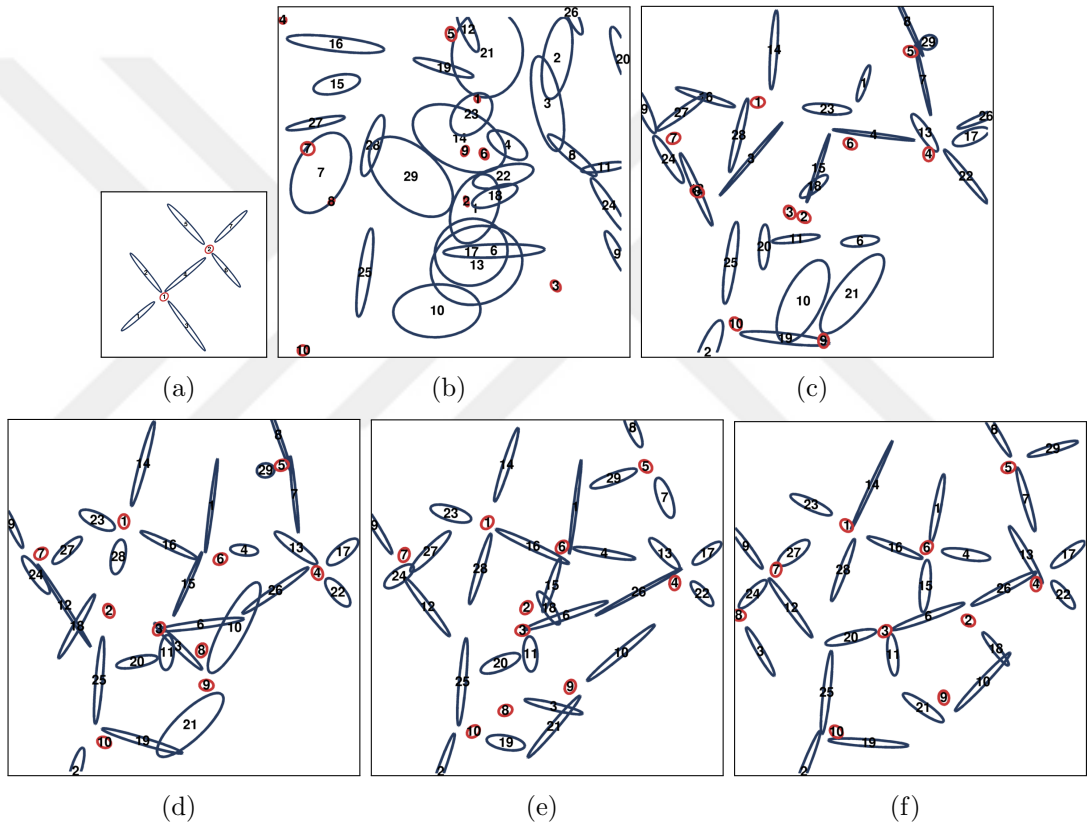


Figure 4.3: Illustration of the Gibbs sampler in Algorithm 2 for two primitive layers. (a) The compound structure V given as input to stochastic gradient ascent in Algorithm 1. Different primitive layers are colored with blue and red, respectively. (b)-(f) Samples $\tilde{V}^{(t)}$ at iterations $t = 0, 50, 250, 600, 1000$ in Algorithm 2.

Chapter 5

Inference and Region Selection

Given a compound structure model with learned parameter vector β, we would like to automatically detect all of its instances in an input image I. We, first, propose a set of candidate primitive regions in the image, and then, an inference algorithm is used to select a coherent subset of those regions that optimize a probability function defined in terms of both appearance and arrangement characteristics of region groups.

5.1 Hierarchical Region Extraction

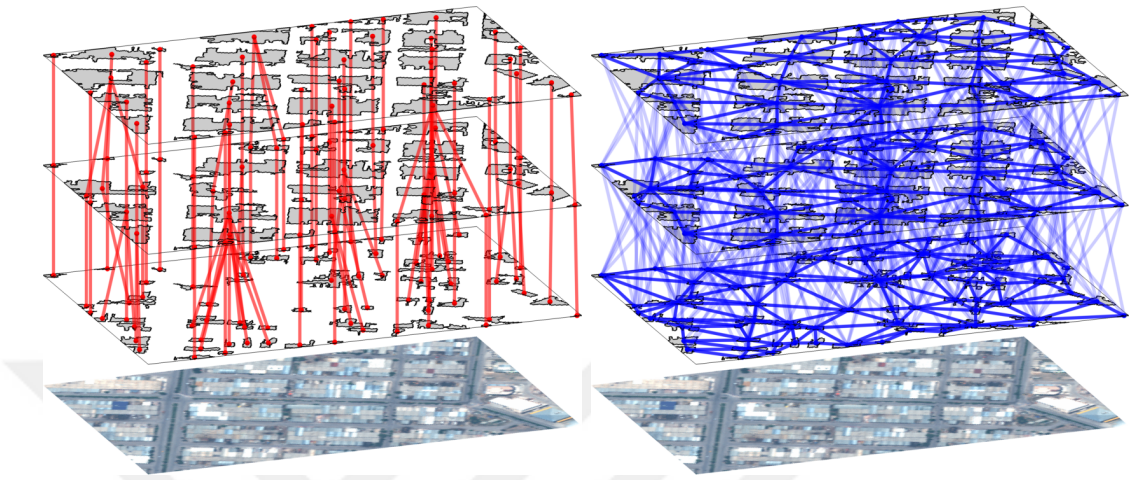
The detection problem is posed as the selection of multiple subgroups of candidate regions $V = \{v_1, \ldots, v_N\}$ coming from multiple hierarchical segmentations where each selected group of regions constitutes an instance of the example compound structure in the large image. Considering the fact that different objects of interest may appear at different scales, the first step in the detection procedure involves the identification of primitive regions for each layer V^r by using a hierarchical segmentation algorithm. The union of these regions from all levels at all layers are treated as candidate primitives, forming the set $V = \bigcup_{r=1,\ldots,R} V^r$.

The next step is to connect the potentially related vertices at all levels to

represent the neighbor relationships. Since the candidate regions are fixed at the segmentation step, the set of neighbors for each region can also be fixed, with no need for the dynamic neighborhood definition for the sampling problem in Section 4.2. Thus, we use Voronoi tessellations of boundary pixels of regions at each level to identify the neighbors of each region at that level. Voronoi-based neighborhood definition is preferred at this step as it does not require any parameter like the proximity threshold or the number of neighbors as in the proximity-based and k-nearest neighbor-based definitions, respectively. For each layer V^r, we use Voronoi tessellations of boundary pixels of regions at each level to identify the within-level edges $(v_i, v_j) \in E^{rr}$ at that level. Furthermore, a between-level edge $(v_i, v_j) \in E^{rr}$ is also formed if $v_j \in V^r$ is at a higher level compared to $v_i \in V^r$ and if any descendant of v_j that is at the same level as v_i is a Voronoi neighbor of v_i. For each pair of layers V^{r_1} and V^{r_2}, vertices $v_i \in V^{r_1}$ and $v_j \in V^{r_2}$ are connected with a between-layer edge $(v_i, v_j) \in E^{r_1 r_2}$ if the distance between the closest pixels of these objects is less than a proximity threshold. The union of these within-level, between-level and between-layer edges form the set of edges $E = \bigcup_{r_1, r_2 = 1, \ldots, R} E^{r_1 r_2}$. Figures 5.1 and 5.2 illustrate example hierarchies.

5.2 Inference without Constraints

In this section, the proposed detection algorithm tries to find an unknown number of regions in the new image data that are similarly arranged as the regions in the reference compound structure. The similarity measure is defined probabilistically rather than by imposing hard constraints on the output regions. The candidate regions are assumed to be distributed according to the previously learned probability model and the identification of the meaningful regions is performed by sampling the input regions in groups.



(a) (b)

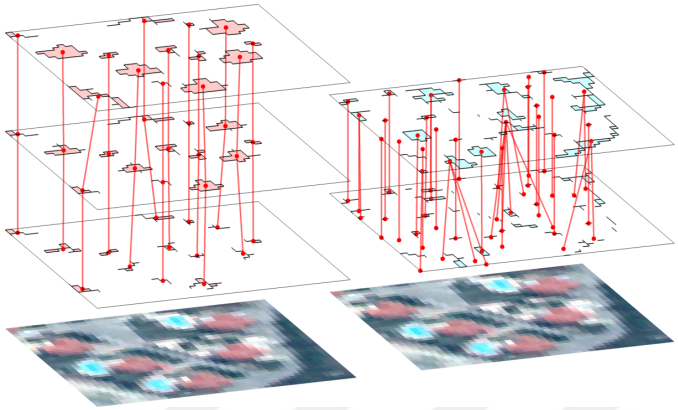
Figure 5.1: Graph construction for a single primitive layer. The hierarchical candidate regions (V) at three levels are shown in gray. (a) The edges that represent parent-child relationship are shown in red. (b) The edges E that represent the within-level and between-level neighbor relationship are shown in blue. For clarity, we do not show the edges between two levels that are not consecutive even though there are edges between all level pairs.

5.2.1 Bayesian Formulation

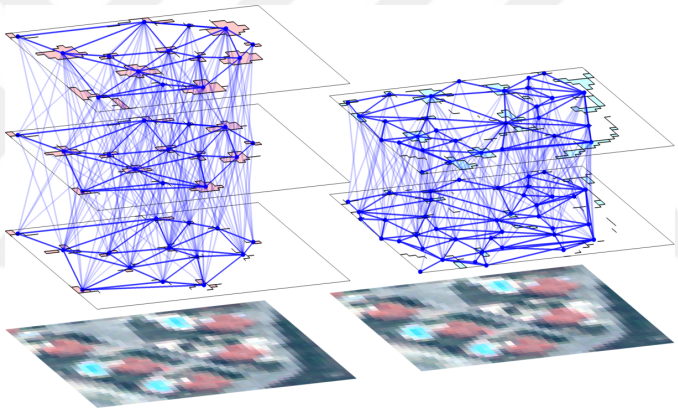
Given a graph $G = (V, E)$ that represents the candidate regions and their neighbor relationships in image I, our goal is to search for coherent groups of regions that attain high probability explanations of instances of compound structures of interest in the image. This problem can be formulated as the selection of a subset V^* among all regions V as

$$V^* = \arg\max_{V' \subseteq V} p(V'|I) = \arg\max_{V' \subseteq V} p(I|V')p(V') \tag{5.1}$$

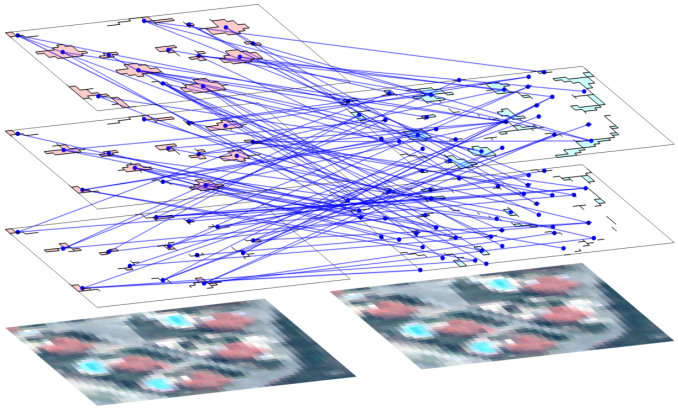
where $p(I|V')$ is the observed spectral data likelihood for the compound structure in the image, and $p(V')$ acts as the spatial (both shape and arrangement) prior according to the model defined in Section 3. We use a simple spectral appearance model where the spectral content of each primitive region in a particular layer V^r is assumed to be independent and identically distributed according to a Gaussian with mean μ^r and covariance Σ^r, so that $p(I|V') = \prod_{r=1}^{R} \prod_{v_i \in V' \cap V^r} p(y_i|\mu^r, \Sigma^r)$ where y_i is the average spectral vector for the pixels inside the i'th region v_i. This



(a)



(b)



(c)

Figure 5.2: Graph construction for two primitive layers (i.e., building and pool). The hierarchical candidate regions at three and two levels for these layers are shown in red and light blue, respectively. (a) The edges that represent parent-child relationship for both layers are shown. (b) The edges that represent the within- and between-level neighbor relationship within the same layer are shown. (c) The edges that represent the neighbor relationship between the layers are shown. For better visualization of edges, only 20 percent of all between-layer edges are shown.

formulation assumes that the primitives in a particular layer of the compound structure have similar spectral characteristics as the focus of this dissertation is to develop a novel spatial data model. The spatial appearance probability $p(V')$ is computed as in (3.4) using ellipses that have the same second moments as the regions in V'.

5.2.2 CRF Formulation

The selection problem in (5.1) can be formulated as a conditional random field (CRF). Let $X = \{x_1, \ldots, x_N\}$ where $x_i \in \{0, 1\}, i = 1, \ldots, N$, be the set of indicator variables associated with the vertices V of G so that $x_i = 1$ implies region v_i being selected. Our CRF formulation defines a posterior distribution for hidden random variables X given regions V and their observed spectral features $Y = \{y_1, \ldots, y_N\}$ in a factorized form as

$$\begin{aligned} p(X|I, V) &\propto p(I|X, V)p(X, V) \\ &= \frac{1}{Z_x} \prod_{v_i \in V} \exp\left\{\left(\psi_i^c + \psi_i^s\right)x_i\right\} \prod_{(v_i, v_j) \in E} \exp\left\{\psi_{ij}^a x_i x_j\right\} \end{aligned} \tag{5.2}$$

where the vertex bias terms ψ^c and ψ^s representing color and shape, respectively, and edge weights ψ^a representing arrangement are defined as

$$\psi_i^c = \frac{-1}{2}(y_i - \mu^r)^T(\Sigma^r)^{-1}(y_i - \mu_r), \qquad \forall v_i \in V^r, r = 1, \ldots, R \tag{5.3}$$

$$\psi_i^s = \sum_{k=6}^{9} \beta^r_{k, \mathbb{I}_k^r\left(\phi_k(v_i)\right)}, \qquad \forall v_i \in V^r, r = 1, \ldots, R \tag{5.4}$$

$$\psi_{ij}^a = \sum_{k=1}^{5} \beta^{r_1 r_2}_{k, \mathbb{I}_k^{r_1 r_2}\left(\phi_k(v_i, v_j)\right)}, \qquad \forall (v_i, v_j) \in E, r_1, r_2 = 1, \ldots, R. \tag{5.5}$$

The feature ϕ_k is computed by using the parameters of the ellipse that has the second moments as the input region, $\mathbb{I}_k^r$ is the index of the histogram bin to which a given feature value belongs in H_k^r, and $\beta_{k,j}^r$ denotes the j'th component of the parameter vector β_k^r controlling H_k^r. $h_k^{r_1 r_2}$ and $\beta_{k,j}^{r_1 r_2}$ are defined similarly. Then, selecting V^* in (5.1) is equivalent to estimating the joint MAP labels given by

$$X^* = \arg\max_X p(X|I, V). \tag{5.6}$$

5.2.3 CRF Inference

Exact inference of (5.6) is intractable in general graphs but an approximate solution can be obtained by an MCMC sampler. However, Gibbs sampling that updates one variable at a time can be slow in such models requiring many updates to produce significant changes in the global state, especially when there is strong dependence between the components [35]. On the contrary, the Swendsen-Wang algorithm [37] mixes much faster by updating the labels of many variables at once.

In this dissertation, we adapt the Swendsen-Wang algorithm that was designed for the Ising model parameterization, i.e., $\{-1, +1\}$ variables, to sample $\{0, 1\}$ variables. First, the original $\{0, 1\}$ indicator variables X are converted to $\{-1, +1\}$ variables $Z = \{z_i = 2x_i - 1, i = 1, \ldots, N\}$. Then, the objective (5.2) is reformulated by variable substitution as

$$\begin{aligned} p(Z|I,V) &\propto p(I|Z,V)p(Z,V) \\ &= \frac{1}{Z_z}\prod_{v_i \in V} \exp\Big\{\big(\frac{1}{2}\psi_i^c + \frac{1}{2}\psi_i^s + \frac{1}{4}\psi_i^w\big)z_i\Big\} \prod_{(v_i,v_j)\in E} \exp\Big\{\frac{1}{4}\psi_{ij}^a z_i z_j\Big\} \end{aligned} \tag{5.7}$$

where a new term $\psi_i^w = \sum_{v_j \in V} \psi_{ij}^a$ is added to the vertex biases. We are interested in samples from $p(Z|I,V)$ so that the most likely configuration for Z can be found.

The motivation behind the Swendsen-Wang algorithm is that, sampling can sometimes be made easier by adding more variables. Suppose we introduce auxiliary variables $U = \{u_{ij} : (v_i, v_j) \in E\}$, one per edge, and define the extended model

$$p(Z, U|I, V) \propto p(I|Z, V)p(Z, V)p(U|Z, I, V). \tag{5.8}$$

A careful selection of $P(U|Z, I, V)$ can make the conditionals $P(U|Z, I, V)$ and $P(Z|U, I, V)$ easy to sample from, and samples for the joint model $P(Z, U|I, V)$ can be obtained by alternately sampling these conditionals with conventional MCMC techniques [38]. Then, marginalization will produce valid Z samples from the original distribution because $\sum_U p(Z, U|I, V) = p(Z|I, V)$.

In the extended model in (5.8), we assume that u_{ij}'s are conditionally independent given the vertex variables and are uniformly distributed between 0 and $\exp\left\{\frac{1}{4}\psi_{ij}^a z_i z_j\right\}$. The conditional distribution of the auxiliary variables can be obtained as

$$p(U|Z,I,V) = \prod_{(v_i,v_j)\in E} \frac{1}{\exp\left\{\frac{1}{4}\psi_{ij}^a z_i z_j\right\}} \mathbb{1}\left[0 \le u_{ij} \le \exp\left\{\frac{1}{4}\psi_{ij}^a z_i z_j\right\}\right] \tag{5.9}$$

where $\mathbb{1}$ is an indicator function that is 1 when its argument is true, and 0 otherwise. Our choice of this $p(U|Z,I,V)$ leads to the joint distribution

$$p(Z,U|I,V) \propto \prod_{v_i\in V} \exp\left\{\left(\frac{1}{2}\psi_i^c + \frac{1}{2}\psi_i^s + \frac{1}{4}\psi_i^w\right) z_i\right\} \prod_{(v_i,v_j)\in E} \mathbb{1}\left[0 \le u_{ij} \le \exp\left\{\frac{1}{4}\psi_{ij}^a z_i z_j\right\}\right]. \tag{5.10}$$

The conditional distribution of the vertex indicator variables Z given the auxiliary variables U is also obtained as

$$p(Z|U,I,V) \propto p(Z,U|I,V). \tag{5.11}$$

That is, $p(Z|U,I,V)$ is equal to the product of the selected vertex biases, restricted to the regions where all constraints

$$\left\{0 \le u_{ij} \le \exp\left\{\frac{1}{4}\psi_{ij}^a z_i z_j\right\},\ \forall (v_i,v_j) \in E\right\} \tag{5.12}$$

are satisfied, and is 0 elsewhere.

In the following, we describe how we sample the extended model via Gibbs sampling from $p(U|Z,I,V)$ and $p(Z|U,I,V)$ alternately. Note that the terms involving the edge weights in (5.9) can only take two values according to the choice of Z, i.e.,

$$\exp\left\{\frac{1}{4}\psi_{ij}^a z_i z_j\right\} = \begin{cases} \exp\left\{\frac{1}{4}\psi_{ij}^a\right\} & \text{if } z_i = z_j, \\ \exp\left\{\frac{-1}{4}\psi_{ij}^a\right\} & \text{if } z_i = -z_j. \end{cases} \tag{5.13}$$

Consequently, when conditioning on U in (5.11), the terms $\mathbb{1}\left[0 \le u_{ij} \le \exp\left\{\frac{1}{4}\psi_{ij}^a z_i z_j\right\}\right]$ may constrain the allowed combinations of Z. In particular, when $\psi_{ij}^a > 0$:

- if $u_{ij} > \exp\left\{\frac{-1}{4}\psi_{ij}^a\right\}$, we must have $z_i = z_j$,
- if $u_{ij} \leq \exp\left\{\frac{-1}{4}\psi_{ij}^a\right\}$, there is no constraint on (z_i, z_j).

Similarly, when $\psi_{ij}^a < 0$:

- if $u_{ij} > \exp\left\{\frac{1}{4}\psi_{ij}^a\right\}$, we must have $z_i = -z_j$,
- if $u_{ij} \leq \exp\left\{\frac{1}{4}\psi_{ij}^a\right\}$, there is no constraint on (z_i, z_j).

Hence, the selection of U introduces constraints to the distribution giving rise to form connected components of vertices to act as a single bonded unit.

To simplify the notation, we replace each u_{ij} with a binary indicator variable $b_{ij} = \mathbb{1}\left[u_{ij} > \exp\left\{\frac{-1}{4}|\psi_{ij}^a|\right\}\right]$ that denotes the presence of a bond. The conditional $p(B|Z, I, V)$ for the set of all bond variables $B = \{b_{ij} : (v_i, v_j) \in E\}$ factorizes over the edges as $p(B|Z, I, V) = \prod_{(v_i,v_j)\in E} p(b_{ij}|z_i, z_j, I, v_i, v_j)$. From (5.13), when $\psi_{ij}^a > 0$:

$$p(b_{ij} = 1|z_i, z_j, I, v_i, v_j) = \begin{cases} \frac{\exp\{\frac{1}{4}\psi_{ij}^a\} - \exp\{\frac{-1}{4}\psi_{ij}^a\}}{\exp\{\frac{1}{4}\psi_{ij}^a\}} = 1 - \exp\left\{\frac{-1}{2}\psi_{ij}^a\right\} & \text{if } z_i = z_j, \\ 0 & \text{if } z_i = -z_j. \end{cases} \tag{5.14}$$

When $\psi_{ij}^a < 0$:

$$p(b_{ij} = 1|z_i, z_j, I, v_i, v_j) = \begin{cases} \frac{\exp\{\frac{-1}{4}\psi_{ij}^a\} - \exp\{\frac{1}{4}\psi_{ij}^a\}}{\exp\{\frac{-1}{4}\psi_{ij}^a)} = 1 - \exp\left\{\frac{1}{2}\psi_{ij}^a\right\} & \text{if } z_i = -z_j, \\ 0 & \text{if } z_i = z_j. \end{cases} \tag{5.15}$$

Sampling from $p(B|Z, I, V)$, and equivalently from $p(U|Z, I, V)$, is done by randomly selecting a subset of the bond variables based on $p(b_{ij}|z_i, z_j, I, v_i, v_j)$ and forming sets of connected components $\mathcal{C}$ that are connected by edges with $b_{ij} = 1$. The individual vertices that are not connected to any other vertex are

also included in this set. Then, sampling from $p(Z|U, I, V)$ is done by randomly selecting some of these connected components and simultaneously flipping the labels of all vertices within these components so that the constraints

- $z_i = z_j$ if $\psi^a_{ij} > 0$,
- $z_i = -z_j$ if $\psi^a_{ij} < 0$

for $b_{ij} = 1$ are still satisfied. When sampling a connected component $C' \in \mathcal{C}$ from $p(Z|U, I, V)$, the acceptance probability for flipping the labels is given by

$$\gamma(C') = \frac{p(-Z|U, I, C')}{p(-Z|U, I, C') + p(Z|U, I, C')} \tag{5.16}$$

where

$$p(-Z|U, I, C') = \prod_{v_i \in C'} \exp\Big\{\big(\frac{1}{2}\psi^c_i + \frac{1}{2}\psi^s_i + \frac{1}{4}\psi^w_i\big)(-z_i)\Big\} \tag{5.17}$$

is the likelihood of the vertices in C' when their labels are flipped ($z_i \leftarrow -z_i$), and

$$p(Z|U, I, C') = \prod_{v_i \in C'} \exp\Big\{\big(\frac{1}{2}\psi^c_i + \frac{1}{2}\psi^s_i + \frac{1}{4}\psi^w_i\big)z_i\Big\} \tag{5.18}$$

is the likelihood when the labels stay the same.

The proposed region selection algorithm is summarized in Algorithm 3 and is illustrated in Figure 5.3. We use a simulated annealing procedure [35] as described in Section 6 to guide the sampling iterations. The sampling procedure continues until the change in the value of the objective (5.2) between two consecutive iterations is significantly small, and a solution to (5.6) is obtained by taking the most likely configuration X^* across all samples. Finally, the marginal probabilities for the individual regions in the set V^* that corresponds to this solution are obtained from the frequency of observation of each primitive region during the sampling process.

Algorithm 3 Swendsen-Wang sampler for CRF inference for estimating X^*. The number of iterations T is determined by simulated annealing.

Input: $\psi_i^c, \psi_i^s, \psi_{ij}^a,\ i, j = 1, \ldots, N$
Output: X^*

Initialize labels $Z = \{z_i = -1,\ i = 1, \ldots, N\}$
for $t \leftarrow 1, 2, \ldots, T$ **do**
    for all $(v_i, v_j) \in E$ **do**
        $b_{ij} \leftarrow$ SampleBondGivenVertices(z_i, z_j, ψ_{ij}^a)
    end for
    Form connected components $\mathcal{C}$ using bonds $b_{ij} = 1$
    Pick component $C' \in \mathcal{C}$ uniformly at random
    Flip labels for all $v_i \in C'$ with probability $\gamma(C')$
    Compute $X^t = \{x_i = (z_i + 1)/2,\ i = 1, \ldots, N\}$
end for
$X^* \leftarrow \underset{X \in \{X^1, \ldots, X^T\}}{\arg\max}\ p(X|I, V)$
procedure SampleBondGivenVertices(z_i, z_j, ψ_{ij}^a)
    if $(z_i = z_j\ \&\ \psi_{ij}^a > 0)$ **or** $(z_i = -z_j\ \&\ \psi_{ij}^a < 0)$ **then**
        Sample $q \sim U(0, 1)$
        if $q < 1 - \exp\left\{\frac{-1}{2}|\psi_{ij}^a|\right\}$ **then**
            return 1
        end if
    end if
    return 0
end procedure

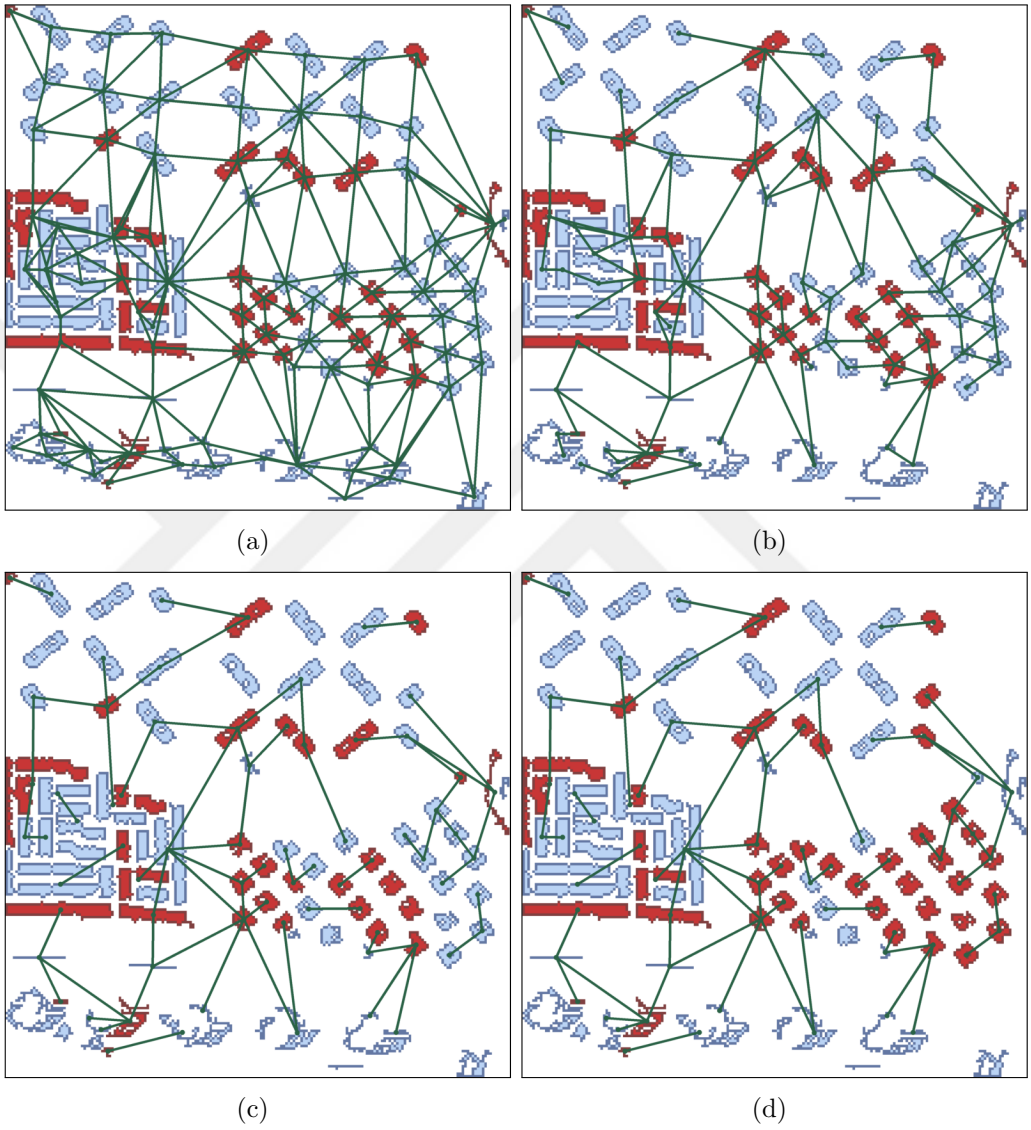


Figure 5.3: Illustration of the Swendsen-Wang procedure in Algorithm 3. In each figure, the labels of the primitives are shown in red for selected ($z_i = +1$) and blue for not selected ($z_i = -1$). (a) The labels at the beginning of a particular sampling iteration. The Voronoi edges (E) are shown in green. (b) The edges with positive bond probabilities as candidates for forming connected components of their corresponding vertices. (c) The sampled edges that form connected components of vertices bonded together. (d) The result of randomly flipping the labels of the primitives in some of these components. A single scale is shown for simplicity even though the algorithm normally runs on the graph for the whole candidate region hierarchy for all layers.

5.3 Inference with Constraints

In Section 5.2, the probability distribution favors the spatial arrangements that are present in the reference compound structure. This is accomplished by introducing energy terms, that encourage or penalize particular arrangements, in the prior probability. These terms can be considered as soft constraints. In some cases, general hard constraints, strictly prohibiting undesirable variable configurations rather than assigning them low probabilities, may exist over the output regions. In constrained inference, our objective is to obtain the maximum probability estimates of the indicator variables, $x_i, i = 1, \ldots, N$ satisfying such convex inequality and equality constraints. These constraints may be related to the shape, the mutual exclusion or the spatial layout properties of one or more regions.

In [6], hard geometrical constraints are enforced as prior energy terms in the probability density function (i.e., Gibbs distribution). In particular, the intersection between two objects is avoided by giving a "$-\infty$" energy term to each object pair with an overlap greater than a given threshold. Otherwise, the energy is set to 0 (i.e., has no effect on detection). The traditional Markov Chain Monte Carlo sampler is used for inference which would be extremely slow, especially, under the presence of hard constraints containing many variables. Also, those constraints cannot be exploited in a natural way by the modified Swendsen-Wang procedure which only allows pairwise interaction potentials.

Hence, we reformulate the problem as quadratic programming with linear programming relaxation under convex constraints. The resulting non-convex problem is approximately solved through a sequence of strongly convex quadratic programs.

5.3.1 Quadratic Programming Formulation

The problem in Equation (5.1) can be rewritten as

$$V^* = \underset{\substack{V' \subseteq V \\ V' \subseteq \Omega}}{\arg\max} p(V'|I) = \underset{\substack{V' \subseteq V \\ V' \subseteq \Omega}}{\arg\max} p(I|V')p(V'). \tag{5.19}$$

where $\Omega \in R^N$ is a nonempty polyhedral convex set determined by a set of constraints. For a $V' \subseteq V$, $\log p(V'|\beta^*)$ can be written as follows

$$\begin{aligned} \log p(V'|\beta) =& \sum_{k=1}^{5} \sum_{r_1=1}^{R} \sum_{r_2=1}^{R} \beta_k^{r_1 r_2} H_k^{r_1 r_2}(V') + \sum_{k=6}^{9} \sum_{r=1}^{R} \beta_k^r H_k^r(V') - \log Z_X \\ =& \sum_{k=1}^{5} \sum_{r_1=1}^{R} \sum_{r_2=1}^{R} \sum_{(v_i,v_j) \in E^{r_1 r_2}} \beta^{r_1 r_2}_{k,\mathbb{I}_k^{r_1 r_2}\left(\phi_k(v_i,v_j)\right)} x_i x_j \\ &+ \sum_{k=6}^{9} \sum_{r=1}^{R} \sum_{v_i \in V^r} \beta^{r}_{k,\mathbb{I}_k^{r}\left(\phi_k(v_i)\right)} x_i - \log Z_X. \end{aligned} \tag{5.20}$$

where Z_X is the partition function. Let $W = \sum_{k=1}^{5} \sum_{r_1=1}^{R} \sum_{r_2=1}^{R} W_k^{r_1 r_2}$ where $W_k^{r_1 r_2}, k \in \{1, \ldots, 5\}, r_1, r_2 \in \{1, \ldots, R\}$ is an $N \times N$ affinity matrix. Each element of this matrix is calculated as $W_k^{r_1 r_2}(i,j) = -\beta^{r_1 r_2}_{k,\mathbb{I}_k^{r_1 r_2}\left(\phi_k(v_i,v_j)\right)}$. Also, let $q = \sum_{k=6}^{9} \sum_{r=1}^{R} q_k^r$ where $q_k^r, k \in \{6, \ldots, 9\}, r \in \{1, \ldots, R\}$ is an $N \times 1$ potential vector. Each element of this vector is calculated as $q_k^r(i) = \beta^{r}_{k,\mathbb{I}_k^{r}\left(\phi_k(v_i)\right)}$. The problem can be formulated as

$$\begin{aligned} \underset{x}{\text{minimize}} \ -\log p(V'|\beta) =& \frac{1}{2} X^T W X + q^T X + \log Z_X \\ \text{subject to } & X \in \Omega, \\ & X \in \{0,1\}. \end{aligned} \tag{5.21}$$

In Sections 6.4 and 6.5, the abstract representation of the constraint space Ω will be determined by explicit linear inequality and equality constraints written in the general form as

$$\begin{aligned} A^T X &\leq b \\ C^T X &= d. \end{aligned} \tag{5.22}$$

where $A, C \in \mathbb{R}^n \times \mathbb{R}^n, b, d \in \mathbb{R}^n$.

5.3.2 Difference of Convex Programming Formulation

A Difference of Convex (DC) problem is defined on the N-dimensional real-space as

$$P = \min \left\{ f(X) = g(X) - h(X) : X \in \mathbb{R}^N \right\} \tag{5.23}$$

where $g : \mathbb{R}^N \to \mathbb{R}$ and $h : \mathbb{R}^N \to \mathbb{R}$ are convex functions. In case there exists a constraint set Ω such that $D = \min \{ f(X) = g_1(X) - h_1(X) : X \in \Omega \}$, an equivalent DC program can be found by letting $g(X) = g_1(X) + \chi_\Omega(X)$ and $h(X) = h_1(X)$ where $\chi_\Omega(X) = 0$ if $X \in \Omega, +\infty$ otherwise. Consider the dual program

$$D = \min \left\{ f^*(Y) = h^*(Y) - g^*(Y) : Y \in \mathbb{R}^N \right\} \tag{5.24}$$

where h^* is the conjugate function of h. It is proven in [39] that an optimal value to the primal problem is also an optimal value for its dual. Thanks to the symmetry of duality and local optimality conditions [39], an iterative primal-dual algorithm is presented in [40] which constructs two alternating sequences $\{X^{(t)}\}$ and $\{Y^{(t)}\}$ such that $g(X^{(t)}) - h(X^{(t)})$ and $g^*(Y^{(t)}) - h^*(Y^{(t)})$ are decreasing and converging to the optimal solutions, X^* and Y^*, to the primal and dual problems, respectively.

At the t^{th} iteration of the algorithm, the primal problem in (5.23) is replaced with

$$P^{(t)} = \min \left\{ g(X) - [h(X^{(t)}) + (X - X^{(t)})^T Y^{(t)}] : X \in \mathbb{R}^N \right\} \tag{5.25}$$

where $h(X) := h(X^{(t)}) + (X - X^{(t)})^T Y^{(t)}$ is the linearization of the second convex function h and $Y^{(t)} \in \nabla h(X^{(t)})$ is a subgradient of h at $X^{(t)}$. The set of solution to this modified problem is $\nabla g^*(Y^{(t)})$. Then, a solution $X^{(t+1)} \in \nabla g^*(Y^{(t)})$ can be used to replace the dual problem (5.24) similarly

$$D^{(t+1)} := \min \left\{ h^*(X) - [g^*(Y^{(t)}) + (Y - Y^{(t)})^T X^{(t+1)}] : Y \in \mathbb{R}^N \right\} \tag{5.26}$$

where $g^*(X) := g^*(Y^{(t)}) + (Y - Y^{(t)})^T X^{(t+1)}$ is the linearization of the second convex function g^*. Obviously, the set of solutions to this problem is $\nabla h(X^{(t+1)})$ which can be used to find a subgradient for linearizing h in (5.23) in the next

iteration. Thus, the update rules at each two-step iteration can be given as

$$\begin{aligned} Y^{(t)} &\in \nabla h(X^{(t)}), \quad \forall t \geq 0 \\ X^{(t+1)} &\in \nabla g^*(Y^{(t)}), \quad \forall t \geq 0. \end{aligned} \tag{5.27}$$

Starting with a feasible initial solution $X^{(0)}$, minimization with these update rules has known convergence properties [40]. A simplified algorithm is given in Algorithm 4. It is worth emphasizing that lines 4 and 5 are convex optimization problems that can be solved with efficient algorithms.

Algorithm 4 DC algorithm.

Input: Convex functions g, h defined on $\mathbb{R}^N$
Output: X^*
1: $t \leftarrow 0$
2: Initialize $X^{(t)} \in \mathbb{R}^N$
3: **repeat**
4: Choose $Y^{(t)} \in \nabla h(X^{(t)})$
5: Choose $X^{(t+1)} \in \nabla g^*(Y^{(t)})$
6: $t \leftarrow t + 1$
7: **until** $\| X^{(t)} - X^{(t-1)} \|_\infty \leq \epsilon$
8: $X^* \leftarrow X^{(t)}$

5.3.3 Difference of Convex Programming Inference

Now consider the original problem in (5.21). First, a linear programming relaxation is applied to the $0-1$ integer program so that $0 \leq x \leq 1$ in (5.21). Since W is not assumed positive semidefinite, in general, the resulting linearly constrained quadratic problem is not convex. Thus, the objective function and the constraints in (5.21) can be reformulated as a difference of two convex functions defined on the real-space and can be solved with a dedicated DC algorithm described below.

Let $W = Q\Lambda Q^T$ be the eigenvalue decomposition of W, and $\Lambda^+ = diag(\Lambda_1^+, \ldots, \Lambda_N^+)$ (respectively, $\Lambda^- = diag(\Lambda_1^-, \ldots, \Lambda_N^-)$) be the positive semidefinite diagonal matrix (respectively, negative semidefinite diagonal matrix) of Λ where $\Lambda_i^+ = \max\{0, \Lambda_i\}$ (respectively, $\Lambda_i^- = \min\{0, \Lambda_i\}$). The main idea is to rewrite the non-convex symmetric quadratic objective function as the difference

of two semidefinite quadratic functions [41] as

$$\begin{aligned} \frac{1}{2}X^T W X + q^T X &= g(X) - h(X) \\ g(X) &= \frac{1}{2}X^T W^+ X + q^T X + \chi_\Omega(X) \\ h(X) &= -\frac{1}{2}X^T W^- X \end{aligned} \tag{5.28}$$

where $W^+ = Q\Lambda^+ Q^T$, $W^- = Q\Lambda^- Q^T$, and $\chi_\Omega(X)$ is assumed to cover the space enclosed by the constraint $X \in \{0, 1\}$ together with the remaining constraints Ω. Based on these function assignments, update equations for a DC algorithm can be derived as

$$\begin{aligned} Y^{(t)} &\in W^- X, && \forall t \geq 0 \\ X^{(t+1)} &\in \underset{\substack{X \in \Omega \\ 0 \leq X \leq 1}}{\arg\min} \frac{1}{2}X^T W^+ X + q^T X - X^T Y^{(t)}, && \forall t \geq 0. \end{aligned} \tag{5.29}$$

These update equations give rise to Algorithm 5 for solving the constrained quadratic programming problem in (5.21). Note that steps 3 and 6 are convex quadratic minimization problems with convex constraints each of that can be optimized effectively by particular methods. In this study, we prefer use the well-known primal-dual interior-point algorithm [42] which is considered generally more fast for large-scale sparse problems. Finally, the regions that should be added to the set of detected regions are determined by thresholding the optimal X variables. The constrained problems that are solved by using Algorithm 5 are demonstrated in detail in Section 6.4.

Algorithm 5 DC algorithm for constrained quadratic programming formulation for estimating X^*.

Input: W, q, Ω
Output: X^*

$t \leftarrow 0$
Decompose W into W^+ and W^-
Initialize $X^{(t)} = \underset{\substack{0 \leq X \leq 1 \\ X \in \Omega}}{\arg\min} \frac{1}{2} X^T W^+ X + q^T X$
repeat
    $Y^{(t)} \leftarrow W^- X^{(t)}$
    $X^{(t+1)} \leftarrow \underset{\substack{0 \leq X \leq 1 \\ X \in \Omega}}{\arg\min} \frac{1}{2} X^T W^+ X + q^T X - X^T Y^{(t)}$
    $t \leftarrow t + 1$
until $\| X^{(t)} - X^{(t-1)} \|_\infty \leq \epsilon$
$X^* \leftarrow X^{(t)}$

Chapter 6

Experiments

First of all, we explain our evaluation criteria that are used to evaluate the accuracy of the algorithm quantitatively in Section 6.1 and then, present detailed results under the following subsections:

- Using a single layer without imposing any constraint (Section 6.2),
- using multiple layers without imposing any constraint (Section 6.3),
- using a single layer by imposing constraints (Section 6.4),
- using multiple layers by imposing constraints (Section 6.5).

6.1 Evaluation Criteria

The detection scores resulting from the inference procedure consist of the marginal probabilities of the selected regions (primitives) at the end of Algorithm 3 and 5. Thresholding of the score of each region produces a binary detection map. We use precision and recall as the quantitative performance criteria as in [3, 43] to compare the binary detection maps obtained using a uniformly sampled range of thresholds to the validation data for each scenario that is described in Section

6.2.1. Recall (producer's accuracy), that is computed as the ratio of the number of correctly detected pixels to the number of all pixels in the validation data, can be interpreted as the number of true positives detected by the algorithm, while precision (user's accuracy), that is computed as the ratio of the number of correctly detected pixels to the number of all detected pixels, evaluates the algorithm's tendency for false positives. In addition to the precision-recall curves that used a full range of thresholds, we used a particular threshold value of 0.9 to provide example detection results for all scenarios in the following sections. We observed that the particular choice for this threshold was not very critical because, as discussed in the following sections, the inference procedure assigned very high probabilities to most of the selected regions.

Since our selection algorithm detects regions instead of individual pixels, we also performed an object-based evaluation as in [44] in addition to pixel-based evaluation. This strategy, which is called focus-of-attention, assumes that a single correctly detected pixel inside a target object is sufficient to attract the operator's attention to that target and label it as correctly detected, but any pixel outside the target is a false alarm because it diverts attention away from true targets. Given the binary detection map for a particular threshold, the union of one or more pixels inside the mask of a validation (ground truth) region was counted as a true positive, while the number of connected components of pixels that did not overlap with any validation region was counted as false positives. Precision and recall used counts of connected groups of pixels instead of individual pixels for object-based evaluation.

6.2 Unconstrained Single-layer Experiments

In the first set of experiments, the compound structure consists of primitive objects from a single primitive layer and we do not assume any constraint on these primitives. The following subsections present the data set used in these experiments (Section 6.2.1), the details of the experimental protocol performed to generate the results (Section 6.2.2), the baseline methods used for comparison

Figure 6.1: Ankara data set used for quantitative evaluation.

with the proposed approach (Section 6.2.3), and finally, the results (Section 6.2.4).

6.2.1 Data Set

The main experiments for quantitative and qualitative evaluation were performed using a multispectral WorldView-2 image of Ankara, Turkey. The test scene consisted of 4000×2500 pixels and 2 m spatial resolution covering various kinds of residential and industrial areas as shown in Figure 6.1.

The proposed compound structure detection algorithm was evaluated using six scenarios where the first five scenarios correspond to residential structures and the sixth one corresponds to an industrial structure as shown in Table 6.1. All scenarios were formed by various arrangements of four buildings used as the main primitive object of interest in the urban test scene. In particular, the first scenario aimed the detection of rectangular buildings that are spatially aligned with respect to their major axes. The second scenario aimed the detection of

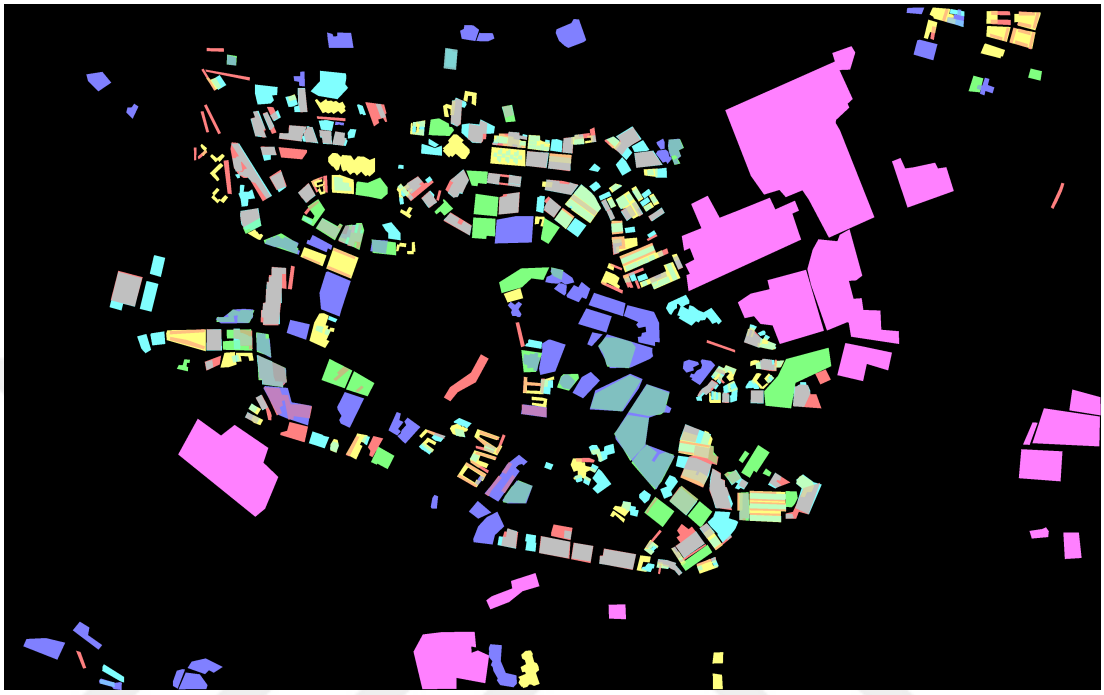
Table 6.1: Detection scenarios for the experiments. Example primitives used for learning the compound structure model for each scenario are shown in a different color. The number of polygons and buildings in the validation data are also given.

Scenario	1	2	3	4	5	6
Example primitives						
# polygons	162	98	48	195	60	16
# buildings	1519	870	1117	1796	771	219

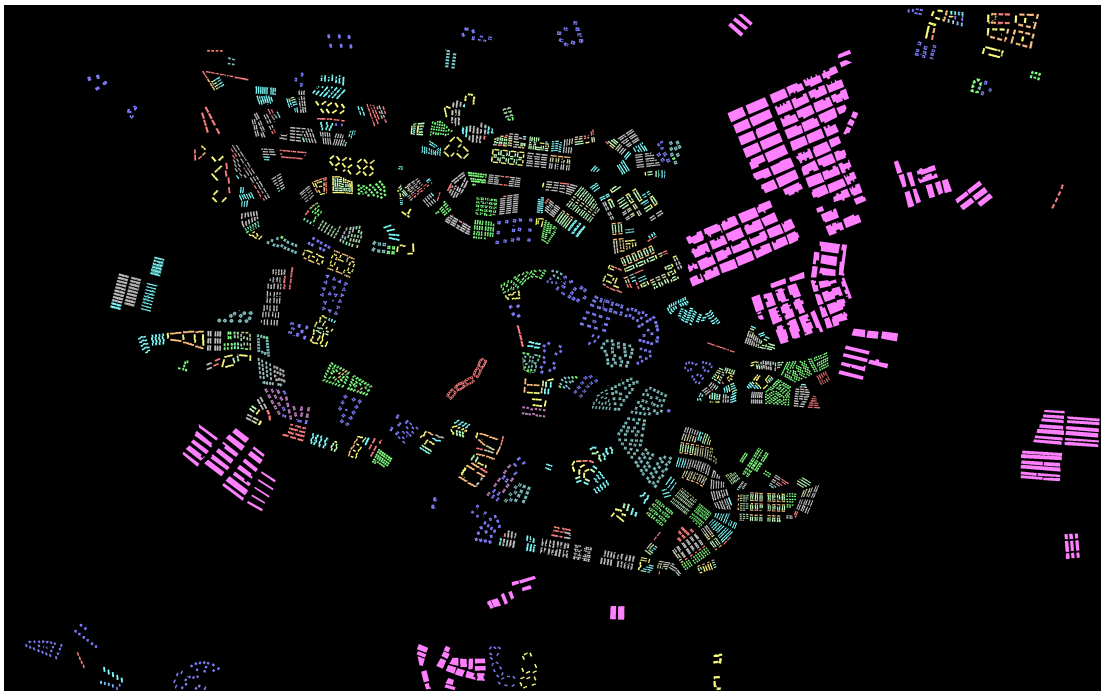
a structure composed of buildings placed in a diamond formation. The third scenario aimed the detection of relatively small, dense, regularly arranged, square-like buildings. The fourth scenario aimed the detection of parallel rectangular buildings that are aligned with respect to their minor axes. The fifth scenario aimed the detection of sparse, randomly located, square-like buildings that are slightly larger than those in scenario three. The sixth scenario aimed the detection of a structure composed of regularly arranged large industrial buildings.

The validation data that were used to evaluate the performance of the method on these scenarios were obtained by manual delineation of polygons corresponding to compound structures (Figure 6.2(a)) as well as buildings inside these polygons as primitive objects (Figure 6.2(b)). Table 6.1 presents the number of compound structures (polygons) and the corresponding primitives (buildings) in the validation data for each scenario. The learning process for building the compound structure model uses manual selection of four of these primitives for each structure of interest. This corresponds to triggering the whole learning and inference process using only four individual objects, and can be considered a very moderate requirement as only a few individual objects need to be delineated as opposed to relatively large training sets needed for supervised detection and classification algorithms.

In addition to the quantitative experiments using the urban scene in the WorldView-2 image presented in this section, we performed qualitative evaluation by using two additional very high spatial resolution images to illustrate the effectiveness of the proposed approach in detecting different compound structures that are composed of different primitive objects in other types of settings such as



(a)



(b)

Figure 6.2: (a) Manually delineated polygons reflecting compound structures of interest. (b) Manually delineated buildings inside these polygons. These buildings are used as the primitives in the validation data. The colors of the polygons and buildings correspond to the scenarios given in Table 6.1.

agricultural and rural scenes. In particular, we used a multispectral 8000×3000 pixel WorldView-2 image of Kusadasi, Turkey (Figure 6.3) for the detection of fruit orchards as agricultural structures composed of trees as the primitive objects, and a panchromatic 1102×971 pixel GeoEye-1 image of Darfur, Sudan with 0.5 m spatial resolution for the detection of refugee camps as rural structures composed of fences as the primitive objects.

6.2.2 Experimental Protocol

The experimental procedure for building the example compound structure model (Section 3) and learning its parameters (Section 4) used a single example structure ($M = 1$) with only four primitive objects ($N = 4$) as described above. The proximity threshold δ was set to 100 pixels. The corresponding arrangement and shape histograms were constructed with five equal length bins between the minimum and maximum possible values for each feature. Thus, the total number of bins in $H(V)$ was 45. The minimum and maximum major and minor axis lengths (s^h_{min}, s^h_{max}) and (s^w_{min}, s^w_{max}) for sampling the ellipses were both set to $(2, 80)$. This interval was chosen so that it covered the expected smallest and largest primitive axis lengths. The parameters of the maximum entropy model $p(V|\beta)$ were obtained using Algorithm 1. The number of samples S that were used to approximate the expectation $\mathbb{E}_p[H(V)]$ was set to 20. The number of Gibbs sampler iterations T in Algorithm 2 was set to 100.

The experimental procedure for inference and region selection (Section 5) starts with morphological profiles for hierarchical region extraction. For residential structures, disk structuring elements with radii 2 and 3 were used for constructing the closing profile of the saturation band of the HSV color space computed from the RGB bands of the multispectral image, and for the industrial structures, disk structuring elements with radii from 5 to 10 were used for constructing the opening profile of the HSV value band, as these bands gave good contrast for the primitives of interest (i.e., red roof buildings and industrial buildings, respectively) in our image. A tree structure was constructed from the corresponding

Figure 6.3: Kusadasi data set used for qualitative evaluation.

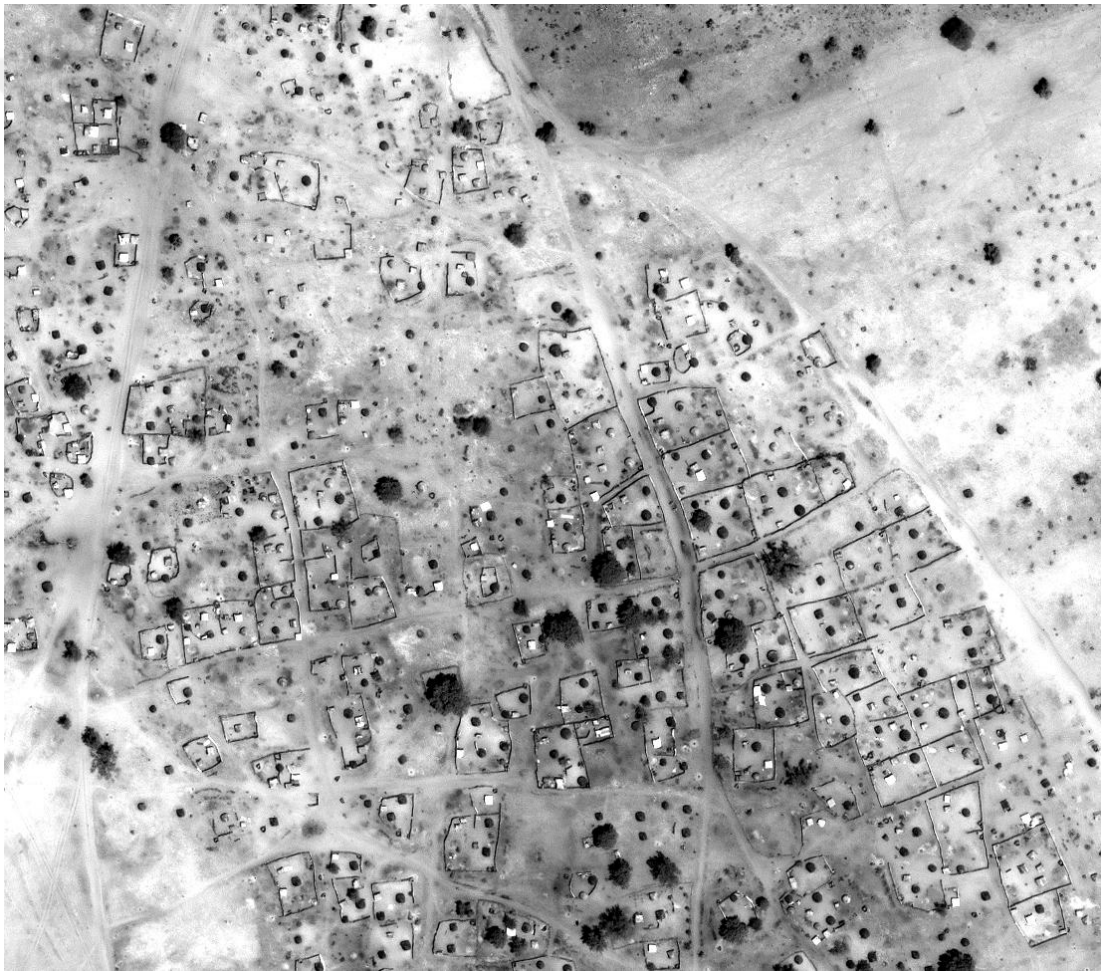


Figure 6.4: Darfur data set used for qualitative evaluation.

profile to extract candidate regions for each scenario. For the residential structures, the number of candidate regions N in two scales was 70,644, and for the industrial structures, the number of candidate regions in six scales was 22,195. This makes a very large pool of candidate regions that we should select from as shown in Figure 6.5. A Voronoi neighborhood between regions was constructed for each scale and the neighbors of a region at lower scales were obtained through its descendants in these scales. The resulting graph constructed for the residential scenarios contained 752,754 edges whereas the graph constructed for the industrial scenario contained 490,222 edges. Considering the total number of the candidate regions in all scales and the number of regions in the validation data, the challenge for the selection problem is that it is expected to select a significantly small fraction of these candidate regions; hence, it should be very selective. Finally, the simulated annealing procedure that was used to help the convergence of Algorithm 3 divided the exponents in the posterior probability in (5.7) by a certain power called temperature. This temperature was slowly decreased in each iteration according to a cooling schedule such that $\tau_k = 0.995\,\tau_{k-1}$ where the initial temperature τ_0 was set to 1.

6.2.3 Baselines for Comparison

The first baseline method used sliding windows similar to the tile-based classification tasks in the literature. In particular, we used overlapping 150×150 pixel windows, and using all primitive objects in each window, we extracted marginal histograms, $H(V)$, as described in Section 3.2, that modeled the shape and arrangement characteristics of the primitives at each scale of the hierarchy. Then, we computed the probability that a particular spatial arrangement existed in that window by using $p(V|\beta)$, as described in Section 3.3, for each scale, and obtained the overall probability for each window as the maximum of the probabilities obtained from all scales. Finally, the marginal probability for each primitive object was obtained as the maximum of the probabilities of the windows that it appeared. This baseline method aimed to evaluate the effectiveness of the proposed selection process by combining the shape and arrangement information from all

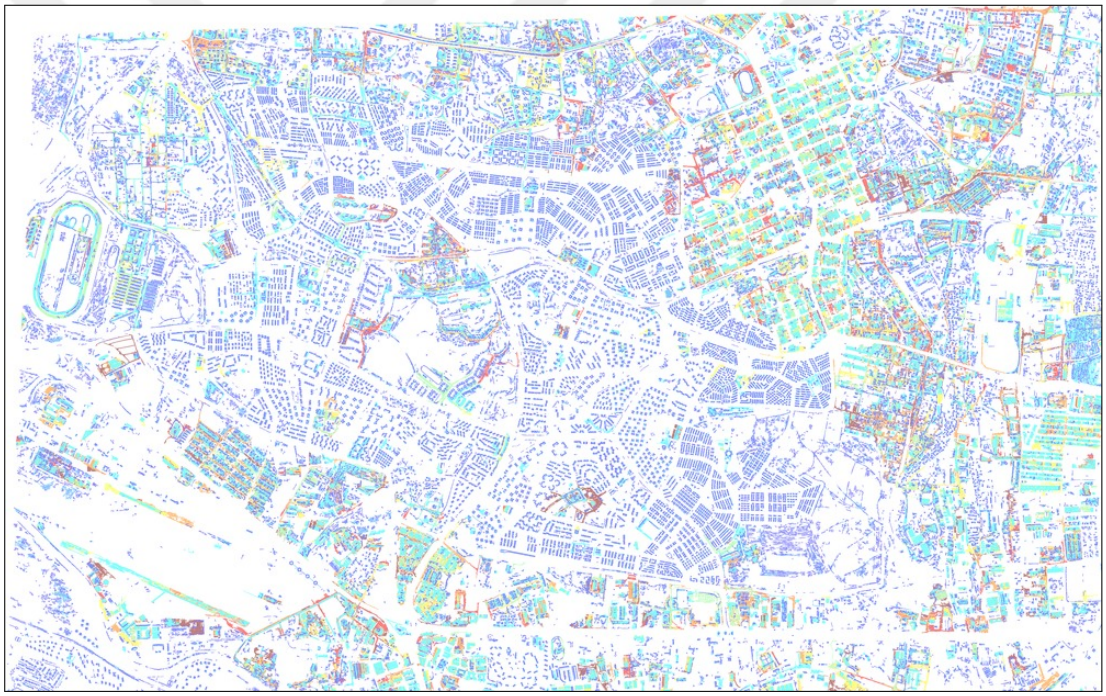


Figure 6.5: Candidate regions obtained by the morphological profile hierarchy. Regions appearing in different levels of the hierarchy are shown with different pseudocolors.

primitives.

The second baseline method performed selection of regions satisfying only color and shape properties by dropping the arrangement terms in the maximum entropy model. Thus, the baseline result was obtained by computing the probability of the candidate regions as $p(X|I,V) \propto \frac{1}{Z_x} \prod_{v_i \in V} \exp\left\{ \left(\psi_i^c + \psi_i^s\right) x_i \right\}$ instead of (5.2). This choice for the baseline aimed to evaluate the effectiveness of the generic spatial arrangement model in the proposed probabilistic region process compared to the commonly used color and shape-only detectors.

6.2.4 Results

The learning and inference procedures summarized in Algorithms 1 and 3, respectively, were run for each of the six scenarios on the data set described in Section 6.2.1. The number of selected regions were 3191, 1828, 3819, 3201, 2027, and 1612 for each scenario, respectively. To reconcile selection of overlapping regions from multiple scales, we computed the maximum of the marginal probability values for each pixel along all scales that it was selected. This operation reduced the number of resulting regions to 1920, 1114, 2648, 1934, 1399, and 357, respectively. These numbers showed that, on the average, only 4% of all candidate regions in all scales were selected for all scenarios. This meant that most of the regions in the input hierarchy were considered as irrelevant by the proposed method that behaved very selectively even when trained with a single example structure that contained only four buildings for each scenario.

Figures 6.6, 6.7 and 6.8 show the marginal probabilities of the selected regions for each scenario. Recall that concurrent selection of overlapping regions at multiple levels was allowed by the inference algorithm. So, the selected regions at multiple levels were merged in one level by assigning each pixel the largest marginal probability of the regions that it belonged to. This caused assignment of different marginal probabilities to different pixels in the same connected component. The merged results showed that our selection algorithm was able to

detect coherent regions in the image that had arrangements similar to the example structures. Note that a region may belong to more than one type of compound structure as it may form different arrangements with different neighbors. For example, a region may have both close and distant neighbors, and may be aligned with different neighbors according to the major and the minor axes at the same time.

We observed high marginal probabilities, e.g., greater than 0.9, for most of the selected regions. This indicated that most of the selected regions appeared in most of the sampling iterations, and showed the power of our sampling procedure compared to the traditional Gibbs sampler that samples an individual region at a time by considering only its neighbors. The latter has a potential problem for regions with several irrelevant neighbors that increase the uncertainty in the decision to flip the selection label of a region, whereas our sampling algorithm that sampled connected components and made the decision for a particular region by the contribution of a larger context that contained other regions that might be part of the same structure behaved very selectively. This difference was especially more clear for the boundary regions of compound structures where the marginal probabilities of the boundary regions were as high as the ones in the middle since their decisions were made together through their corresponding connected components.

The next set of experiments was done to compare the performances of the proposed detection algorithm and the baseline methods as described in Sections 6.2.2 and 6.2.3, respectively. Figures 6.9 and 6.10 shows precision versus recall curves obtained by applying different thresholds to the marginal probabilities. The results showed that the proposed algorithm that jointly exploited spectral, shape and arrangement information performed significantly better than the baselines that did not use either selection or arrangement. Even though the two less restricted baselines could approach higher recall levels (bottom right corner of the precision-recall curves) with a sacrifice of substantially reduced precision by accepting more buildings in the output, the proposed method could achieve significantly higher precision values at the same level of recall. The observation that the baseline that used shape and arrangement without selection performed worse

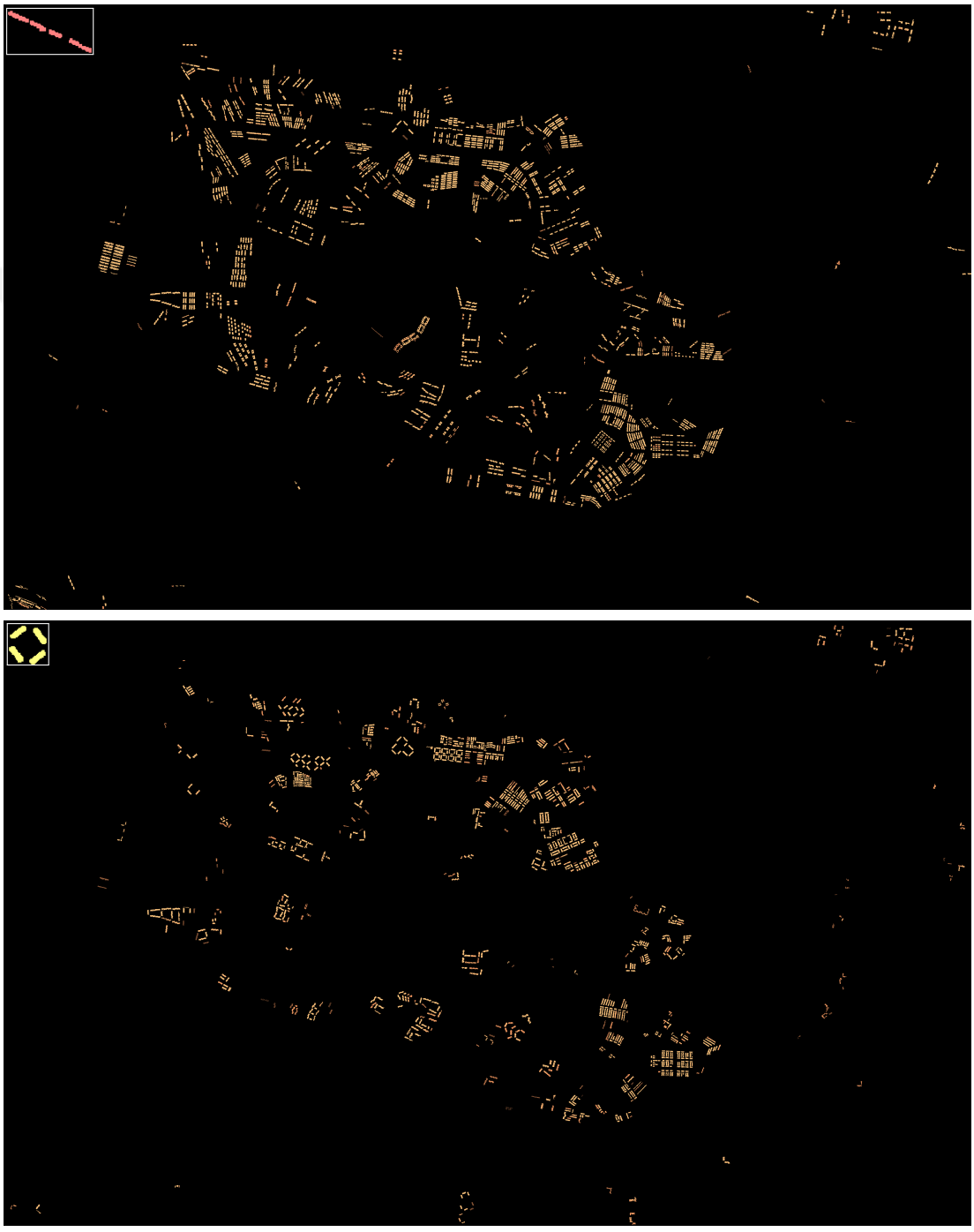


Figure 6.6: Marginal probabilities for the selected regions for the first and second scenarios. Brighter values indicate higher probabilities. The example primitives are also shown.

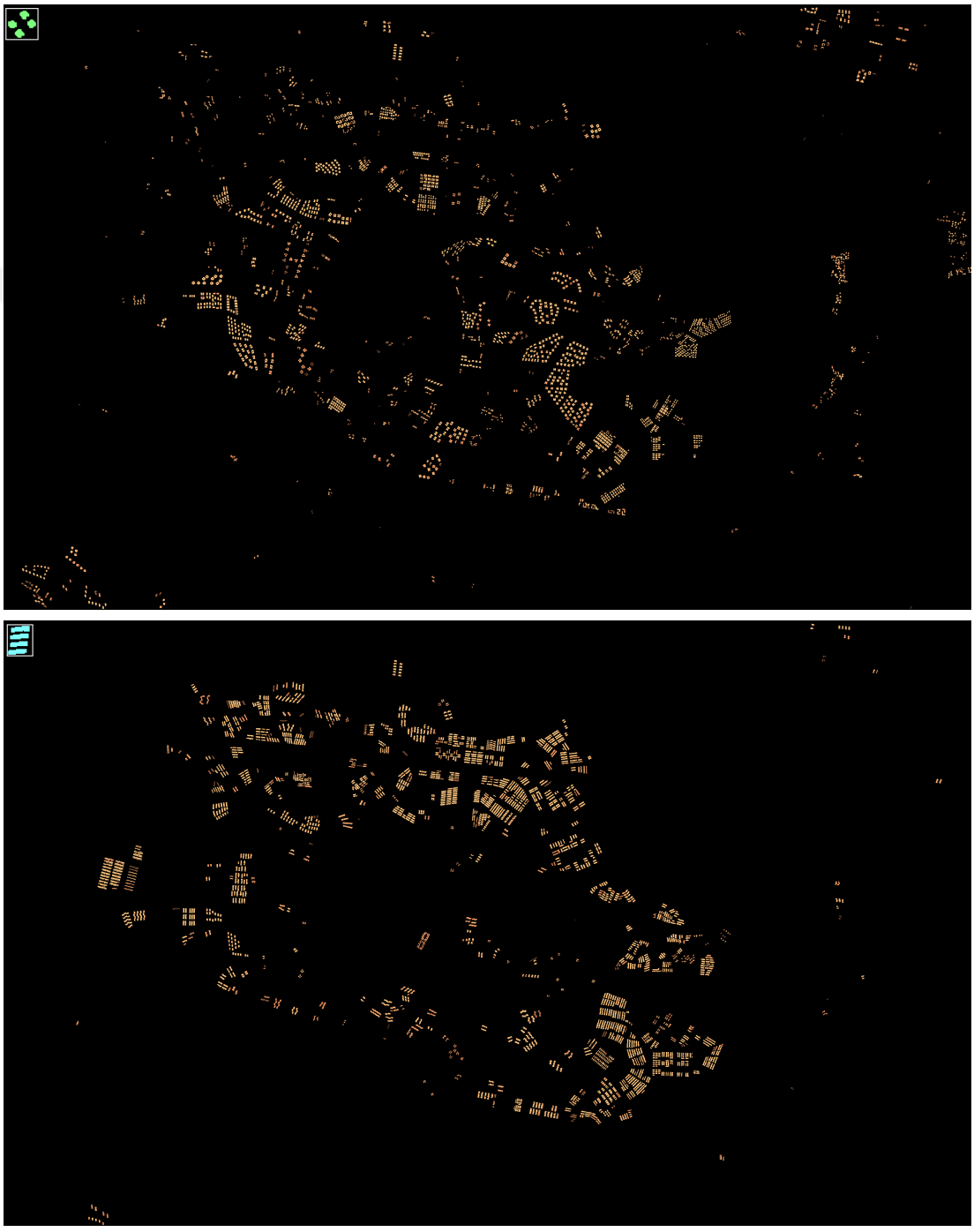


Figure 6.7: Marginal probabilities for the selected regions for the third and fourth scenarios. Brighter values indicate higher probabilities. The example primitives are also shown.

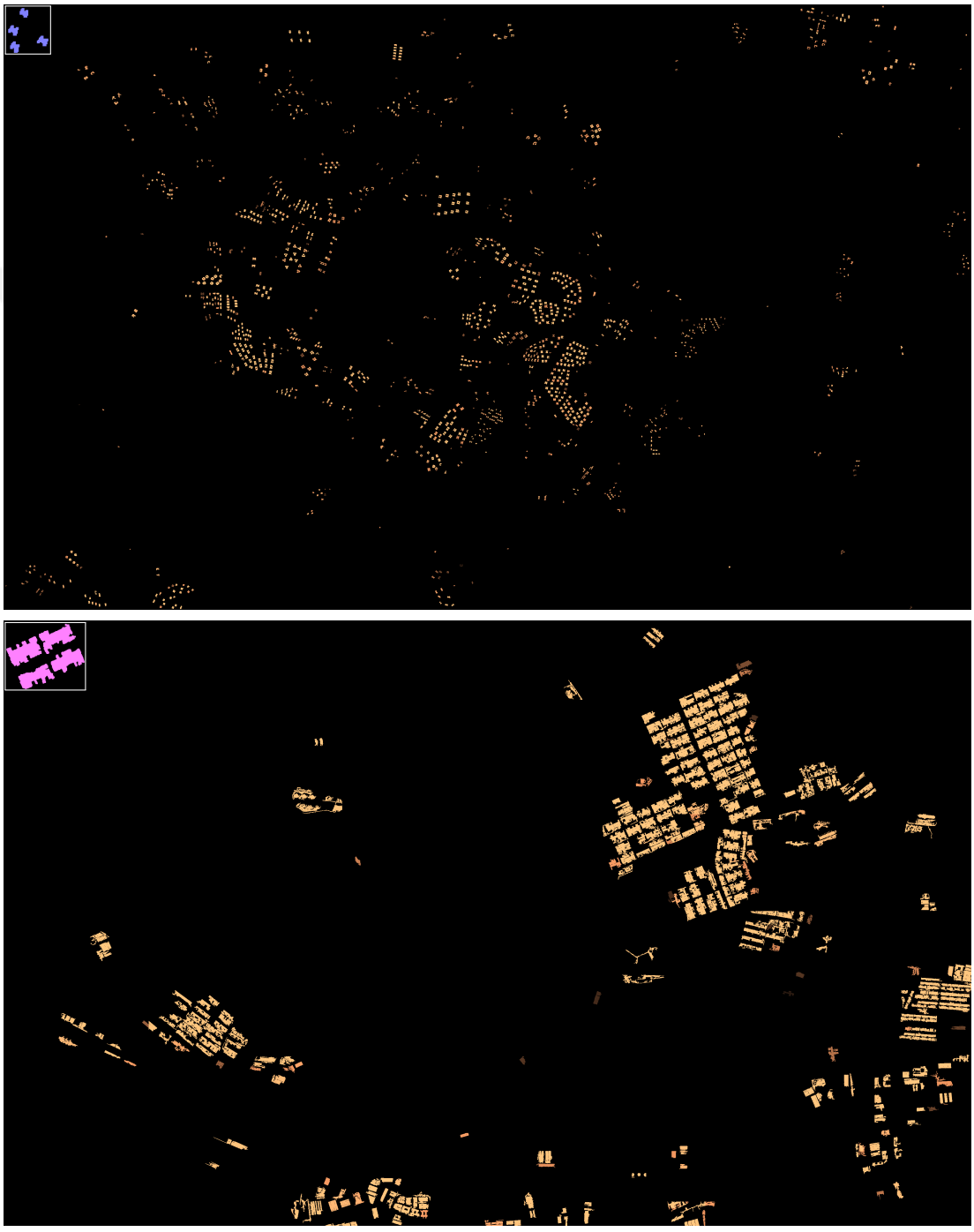


Figure 6.8: Marginal probabilities for the selected regions for the fifth and sixth scenarios. Brighter values indicate higher probabilities. The example primitives are also shown.

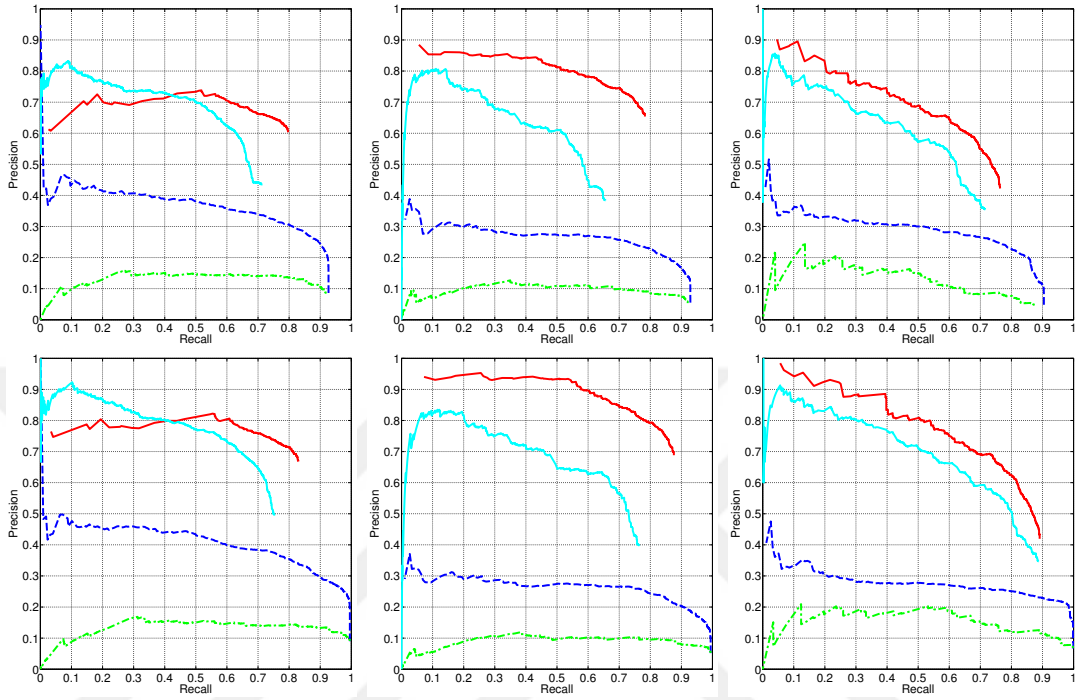


Figure 6.9: Precision-recall curves for the first three scenarios. The columns correspond to the scenarios one to three from left to right. The top row corresponds to the pixel-based evaluation and the bottom row is for the object-based evaluation. The solid red curves correspond to the proposed approach, dashed green ones are for the first baseline (shape and arrangement without selection), and dashed blue ones are for the second baseline (color and shape-only selection with no arrangement). The solid cyan curves will be explained in Section 6.4.

than the one that used color and shape-only selection with no arrangement also confirmed the effectiveness of the proposed selection algorithm. When we compared the results for different scenarios, we could observe that the decreases in precision in the third and fifth scenarios were faster than the others for increasing recall (corresponding to decreasing detection threshold). This could be explained by the observation that orientation-based features for square-like buildings could be noisy so that more building groups that were not in the validation data appeared in the output as we decreased the detection threshold. This result could also be justified by a smaller ratio of the number of buildings in the validation data versus the number of selections for each of these scenarios.

We also observed that the quantitative evaluation did not always reflect the

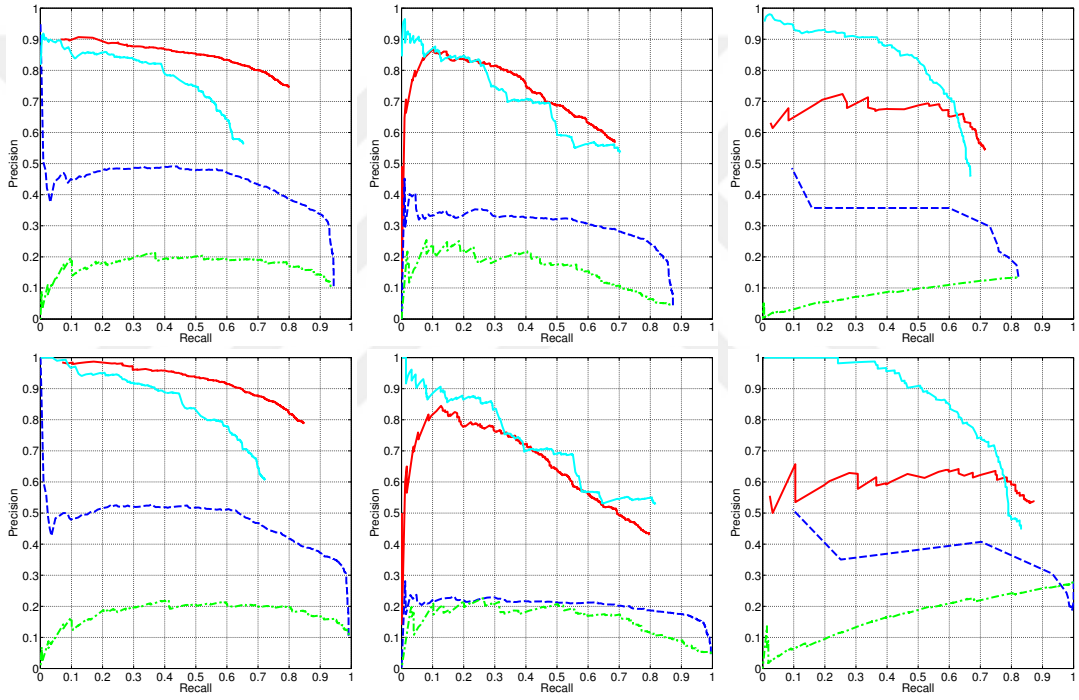


Figure 6.10: Precision-recall curves for the last three scenarios. The columns correspond to the scenarios four to six from left to right. The top row corresponds to the pixel-based evaluation and the bottom row is for the object-based evaluation. The solid red curves correspond to the proposed approach, dashed green ones are for the first baseline (shape and arrangement without selection), and dashed blue ones are for the second baseline (color and shape-only selection with no arrangement). The solid cyan curves will be explained in Section 6.4.

quality of the results very precisely because the validation data remained approximate. We present zoomed versions of the results for example areas to better illustrate the details for high-resolution imagery. Figure 6.11 shows example region hierarchies and selection results. As can be seen in the hierarchies, different regions had better arrangements with their neighbors and had better appearances in different scales with respect to the structure of interest. This fact was reflected in the algorithm by selecting only an appropriate subset of the regions on a path from a leaf region to the highest scale region. Note that misdetections would have occurred if we had manually selected only one scale or attempted to find the single best scale for all the regions. An important property of our algorithm was that it could automatically select regions from different scales. It also did not require a priori knowledge of the number of regions to be selected.

Figures 6.12 and 6.13 give a visual comparison of the results for our algorithm and the second baseline method. Figure 6.14 shows more examples of the marginal probabilities and the detections after thresholding these probabilities. The marginal probability values were very strong indicators of the goodness of the detections as the highest likelihood values were obtained for the regions that were very similar to the individual primitives in the example structures but also satisfied the spatial arrangements. On the other hand, the baseline method shown detected a wide range of individual objects without any consideration of their spatial arrangements as expected. This led to very low precision levels as well as unsatisfactory localization of the structures of interest. Furthermore, our method could select regions that would have normally been misdetected if only individual properties were used. For example, structures with diamond formation involved some candidate regions with shorter major axes than the example primitives. The baseline could not detect these regions whereas our algorithm selected them since their selection along with the others satisfied the arrangement distribution. This was a good example for demonstrating the importance of the local spatial context in the selection problem.

We also analyzed different sources of errors in the detections. One of the main reasons for the misdetections was the errors in the input hierarchical segmentation. Some target primitives were never selected because a corresponding

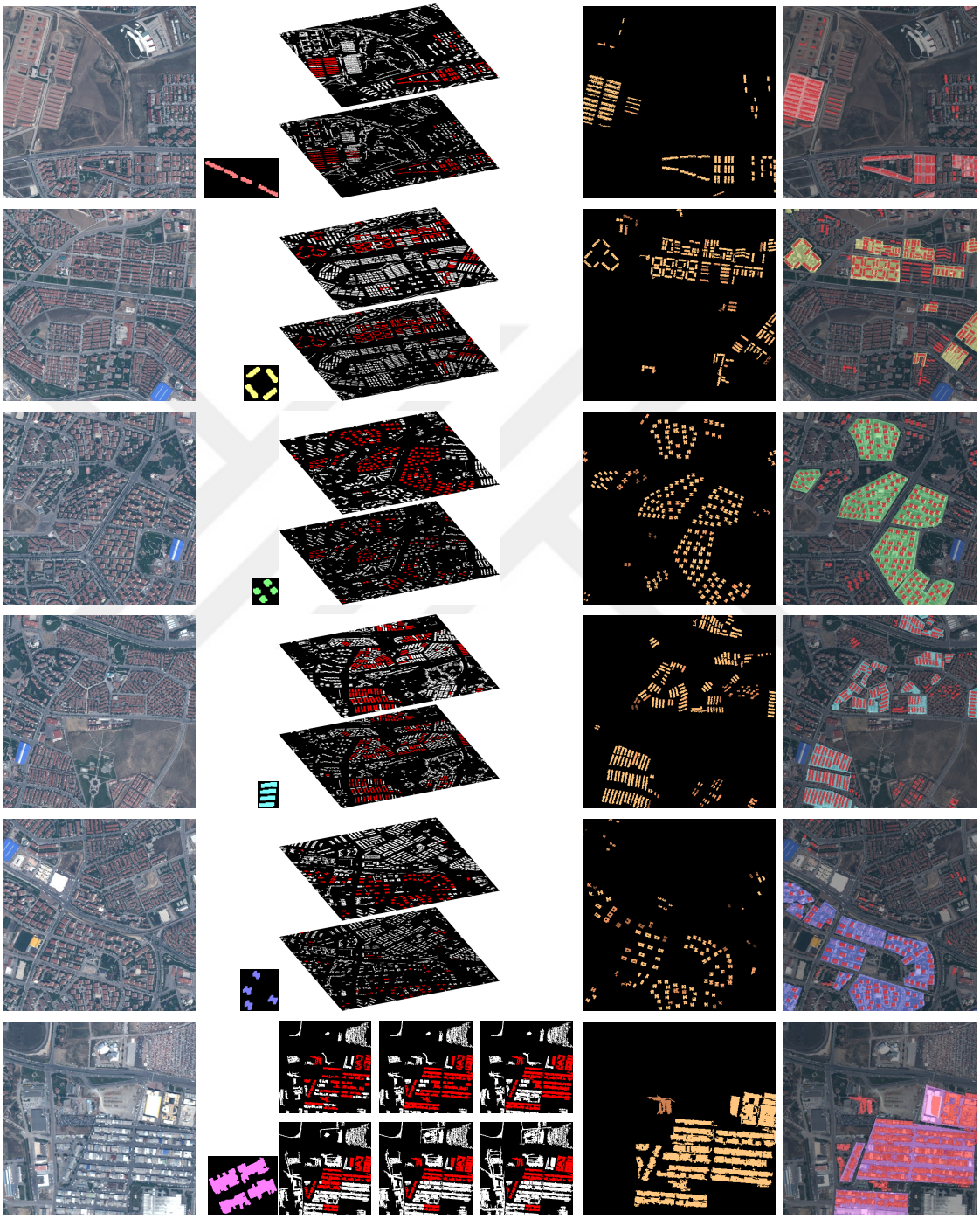


Figure 6.11: Zoomed detection examples. The first column shows the 500×500 sub-scenes. The second column shows the two-level hierarchical candidate regions for the first five rows and six-level hierarchy from left to right and bottom to top for the last row. The selected regions are colored with red. The third column shows the marginal probabilities at the end of selection. The fourth column shows the thresholded detections overlayed as red and the validation polygons overlayed with the corresponding colors in Table 6.1.

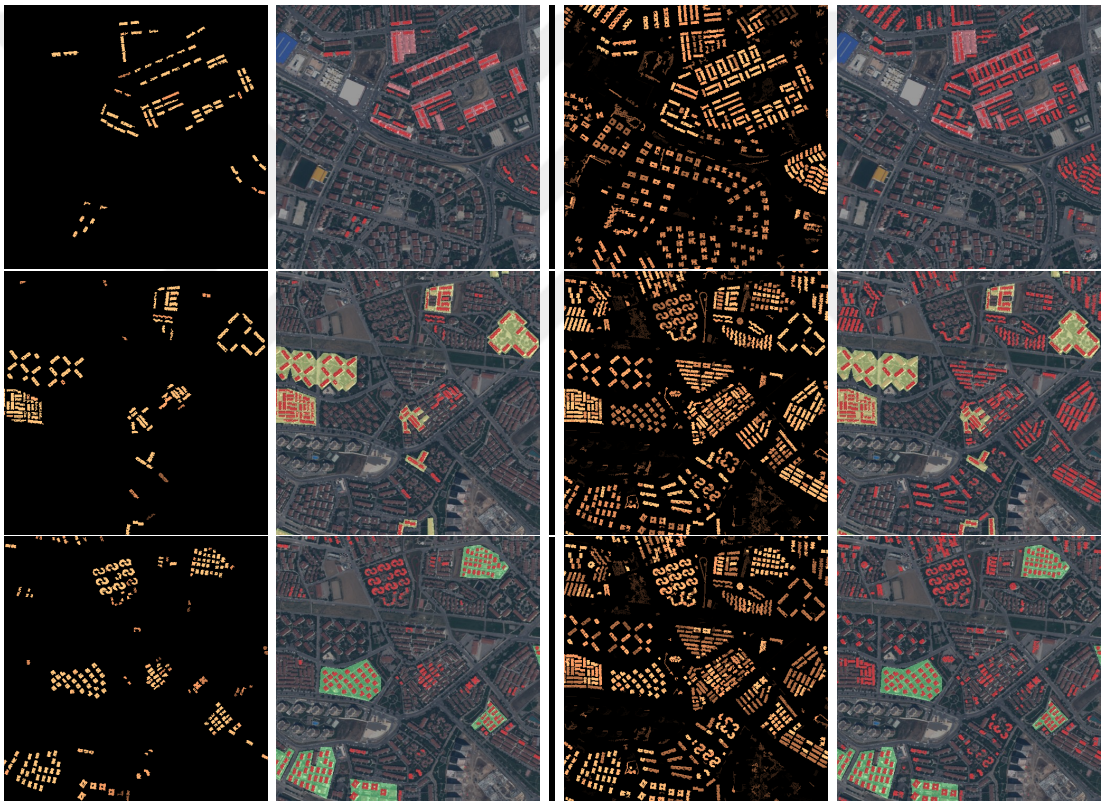


Figure 6.12: Zoomed detection examples. The image pairs show the marginal probabilities and the overlayed detection results. The first and last pairs in each row show the results of our algorithm and the second baseline, respectively.

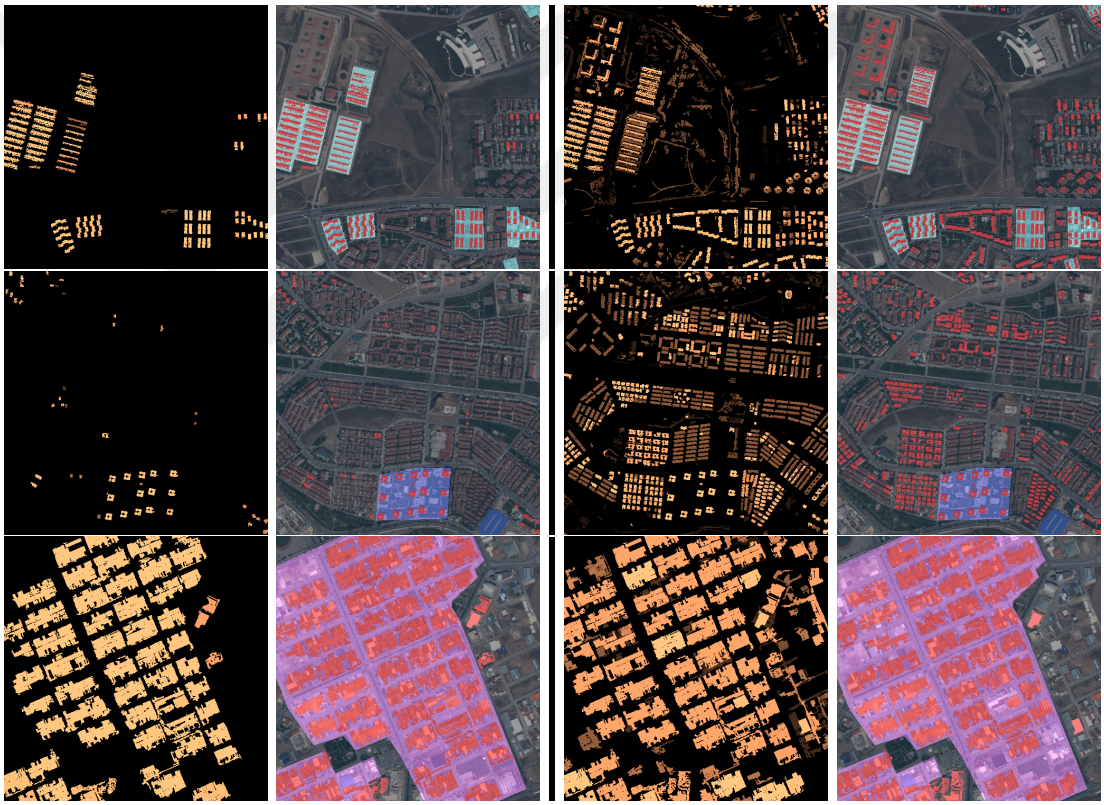


Figure 6.13: Zoomed detection examples. The image pairs show the marginal probabilities and the overlayed detection results. The first and last pairs in each row show the results of our algorithm and the second baseline, respectively.

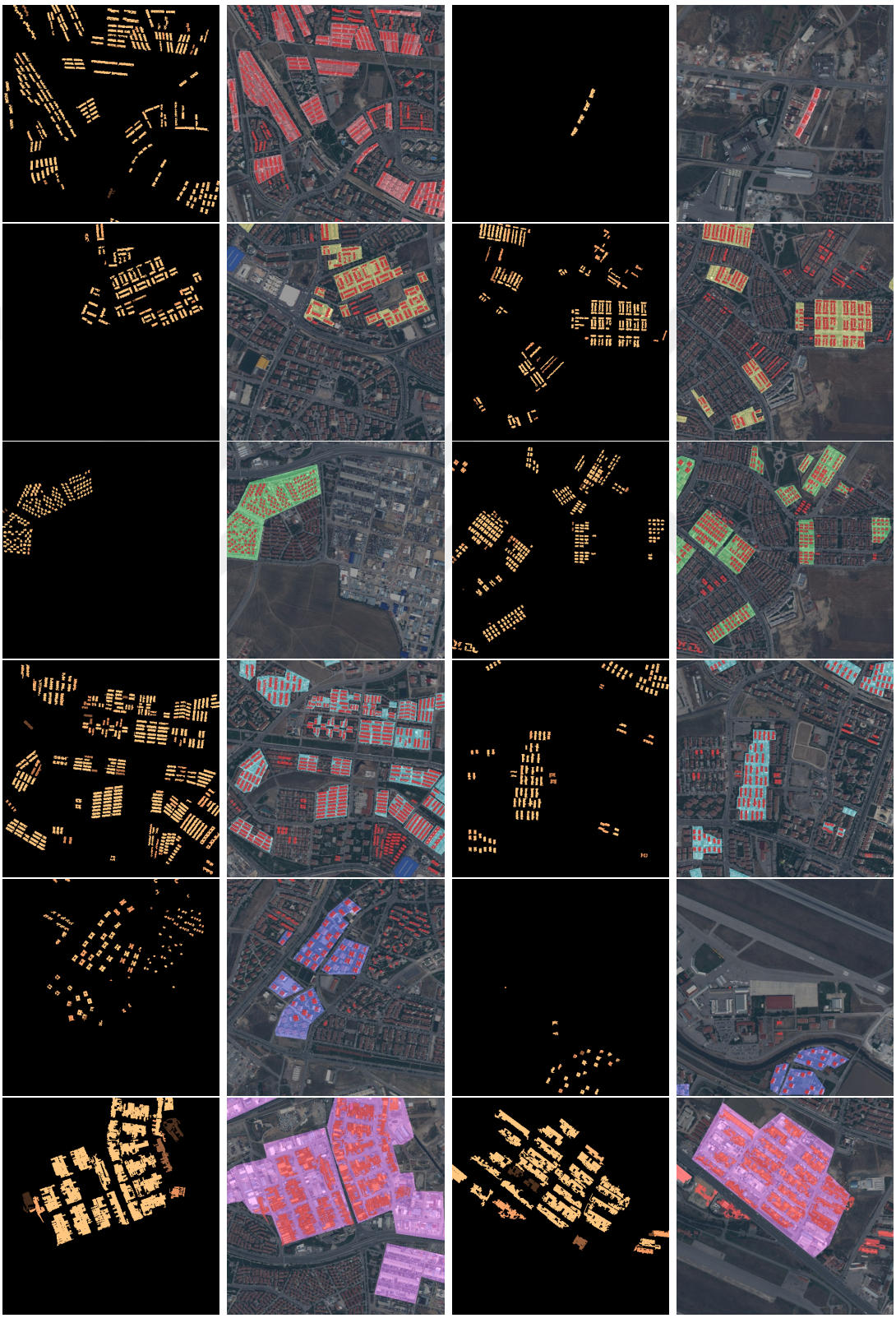


Figure 6.14: Zoomed detection examples. The image pairs show the marginal probabilities and the overlayed detection results of our algorithm. Each row corresponds to a particular scenario.

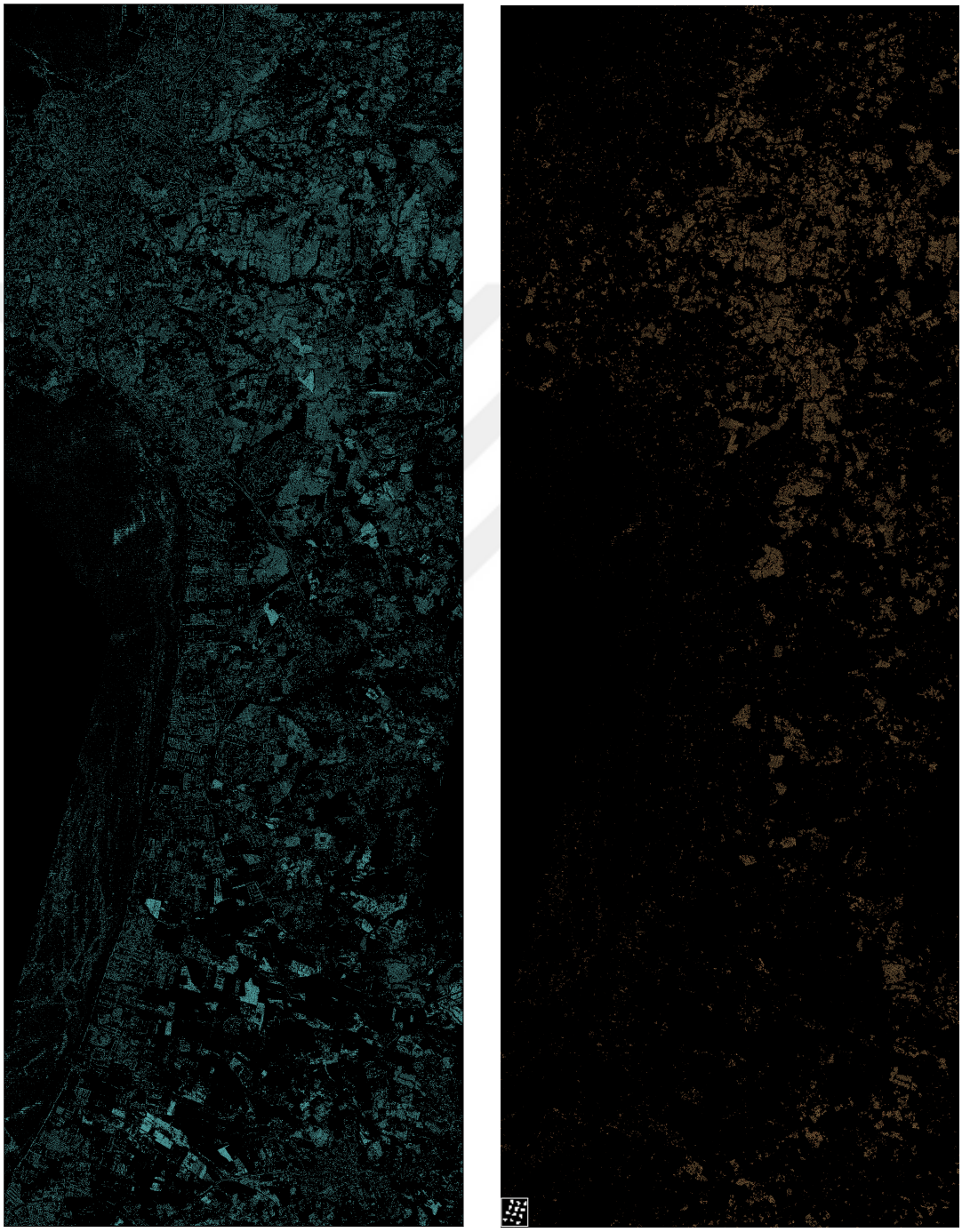
candidate region never appeared clearly in the hierarchy. That is, the candidate regions stayed too small until they merged with their surroundings and got completely lost. For example, the industrial regions had complex surfaces that made the morphological operations unable to find some of these regions precisely and prohibited the selection procedure from selecting them. Using additional hierarchical segmentations obtained by different algorithms and/or parameters can overcome this problem by introducing more than one possible set of candidates. Detailed analysis of the results revealed another reason for the misdetections where even though the arrangements of the candidate regions were satisfying the arrangement distribution of an example scenario, their color and shape properties were not supportive enough for the decision for being selected. Also, in particular, some of the misdetections for the fifth scenario occurred because the primitives were relatively distant from each other. For a candidate region in the image, its closer neighbors might have prevented the distant neighbors to appear in its Voronoi neighbors set. Then, this region was not selected in the result because it could not connect to the neighbors of interest. Some of the false alarms were caused by single individual regions that had individual statistics that were very similar to those of the example primitives so that the arrangement cues were dominated by the appearance cues. However, since the validation data were subjective, most of the regions that were reflected as false alarms could actually be accepted as true positives under different applications.

Figure 6.15 shows the input candidate regions and the marginal probabilities of the selected regions for orchard detection. Target orchards are made up of circularly shaped tree primitives appearing in a near-regular repetitive arrangement (e.g., bottom left corner of Figure 6.15(b)). Individual trees were localized as candidate regions by using the top-hat transform of the normalized difference vegetation index that had sufficient contrast between the trees and the background. We used a disk structuring element with a radius of 1 pixel in the opening operation. 179,721 out of 271,320 candidate regions were left out in the final selection results for having unlikely arrangements according to the learned probability model. The main challenge in learning and detection of orchards was

that a slight change in the number of pixels of the individual trees caused dramatical changes in their shapes due to their small sizes. This leaded to unreliable computations of some shape-oriented features such as eccentricity (ϕ_7), relative orientation (ϕ_2), antipodal proximity (ϕ_4) and relative size (ϕ_5). In general, the histograms for these features were distributed uniformly among the corresponding bins and so, were not very informative in modeling of orchards. On the other hand, the most informative feature was regularity (ϕ_9) which uses the centroids of the primitives for computation and hence, is completely shape-invariant.

Zoomed examples are presented in Figures 6.16, 6.17 and 6.18. Examination of local details show that the method was very successful in identifying the regions corresponding to orchards with only minor misdetections due to a few missing trees in the top-hat transform outputs. The boundaries of neighboring orchards were accurately detected even when no sharp boundaries existed in the image data since individual trees were selected as being member of an orchard preventing the shape assumption imposed by window-based approaches. The restrictions of the terrain on the planting pattern were successfully handled by using only the candidate regions for feature extraction excluding most of the outlier pixels. Furthermore, the granularity differences between the trees can be overcome by the multi-scale nature of the algorithm.

Example results for the detection of refugee camps are shown in Figure 6.19. The goal was to identify the refugee camps consisting of dwellings surrounded by fences made of clay or straw. The fences appear as dark rectangular outlines with one or more entrances (so that the outlines are not closed). More information about the test scene can be obtained from [45]. We aimed to model the fences in terms of spatial arrangements of line segments. Thus, we performed line fitting to the edge detection outputs, and the resulting line segments were considered as candidate primitives in the selection process. The results show that the proposed method could identify the perpendicular arrangement of the fence segments with only a few false positives. A few fence segments could not be detected because they were missing in the line fitting result. Overall, these examples illustrate that the ellipse-based primitive representation and the generic spatial arrangement model together with the proposed learning and inference algorithms were



(a) (b)

Figure 6.15: (a) Candidate regions for orchard detection obtained by the morphological profile hierarchy. (b) Marginal probabilities for the selected regions. The example primitive mask is also shown.

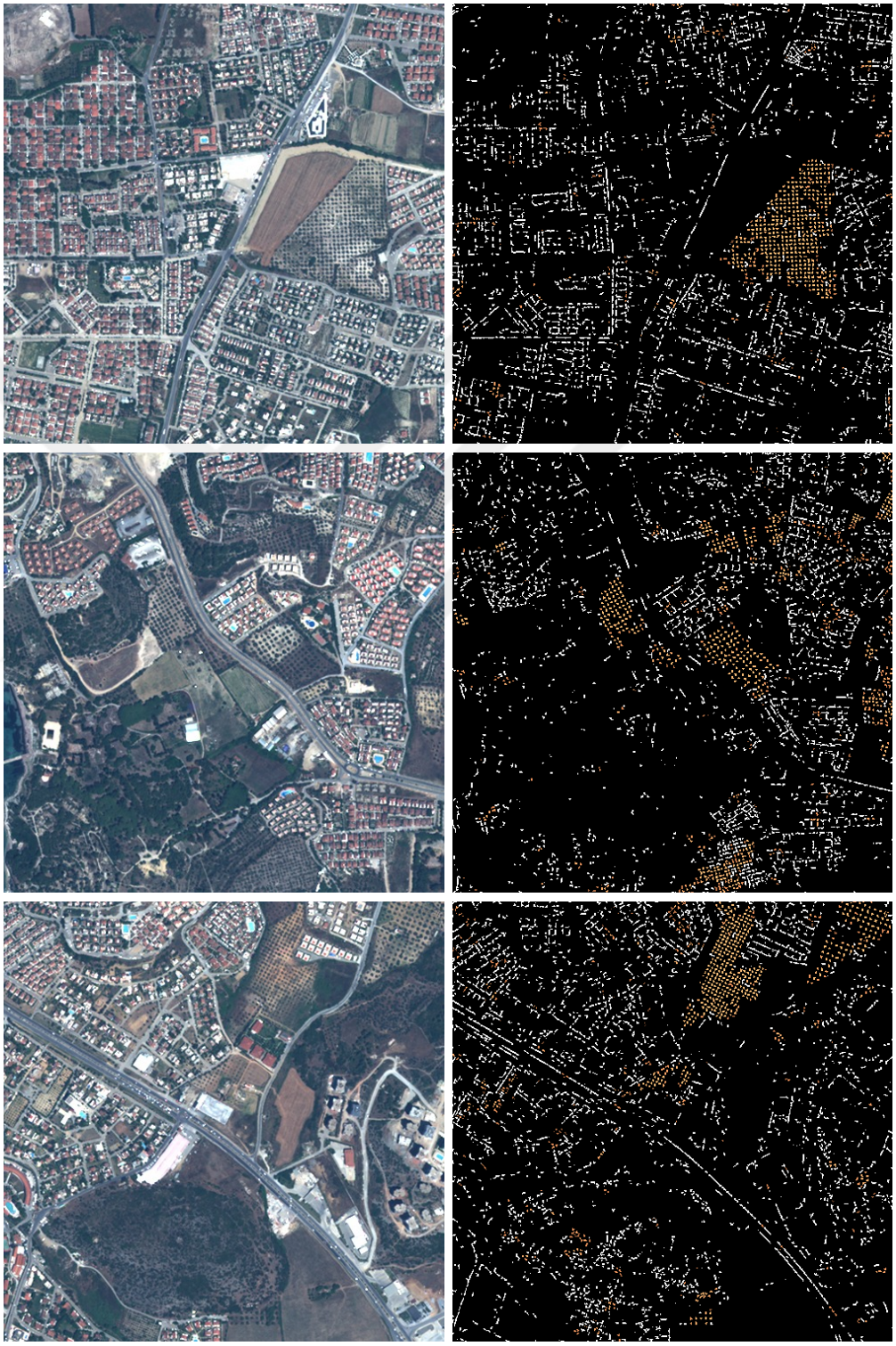


Figure 6.16: Example results for the detection of orchards as agricultural structures in three Kusadasi subimages on the left column. The right column shows the corresponding marginal probabilities of the selected regions (emphasized with the copper colormap) as well as the discarded input candidate regions (shown in white).

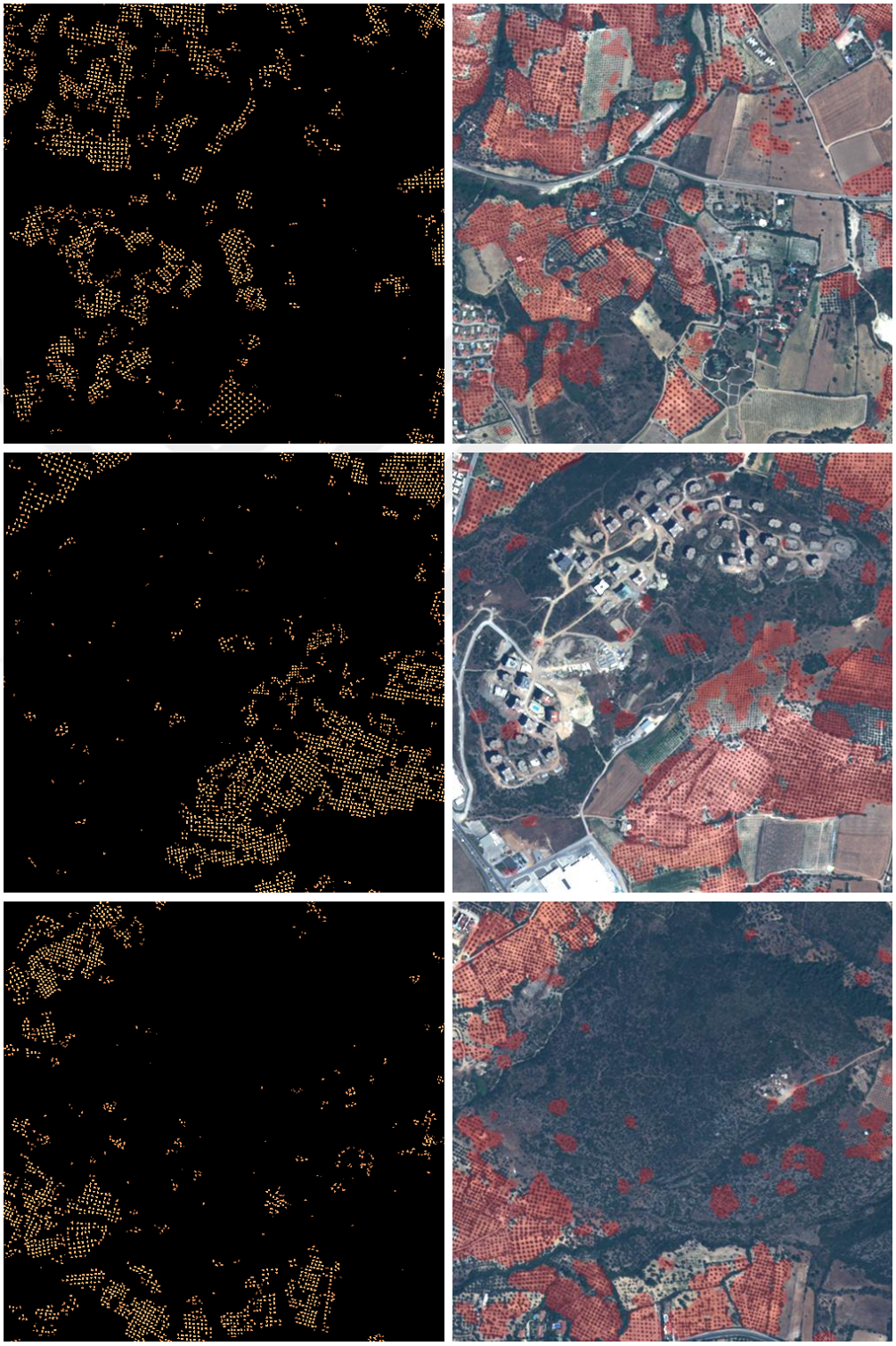


Figure 6.17: Example results for the detection of orchards. The left column shows the marginal probabilities at the end of selection. The right column shows the thresholded detections overlaid as red. We used a 21×21 pixel Gaussian smoothing filter to enhance the binary detection results before overlaying.

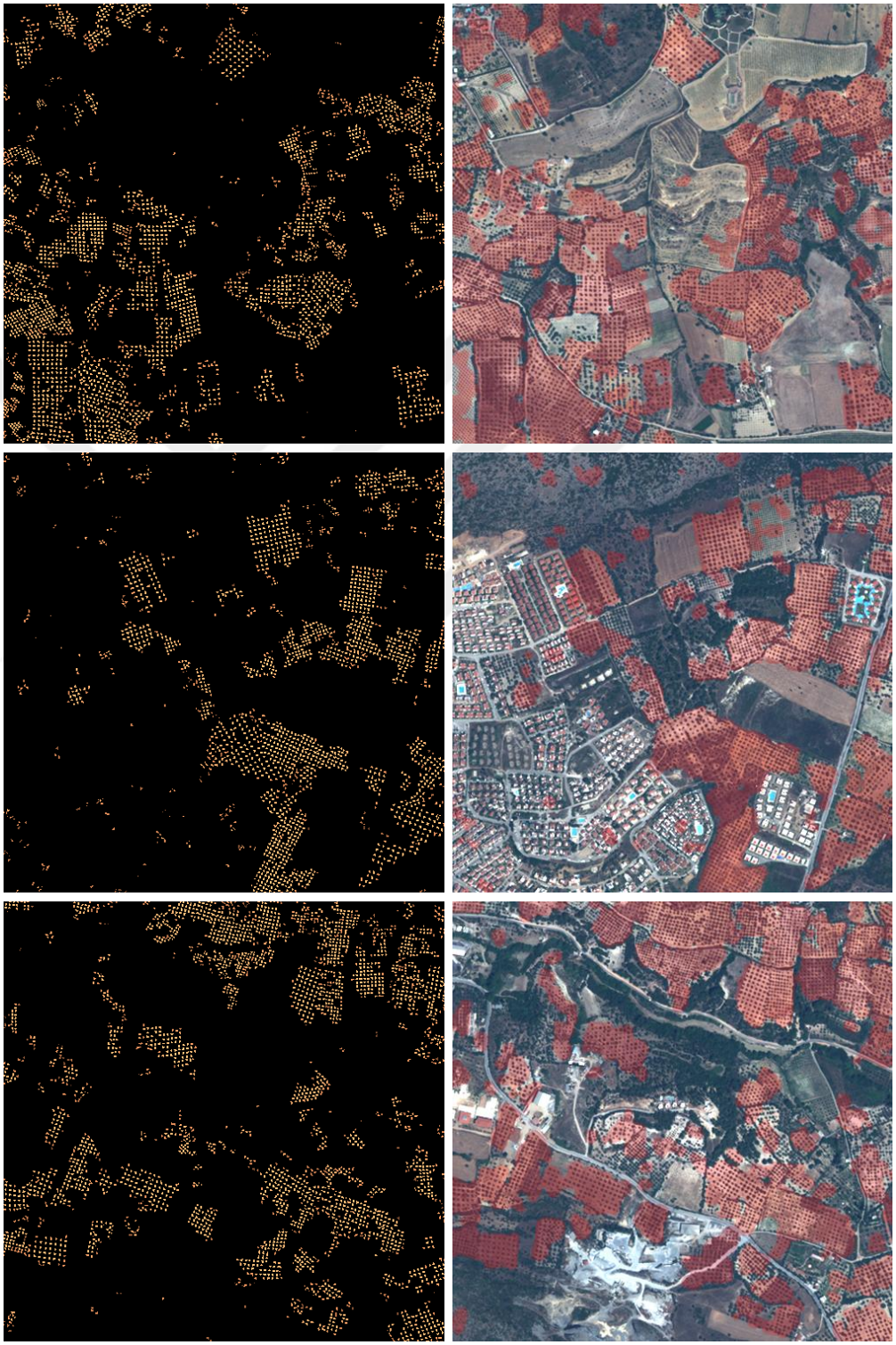


Figure 6.18: Example results for the detection of orchards. The left column shows the marginal probabilities at the end of selection. The right column shows the thresholded detections overlayed as red. We used a 21×21 pixel Gaussian smoothing filter to enhance the binary detection results before overlaying.

successful in the detection and localization of various compound structures in different types of scenes.

We believe that the output of the proposed method can be particularly useful when the goal is to perform image mining when we do not have a detailed labeling of example target structures but are interested in finding similar structures using a single example. The localization ability of the algorithm is valuable when there is no clear boundary with respect to low-level cues such as color and texture for the structure of interest. This also conforms to the focus-of-attention strategy that assumes that a single correctly detected pixel inside a target object is sufficient to attract the operator's attention to that target. These results can also be given as input to other algorithms so that more detailed labeling of the image can be produced. For example, the algorithm in [46] aims to estimate the spatial extents of complex geospatial objects that are composed of multiple land use and land cover classes. However, the method requires that at least a single known pixel is given as input for each object so that the procedure can be initialized and the model that was learned from multiple examples can compute its extent. The proposed method can provide the initializations and the models for such complex structures. As another example, the algorithm in [47] performs detailed classification of urban land use according to the shapes and spatial characteristics of buildings, but requires that complete GIS data with individual parcel boundaries and building polygons with detailed attributes are given as input for parcel-based classification where each parcel is assumed to belong to a single, homogeneous land use class. The proposed method can localize different urban, rural, or agricultural structures so that the availability of parcel boundaries is no longer a requirement for high-level semantic land use classification.

Finally, we analyzed the execution times for different steps of the proposed algorithm. The proposed learning and inference algorithms were implemented in Matlab with the only exception of the Swendsen-Wang sampling step in Section 5.2.3 implemented in C. We performed a code profile analysis to investigate the time spent in different steps. For the first scenario used in the experiments, the learning process for the example compound structure with four primitives took 774 seconds using the unoptimized Matlab code on a laptop with a 2.67 GHz Intel

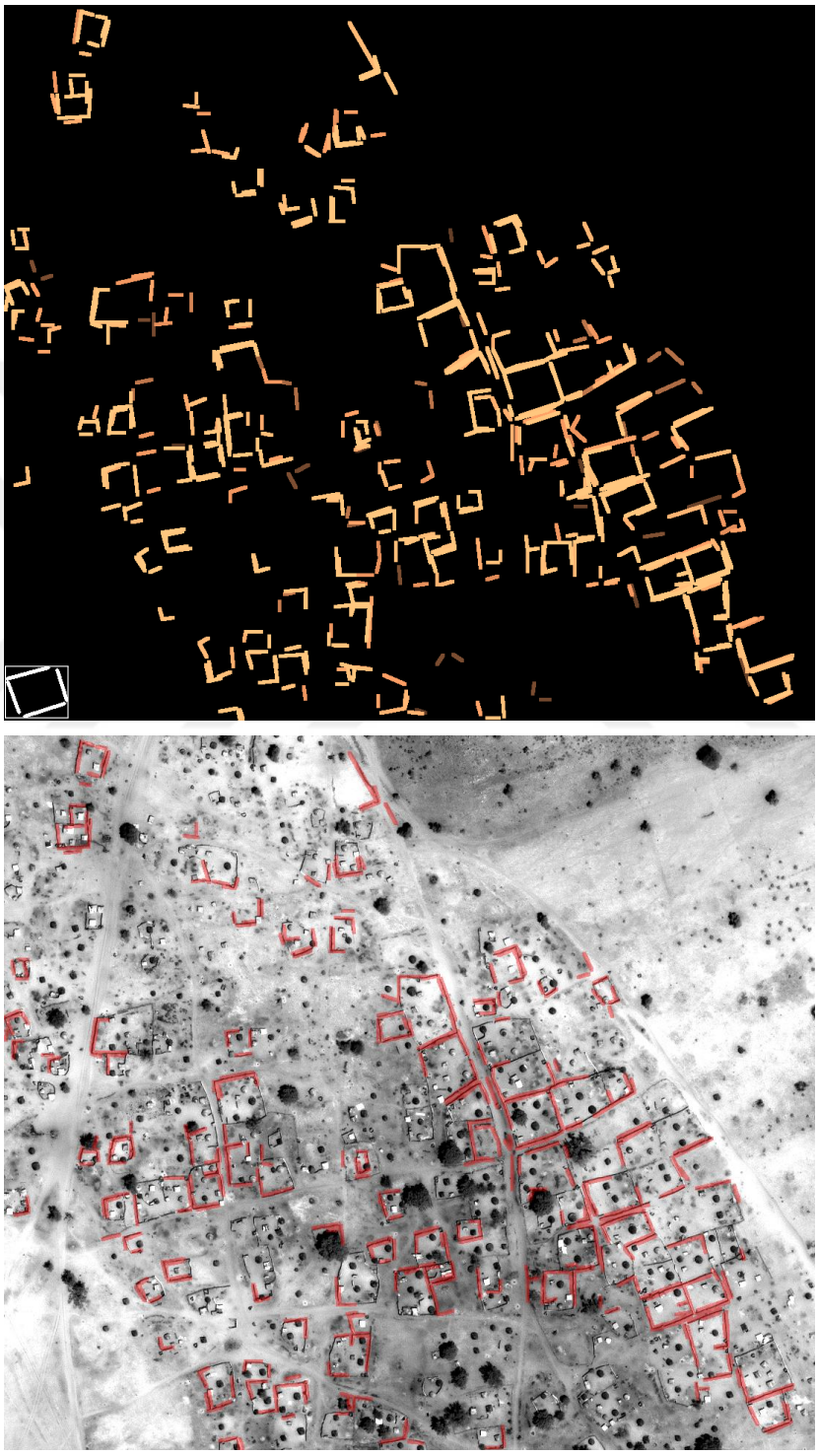


Figure 6.19: Example results for the detection of refugee camps as rural structures. The top image shows the marginal probabilities as well as the example primitives used for learning on the bottom left corner. The bottom image shows the thresholded detections overlayed as red. We used dilation with a disk with radius of 3 pixels to enhance the line segments for display.

Core i5 processor. The sampling process in Section 4 took 99% of the time where the number of samples was empirically set as described in Section 6.2.2. More samples can take longer but can produce a better model with a higher likelihood. The inference and selection process for an example 500×500 pixel image took 162 seconds. Of the total time, hierarchical region extraction in Section 5.1 took 1% of the time to produce 854 candidate regions, feature extraction in Section 3.2 took 2.5% of the time, finding the Voronoi neighbors in Section 5.1 took 26% of the time, and region selection in Section 5.2.3 took 62% of the time. The hierarchical region extraction step can take longer if the number of scales in the segmentation increases. Using faster algorithms for computing the Voronoi tessellation for the neighborhood graph could decrease the running time.

Another note regarding the implementation is that the proposed algorithm can directly run on large images as the selection algorithm considers only local interactions between the regions within connected components of the large scene graph. However, if the resulting segmentation tree structures are very large with resulting large number of vertices and edges in the neighborhood graph, sliding windows can be used to process the image where the window size can be selected based on the expected sizes of compound structures. Since the decision in the selection algorithm is based on individual connected components of the graph, it also does not matter how many different structures exist in the same window as the decision for each compound structure is made independently from other structures. This is one of the major advantages ovser traditional tile-based approaches with tiling being at the core of the image representation where each tile is assumed to correspond to a compound structure and the features are extracted from whole tiles.

6.3 Unconstrained Multi-layer Experiments

In this section, the aim is to detect housing estates that are composed of relatively small-sized, regularly-arranged, red roof buildings with similar shapes using the WorldView-2 image of Kusadasi, Turkey (Figure 6.3). First, a housing estate

model was learned using example building arrangements. However, the results showed that the selection algorithm that used this single-layer model could not detect several housing estates due to large variations in the spectral appearances of the primitives. An important observation was that some of the housing estates also have pools close to the buildings. The idea was to add a pool layer that can provide additional cues for finding the missed buildings. Then, this initial model was extended by learning the arrangements of buildings with respect to pools as well.

The base experimental procedure used two example structures ($M = 2$) for learning that involve four and five primitive objects from a single building primitive layer. Then, candidate building regions were extracted by thresholding the top-hat transform of the saturation band with a disk structuring element of 2 pixels radius. Hence, the input to the inference algorithm was a housing estate arrangement model and a set of candidate buildings. The output was a subset of these candidates selected by Algorithm 3 respecting the single layer compound structure model.

The multi-layer experiment used two primitive layers ($R = 2$) by including a pool layer along with the existing building layer. This new layer introduced a new pool primitive in each example structure used for learning. Hence, the learned probability function modeled both the building-building and building-pool arrangements. What we expect was that the arrangements between buildings and pools would assist in finding some of the estates near pools that were missed by using only the arrangements of buildings. For inference, candidate pool regions were obtained in addition to the building candidates by thresholding the normalized difference water index [48]. The inference algorithm constructed a graph from these candidate building and pool regions, and then, selected region groups that obey the learned multi-layer compound structure model. Note that these groups may consist of only buildings or buildings and pools together.

Candidate building and pool regions are shown in Figure 6.20(a). As expected, the number of candidates in the pool layer was considerably small compared to the building layer (Table 6.2). Since most of the pool (i.e., water) candidates were

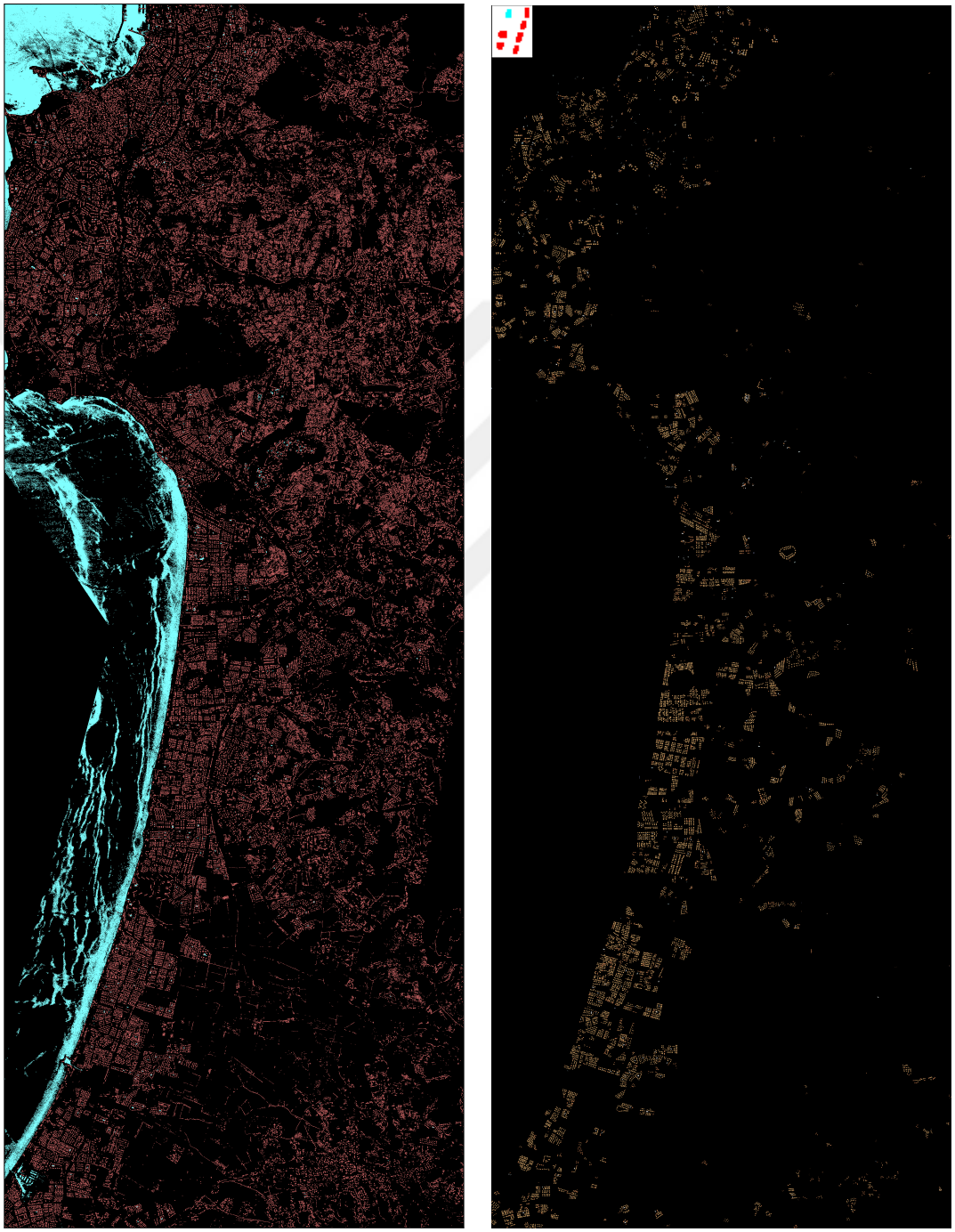
Table 6.2: The number of candidate and detected regions for single and multi-layer selection scenarios.

	Single-layer		Multi-layer	
	Candidates	Detected	Candidates	Detected
Building	67,983	11,173	67,983	11,871
Pool	-	-	16,276	436
Total	67,983	11,173	84,259	12,307

located on the sea, only 3 percent of the candidate pool regions were selected. Also, 698 new building regions were detected by the joint selection of building and pool candidates. Figure 6.20(b) shows the regions selected with multi-layer compound structure model. Not surprisingly, the detected housing estates were mostly located close to the sea.

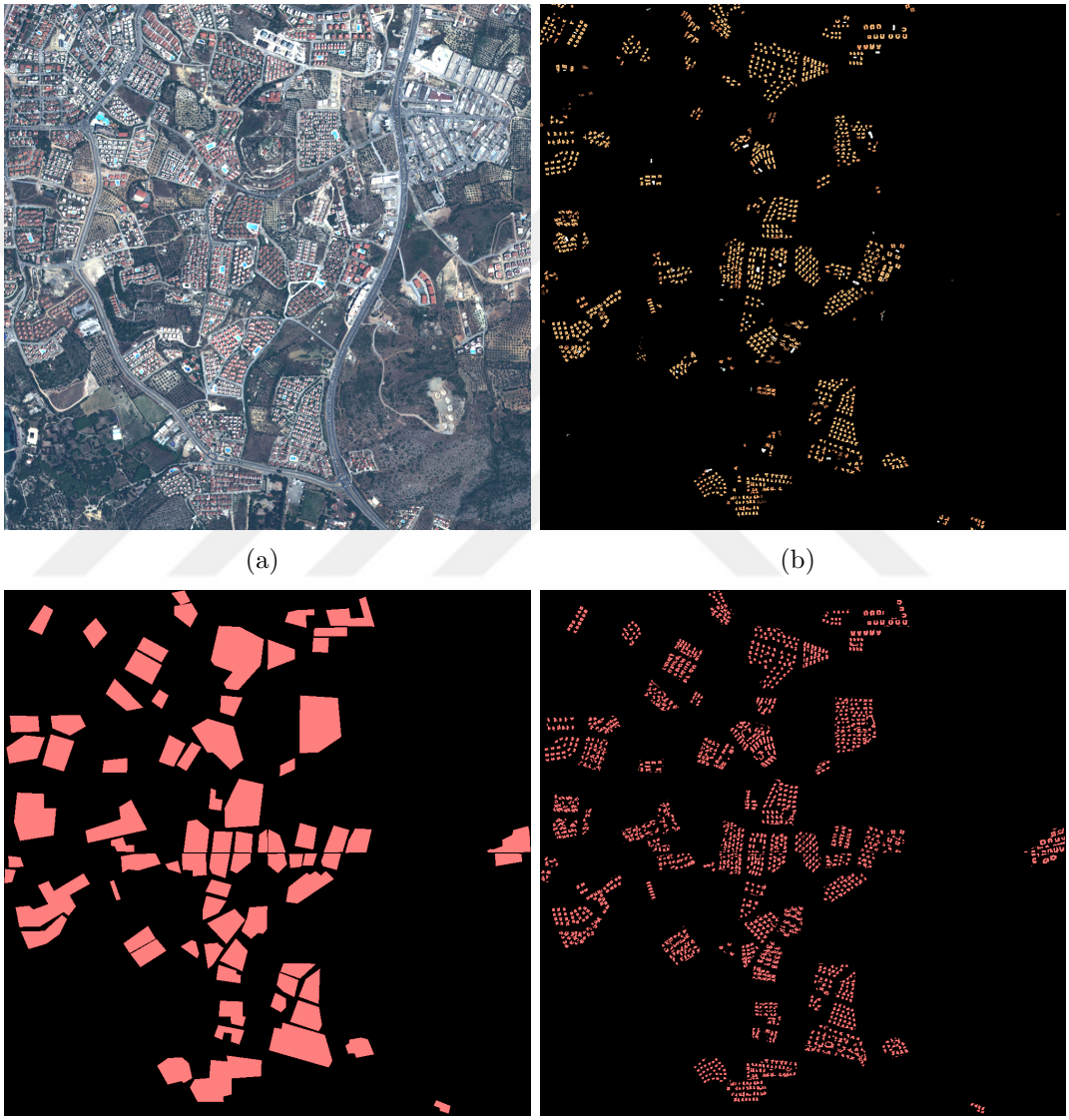
The proposed idea was tested quantitatively on a 1000×1000 pixel subimage (Figure 6.21(a)) by manual delineation of polygons corresponding to compound structures (Figure 6.21(c)) as well as buildings inside these polygons as primitive objects (Figure 6.21(d)). Detailed examination of the selected groups of buildings and pools in Figure 6.21(b) shows that some of the selected pool regions could be considered as the facilitators of the selection of new building regions. Some of them appeared alone in the selection results without any building candidates around being selected even though they existed. Mostly, these missed buildings had almost untiled complex roofs that were comprised of solar power systems and concrete teraces. Typical roof examples for the Kusadasi data set are illustrated in Figure 6.22. The remaining selected pools were members of housing estates that were already detected using only the building layer. Figure 6.23 shows precision versus recall curves obtained by applying different threshold values to the marginal-based detection scores. As expected, the multi-layer selection showed minor improvements over the single-layer selection for both pixel-based and object-based evaluation. This could be explained by the observations that most of the misdetected housing estates still did not have pools close to them and some pools were not adequate to compensate the major spectral divergence of nearby target structures from the assumed color model.

Figure 6.24 illustrates the effect of using multiple layers on an example housing



(a) (b)

Figure 6.20: (a) Candidate regions for the detection of housing estates. (b) Marginal probabilities for the regions selected from the building and pool layers. The marginals for the selected pool regions are depicted with a blue colormap. The example primitive mask is also shown.



(a) (b)

(c) (d)

Figure 6.21: (a) Test site. (b) Marginal probabilities of the selected buildings and pools. (c) Manually delineated polygons reflecting compound structures of interest. (d) Manually delineated buildings inside these polygons. These buildings are used as the primitives in the validation data.

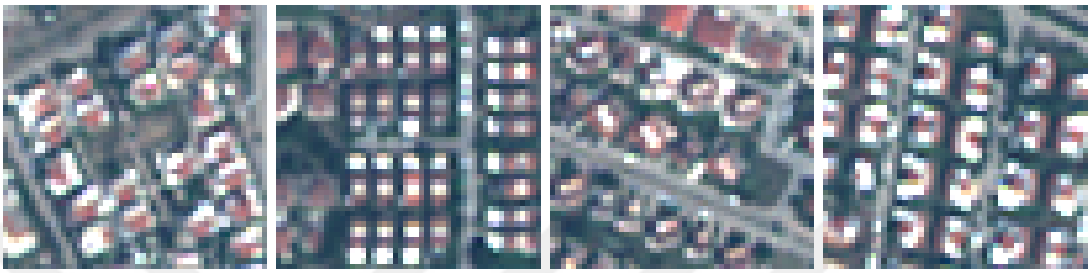


Figure 6.22: Examples of local details of red building rooftops.

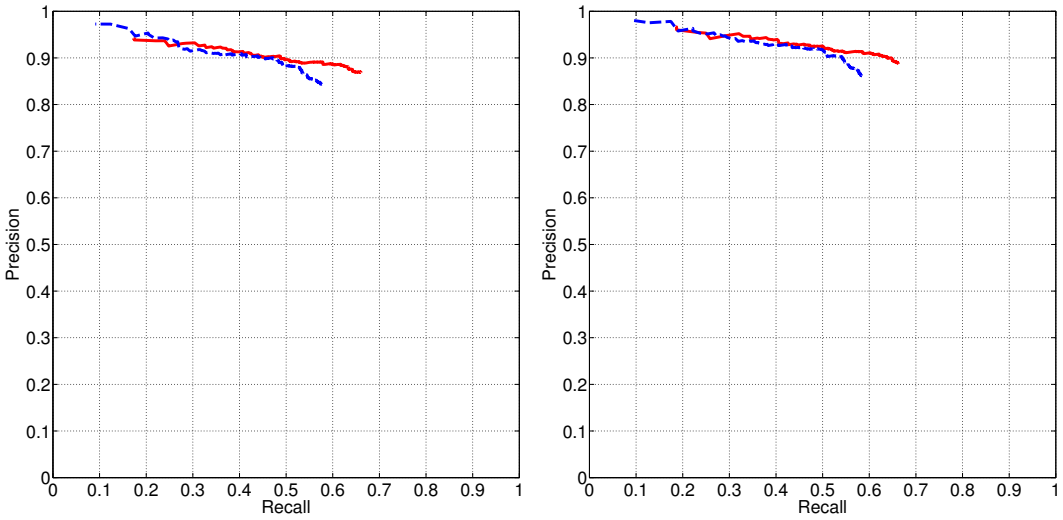


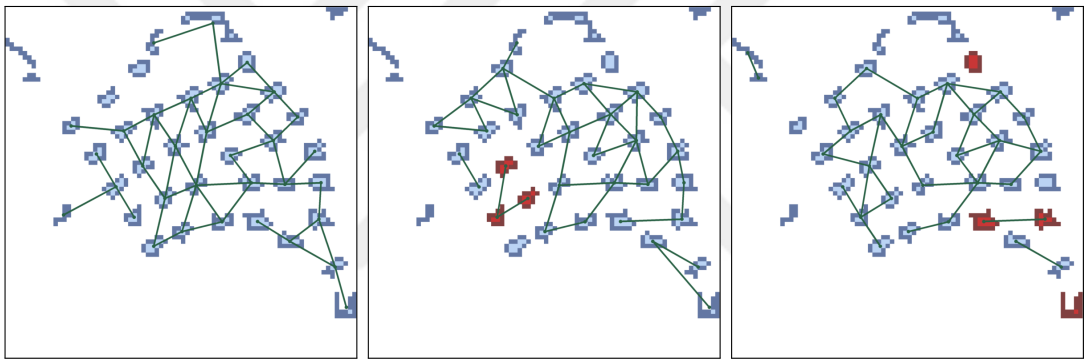
Figure 6.23: Precision-recall curves for housing estate detection. The left plot corresponds to the pixel-based evaluation and the right plot is for the object-based evaluation. The solid red curves correspond to the multi-layer approach, and dashed green ones are for the single-layer approach.

estate area. The top and bottom rows show evolutions of the samples at different iterations of the selection algorithm ran on a single layer and multiple layers, respectively. At the start of both runs, the regions were initialized as not selected. In the single-layer run, almost none of the connected components that were composed of only building candidates could not be labeled as selected because the probability in Equation (5.16) for flipping the label of each such group was very low. This was consistent with the previous observation that even though the arrangements of the buildings inside the housing estate seemed appropriate, the colors of individual buildings appeared far different from the learned color model. However, utilization of a pool layer in the detection process triggered the selection of those connected components that included the only pool candidate in the image. Although the single-layer model could not learn and detect the buildings, the multi-layer model could identify all of the neighboring buildings because of the enhanced likelihood due to the joint use of building-building and building-pool arrangement information. In other words, the positive weights added by the building-pool arrangements evidently overcame the negative weights caused by the color component in Equation (5.17).

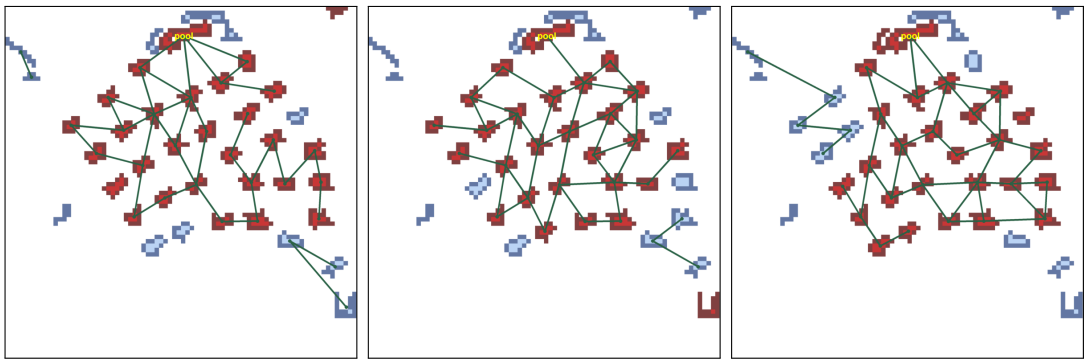
Figures 6.25 and 6.26 show how the additional pool-building arrangement information accompanied the building-building arrangements on selecting new buildings. These buildings were either members of completely misdetected housing estates or missing members of partly detected housing estates in single-layer selection results.

Figure 6.27 shows more examples of multi-layer selection. Misdetections occured mainly because the target objects of interest had an average size of 13 pixels, whereas the target primitives in the Ankara image had an average size of 120 pixels, making the former a much more difficult learning and detection task. That is, morphological operations could not accurately reveal many buildings as candidate regions due to their sizes and complicated roofs (Figure 6.28). In particular, a region appeared for a small SE size remained one to two pixels wide, disallowing reliable feature computations. On the other hand, a region that corresponded to a true object started merging with its surroundings and appeared as part of other regions at larger SE sizes. This resulted in heterogeneous and

(a)

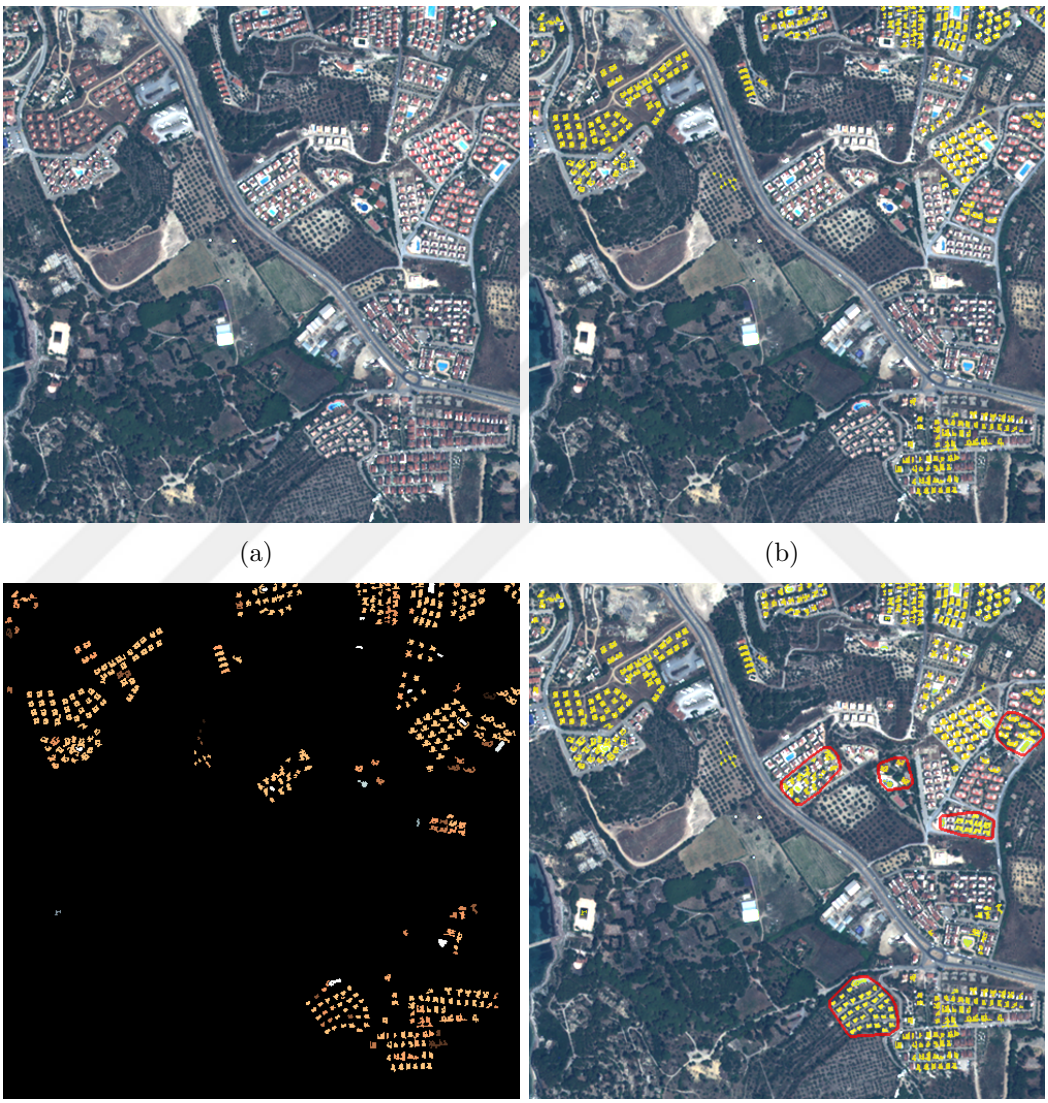


(b)



(c)

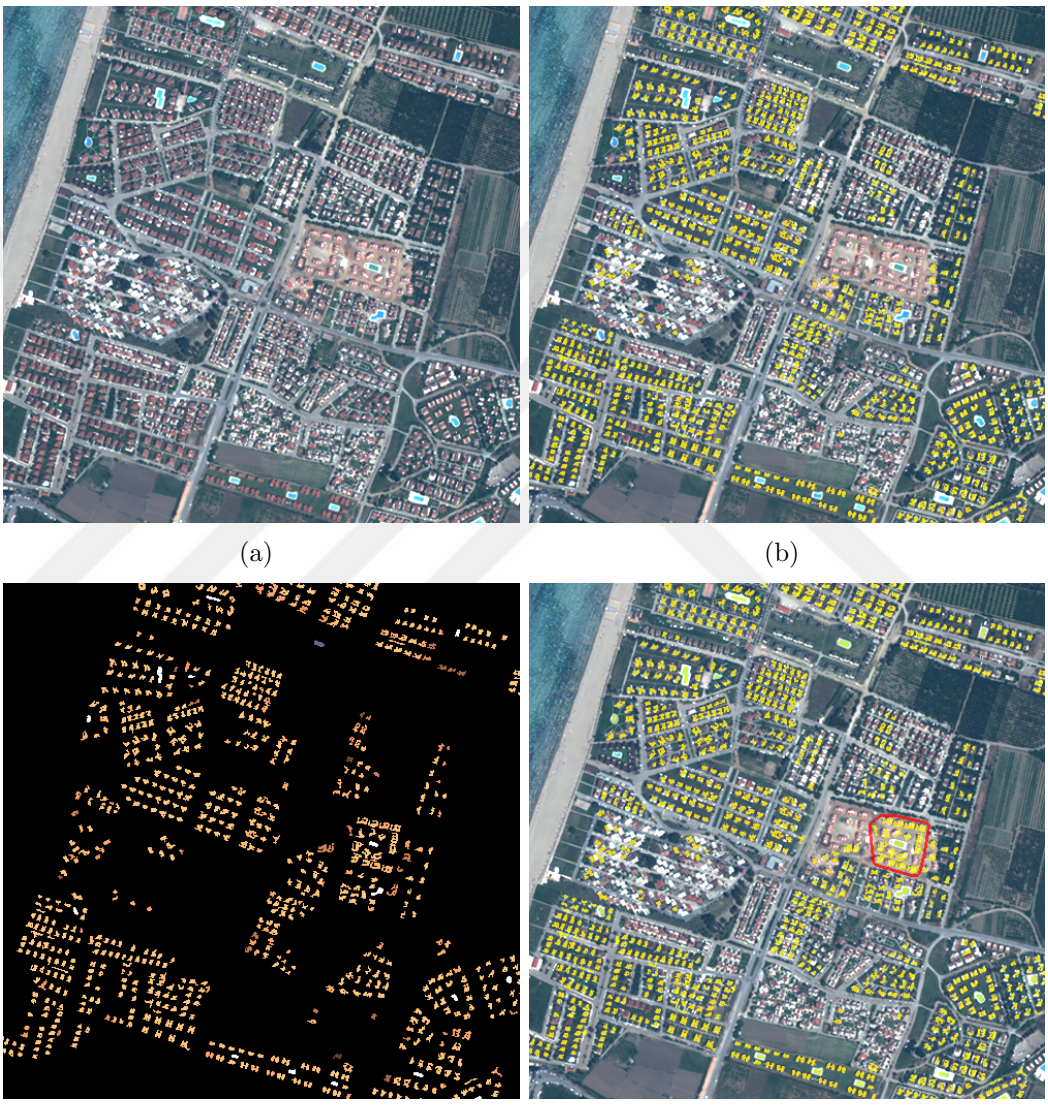
Figure 6.24: Samples obtained by the Swendsen-Wang procedure ran on single and multiple layers. (a) RGB image containing a housing estate with a pool. (b) and (c) show the sampled edges and primitive labels at iterations $1, 10$ and 20 for a single layer (i.e., building) and two layers (i.e., building and pool), respectively. The labels of the primitives are shown in red for selected and blue for not selected. Notice that in the bottom middle sample, a single edge sampled from the pool candidate to a building candidate was able to turn the color of the corresponding connected component including many buildings to red.



(a) (b)

(c) (d)

Figure 6.25: Example results for the detection of housing estates. (a) RGB image. (b) The thresholded selected regions, using only the building layer, overlayed as yellow. (c) The marginal probabilities obtained by selection with building and pool layers. (d) The selected regions after thresholding. A newly detected housing estate that was missed with single layer selection is enclosed by a red convex hull.



(a) (b)

(c) (d)

Figure 6.26: Example results for the detection of housing estates. (a) RGB image. (b) The thresholded selected regions, using only the building layer, overlayed as yellow. (c) The marginal probabilities obtained by selection with building and pool layers. (d) The selected regions after thresholding. A newly detected housing estate that was missed with single layer selection is enclosed by a red convex hull.

geometrically complex candidate regions.

6.4 Constrained Single-layer Experiments

Next, we demonstrate the results of adding constraints to the unconstrained selection formulation. First, we give an example constraint imposed on the selected set of regions from a single primitive layer (Section 6.4.1). Then, a comparative analysis of the results are presented (Section 6.4.2).

6.4.1 Problem Definition

In the segmentation literature, a segmentation is said valid if and only if it assigns at most one region label to each pixel [27]. However, a problem with the selection results in Section 6.4 was that they involved overlapping regions at different levels of the hierarchy (Figure 6.11). In most cases, only one of these regions actually corresponded to the true object while the remaining ones caused some sort of redundancy as they were either parts of or larger than the preferred structure. To overcome this problem, we require at most region should be selected per path where a path corresponds to the set of vertices from a leaf to the root and has a maximum length equal to the height of the corresponding tree. However, different from our previous work in [23], there is no obligation, in this study, that every path must include a selected vertex. Formally, we select an optimal subset $V^* \subseteq V$ such that

- $\forall v_i, v_j \in V^*, v_i \notin \text{descendant}(v_j)$ and $v_j \notin \text{descendant}(v_i)$.

This condition states that any two vertices in V^* cannot be on the same path (i.e., the corresponding segments cannot overlap in the hierarchical segmentation). Note that not every path must include a node that is in V^*.

Here, we make the constraint abstraction Ω in Section 5.3.1 concrete by writing

Figure 6.27: Zoomed detection examples. The image pairs show the marginal probabilities and the overlayed detections after thresholding these probabilities.

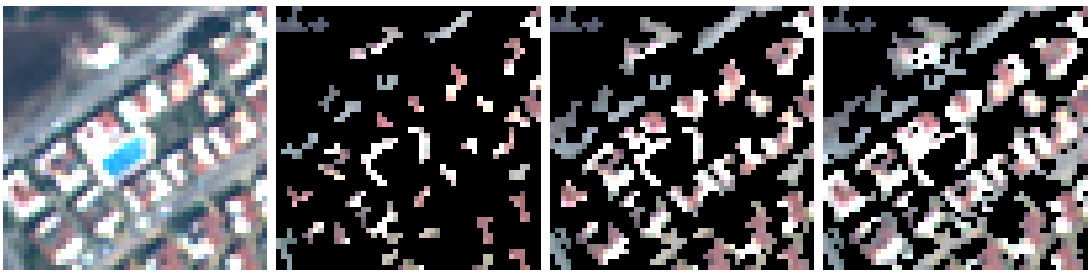


Figure 6.28: An example hierarchy .

the constraint in its usual form $A^T X \leq b$. Let A be a $|P| \times |\mathcal{V}|$ matrix where P denotes all the paths from the leaves to the roots of the input hierarchical forest and $A(i,j) = 1$ implies $v_i \in p_j \in P$. Note that the number of 1's in each row is equal to the height of the hierarchical tree structure. The problem in (5.21) can be reformulated as

$$\begin{aligned} \underset{x}{\text{minimize}}\ & \frac{1}{2} X^T W X + q^T X + \log Z_X \\ \text{subject to}\ & AX \leq \mathit{1}, \\ & 0 \leq X \leq 1. \end{aligned} \tag{6.1}$$

Recall that the binary X variables were previously relaxed to take continuous values in $[0, 1]$. The resulting problem is solved by the DC inference algorithm described in Section 5.3.3.

6.4.2 Results

We applied the proposed constrained selection algorithm to all six scenarios described in Table 6.1. The same parameters β and graphs in Section 6.2.2 were also given as input to the DC inference. In particular, these variables were used to determine the quadratic programming variables W, q and A in Equation (6.1). An important observation was that the optimized constrained solution X^* took discrete values (i.e., either, 0 or 1) even though they were relaxed to take real values between 0 and 1.

Figures 6.29, 6.30 and 6.31 show the selected regions for each scenario. Each region r_i was weighted by its conditional log-probability computed as

$p(v_i|nb(v_i)) = \sum_{j=1}^{N} W(i,j)x_j + q(i)$. Note that these results do not contain overlapping regions contrary to the results in Figures 6.6, 6.7 and 6.8. Hence, it is easier to show the selected regions at multiple levels in a figure.

The solid cyan precision-recall curves in Figures 6.9 and 6.10 corresponded to the results for the constrained selection. The conditional probabilities as goodness measures for the constrained selection were distributed uniformly in contrast to the marginal probabilities in unconstrained selection that were piled at high values. So, we can conclude that there was more uncertainty in the final decision of the constrained selection. This also explains the higher slope in the precision-recall curves of the constrained selection. The constrained selection algorithm, which did not allow overlapping regions, performed worse than the unconstrained algorithm, which allowed them, for the first four scenarios whereas the fifth and sixth scenarios achieved comparable results. Lower precisions for the same recall values were mainly due to the false alarms where the most dramatic decrease was seen in the second scenario. These false detections could be distinguished by visual inspection as groups of two or three perpendicular primitives that were not included in the ground truth. Missed detections, especially for the fourth scenario, caused the recall values to stuck at values lower than the maximal recall values of the unconstrained selection. The number of selected regions were 1485, 856, 2562, 1740, 811, and 263 for each scenario, respectively. Hence, at the sacrifice of increased misdetection rate, we reduced the amount of region redundancy to zero by eliminating nearly half of the output regions for all scenarios.

The inference and selection process for an example 500×500 pixel image took 296 seconds. This is approximately 2 times slower than the unconstrained selection process. This is mainly due to the fact that the constrained selection algorithm makes a global search through the whole variable space without considering the structure of the graph while the unconstrained selection makes local heuristic decisions by treating the connected components of the graph at a time.

Figures 6.32-6.37 show zoomed constrained selection examples. For the first five scenarios, the length of a path was 2 and at most one region was selected on each path. Particularly, if a selected region was in the first level, its parent was

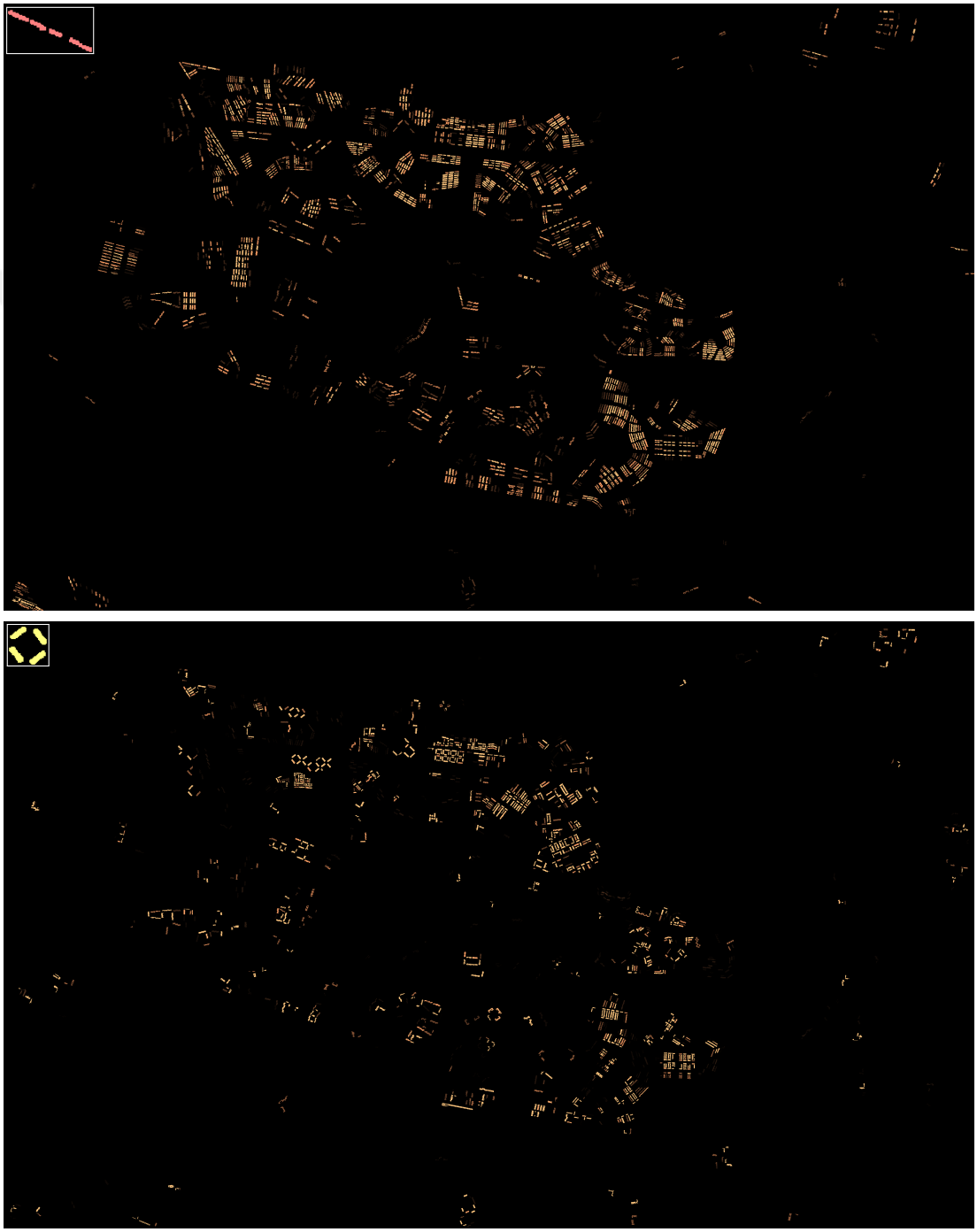


Figure 6.29: Conditional probabilities for the selected regions for the first and second scenarios. Brighter values indicate higher probabilities. The example primitives are also shown.

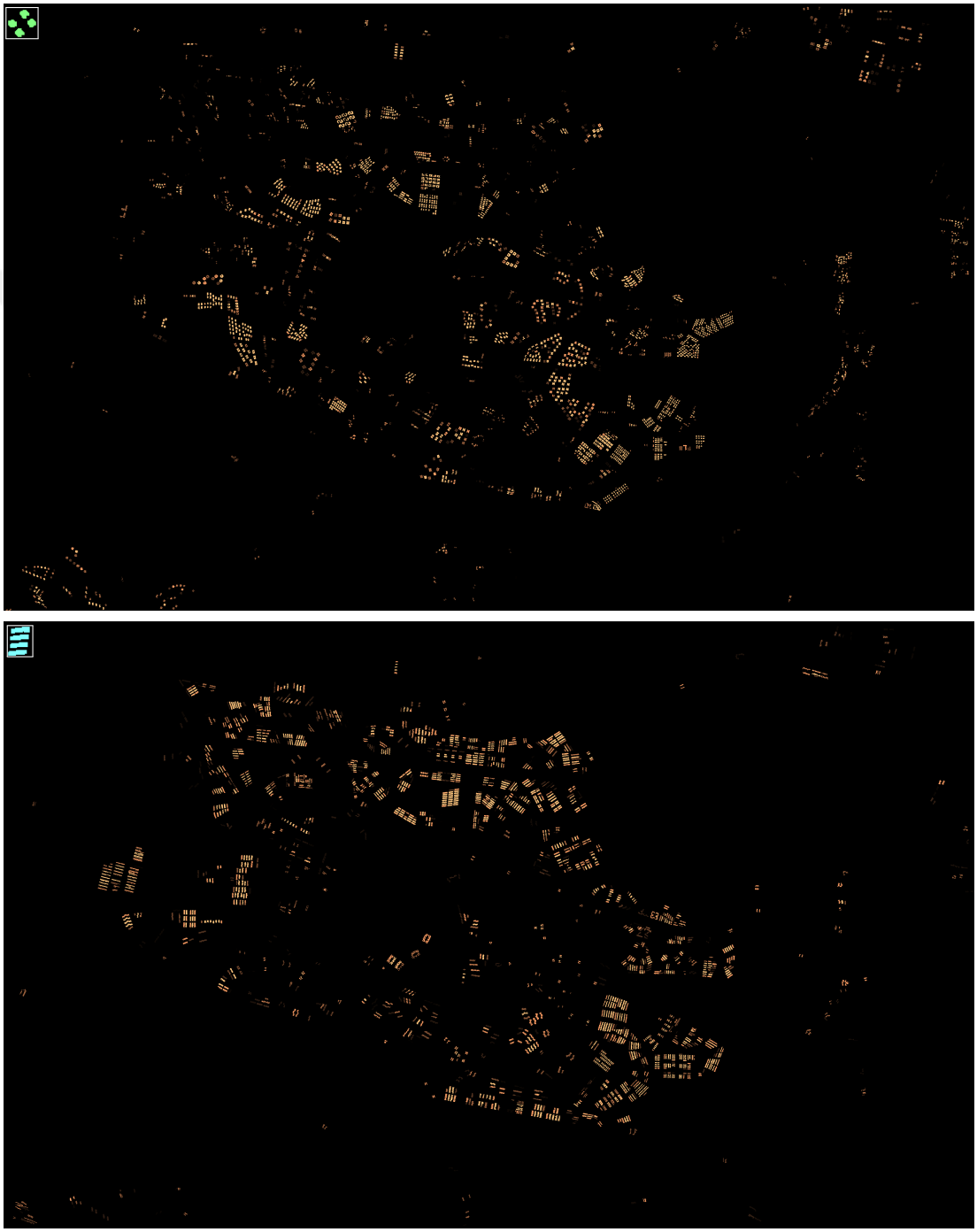


Figure 6.30: Conditional probabilities for the selected regions for the third and fourth scenarios. Brighter values indicate higher probabilities. The example primitives are also shown.

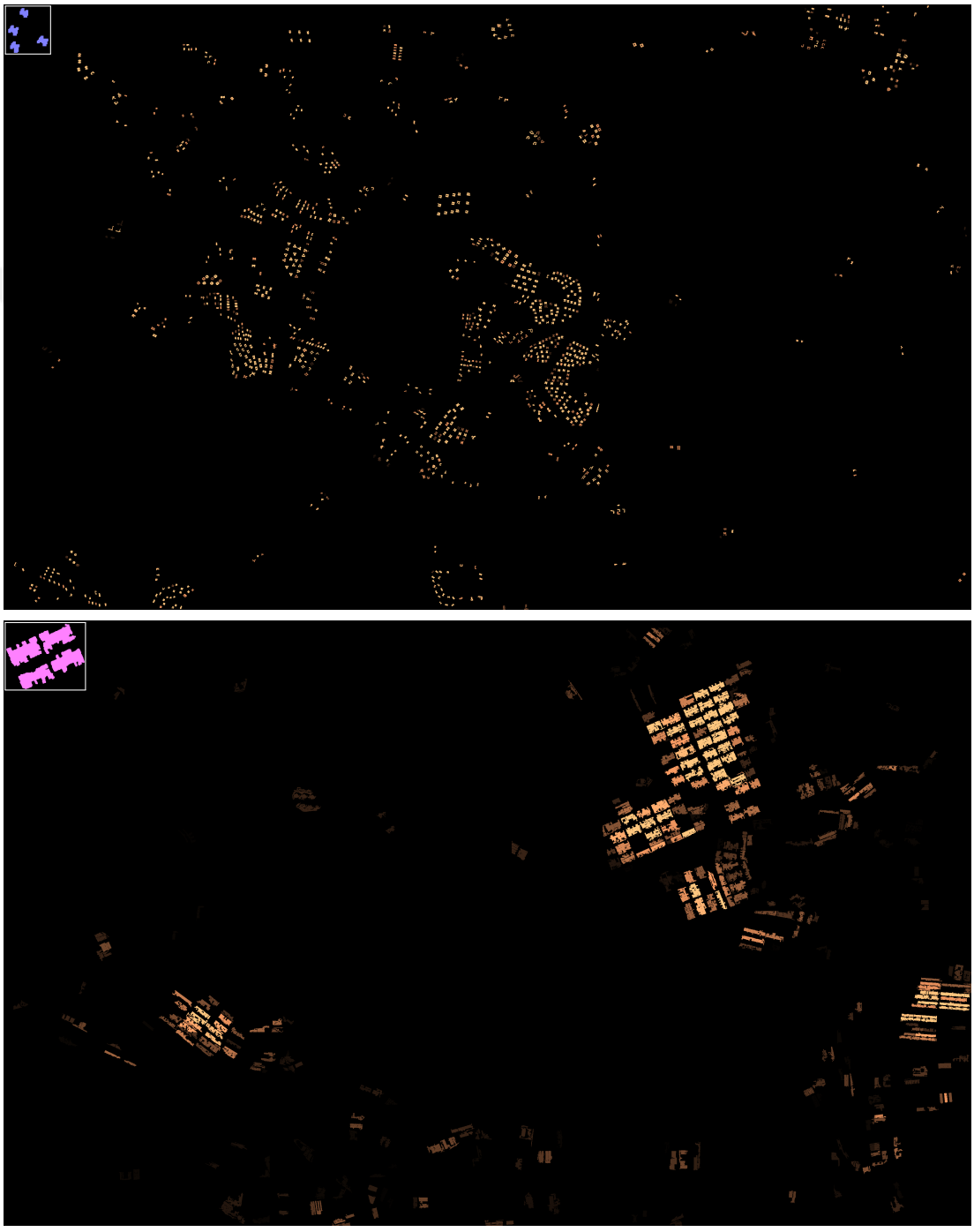
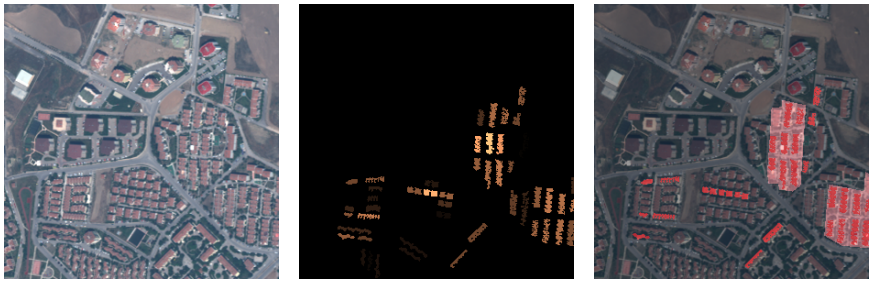
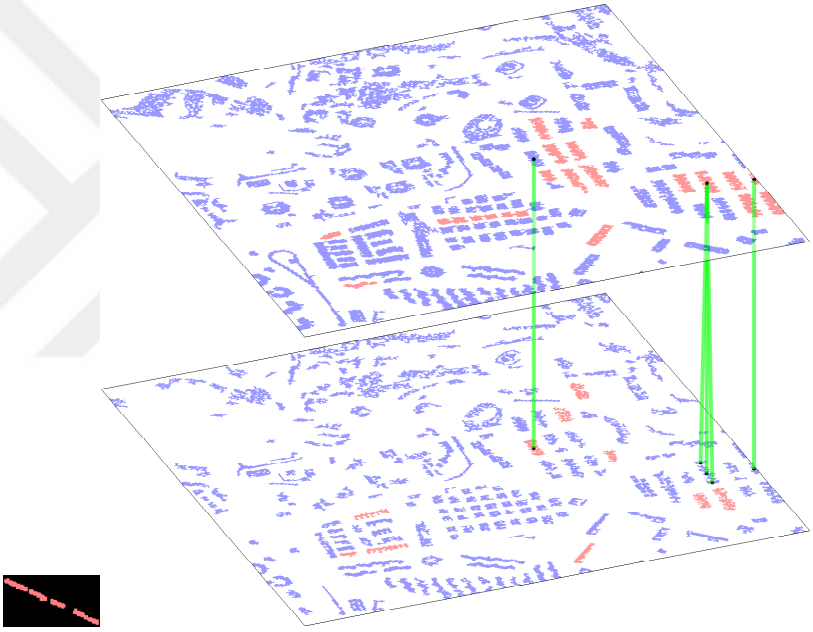


Figure 6.31: Conditional probabilities for the selected regions for the fifth and sixth scenarios. Brighter values indicate higher probabilities. The example primitives are also shown.



(a) RGB image. (b) Conditional probabilities. (c) Thresholded selections.

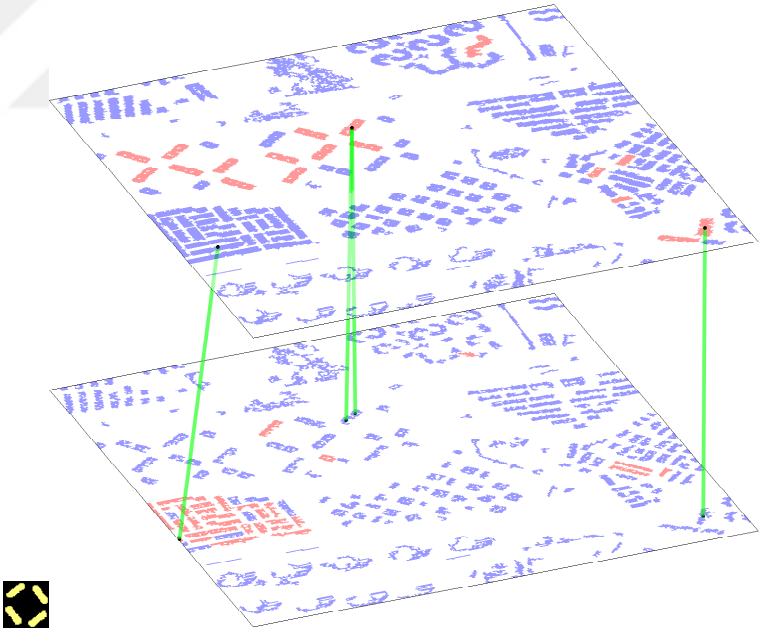


(d) Selected regions in two-level hierarchy.

Figure 6.32: Zoomed detection examples. The selected regions are colored with red in (d). For three selected regions, all the paths involving them are drawn.

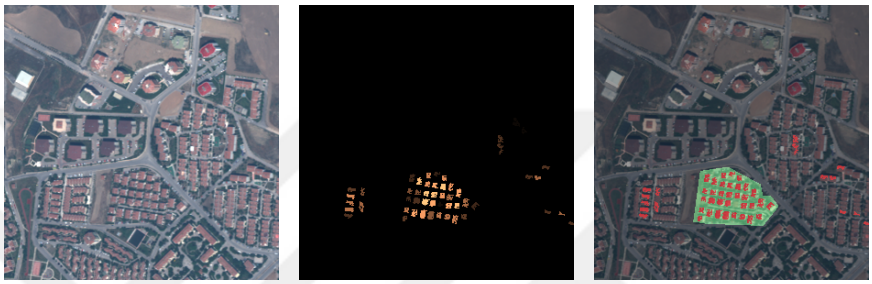
not selected. Conversely, if a selected regions was in the second level, none of its children were selected. For the last scenario, the length of a path was 6. Likewise, if a region was selected, none of its descendants nor ancestors were selected.

(a) RGB image. (b) Conditional probabilities. (c) Selected regions.

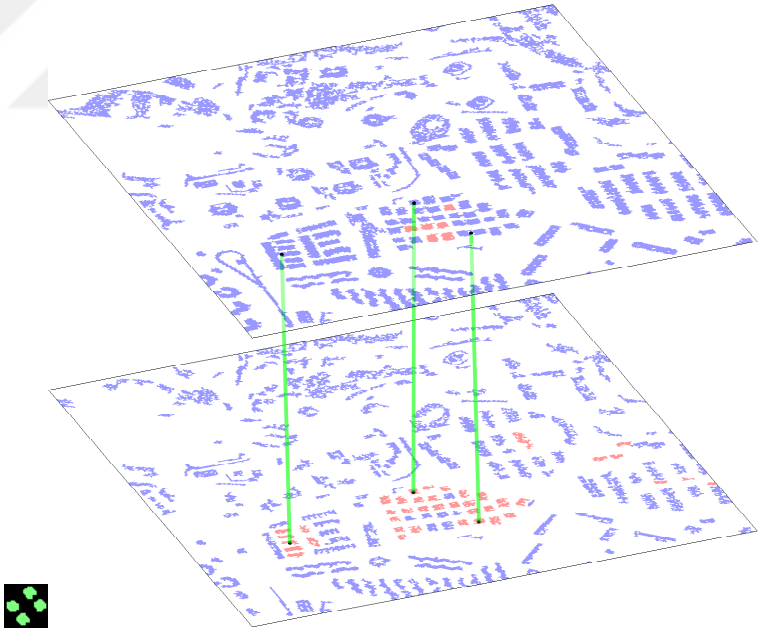


(d) Selected regions in two-level hierarchy.

Figure 6.33: Zoomed detection examples. The selected regions are colored with red in (d). For three selected regions, all the paths involving them are drawn.

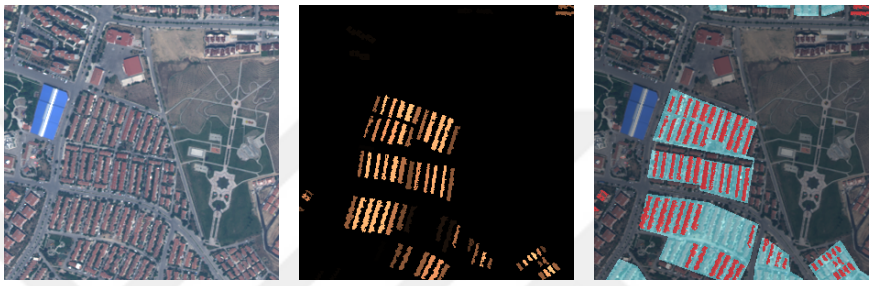


(a) RGB image. (b) Conditional probabilities. (c) Thresholded selections.

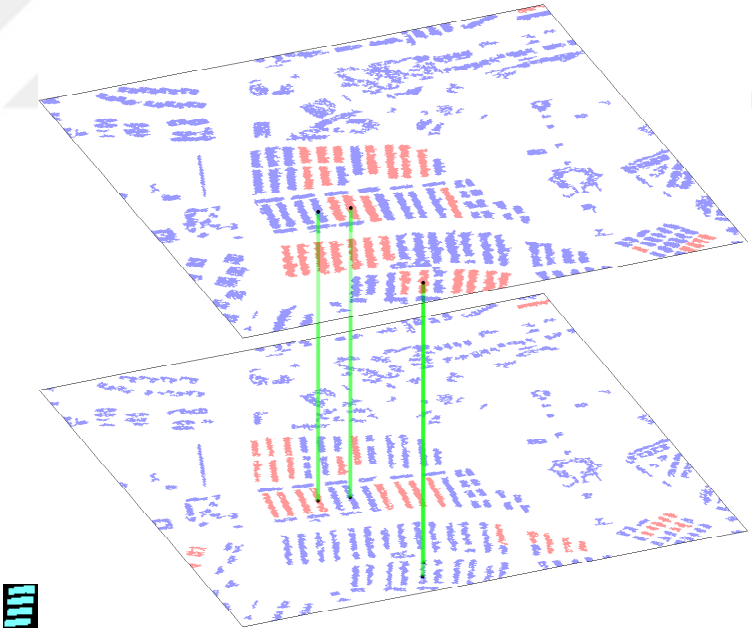


(d) Selected regions in two-level hierarchy.

Figure 6.34: Zoomed detection examples. The selected regions are colored with red in (d). For three selected regions, all the paths involving them are drawn.

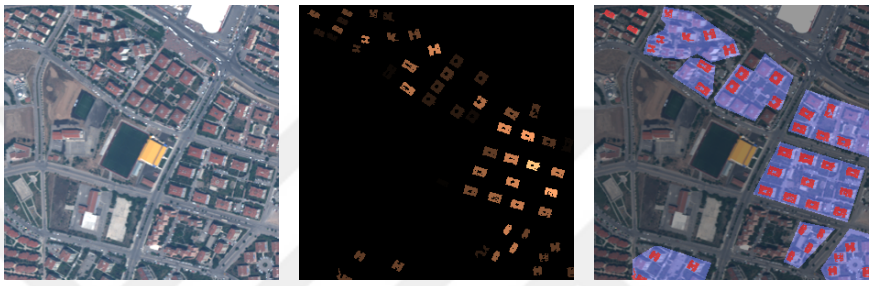


(a) RGB image. (b) Conditional probabilities. (c) Thresholded selections.

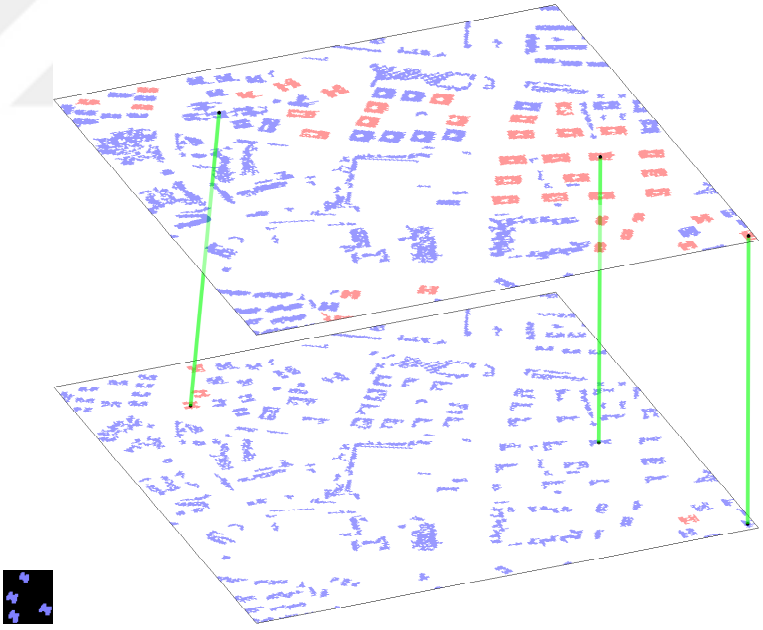


(d) Selected regions in two-level hierarchy.

Figure 6.35: Zoomed detection examples. The selected regions are colored with red in (d). For three selected regions, all the paths involving them are drawn.

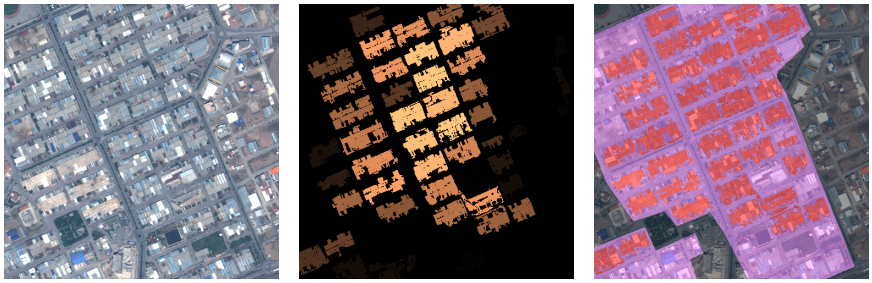


(a) RGB image. (b) Conditional probabilities. (c) Thresholded selections.



(d) Selected regions in two-level hierarchy.

Figure 6.36: Zoomed detection examples. The selected regions are colored with red in (d). For three selected regions, all the paths involving them are drawn.



(a) RGB image. (b) Conditional probabilities. (c) Thresholded selections.

(d) Selected regions in six-level hierarchy.

Figure 6.37: Zoomed detection examples. The selected regions are colored with red in (d). For two selected regions, all the paths involving them are drawn.

6.5 Constrained Multi-layer Experiments

The last set of experiments enforces that the selected set of regions from two primitive layers should obey a particular spatial configuration. First, we give the detailed definition of this spatial constraint (Section 6.5.1) and then, show the results (Section 6.5.2).

6.5.1 Problem Definition

The next set of experiments uses two primitive layers and enforces geometrical constraints between them. In particular, we search for nearby alike buildings and green areas where each building group must have a green area in the middle. An example arrangement is shown in Figure 6.38. This type of arrangements usually emerge as new settlement sites located outside the city center. Before detecting such compounds, we learn a probability model for the arrangements between the

buildings and between the buildings and green area as well as the individual properties of the individual buildings and green areas as in Section 4. Inference by maximizing this model without any hard spatial layout constraints may return groups of buildings and green areas that are close to each other, but the green area is located outside the building group. For example, structures other than what we require such as building sites close to a soccer field or a park will also give high probabilities in this model although the green area is not surrounded by the buildings, but close to them. Therefore, in inference, we strictly require that the distance between the centroid of the centroids of a selected group of similar buildings and the centroid of a selected large green area cannot exceed a distance threshold δ'. Let V^1 and V^2 represent the building and green area layers, respectively. The primitives of the desired compound structure can be selected by solving the following formulation

$$
\begin{aligned}
&\underset{x}{\text{minimize}}\ \frac{1}{2}X^T W X + q^T X + \log Z_X \\
&\text{subject to } AX \leq \mathit{1}, \\
&\qquad \left| \frac{1}{k_1} \sum_{v_i \in V^1} s_i^h x_i - \frac{1}{k_2} \sum_{v_j \in V^2} s_j^h x_j \right| \leq \delta' \\
&\qquad \left| \frac{1}{k_1} \sum_{v_i \in V^1} s_i^w x_i - \frac{1}{k_2} \sum_{v_j \in V^2} s_j^w x_j \right| \leq \delta' \\
&\qquad \sum_{v_i \in V^1} x_i = k_1 \\
&\qquad \sum_{v_j \in V^2} x_j = k_2 \\
&\qquad 0 \leq X \leq 1.
\end{aligned}
\tag{6.2}
$$

where $|\cdot|$ denotes the absolute value operator, (s_i^h, s_i^w) is the centroid of region v_i, k_1 and k_2 denote the number of buildings and the number of green areas to be selected, respectively. The number of regions k_1 and k_2 to be selected for each layer is given explicitly in the third and fourth constraints to be able to compute the mean values of their centroids in the second constraint. The resulting linearly constrained quadratic program is solved by the DC algorithm in Section 5.3.3.

6.5.2 Results

We conducted experiments for qualitative evaluation of the constrained selection using the Ankara data set. The learning and graph construction protocols were followed as in Section 6.2.2. We set $N = 6$ where one of the primitives is a green area. In addition to the building candidates, the candidate regions for the green area layer were extracted by thresholding the normalized difference vegetation index and then, applying a connected components analysis. We set the distance threshold δ' to 8 pixels. We varied the number of selected buildings such that $k_1 \in 4, 6, 8$. For each value of k_1, we select only one green area (i.e., $k_2 = 1$). To detect multiple instances of the target compound structure in the image, we initialize the DC algorithm multiple times with different values of X. Figure 6.38 shows the selected regions for all values of k_1. 255 building regions and 42 green area regions were selected out of 271,320 and 21,000 candidates, respectively. Note that a region can be selected by different values of k_1 and different initializations for each k_1. Each selected region was colored according to the largest value of the objective function (i.e., denoted by $fVal$) in (6.2) where it is included in the solution set.

Detailed examples are presented in Figures 6.39, 6.40 and 6.41. Notice that the constraint of the previous section that enforced at most one selected region on a path was in effect as well as the spatial layout constraint. Visual analysis of the results show that the proposed algorithm was able to localize the members of the target compound structure of interest in an image across a large number of candidate regions. Additional selection results were given in Figures 6.42, 6.43 and 6.44. The number of building regions k_1 to be selected and the returned locally optimal objective function values $fVal$ were given for each result. It is important to note that different than the variables X^* in Section 6.4.2 that took binary values (i.e., either, 0 or 1), the variables X^*, in this program, may take real values between in $[0, 1]$ in order to satisfy the spatial layout constraint. Two examples for $k_1 = 4$ are shown in Figures 6.42-(g) and 6.42-(j). In these examples, some regions took real values other than 0 or 1 so that $\sum_{i=1}^{N} x_i = 4$. If otherwise, the spatial layout constraint would not have been satisfied. Another interesting

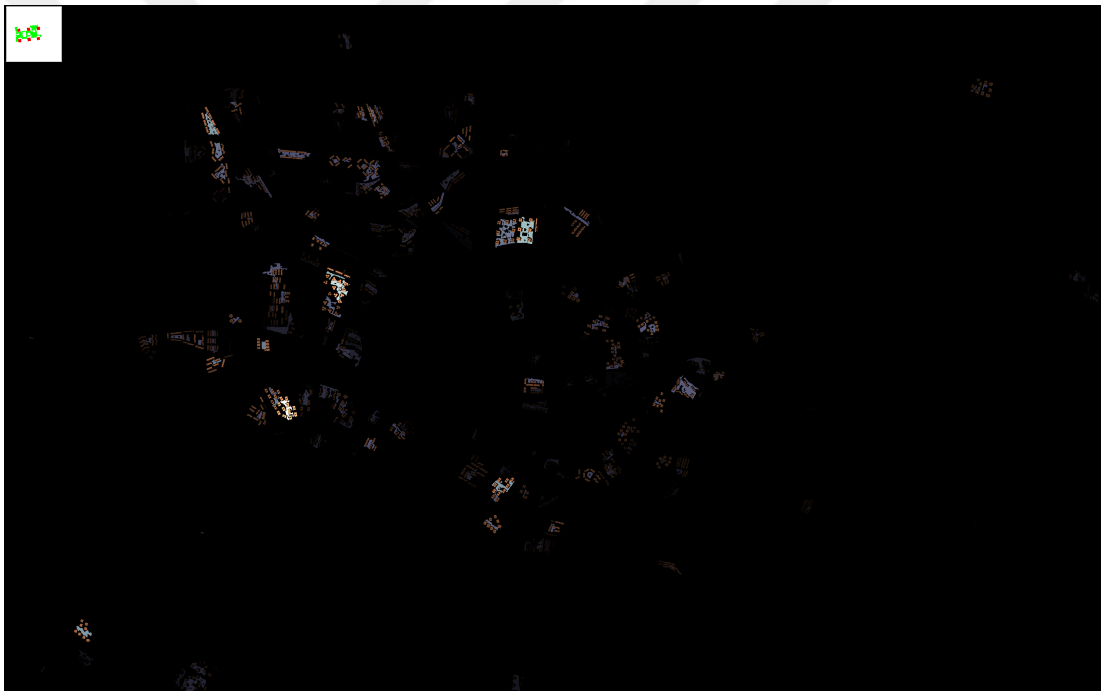


Figure 6.38: Selected regions for the green areas surrounded by building sites scenario. Brighter values indicate higher objective values. The example primitives are also shown on the top left corner.

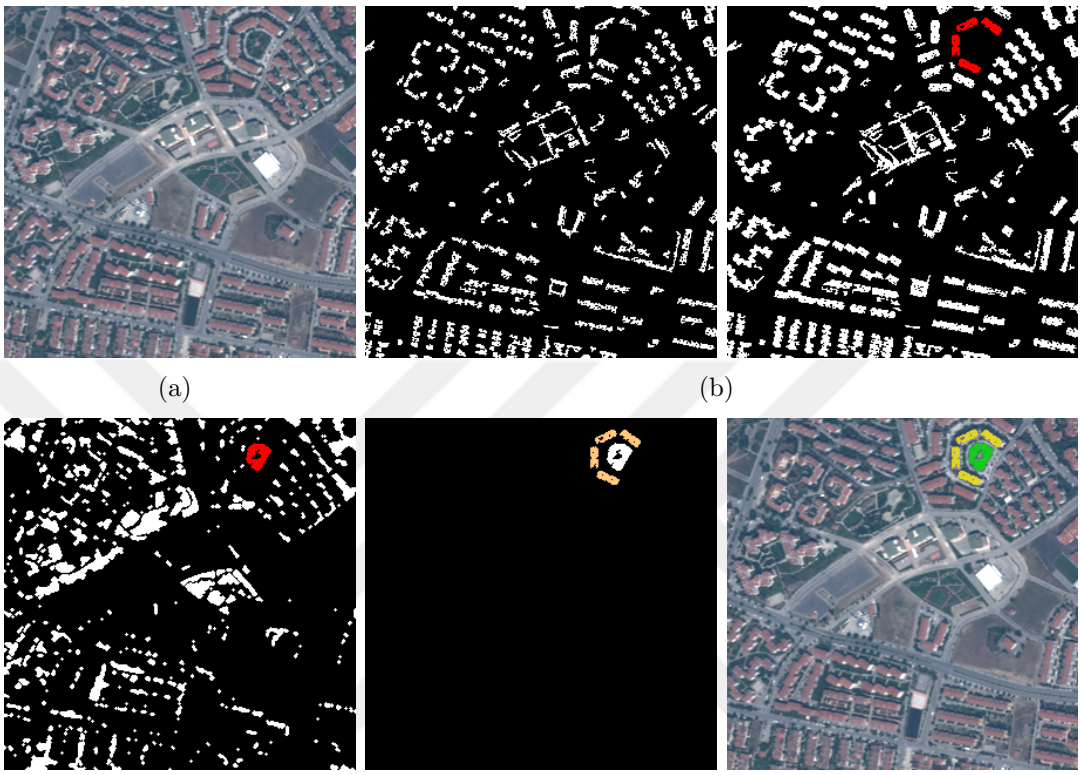
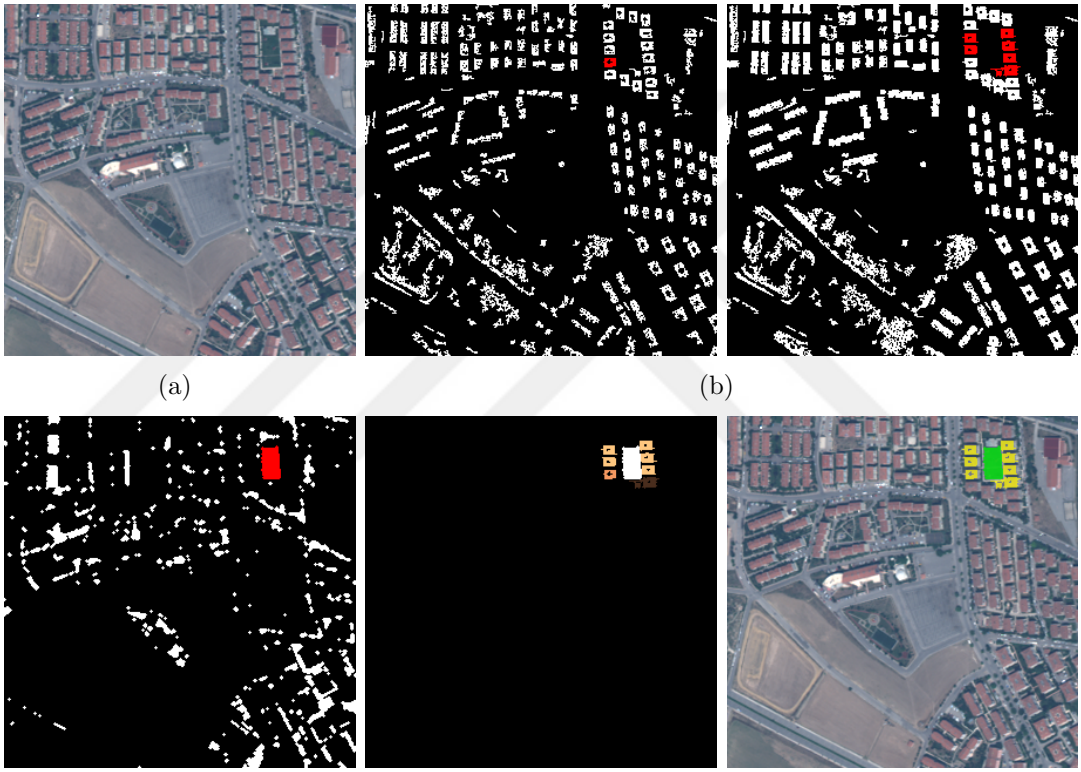


Figure 6.39: A zoomed detection example for a 300×300 sub-scene in (a). k_1 is set to 4. (b) shows the two-level hierarchy of building candidate regions. (c) shows single-level green area candidate regions. The selected regions are colored with red. (d) shows the X^* values at the end of selection ($fVal = -2.97$). Selected buildings and green area are colored by the copper and the pink colormap, respectively. Brighter colors indicate higher values. (e) shows the thresholded detections overlayed as yellow (buildings) and green (green area).

result was encountered in 6.42-(i) where k_1 was set o 8. In order to force the centroid of the building regions' centroids to remain close to the centroid of the green area, two distant building regions were added to the final set of selected regions. As a result, these regions increased the objective function value.



(a) (b)

(c) (d) (e)

Figure 6.40: A zoomed detection example for a 300×300 sub-scene in (a). k_1 is set to 6. (b) shows the two-level hierarchy of building candidate regions. (c) shows single-level green area candidate regions. The selected regions are colored with red. (d) shows the X^* values at the end of selection ($fVal = -3.01$). Selected buildings and green area are colored by the copper and the pink colormap, respectively. Brighter colors indicate higher values. (e) shows the thresholded detections overlayed as yellow (buildings) and green (green area).

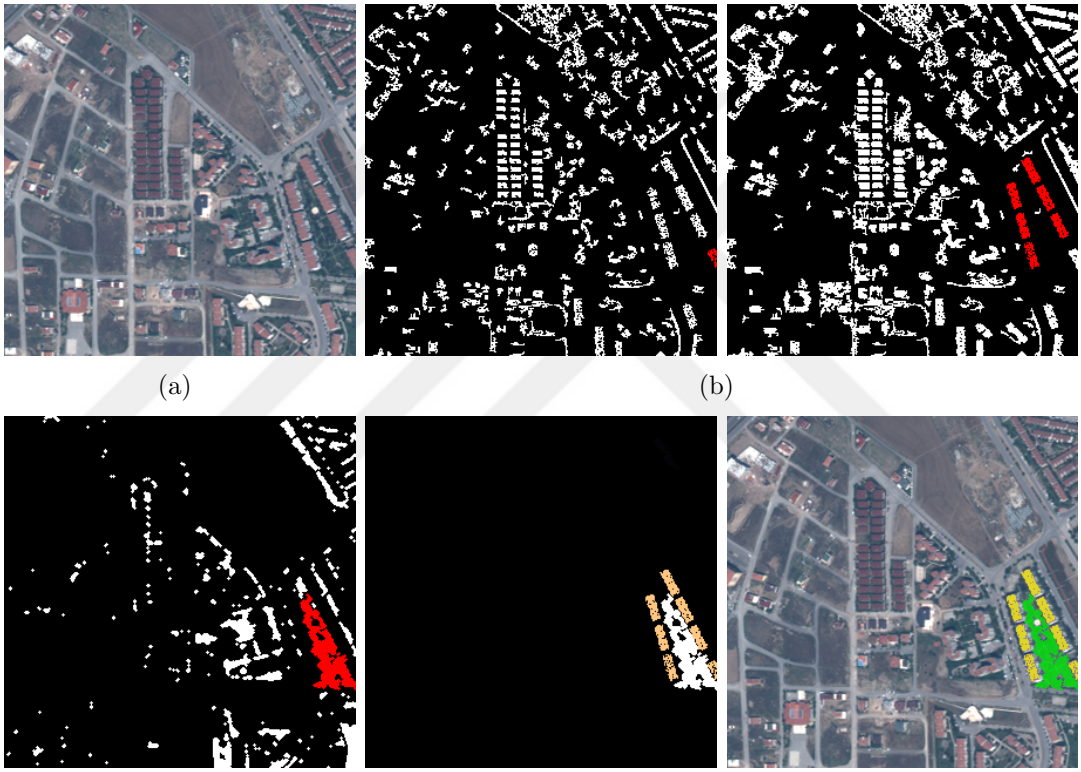


Figure 6.41: A zoomed detection example for a 300×300 sub-scene in (a). k_1 is set to 6. (b) shows the two-level hierarchy of building candidate regions. (c) shows single-level green area candidate regions. The selected regions are colored with red. (d) shows the X^* values at the end of selection ($fVal = -3.05$). Selected buildings and green area are colored by the copper and the pink colormap, respectively. Brighter colors indicate higher values. (e) shows the thresholded detections overlayed as yellow (buildings) and green (green area).

(a) $k_1 = 4, fVal = -2.90$ (b) $k_1 = 4, fVal = -3.13$

(c) $k_1 = 6, fVal = -3.16$ (d) $k_1 = 4, fVal = -3.01$

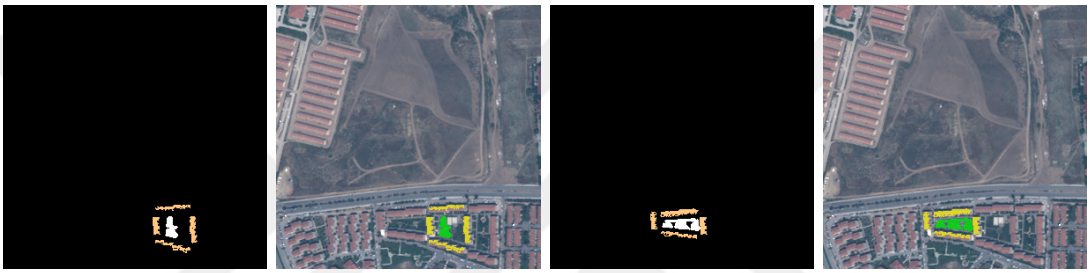
(e) $k_1 = 6, fVal = -2.92$ (f) $k_1 = 4, fVal = -2.96$

(g) $k_1 = 4, fVal = -2.95$ (h) $k_1 = 8, fVal = -2.89$

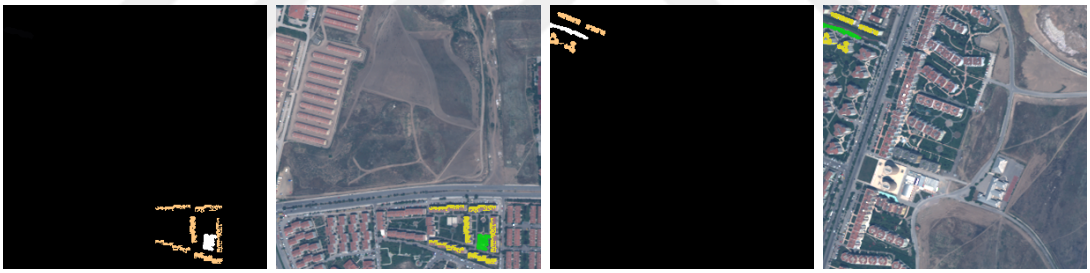
(i) $k_1 = 8, fVal = -2.64$ (j) $k_1 = 4, fVal = -2.97$

Figure 6.42: Zoomed detection examples for different values of $k_1 = 4, 6, 8$.

(a) $k_1 = 8, fVal = -3.23$ (b) $k_1 = 4, fVal = -2.91$



(c) $k_1 = 4, fVal = -2.66$ (d) $k_1 = 4, fVal = -2.77$

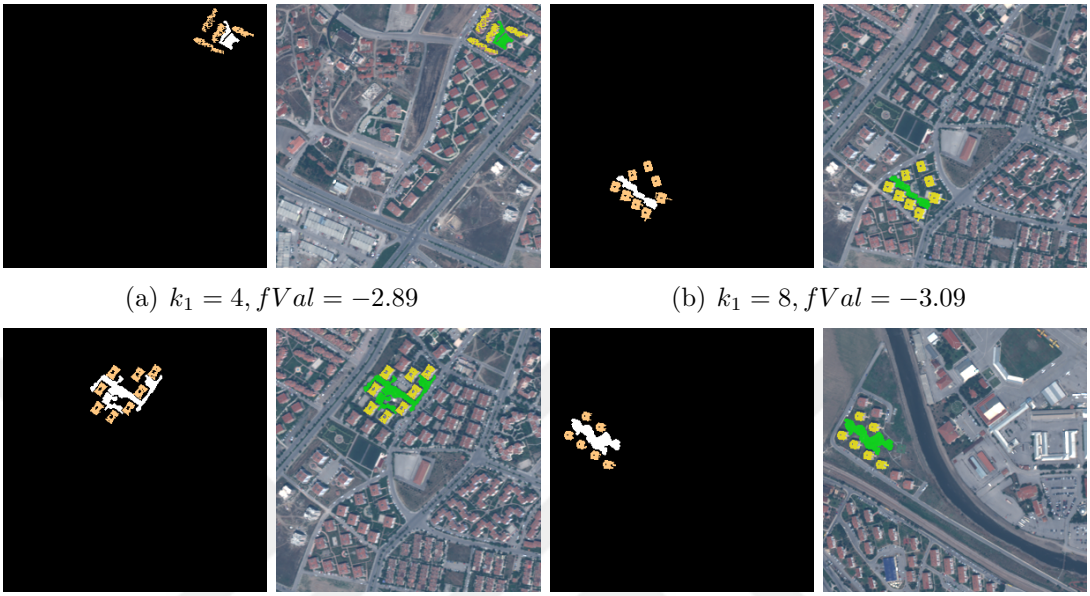


(e) $k_1 = 6, fVal = -2.67$ (f) $k_1 = 4, fVal = -2.92$

(g) $k_1 = 8, fVal = -3.62$ (h) $k_1 = 8, fVal = -3.60$

(i) $k_1 = 4, fVal = -3.06$ (j) $k_1 = 8, fVal = -3.78$

Figure 6.43: Zoomed detection examples for different values of $k_1 = 4, 6, 8$.



(a) $k_1 = 4, fVal = -2.89$ (b) $k_1 = 8, fVal = -3.09$

(c) $k_1 = 8, fVal = -3.37$ (d) $k_1 = 6, fVal = -3.35$

Figure 6.44: Zoomed detection examples for different values of $k_1 = 4, 6, 8$.

6.6 Discussion

The proposed framework has several parameters that may affect the performance. In representation of the primitives, the parameters (s^h_{min}, s^h_{max}) (respectively, (s^w_{min}, s^w_{max})) representing the minimum and maximum major (respectively, minor) axis lengths were determined wide enough so that they are smaller and larger than the smallest and largest primitives observed in the reference structures. The maximum possible value for the proximity threshold δ can be set to the maximum of the image height and image width. Since we use a relatively small number of primitives N for each layer, the resulting graphs are mostly complete. However, for large M increasing and decreasing δ would change the number of neighbors each primitive has. In particular, a larger value of δ may increase the quality of the observed histograms while increasing the time to find the neighbors of each primitive object in the compound structure. Since the observed histograms are computed once, it might be better to have many neighbors to better generalize the distribution in the compound structure. In some cases where, for example, regularity feature has more importance than the others, using only the close

neighbors for each object might be more helpful in representing the compound structure. In other words, increasing the number of neighbors may bring many irrelevant neighbors that result in noisy histograms. In such cases, setting δ very small is generally preferred. We observed that setting the number of bins around 5 for each marginal histogram was fairly enough to produce samples that have similar statistics with the example compound structure. Using much smaller and larger number of bins underrepresented and overrepresented the target structure, respectively. Clearly, since we discretize the features with histograms, a perfect match with the desired distribution would never occur.

The learning algorithm was able to model a large number of potential relationships using a very small number of examples M that are used to model the statistics of the compound structures. However, using more examples with a larger number of primitive objects N can provide a better generalization of the distribution of the target compound structure. The current study relied on the manual extraction of the reference structures used in learning. Hence, there is a trade-off between the effort to obtain more examples and the quality of the representation of the compound structure. Moreover, we found that the time took by the learning process was mostly consumed by sampling new images, which can be made tractable through a number of optimizations. Obviously, the larger the number of samples S for estimating β, the more representative the average for the sampled histograms become. Furthermore, the number of sampling iterations T can be crucial to obtain better samples. We observed that T should not typically be too small (e.g., less than 50), otherwise, the sampled histograms would not be representing the distribution.

Another important advantage of the proposed algorithm was that it did not need windows for feature extraction where the window size is usually a critical parameter in window-based approaches. In the literature, the size of the window is mostly chosen empirically according to the expected compound object size. Generally, the window size should be chosen so that the distribution of the features are sufficient to discriminate the structure of interest and the number of irrelevant features is reduced as much as possible. For example, the structures of interest in the Ankara image occupied typically a $300 \times 300m^2$ area which

corresponds to 150×150 pixels. However, whatever the window size is, there would be unnecessary features that are possibly much more than the relevant features inside the window. In this study, our goal was mainly to leave out these irrelevant features. For any given window size, we believe that the probability of the features would get higher even after a rough selection procedure. One of our baseline methods also aimed to demonstrate this. The number of hierarchy levels can be considered a critical parameter in the detection procedure. However, since objects in each layer have similar sizes and each layer has its own set of hierarchical segmentations, the number of hierarchy scales can be optimized for each scale according to the smallest and largest object sizes. Finally, we consider the parameters k_1 and k_2 as input arguments to the procedure that should be given by the user rather than real parameters that should be optimized. In particular, having k_1 number of buildings and having k_1' number of buildings in a compound structure are equally likely events in an image where each possible value of k_1 should be given as input if it is desired to find all the occurrences of the buildings with a green area in the middle. Furthermore, results obtained by smaller k_1 values can also be used to limit the parts of the image for which the algorithm will be run for a larger k_1 value. Because, the regions that are selected for a large k_1 value will probably be selected for a smaller k_1 value as well.

Chapter 7

Summary and Conclusions

We described a generic method for the modeling and detection of compound structures that consisted of arrangements of unknown number of primitive objects from different primitive object layers in large scenes. Each primitive object in a compound structure is represented as an ellipse whose centroid, major-minor axis lengths and orientation parameters were treated as random variables. Given a relatively small number of example structures, the modeling process built an MRF-based contextual model on the vectors of random variables composing the compound structure of interest. The proposed MRF model was different from the conventional pixel-based MRF structures used in the computer vision literature in the sense that the neighborhood structure was not fixed and the number of edges connecting primitives were much smaller than the number of edges connecting pixels. The parameters of this model were learned via sampling from the corresponding maximum entropy distribution. We also showed that were able to sample realistic arrangements from the learned model. Our training set was very small and hence, the human effort in learning was very little.

The detection problem was posed as a combinatorial selection problem where multiple subsets of a large pool of candidate regions from multiple hierarchical segmentations were selected via joint sampling of groups of regions by maximizing the likelihood of their individual appearances and relative spatial arrangements.

For this aim, the vertices at different levels of each hierarchy tree as well as the vertices across different trees were connected by edges to form an adjacency graph of neighboring vertices. The sampling of the resulting graph was done as clusters of vertices flipping together with equal probabilities.

Furthermore, we extended the selection formulation in order to handle possible hard constraints imposed on the candidate regions. The selection problem was reconstructed as a quadratic optimization problem with linear equality and inequality constraints. This extension of the unconstrained selection formulation necessitated a new algorithm which could consider the added constraints while maximizing the likelihood of the selected regions. We utilized the Difference of Convex algorithm which was proposed to solve the general non-convex problems that could be written as the difference of two convex problems.

Large-scale experiments using very high spatial resolution imagery showed that the proposed method could effectively localize unknown number of instances of different compound structures that could not be detected by using spectral and shape features alone or by using a tile-based approach. Selection results in different types of landscapes such as residential, industrial, agricultural and rural areas also demonstrated the scope of applicability of the proposed technique. The scenarios chosen in the experiments involved structures that had some particular arrangement in them. We investigated that other complex structures such as stadium, golf courses, nuclear power plants mostly involve co-occurrences rather than spatial arrangements as the most dominant contextual cues. However, the proposed selection was generic so that, first, a model of co-occurrence histograms of a large set of primitives can be learned and then, can be given to the selection algorithm to detect such structures as well. Moreover, comparing the size of the training set with the approaches that need a very large amount of training data to learn the appearances of compound structures and considering the large-scale outputs of our approach, our approach can be considered very effective in terms of human intervention and returned output. This is mainly due to the stochastic nature of the approach that can handle low probability explanations as opposed to deterministic approaches such as SVM and neural networks that probably overfit with such a small data.

We also observed that using multi-scale segmentations, rather than a single segmentation, as input to the detection procedure removed the assumption that the initial segmentation is correct since some objects were selected from only one scale of the hierarchy due to their erroneous segmentations in other scales. But still, detailed analysis of the overall results revealed that one of the most important bottlenecks for the proposed method in terms of accuracy was the errors in the input hierarchical segmentations. Even though we use a hierarchical segmentation for each layer to minimize the risk of missing an object, some objects still did not appear in the hierarchy. However, the proposed approach is suitable for adding multiple hierarchical trees to each layer obtained by different segmentation algorithms in order to increase the possibility of accurately obtaining a region in at least one of these candidate segmentations. Furthermore, complex models for each layer such as SVM can be used to obtain better candidates. However, this will require a training data for learning the model for each layer. We think that this is not one of the main scopes of this dissertation. Additionally, the multi-layered experiments showed that selection of some objects required the selection of objects in other layers that had spatial relation with them. Furthermore, in the constrained selection experiments, we demonstrated that we could combine the soft arrangement and appearance constraints appeared in the probability distribution and general hard constraints appeared as linear equality and inequality functions in a single detection framework.

As a future work, the proposed sampling algorithm can be used to predict new instances of objects that are missed due to several reasons in the detection process. As stated in the experiments, one of the most important reasons was that the missed objects never appeared in the candidate hierarchies because of under- or over-segmentation of the object in the hierarchy. Whenever we detect a compound structure to which the missed object should have actually belonged to, we may predict additional objects by Gibbs sampling ellipses inside the window enclosing the compound structure. These object proposals may be added to the candidate hierarchy for a final round of detection process. Another future direction can be to discover the hidden primitive layers inside a compound structure each of which has an unknown number of similar primitives. This corresponds to either estimating

the candidates in an unsupervised manner before selection or jointly estimating which candidate belongs to which layer and the arrangement parameters. The former can be stated as the problem of interesting co-occurrences in the image that can be found by graph clustering. Obviously, the latter is more difficult to solve where the same hierarchical segmentation can be given as candidates for all layers and the layer of each candidate is unknown. The task will be to learn which selected region belongs to which layer as well as the spatial arrangements. Finally, we believe that this work can also be used for different applications, especially in the medical domain. For this aim, one may define primitives of the complex structure of interest and learn its spatial layout to detect new instances.

Bibliography

[1] D. Zamalieva, S. Aksoy, and J. C. Tilton, "Finding compound structures in images using image segmentation and graph-based knowledge discovery," in *IEEE International Geoscience and Remote Sensing Symposium*, vol. V, pp. 252–255, 2009.

[2] H. G. Akcay and S. Aksoy, "Detection of compound structures using multiple hierarchical segmentations," in *IEEE International Geoscience and Remote Sensing Symposium*, pp. 6833–6836, 2012.

[3] C. Ari and S. Aksoy, "Detection of compound structures using a Gaussian mixture model with spectral and spatial constraints," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, pp. 6627–6638, October 2014.

[4] G. Moser and S. B. Serpico, "Combining support vector machines and markov random fields in an integrated framework for contextual image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, pp. 2734–2752, May 2013.

[5] J. Porway, Q. Wang, and S. C. Zhu, "A hierarchical and contextual model for aerial image parsing," *International Journal of Computer Vision*, vol. 88, pp. 254–283, June 2010.

[6] C. Benedek, X. Descombes, and J. Zerubia, "Building development monitoring in multitemporal remotely sensed image pairs with stochastic birth-death dynamics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 33–50, January 2012.

[7] H. G. Akcay and S. Aksoy, "Automatic detection of compound structures by joint selection of region groups from a hierarchical segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, pp. 3485–3501, June 2016.

[8] Z. Sun, H. Fang, M. Deng, A. Chen, P. Yue, and L. Di, "Regular shape similarity index: A novel index for accurate extraction of regular objects from remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, pp. 3737–3748, July 2015.

[9] Z. Li, W. Shi, Q. Wang, and Z. Miao, "Extracting man-made objects from high spatial resolution remote sensing images via fast level set evolutions," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, pp. 883–899, February 2015.

[10] J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, "Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, pp. 3325–3337, June 2015.

[11] H. Duan and L. Gan, "Elitist chemical reaction optimization for contour-based target recognition in aerial images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, pp. 2845–2859, May 2015.

[12] R. R. Vatsavai, A. Cheriyadat, and S. Gleason, "Supervised semantic classification for nuclear proliferation monitoring," in *IEEE Applied Imagery Pattern Recognition Workshop*, 2010.

[13] J. Graesser, A. Cheriyadat, R. R. Vatsavai, V. Chandola, J. Long, and E. Bright, "Image based characterization of formal and informal neighborhoods in an urban landscape," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, pp. 1164–1176, August 2012.

[14] L. Gueguen, "Classifying compound structures in satellite images: A compressed representation for fast queries," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, pp. 1803–1818, April 2015.

[15] X. Yao, J. Han, G. Cheng, X. Qian, and L. Guo, "Semantic annotation of high-resolution satellite images via weakly supervised learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, pp. 3660–3671, June 2016.

[16] Y. Yang and S. Newsam, "Geographic image retrieval using local invariant features," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, pp. 818–832, February 2013.

[17] B. Demir and L. Bruzzone, "A novel active learning method in relevance feedback for content-based remote sensing image retrieval," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, pp. 2323–2334, May 2015.

[18] B. Demir and L. Bruzzone, "Hashing-based scalable remote sensing image search and retrieval in large archives," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, pp. 892–904, February 2016.

[19] C. Vaduva, I. Gavat, and M. Datcu, "Latent Dirichlet allocation for spatial analysis of satellite images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, pp. 2770–2786, May 2013.

[20] C. Kurtz, N. Passat, P. Gancarski, and A. Puissant, "Extraction of complex patterns from multiresolution remote sensing images: A hierarchical top-down methodology," *Pattern Recognition*, vol. 45, pp. 685–706, February 2012.

[21] R. Gaetano, G. Scarpa, and G. Poggi, "Hierarchical texture-based segmentation of multiresolution remote-sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, pp. 2129–2141, July 2009.

[22] M. C. Vanegas, I. Bloch, and J. Inglada, "Alignment and parallelism for the description of high-resolution remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, pp. 3542–3557, June 2013.

[23] H. G. Akcay and S. Aksoy, "Automatic detection of geospatial objects using multiple hierarchical segmentations," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, pp. 2097–2111, July 2008.

[24] M. Dalla Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone, "Morphological attribute profiles for the analysis of very high resolution images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, pp. 3747–3762, October 2010.

[25] Y. Tarabalka, J. C. Tilton, J. A. Benediktsson, and J. Chanussot, "A marker-based approach for the automated selection of a single segmentation from a hierarchical set of image segmentations," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, pp. 262–272, February 2012.

[26] L. Gueguen, G. K. Ouzounis, M. Pesaresi, and P. Soille, "Tree based representations for fast information mining from VHR images," in *ESA-EUSC-JRC Conference on Image Information Mining*, vol. 1, pp. 15–20, 2012.

[27] B. R. Kiran and J. Serra, "Global-local optimizations by hierarchical cuts and climbing energies," *Pattern Recognition*, vol. 47, pp. 12–24, January 2014.

[28] X. Zhang, P. Xiao, X. Feng, L. Feng, and N. Ye, "Toward evaluating multiscale segmentations of high spatial resolution remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, pp. 3694–3706, July 2015.

[29] Y. Zhong, R. Gao, and L. Zhang, "Multiscale and multifeature normalized cut segmentation for high spatial resolution remote sensing imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, pp. 6061–6075, October 2016.

[30] J. Inglada and J. Michel, "Qualitative spatial reasoning for high-resolution remote sensing image analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, pp. 599–612, February 2009.

[31] M. C. Vanegas, I. Bloch, and J. Inglada, "Fuzzy constraint satisfaction problem for model-based image interpretation," *Fuzzy Sets and Systems*, vol. 286, pp. 1–29, March 2016.

[32] S. Sarkar and K. L. Boyer, "Perceptual organization in computer vision: A review and a proposal for a classificatory structure," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, pp. 382–399, March/April 1993.

[33] A. Y.-S. Chia, D. Rajan, M. K. Leung, and S. Rahardja, "Object recognition by discriminative combinations of line segments, ellipses, and appearance features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 1758–1772, September 2012.

[34] A. L. Berger, V. J. Della Pietra, and S. A. Della Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, pp. 39–71, March 1996.

[35] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[36] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, pp. 1087–1092, June 1953.

[37] R. H. Swendsen and J.-S. Wang, "Nonuniversal critical dynamics in Monte Carlo simulations," *Physical Review Letters*, vol. 58, pp. 86–88, January 1987.

[38] D. M. Higdon, "Auxiliary variable methods for Markov chain Monte Carlo with applications," *Journal of the American Statistical Association*, vol. 93, pp. 585–595, June 1998.

[39] R. Horst and N. V. Thoai, "Dc programming: Overview," *Journal of Optimization Theory and Applications*, vol. 103, pp. 1–43, October 1999.

[40] H. A. L. Thi, T. P. Dinh, and N. D. Yen, "Properties of two dc algorithms in quadratic programming," *Journal of Global Optimization*, vol. 49, pp. 481–495, March 2011.

[41] T. P. Dinh, H. A. L. Thi, and F. Akoa, "Combining dca (dc algorithms) and interior point techniques for large-scale nonconvex quadratic programming," *Optimization Methods and Software*, vol. 23, pp. 609–629, August 2008.

[42] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[43] S. Aksoy, I. Z. Yalniz, and K. Tasdemir, "Automatic detection and segmentation of orchards using very high-resolution imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, pp. 3117–3131, August 2012.

[44] N. R. Harvey and J. Theiler, "Focus-of-attention strategies for finding discrete objects in multispectral imagery," in *SPIE Imaging Spectrometry X*, vol. 5546, pp. 179–189, 2004.

[45] T. Kemper, M. Jenerowicz, M. Pesaresi, and P. Soille, "Enumeration of dwellings in Darfur camps from GeoEye-1 satellite images using mathematical morphology," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 4, pp. 8–15, March 2011.

[46] Y. Yang and S. Newsam, "Estimating the spatial extents of geospatial objects using hierarchical models," in *IEEE Workshop on Applications of Computer Vision*, pp. 305–312, 2012.

[47] S. Wu, J. Silvan-Cardenas, and L. Wang, "Per-field urban land use classification based on tax parcel boundaries," *International Journal of Remote Sensing*, vol. 28, pp. 2777–2800, June 2007.

[48] B. C. Gao, "NDWI-A normalized difference water index for remote sensing of vegetation liquid water from space," *Remote Sensing of Environment*, vol. 58, pp. 257–266, December 1996.