A Hierarchical and Contextual Model for Aerial Image Understanding

Jake Porway, Kristy Wang, and Song Chun Zhu
University of California
Los Angeles, CA
{jporway, qcwang, sczhu}@stat.ucla.edu

Abstract

In this paper we present a novel method for parsing aerial images with a hierarchical and contextual model learned in a statistical framework. We learn hierarchies at the scene and object levels to handle the difficult task of representing scene elements at different scales and add contextual constraints to resolve ambiguities in the scene interpretation. This allows the model to rule out inconsistent detections, like cars on trees, and to verify low probability detections based on their local context, such as small cars in parking lots. We also present a two-step algorithm for parsing aerial images that first detects object-level elements like trees and parking lots using color histograms and bag-of-words models, and objects like roofs and roads using compositional boosting*, a powerful method for finding image structures. We then activate the top-down scene model to prune false positives from the first stage. We learn this scene model in a minimax entropy framework and show unique samples from our prior model, which capture the layout of scene objects. We present experiments showing that hierarchical and contextual information greatly reduces the number of false positives in our results.*

1. Introduction and Related Work

Aerial image understanding is a widely studied topic of great importance for military, navigational, and surveillance tasks. Aerial images have two prominent features that differentiate them from other natural images:

Long Range: Objects of interest in aerial images exist at very different sizes, from large blocks of buildings to small, individual cars. It is nearly impossible to model and detect these objects successfully at a single scale.

Wide View: Unlike many images used for object detection that have a few objects present in consistent configurations, aerial images can have hundreds of objects present, creating a countless number of potential spatial layouts.

Work in aerial image understanding has commonly addressed the problems above in one of two ways. One simplification of the problem is to work in a narrow depth range and detect just one type of object, such as rooftops [9, 15, 16] or cars [18]. In this domain, higher level cues, such as context, are of little benefit, as researchers need only concern themselves with intraclass context, such as whether two of the same object overlap. This line of study has produced good results for single objects, but generally ignores multi-category situations.

An improvement over the method above is to extend the task to identifying multiple object types, but to code spatial context using a hardcoded logic-based model [10, 12]. This

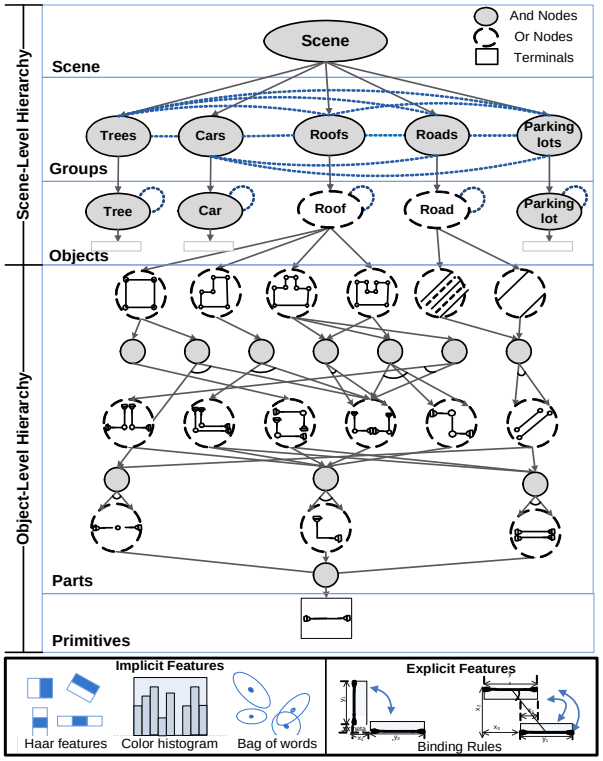

Figure 1. Our three-level hierarchy. The scene decomposes into sets of group nodes, which in turn decompose into sets of individual objects, which are represented either at that level or by further hierarchical decomposition. The features at the bottom are used to detect the objects during the inference stage.

work approaches the goal of image understanding much more closely than the single-class case, but relies on hand-coded models and relationships, which are non-scalable and require human intervention should the model need to change. The work in [13] proposes probabilistic relationships between objects, but the hierarchical grouping and instantiation of these relationships is still fixed.

The field of object recognition has recently begun focusing on hierarchies and context information for object and scene classification [3, 5, 7]. We adopt some of these ideas to apply to aerial image understanding:

Multi-category hierarchy - We propose a novel two-layer hierarchical model that represents the image from the scene level down to the pixel level. Figure 1 shows a depiction of this hierarchy, in which the scene level decomposes first into groups of objects. Groups, like blocks of buildings or rows of cars, are fairly unique to aerial images, as there are few image domains in which multiple instances of the same object exist in large groups. These groups decompose into single objects, some of which, like roofs, decompose further into parts and primitives.

Context learned from real data - We model context as constraints on the attributes of objects in the scene. For example, cars are associated with roads and appear contained within them at the appropriate scale. This context also lets us resolve ambiguities across different object scales, for example ruling out vents on roofs that are often detected as cars.

Our two-layer hierarchical model helps capture the **long range** of object sizes by representing scene elements at different scales, while the contextual part of the model captures the interactions across the **wide view** of objects present in the scene.

Our hierarchy also models the different characteristics of the scene at varying scales. At the scene level we observe loosely constrained groups of objects, easily modeled by the soft, descriptive constraints of an MRF model [6, 11]. At the object level, however, we observe tightly constrained parts, such as the edges forming the boundary of a roof. These require explicit bindings. There is still variation at the object level, modeled by the "Or" nodes in Figure 1. A roof can take many different shapes, each of which can be formed from many different combinations of subparts. The Or nodes model the possibility for an object to be modeled as one of many part compositions.

We implement a two step inference algorithm that takes advantage of the hierarchy and context in our model. In the first phase, we use compositional boosting [17] to detect roofs and roads, while we use low-level features, like those shown at the bottom of Figure 1 to detect the remaining object categories, parking lots, trees, and cars. Compositional boosting is a hierarchical process that groups edges into larger structures based on weak classifers learned on their geometric and photometric features. This grouping process passes information up and down its hierarchy until objects are finally confirmed. This is a powerful method for object detection that has not yet been applied to aerial image modeling.

The first inference phase is designed to ensure a very high true positive rate, but at the cost of having many false positives. In the second phase of our algorithm we activate the top-down scene-level component of the model to prune inconsistent false positives using local context, resulting in a much improved interpretation of the scene.

Figure 2 shows an example of an aerial image parsed using our model. Figure 2(b) shows the labeled objects detected in the scene, while Figure 2(c) shows the hierarchical decomposition of the scene. In this decomposition, edges have been grouped into buildings, which have been grouped into city blocks. These objects are constrained by contextual relationships, examples of which are shown in Figure 2(d). This figure visualizes which relationships exist between different objects in the parse. For example, Figure 2(d)(1) shows which objects are aligned. Figure 2(d)(2) and Figure 2(d)(3) show which objects are related by the overlap and relative position relationship, respectively. For example, cars obey the constraint that they overlap the road. These relationships have been learned from a training set of parsed aerial images.

In this paper we first discuss the representation of our contextual hierarchy in Section 2. We then discuss how to learn its parameters and show samples from this learned prior in Section 3. Next we describe a greedy inference algorithm in Section 4, which combines bottom-up results from our object model with our top-down scene model to arrive at the most reasonable explanation of the scene. We finally show results where the hierarchical and contextual information greatly improve our pure bottom-up detection.

2. Contextual Hierarchical Model

Figure 1 shows a diagram of our two-layer hierarchical representation consisting of the scene-level hierarchy and the object-level hierarchy model.

2.1. Hierarchical Composition

Scene-Level Hierarchy We can express the decomposition rules for the scene-level in a grammar format:

1. S $\rightarrow$ Roofs(n_1) $\oplus$ Cars(n_2) $\oplus$ Roads(n_3) $\oplus$ Trees(n_4) $\oplus$ Parking Lots(n_5), $n_i \sim p(n_i)$
2. Roofs $\rightarrow$ Roof(m_1), $m_1 \sim p(m_1)$
3. Cars $\rightarrow$ Car(m_2), $m_2 \sim p(m_2)$
4. Roads $\rightarrow$ Road(m_3), $m_3 \sim p(m_3)$
5. Trees $\rightarrow$ Tree(m_4), $m_4 \sim p(m_4)$
6. Parking Lots $\rightarrow$ Parking Lot(m_5), $m_5 \sim p(m_5)$

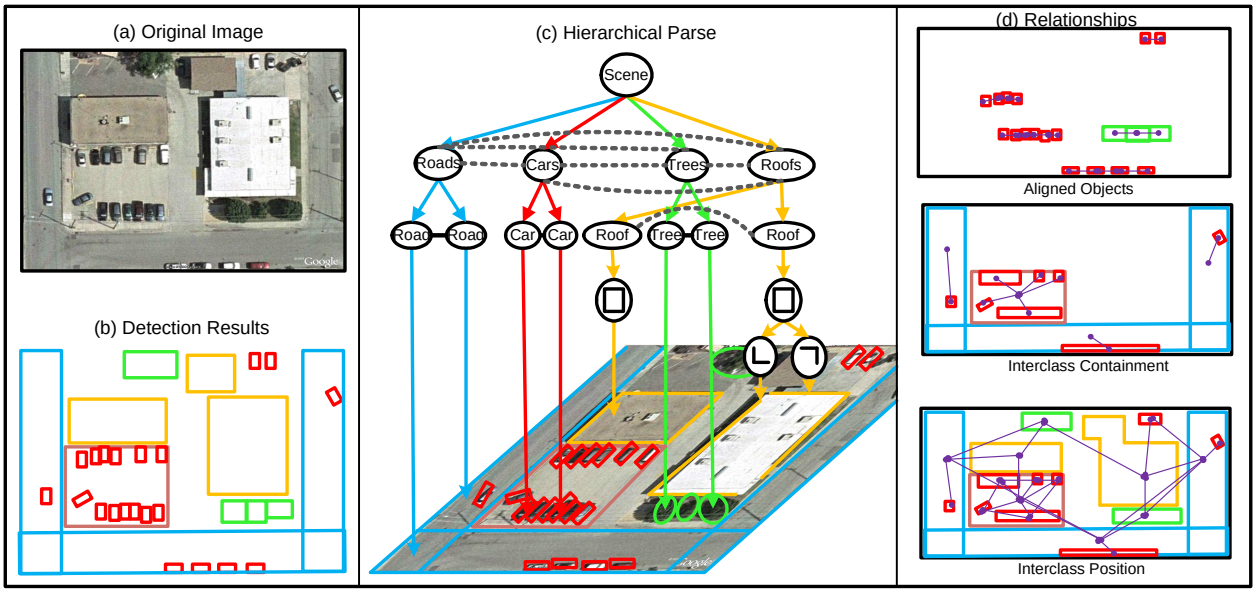

Figure 2. A running example. (a) Original image (b) Detection results on image (c) Hierarchical parse graph g (d) Constraints between objects (1) Aligned objects grouped together (2) Objects that overlap or contain one another (3) Objects related by relative position.

where n_i and m_i are integral values determining the cardinality of each decomposed set.

This portion of the model is very similar to a hierarchical Dirichlet prior [14], in that the scene decomposes into a number of groups, which in turn decompose into a number of single objects. We choose instead to represent these decomposition rules as constraints to keep our formulation unified, which we discuss in Section 3.

Object-Level Hierarchy Nodes in the object-level hierarchy can terminate as implicit representations or decompose into their own hierarchy. Cars, trees, and parking lots are modeled using color histograms and bags of SIFTs, and thus terminate at this level. Roofs and roads, however, are defined by a hierarchy of grouped edge primitives.

Figure 1 shows the object-level decomposition. Roofs can take on one of many shapes, each of which can be formed from simpler edge groups, which in turn can be formed from collections of edges. For example, a rectangle can be formed from two L-junctions, or from two perpendicular sets of parallel lines. The uncertainty in decomposition is modeled by the Or nodes in Figure 1, indicating that a roof can decompose into one of many shapes. Each Or node takes on an integral value during the parse phase that determines which child it decomposes into: $\omega(v^{Or}) = i;\ \ i = 1, 2, \ldots, m$.

Decomposing the scene node down into objects and then into parts creates a "parse graph" g from our model, consisting of a set of nodes V and relations between them. Every node instance $v_i \in V$ can be represented by the following attributes, derived from the boundary points defining it:

$$A(v_i) = \{X_i, \theta_i, \sigma_i\} \quad (1)$$

where X_i is the center of mass, θ_i the orienation, and σ_i the scale. $A(v_i)$ serves as a general set of features for constraint formulation in the next section.

2.2. Contextual Relations

The true power of our model comes from adding context to the existing hierarchy through contextual constraints, which determine the relative appearance of related parts. Contextual constraints model the distributions of certain relationships between objects, for example relative scale. Figure 1 shows these constraints as dashed horizontal lines.

Scene-Level Context A contextual relationship r_i is simply a function of the geometric attributes of one or more nodes $V = \{v_1, v_2, \ldots, v_k\}$, $\phi = r_i(\vec{A}(V))$.

We define a dictionary of relationship functions, Δ_R. For a relationship $r_i \in \Delta_R$, we can compute its value ϕ_{ij} for every realization $V_j \subseteq V$ of a set of nodes in a dataset. For example, to compute the position relationship between the "car" and "road" nodes, we obtain every pair of cars and roads nodes in a set of training data and return the distance between them. We can then model the distribution of these values using a histogram, $H(r_i(\vec{A}(V_j)))$, for each constraint. These loose distributions are similar in spirit to the MRF models proposed in [6] and [11].

Adjacencies We only want to measure relationships across node instances that influence each other. For example, V_j may be $\{Roofs, Trees\}$, but the instances of roofs and trees in each V_j may be so far away as to not influence one another. Thus we add an indicator function for each relationship r_i to determine if a set of nodes is adjacent, and

thus valid to be operated on.

$$I_i(\vec{A}(V_j)) = \begin{cases} 1 & \text{if } f_i(\vec{A}(V_j)) < t_i, \\ 0 & \text{else} \end{cases}$$

where f_i is a function over the node instances in V_j and t_i is its corresponding threshold. Note that "adjacent" here is defined differently for each r_i, and is not necessarily solely a function of distance.

Object-Level Context At the object-level our constraints change slightly. We are now more interested in low-level Gestalt features, such as parallelism, perpendicularity, collinearity, but these can still be modeled as above.

3. Learning

We now learn a probability distribution, $p(g;\Theta)$, on both levels of our hierarchical representation together. $p(g;\Theta)$ is the probability of a parse g and is learned in two steps. We first define the hierarchical component of the whole model $p_0(g;\Theta_0)$, then iteratively add contextual relations to get our final constrained model, $p(g;\Theta)$.

$$p_0(g;\Theta_0) \overset{r_1}{\Rightarrow} p_1(g;\Theta_1) \overset{r_2}{\Rightarrow} \dots \overset{r_k}{\Rightarrow} p_k(g;\Theta_k) \tag{2}$$

where Θ is the parameter vector for the model.

3.1. Probability Model

Given a set of annotated parse graphs of aerial images $g^{obs} = \{g_1^{obs}, g_2^{obs}, \dots, g_n^{obs}\}$, we would like our model, $p(g;\Theta)$, to approximate the true underlying distribution, $f(g;\Theta)$, of these parses. This model needs to match:

1. The distribution of the number of parts the scene and group nodes decompose into.
2. The frequency with which Or nodes decompose into their children.
3. The distribution of the relationships between nodes.

We can use these constraints to derive our probability model using minimax entropy, resulting in a standard Gibbs distribution [19, 20] where $\Theta = \{\lambda_\alpha, \lambda_\beta, \lambda_w, \lambda_i\}$ are Lagrange parameters to be estimated:

$$p(g;\Theta) = \frac{1}{Z(\Theta)} \exp^{-(E_0(g)+E_1(g))} \tag{3}$$

$$E_0(g) = \sum_{i=1}^{5} \lambda_\alpha(|v_i^G|) + \sum_{i=1}^{5} \sum_{j=1}^{|v_i^G|} \lambda_\beta(|v_j^O|)) + \sum_{v_i \in V^{Or}(g)} \lambda_w(\omega(v_i)) \tag{4}$$

$$E_1(g) = \sum_{i=1}^{k} \sum_{V_j \in V} \lambda_i(r_i(\vec{A}(V_j)))I_i(\vec{A}(V_j)) \tag{5}$$

Here $E_0(g)$ is the energy associated with the hierarchical component of our model, including terms for the number of group nodes v^G and object nodes v^O present, as well as for the decomposition of each Or node V^{Or}. $E_1(g)$ is the energy of the k contextual constraints selected for this model. The indicator I_i ensures that only instances that are adjacent are counted towards the energy. We can first learn the hierarchical parameters $\{\lambda_\alpha, \lambda_\beta, \lambda_w\}$ using MLE [1], then iteratively add relations to the hierarchy following a minimax entropy framework [20].

3.2. Relationship Pursuit

Scene-Level Relationship Pursuit We begin with a model $p_0(g;\Theta_0)$ containing only our hierarchical parameters, then augment that model to $p_+(g;\Theta_+)$ one constraint at a time. Keeping with a minimax entropy framework, we select the relationship r_+^* at each step that maximizes the distance between our current model and the augmented model, giving $p_+^*(g;\Theta_+^*)$. Like texture synthesis, we use the squared distance between our current model and the observed histogram for r_i as our metric. Unlike texture synthesis, however, not all constraints may be present between the same sets of nodes in every image, so we must weight this distance metric by the frequency of each relationship, $f(r_i)$.

$$\begin{aligned} r_+^*(g) &= \underset{r_+}{\text{argmax}}\{KL(f(g)|p_+(g)) - KL(f(g)|p(g))\} \\ &= \underset{r_+}{\text{argmax}}\, D(p_+(g)|p(g)) \end{aligned} \tag{6}$$

$$D(p_+(g)|p(g)) \cong f(r_i)|H(r_i)^{obs} - H(r_i)^{syn}| \tag{7}$$

$H(r_i)^{obs}$ is the observed histogram for this relationship, while $H(r_i)^{syn}$ is the histogram created by samples drawn from our current model. The bigger $|H(r_i)^{obs} - H(r_i)^{syn}|$ is, the more information adding this relation would contribute to this model. In this way, we add constraints that produce the most information gain, i.e. bring our new model p_+ maximally far away from our old model p.

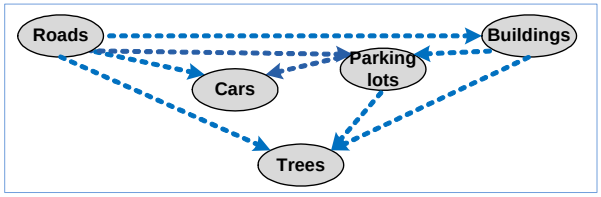

Figure 4. Relationship constraints between groups modeled as a directed acyclic graph. This adjustment is made to the model for sampling.

It bears noting that the relationships at the group level can exist between any pair of objects, but can be rewritten in a partial ordering as a directed acyclic graph where each object's appearance depends only on a set of the other objects. An example is shown in Figure 4. This is necessary

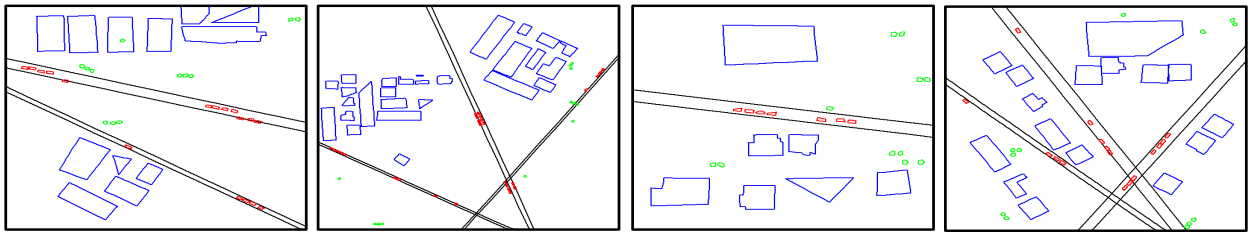

Figure 3. Samples drawn from the scene prior. This analysis-by-synthesis shows the traits our model captures, similar to texture modeling.

for sampling, in which it is intractable, without adapting something like Swendsen-Wang cuts, to arrange all of the parts at once. We can first sample roads, then sample cars given roads, then sample roofs given roads and cars, and so on.

Figure 3 shows samples drawn from our learned scene-level model. Here we model four categories of objects and model the relationships of relative scale, relative position, relative orientation, percentage overlap, aspect ratio, and alignment. The boundaries are sampled from the training data. We can see that the scenes are similar to urban aerial images, marked by roads of consistent size, cars contained within roads, no overlaps, and clustering of objects.

Object-Level Relationship Pursuit Relationships at the object level are not pursued, but instead are all present. We learn thresholds on these energy functions, or "explicit tests", to determine if nodes should be combined during inference. In addition to these explicit tests for nodes, we learn "implicit tests" for single nodes, which are simply strong classifiers learned from Adaboost [2].

4. Inference

Our inference algorithm proceeds in two phases. We first identify single object nodes in the image using specific bottom-up detectors for each object class. We then activate the top-down object/scene level of the model to prune incompatible proposals and arrive at the most likely description of the scene.

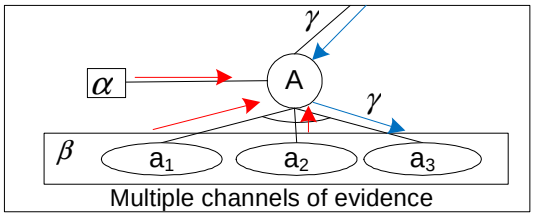

Figure 5. Information about the presence of a node may come from a bottom-up detector, detected children, or a detected parent.

4.1. Bottom-up Object-Level Detection

We first detect single objects in the scene using detectors suited to their representations.

Cars: Cars are represented by Haar filter responses, and are detected using Adaboost [2].

Trees: Trees are represented by color histograms. For every 7x7 window in each image, we compare the window's histogram to a learned category histogram and accept the pixel as belonging to a tree if the product between the two is below some threshold.

Parking Lots: Parking lots are represented using a histogram of SIFT features. Like trees, we move 80x80 windows across the image to find matching parking lot regions.

Compositional Boosting For the more complex cases of roads and roofs we use compositional boosting [17], exploiting the implicit and explicit tests we learned in Section 3. Figure 5 shows the way compositional boosting introduces context during inference. A node A may receive evidence of its existence from one of three channels, named the α, β, and γ channels. The α channel comes directly from pixel-level evidence, such as Adaboost detection results for that node. The β channel submits evidence for A from the existence of its children. The γ channel provides evidence for A due to the existence of its parent. For example, a roof may be detected directly from the pixel-level results of Adaboost, or it may be proposed because two opposing L-junctions exist under certain constraints. Thanks to the γ channel, we can also detect mid and low-level nodes that were previously undetected due to the existence of their parent.

Compositional boosting operates on a primal sketch of an image, which is similar to an edge map [17]. The algorithm strives to encode this sketch with as many composite edge features as possible. In our case, we are trying to find the best "roof encoding" of a sketch of our image. This is done by first searching the input sketch for every possible node in the hierarchy using its implicit representation, the strong classifier learned for that node. Each particle is then weighted by a local posterior probability ratio of how well it encodes a patch relative to other particles. The algortihm then proposes new candidates by binding or decomposing the implicitly detected nodes into higher and lower level structures. These proposals are similarly weighted.

At each iteration we greedily select the candidate from our proposal set with the highest weight. We then reweight the remaining candidates according to whether or not the newly selected particle overlaps their domain or alters the evidence that they exist. For example, if we select a low

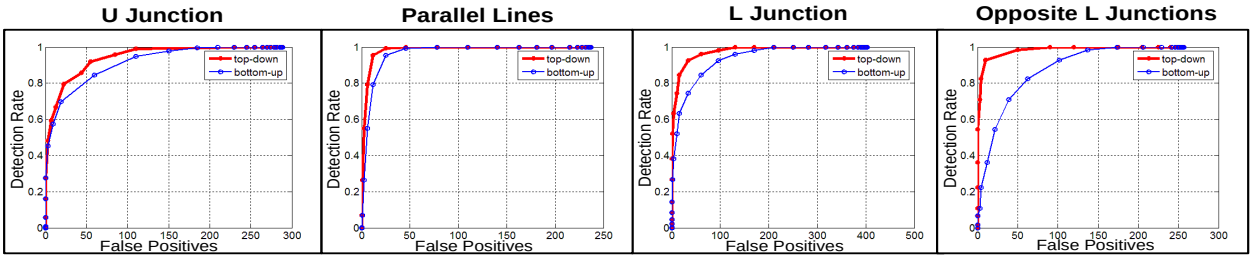

Figure 6. ROC curves for line structures with and without compositional boosting. The blue curve shows results using just one-pass of Adaboost, while the red curve shows the improvement from using top-down information from compositional boosting.

level node, it would increase the weight on the proposal that its parent existed. We refer the reader to [17] for more details on this formulation. Suffice it to say that, given an edge image, we first propose nodes using implicit and explicit tests, then iteratively select and reweight particles that best explain the image. By the end we have a hierarchical decomposition of the sketch of the image, yielding roof and road candidates.

4.2. Top-down Pruning

The previous step produces a huge number of candidate particles for each object category. We now want to pursue the g^* that maximizes our posterior distribution for the scene level:

$$g^* = \underset{g}{\operatorname{argmax}}\, p(I|g;\Theta)p(g;\Theta) \tag{8}$$

We optimize this value by pursuing candidates found in the bottom-up phase, similar to [8]. We greedily add nodes to a running parse, g, initially empty. At every iteration, we reweight each particle c_i from a set of detected candidate particles, $C = \{c_1, c_2, \ldots, c_k\}$ by the change in energy its addition produces, where $g_+ = g \cup \{c_i\}$.

$$w(c_i) = \log \frac{p(I|g_+;\Theta_+)}{p(I|g;\Theta)} + \log \frac{p(g_+;\Theta_+)}{p(g;\Theta)} \tag{9}$$

We model the likelihood for each object c_i based on how well it matches a color histogram for its object type, $H_i(I_{(x,y)})$, relative to the previous explanation of that area, $H_j(I_{(x,y)})$, which may be uniform if g doesn't yet explain those pixels, or may belong to whatever object is currently covering that region. We also measure the energy of the prior on g_+, which is simply the energy of the relationships created due to the addition of c_i.

$$\log \frac{p(I|g_+;\Theta_+)}{p(I|g;\Theta)} = \frac{\sum_{(x,y)\in\Lambda_i} \log H_i(I_{(x,y)})}{\sum_{(x,y)\in\Lambda_i} \log H_j(I_{(x,y)})} \tag{10}$$

$$\log \frac{p(g_+;\Theta_+)}{p(g;\Theta)} = -\sum_{i=1}^{k} \sum_{V_j \ni c_i} \lambda_i(r_i(\vec{A}(V_j))) \tag{11}$$

This proceeds until no candidates remain with $w(c_i) > 0$.

As this is a greedy algorithm, it is not guaranteed to converge to a global minimum. However, we have found that in practice, with good initial conditions, the algorithm achieves sensible parses. Our detectors are reliable enough that we are virtually ensured that the first particles picked are in fact correct objects.

5. Experiments

Training We learned our prior model and bottom-up parameters from 196 hand-labeled, multiresolution training images taken from Google Earth. This dataset included 10477 cars, 973 roofs, 202 roads, 584 parking lots, and 555 tree regions. We implemented relationships for aspect ratio, relative position, relative scale, relative orientation, percentage overlap, and grid alignment. We imposed grouping constraints dictating that single objects be grouped if their relative orientation varied less than 15 degrees from one another and were a distance less than or equal to twice the scale of the object along each axis away from one another. With this information, we were able to reconstruct the parses for each of the labeled images.

Our testing set was comprised of three large Google Earth images that were mosaicked together from many smaller high-resolution images. This allowed us to run our object detectors at multiple scales for each image.

Compositional Boosting Figure 6 shows ROC curves for our compositional boosting results on a subset of the training set. The blue curve shows just the initial implicit testing results of four types of edge structures. This curve is not very peaked, so Adaboost alone is not very effective for detecting these structures. By using compositional boosting to propose higher-level structures and then to re-verify originally missed edge structures, we see a huge improvement. The red curves show the improved detections using the multi-layer evidence from compositional boosting instead of just a single pass. This guarantees that we will have a higher detection rate for roofs and roads using full compositional boosting than simply using implicit detectors.

Top-down + Bottom-up Figure 7 shows results of the different stages of our algorithm on a series of aerial images. The first panel visualizes the compositional boosting

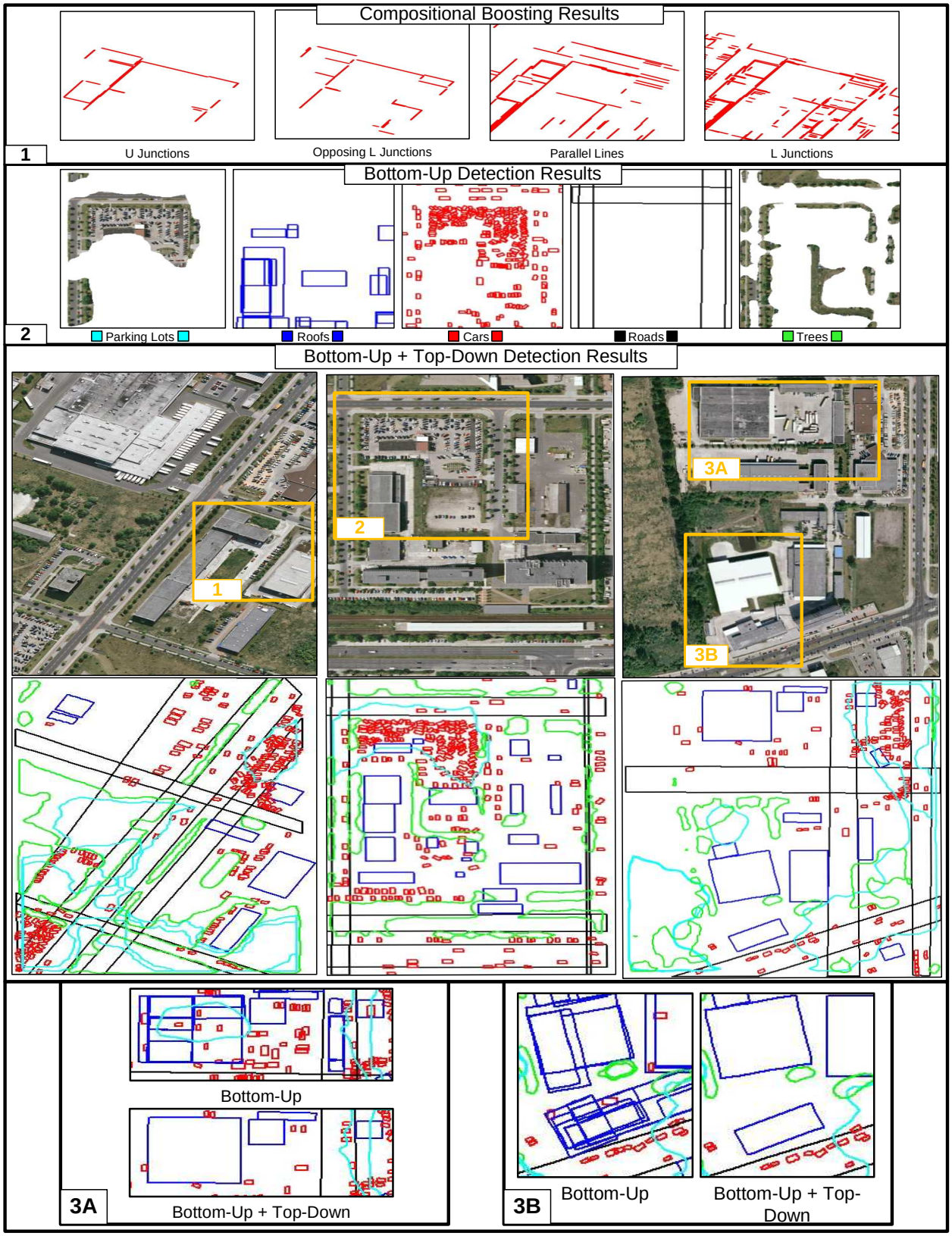

Figure 7. Results of parsed aerial images, with different object categories shown in different colors. (1) shows the bottom-up candidates from compositional boosting. (2) shows typical bottom-up results for each category at the scene level. The central panel shows parsed results for 3 typical images. Panel (3) shows close-up comparisons between bottom-up alone vs. bottom-up + top-down information.

results for the four part types on an area of the image. Panel 2 shows typical bottom-up detection results for an area of the image. Note the abundance of false positives. The center panel shows the final detection results for 3 images, using our top-down model to prune unlikely bottom-up candidates. We see that the majority of the objects are detected correctly and that we have very few inconsistencies. Panel 3 shows a close up of the results before and after top-down pruning. We can see that, beforehand, we have many overlapping inconsistent representations. After top-down information is introduced, these are pruned away.

We do see incorrect labelings as well, however. For example, the rightmost image has decided that the straight lines of buildings are in fact roads, thus ruling out the buildings there. Also, our training data included trees on the medians of roads. Thus, our model learned that trees can overlap roads, so we see proposals where trees block the entire road. These problems can be solved by weighting our likelihood term differently and by including more complex relations in our model.

Table 1 shows the improvement achieved using our top-down model. We compare the number of true positives and false positives in our testing set before and after top-down pruning. Though we lose some true positives during the top-down phase, we see that the false positives are drastically reduced. Looking at the images in Figure 7, these seem to correspond to instances that are fairly difficult even as a human to label. The top-down pruning has then in effect eliminated the majority of the false positives.

		Bottom-Up		Top-Down	
	Ground Truth	TP	FP	TP	FP
Roofs	59	56	117	48	24
Roads	9	9	8	9	6
Cars	806	768	415	651	31
Parking Lots	6	3	15	3	3
Trees	55	53	60	53	11

Table 1. Comparison of results between bottom-up and bottom-up with top-down pruning.

6. Conclusions and Future Work

We have shown a contextual hierarchical model that incorporates bottom-up and top-down information to parse a scene containing multiple object categories. The dual hierarchies succeed in capturing the relations at the object and scene levels and compositional boosting greatly improves our bottom-up detection rate. The top-down scene model is able to prune inconsistent candidates using scene context, producing far better precision than bottom-up detection alone. We hope in the future to improve this model by extending it to handle arbitrary object types and to implement top-down prediction in the scene-level hierarcy to help detect missing objects.

Acknowledgments

This work is supported by the IARPA/ODNI.

References

[1] Z. Chi, S. Geman, "Estimation of probabilistic context-free grammars", *Computational Linguistics*, v.24, June 1998. 4

[2] Y. Freund, R. Schapire, "A Decision-theoretic Generalization of On-line Learning and an Application to Boosting", Journal of Computer and System Sciences, n.55. 1997. 5

[3] F. Han and S.C. Zhu, "Bottom-up/top-down image parsing by attribute graph grammar", *ICCV*, 2005. 2

[4] S. Hinz, A. Baumgartner, "Road Extraction in Urban Areas Supported by Context Objects", Intl. Archives of Photogrammetry and Remote Sensing, volume 33(B3), 2000.

[5] Y. Jin and S. Geman, "Context and hierarchy in a probabilistic image model", *CVPR*, New York, June, 2006. 2

[6] V.P. Kumar and U.B. Desai, "Image interpretation using Bayesian networks", PAMI, 18(1), January 1996. 2, 3

[7] F.F. Li, P. Perona, "A Bayesian Hierarchical Model for Learning Natural Scene Categories", *CVPR* 2005. 2

[8] S. Mallat, and Z. Zhang, "Matching pursuit with time-frequency dictionaries", *IEEE Trans. on Signal Processing*, vol. 41, no. 12, 3397-3415, 1993. 6

[9] M. A. Maloof, P. Langley, T.O. Binford, R. Nevatia, S. Sage, "Improved Rooftop Detection in Aerial Images with Machine Learning". *Machine Learning*, 53, 2003. 1

[10] T. Matsuyama, V. Hang, "SIGMA: A Fromework for Image Understanding Integration of Bottom-up and Top-down Analyses", Plenum, New-York, 1990. 2

[11] J. Modestino, J. Zhang, "A Markov Random Field Model-Based Approach to Image Interpretation", PAMI, Volume 14, Issue 6, 1992. 2, 3

[12] H. Moissinac, H. Ma tre, I. Bloch, "Urban Aerial Image Understanding Using Symbolic Data", *Image and Signal Processing for Remote Sensing, Proc. SPIE*, 1994. 2

[13] A. Singhal, J. Luo, W. Zhu, "Probabilistic spatial context models for scene content understanding", *CVPR* vol.1, 2003. 2

[14] E. Sudderth, A. Torralba, W. Freeman, A. Wilsky, "Describing Visual Scenes Using Transformed Objects and Parts", *IJCV* 2007. 3

[15] C. Vestri, F. Devernay, "Using Robust Methods for Automatic Extraction of Buildings", *CVPR*, vol. 1.2, 2001. 1

[16] L. Wei, V. Prinet, "Building Detection from High-resolution Satellite Image Using Probability Model", *Geoscience and Remote Sensing Symposium, IGARSS*, 25-29 July 2005 1

[17] T.F. Wu, G.S. Xia, and S.C. Zhu, "Compositional Boosting for Computing Hierarchical Image Structures", *CVPR* , June, 2007. 2, 5, 6

[18] T. Zhao, R. Nevatia, "Car detection in low resolution aerial image" *ICCV*. Volume 1, vol.1, 2001. 1

[19] S.C. Zhu, D. Mumford, "Quest for A Stochastic Grammar of Images", *Foundations and Trends in Computer Graphics and Vision*, v.2, n.4, 2006. 4

[20] S. C. Zhu, Y. N. Wu, D. Mumford, "Minimax entropy principle and its application to texture modeling", Neural Computation, v.9 n.9 1997