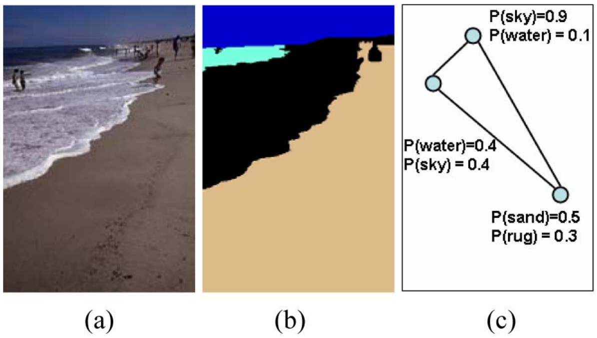
Scene Parsing Using Region-Based Generative Models

Matthew R. Boutell, *Member, IEEE*, Jiebo Luo, *Senior Member, IEEE*, and Christopher M. Brown

***Abstract*—Semantic scene classification is a challenging problem in computer vision. In contrast to the common approach of using low-level features computed from the whole scene, we propose "scene parsing" utilizing semantic object detectors (e.g., sky, foliage, and pavement) and region-based scene-configuration models. Because semantic detectors are faulty in practice, it is critical to develop a region-based generative model of outdoor scenes based on characteristic objects in the scene and spatial relationships between them. Since a fully connected scene configuration model is intractable, we chose to model pairwise relationships between regions and estimate scene probabilities using loopy belief propagation on a factor graph. We demonstrate the promise of this approach on a set of over 2000 outdoor photographs, comparing it with existing discriminative approaches and those using low-level features.**

***Index Terms*—Factor graph, generative models, scene classification, semantic features.**



(a) (b) (c)

Fig. 1. (a) Beach scene. (b) Its manually labeled materials. The true configuration includes *sky above water*, *water above sand*, and *sky above sand*. (c) The underlying graph showing detector results and spatial relations.

I. Introduction

SEMANTIC features, such as the output from object and material detectors, can help classify scenes when those features are available. As semantic (high-level) features have already begun to bridge the "semantic gap" between pixels and image understanding, scene classification using these features should become more accurate, given accurate detectors, than that using low-level features such as color or edges. While a scene classifier using reliable high-level semantic features seems like "standing on the shoulders of giants," the key is how to handle computed semantic features that are imperfect.

Semantic features include the output of object (e.g., buildings, airplanes, faces) and material (homogeneous, e.g., grass, sand, and cloth) detectors: we generically refer to image regions detected by either type of detector as *semantic regions*. A further advantage to approaches using these features is their modularity, allowing one to use independently developed, domain-sensitive detectors. Only recently has object and material detection in natural environments become accurate enough to consider using in a practical system. Recent work using object and material detection for other tasks [7], [12], [17] has achieved some success using primarily their presence or absence alone as evidence. However, despite improvements, the detectors still make errors, presenting a continuing difficulty for this approach.

How does one overcome detector errors? One way is to use a principled probabilistic inference system (versus a rule-based one [12]) to classify a scene based on the presence or absence of certain objects and materials. More importantly, one should extract additional useful evidence from the input image, such as *spatial relationships between the detected semantic regions*, to improve scene classification.

Fig. 1 shows an image; true identities of key regions (color-coded); detector results, expressed as confidence that each region is labeled with a given material; and spatial relationships between the regions. The problem is how to determine which scene type best explains the observed, often imperfect, evidence. As humans, we can easily see that the evidence taken as a whole [Fig. 1(c)], though ambiguous for each region individually, better fits a beach model than a field or city street model; our job is to train a system to do likewise.

We make the following contributions. 1) We present generative models, factor graphs in particular, for *scene configurations*, consisting of semantic regions' identities and their spatial relations, in Section II. 2) We show that such models can be used to classify medium- to long-distance outdoor scenes, given semantic features (Section III) and the scene configurations. Our technique is not applicable to close-up photos of people; face and body detection and recognition would be useful there, but is beyond the scope of this work. The most effective of these models uses pairwise spatial relationships between regions. 3) In Section V, we compare this model with three other generative models: an exact model that models the full joint distribution of

Manuscript received August 15, 2005; revised April 7, 2006. Preliminary findings in this paper were previously presented at the Proceedings of the Workshop on Statistical Relational Learning (SRL 2004). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Stefanos Kollias.

M. R. Boutell is with the Department of Computer Science and Software Engineering, Rose-Hulman Institute of Technology, Terre Haute, IN 47803 USA (e-mail: boutell@rose-hulman.edu).

J. Luo is with the Research and Development Laboratories, Eastman Kodak Company, Rochester, NY 14650 USA (e-mail: jiebo.luo@kodak.com).

C. M. Brown is with the Department of Computer Science, University of Rochester, Rochester, NY 14620 USA (e-mail: brown@cs.rochester.edu).

Color versions of Figs. 1–5 and 7–9 are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2006.886372

the scene type and every semantic region in the image, one that models co-occurrence of these regions while ignoring the actual spatial relations, and one that treats these regions independently. 4) Finally, we compare our model with a discriminative model that uses high-level features and with one that uses low-level features. In particular, the main novelty of the work lies in the *explicit* use of spatial relations in building a *generative model* to parse a scene, distinguishing it from other work using semantic features [7], [12], [17].

II. Generative Models for Scenes

Our generative model is based on the concept of scene configurations. *Scene configurations* consist of two parts. First is the actual *spatial arrangement* of semantic regions (edge labels in the graph of Fig. 1(c). Second is the *material configuration*, the identities of those regions [node labels in Fig. 1(c)]. We use theterm *material* in this paper generically to refer to both materials (grass, sky) and objects (buildings). We use the following terminology to discuss configurations:

n	number of distinct semantic regions detected in the image;
M	small set of semantically critical materials for which detectors are used;
$m_j \in M$	identity of semantic region j;
R	set of spatial relations (e.g., above, beside);
$r_{kl} \in R$	spatial relationship between semantic regions k and l;
G	set of configurations of materials in a scene;
g	single scene configuration.

Under this formalism, $G = m_1 \times m_1 \times \cdots m_n \times r_{11} \times r_{12} \times \cdots \times r_{(n-1)n}$ (we assume complete graphs), and an upper bound on the number of scene configurations, $|G|$, is

$$|G| = |M|^n \cdot |R|^{C(n,2)} \tag{1}$$

in a complete graph. We use the standard notation $C(n,k)$ for the number of combinations of k elements chosen from n. Each of the n regions could have one of M identities, and each of the $C(n,2)$ edges in the graph could be one of the $|R|$ spatial relations. However, some of these enumerated spatial arrangements are inconsistent, e.g., X *above* Y, Y *above* Z, X *below* Z. In our experiments, for example, we have $|M| = 10$ materials of interest (the potential labels for a single region) and $|R| = 7$ spatial relations. An image with $n = 4$ regions has 10^4 material configurations and $C(4,2) = 6$ pairwise spatial relations yielding 7^6 spatial arrangements and a total of $10^4 \cdot 7^6 \approx 1.2$ trillion scene configurations. While this is an overestimate, we clearly need an efficient method to determine which is most likely!

In the inference phase, the spatial arrangement of the regions in the test image is computed yet fixed; thus, its graph need only be compared with those of training images with the same arrangement. The distribution of material configurations with a *fixed* spatial arrangement can be learned independently of those with other spatial arrangements. Each such distribution has $|M|^n$ material configurations. For example, an image with two regions, r_1 *above* r_2, has only $|M|^2$ configurations. In our example above, once the spatial arrangement is known, there would only be $10^4 = 10{,}000$ possible material configurations.

A. Formalizing the Problem of Scene Classification From Configurations

We formalize the scene classification problem as follows: let $S = \{S_i\}$ be the set of scene classes considered, and $E = \{E_1, E_2, \ldots, E_n\}$ be the detector evidence, one for each of the n regions. Each $E_j = \{E_{j1}, E_{j2}, \ldots, E_{j|M|}\}$, in turn, is a confidence or belief vector for the identity of region j. In Fig. 1, for example, if we consider only detectors for sky, water, sand, and rug, $E_1 = \{0.9, 0.1, 0, 0\}$ for region 1 at the top. These beliefs are computed using a list of which material detectors fired on that region and with what belief, as described in Section IV-C.

In this framework, we want to find the scene with *maximum a posteriori* (MAP) likelihood, given the evidence from the detectors, or $\mathrm{argmax}_i P(S_i|E)$. By Bayes' Rule

$$P(S_i|E) = \frac{P(S_i)P(E|S_i)}{P(E)}. \tag{2}$$

At inference time, we have the evidence E, thus $P(E)$ is fixed and does not depend on the scene i. Thus

$$\arg\max_i P(S_i|E) = \arg\max_i P(S_i)P(E|S_i) \tag{3}$$

because the value of the argmax does not change when multiplying by the constant $1/P(E)$. Taking the joint distribution of $P(E|S_i)$ with the set of scene configurations G yields

$$\arg\max_i P(S_i) \sum_{g \in G} P(E, g|S_i). \tag{4}$$

Conditioning on g gives

$$\arg\max_i P(S_i) \sum_{g \in G} P(g|S_i) P(E|g, S_i). \tag{5}$$

B. Graphical Model

While all graphical models may have the same representational power, not all are equally suitable for a given problem. A two-level Markov random field (MRF) is a suitable choice for a region-based approach, due to the similarity to its use in low-level vision problems. However, we are solving a fundamentally different problem than those for which MRFs are used. MRFs are typically used to regularize input data [5], [9], finding $P(g|E)$, the single configuration (within a single scene model, S_i) that best explains the observed faulty evidence. In contrast, we are trying to perform *cross-model comparison*, $P(S_i|E)$, comparing how well the evidence matches each model S_i in turn. To do this, we need to sum across *all possible* configurations of the scene nodes (5). We formalized our work in this fashion in previous work [1].

Another alternative is to use a factor graph [10]. A factor graph is a graphical model that encodes functions, in particular, probability distributions, over a set of variables using a bipartite graph. We assume the distribution, $h(X)$, can be written as a product of local factors: $h(X) = \prod_{a \in A} f_a(X_a)$, where A is the set of factors and each $X_a \subseteq X$. A key observation about factor graphs is that they also encode the calculations needed to compute the marginal functions of $h(X)$ for each variable. This leads directly to the derivation of the sum-product algorithm, a version of belief propagation that generalizes the forward algorithm for hidden Markov models and Pearl's algorithm [14].

We use the factor graphs shown in Fig. 2 to efficiently encode and calculate $\arg\max_i P(S_i|E)$ in (5). The observations $P(E|g, S_i)$ are encoded in the set of detector factors; there is one factor E_j for each region. We describe these in detail for real and simulated detectors in Section IV-C. The set of scene-to-region factors in the graph encode $P(g|S_i)$, the compatibilities between the scene type and the scene configurations. The prior factor encodes the prior distribution, $P(S)$. There is one variable node for the scene class S and n for the material identities M_j (one per region j).

We efficiently propagate evidence through the network using the standard *sum-product* message-passing rules given in [10] (details omitted here due to lack of space). After the message passing algorithm has completed, we find the scene class by taking the value with the highest marginal probability at the scene node. We now discuss our simple technique for learning the factors.

C. Learning the Model Parameters

For efficiency, we learn each factor by counting frequencies in the training set, rather than using a maximum likelihood technique on the factor graph.

Learning $P(E|g, S_i)$ is relatively easily. As is standard with probabilistic models used in low-level vision [8], we assume a locality condition: a detector's output on a region depends only on the material present in that region and not on other materials in the scene nor any spatial relations between the objects. We refer to "semantic regions" as those parts of the image where one or more material detectors have detection with nonzero beliefs. Furthermore, we assume that the detector's output is independent of the class of the scene (again, given the material present in that region). This allows us to factor the distribution as

$$P(E|g, S_i) = P(E|g) = \prod_{j=1}^{n} P(E_j|m_j) \quad (6)$$

in which each factor on the equation's right-hand side describes a single detector's characteristics. These characteristics can be learned by counting detection frequencies on a training set of regions or fixed using domain knowledge. This distribution is used in the belief calculations given in Section IV-C.

$P(S)$ models the prior distribution of scene types across the image population. We currently do not take advantage of prior information and simply use a flat prior, but priors could be learned in the future.

Learning $P(g|S_i)$, the set of scene to region factors, is more difficult. At this coarse level of segmentation, even distant (with respect to the underlying image) nodes may be strongly correlated, e.g., sky and pavement in urban scenes. Thus, we must assume that the underlying graph of regions is *fully connected*, prohibiting us from factorizing the distribution $P(g|S_i)$, as is typically done in low-level vision problems. Fortunately, for scene classification, and particularly for landscape images, the number of critical material regions of interest, n, is generally small ($n < 7$ in the ground truth labeled by humans): over-segmentation is rare because the material detectors can be imbued with the ability to merge regions. Thus a brute-force approach to maximizing (5) can be tractable.

One difficulty with learning and inference in this approach is that each feature and relation is discrete. Disparate materials such as *grass*, *sand*, and *foliage* cannot be parameterized on a continuum. Even while *rocks* and *pavement* might be considered similar, their frequencies of occurrence in various scene types are dramatically different: rocks occur primarily on mountains and beaches, while pavement occurs primarily in urban and suburban scenes. Relations such as *above* and *enclosing* are discrete as well. Therefore, we learn the set of scene-configuration factors by counting instances from the training set and populating matrices. The number and type of these factors can vary, depending on our independence assumptions, as we now discuss.

D. Factor Graph Variations for Between-Region Dependence

We experiment with a number of methods to enforce the compatibility $P(g|S_i)$ between the scene and set of regions, as given in the set of *scene-to-region* factors. The exact topology is *dynamic*, depending on the number of regions in the image. Furthermore, for a given number of regions, we can change the network topology to enforce or relax independence assumptions in the model and observe the effects of these assumptions. We present four generative models of *between-region dependence, given the scene type*.

1) *Exact*: generative model in which the full scene (material and spatial) configuration is taken as a single, dependent unit.
2) *Spatial Pairs*: same as *Exact*, but an approximation using pairwise spatial relationships.

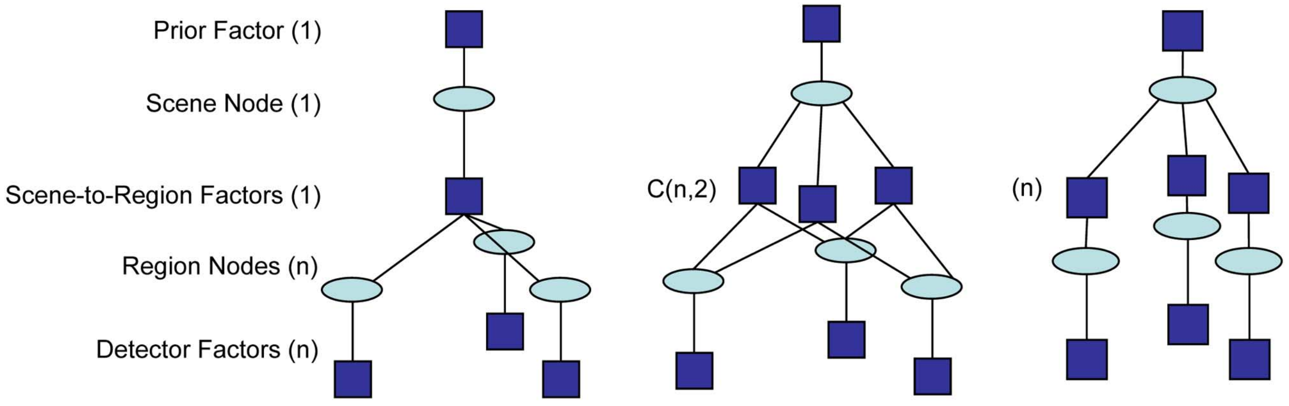
The next two are used are baselines for comparison.

1) *Material Pairs*: dependent only on the pairwise co-occurrence of materials.
2) *Independent*: each region is independent.

We now discuss the scene-to-region factors and the factor graph topology for each option.

1) Exact: Recall that we cannot factor the distribution $P(g|S_i)$ into individual components because of the strong dependency between regions. We model it with a fully connected structure, i.e., each pair of region nodes is adjacent. If we want an *exact* MAP solution to the distribution given in (5), we must use the factor graph shown in Fig. 2(a).

Here, the single scene-to-region factor encodes the conditional probability $P(g|S_i)$, the distribution of all region identities for a given scene. We model this distribution using a sparse matrix representation. We first populate a matrix of counts $N(S_i, g)\,\forall\, i$ by counting instances of each configuration g in the training set. Consider only images with n regions in the same spatial arrangement. Then each single configuration can be represented as a point in an $(n+1)$-dimensional space, in which the first n dimensions each encode the identity of a single region and the last dimension encodes the image class. For example, when $n = 3$, each scene can be represented by the point (m_1, m_2, m_3, S) in the four-dimensional (4-D)



(a) (b) (c)

Fig. 2. Factor graphs modeling various independence assumptions for $n = 3$ regions. The difference between the models is the number and type of Scene-to-Region factors, described in the text. (a) *Exact* model. Due to its tree structure, we can perform exact inference on it. However, the complexity of the model is hidden in the spatial configuration factor; learning it is problematic. (b) *Spatial pairs* model. While it is not exact due to the loops, each spatial factor's parameters are easier to learn than the joint one proposed in the exact model. Furthermore, its dynamic structure allows it to work on any image. (c) *Independent* model. This is equivalent to a tree-structured Bayesian network.

matrix; a typical beach scene would be represented by (sky, water, sand, Beach). We then normalize the matrix such that $\forall\, i, \sum_{g\in G} N(S_i, g) = 1$, to obtain the conditional probability $P(g|S_i)$ to use in the factor graph. This matrix has $|M|^n \cdot |S|$ elements.

The main benefits of this model are both due to its loopless topology: it can give an exact solution and it provides for efficient inference. However, it suffers from drawbacks. The distribution $P(g|S_i)$ is sparsely populated: the number of training images (call it $|T|$) is typically much less than the number of entries $|M|^n \cdot |S|$. (Consider that $|T| = 1000$ is considered large in the literature, and that for $|M| = 10$ and $|S| = 6$, a factor for a matrix with five regions has 600 000 entries.) The sparseness is exacerbated by correlation between objects and scenes, causing some entries to receive many counts and most entries to receive none: for example, open-water scenes always have water and sky regions, but never pavement.

Recall that each feature and relation is discrete, so we cannot interpolate between training examples to smooth the distribution (as can be done with parameterized distributions such as mixtures of Gaussians). In previous work [1], we proposed a graph-based smoothing technique to deal with this sparse distribution problem.

2) Spatial Pairs: Even after smoothing, using the full joint distribution has a major limitation keeping it from being used in practice: the distribution of regions is also a function of the spatial arrangement of the regions in the image, which includes the number of regions in the image. For example, the distribution of regions in the arrangement: "X *beside* Y, Y *beside* Z and X *beside* Z " (three horizontally arranged regions) is distinct from those in the arrangement "X *above* Y, Y *above* Z, and X *above* Z " (three vertically stacked regions), which is distinct from those in the arrangement of four vertically stacked regions W, X, Y, and Z.

Thus, a new matrix must be learned for *each* spatial arrangement, the number of which equals the number of edge labelings in the spatial configuration graph, $|R|^{C(n,2)}$ for $|R| = 7$ spatial relations and n regions, as we showed in the beginning of Section II. Granted, some are inconsistent and many are unlikely to appear in practice. However, for a training set of 1400 images and only the relations, *above*, *below*, and *beside*, we counted 133 distinct spatial arrangements of objects, many of which only occur once in the training set. Even though we can enumerate all possible arrangements, we do not have nearly enough data even to think about training all the models.

However, approximating the joint distribution as the product of pairwise distributions, as shown in Fig. 2(b), means we only have to learn *pairwise* relationships. Each conditional probability $P(m_i \text{ rel } m_j|S)$ for materials m_1 and m_2 and spatial relation *rel* is only a three-dimensional (3-D) matrix with only elements. For example, m_j above m_k in S_i is represented by the point (m_j, m_k, S_i) in the matrix for *above*. Furthermore, there are only $|R| = 7$ of them, and because each training example contains multiple pairwise relations, the pairwise distributions will be much more densely populated: pairwise relations such as "sky above grass" occur frequently, compared to full configurations such as "sky above foliage above pavement enclosed by grass".

This model does contain loops, and so encounters the typical problems with loopy belief propagation: it can be slow and may not converge. However, we have found that n tends to be small in practice, thus speed is not an issue, and empirically it converges most of the time.

Furthermore, we can create the graph structure on the fly once the number of regions is known, and once each pairwise relationship is modeled (X *above* Y, X *beside* Y, X *enclosed by* Y and their inverse relationships), any spatial arrangement containing those relations can be modeled. Thus, for the set of 1400 images described above, we would need to learn only seven factors and use one dynamically created network using combinations of these densely populated factors rather than needing 133 networks with sparsely populated factors.

3) Material Pairs: One may argue that the spatial relationships do not provide information about relative position—perhaps they just encode which materials tend to co-occur. Our model, *Material Pairs*, is designed to test this hypothesis. We use the same network topology as shown in Fig. 2(b), but replace the spatial relation factors with co-occurrence factors. We obtain these co-occurrence factors by counting the number of times a pair of materials j and k occur together regardless of their spatial relationship in the training set for a scene S_i. Normalizing the matrix gives $P(m_j \text{ co-occur } m_k|S_i)$ for each scene S_i. Note that, as before, the materials are encoded as indices in the matrix.

4) Independent: In our final model, *Independent*, we consider each of the n regions independently of each other region and any spatial relationships given the scene, $P(g|S_i) \approx \prod_{j=1}^{n} P(m_j|S_i)$. This yields the factor graph shown in Fig. 2(c). Each scene-to-region factor is just a normalized two-dimensional (2-D) matrix of counts of the number of times each material occurs in each scene type.

E. Computing the Spatial Relationships

Singhal, *et al.* found that $|R| = 7$ distinct spatial relations were sufficient to model the relationships between materials in outdoor scenes; we use the same relations (*above*, *far_above*, *below*, *far_below*, *beside*, *enclosed*, and *enclosing*). We also use the same hybrid technique to compute these relations: distant regions are compared efficiently by using their bounding boxes, while ones with overlapping bounding boxes are compared using a lookup table based on the directional weights of the pixels in the two regions [15].

For simplicity, we compute the spatial relations between every pair of regions in the image, adjacent or not. The current model also ignores the shape and size of the regions and occlusions that cause regions to be split. While any of these may be useful features in a full-scale system, we ignore them in this work.

III. Discriminative Approach

An alternative to the generative model-based approach is a discriminative approach, in which a classifier is trained to learn the distribution $P(S|E)$ directly, using spatial relations to help overcome detector errors. One direct way to encode the location of the semantic regions (objects) and the spatial relations between them is to use a grid-based approach. We divide the image into 49 blocks using a 7 × 7 grid; the grid size was chosen empirically based on experience with solving a similar problem; in that case the results were robust with respect to the size of the grid, similar to the finding in [22]. We compute the belief in each material for each block by weighting each region's beliefs output by the combined detectors ($\lambda(B)$ in (7) by the percentage of the area the region occupies in a block. For $|M| = 10$ materials, this always yields a 490-dimensional feature vector, regardless of the number of regions in the image. We then train a set of support vector machines [21] for doing multiclass classification. Support vector machines (SVMs) are binary classifiers that operate by maximizing the margin between the examples in the two classes by projecting the data into a higher dimension via a kernel function. To obtain multiclass classification, we trained a SVM for each class to distinguish it from all others, and classified the image with the class whose SVM gave the maximum output. Further details are given in [3].

IV. Semantic Detectors

While most past approaches to scene classification used low-level features, semantic features, such as the output from object and material detectors, provide strong evidence for some scene types when the features are available. We define semantic, or *high-level*, features to be labeled image regions. For outdoor scenes, the ten labels of highest interest include *sky*, *cloud*, *grass*, *foliage*, *sand*, *snow*, *water*, *rocks*, *pavement*, and *buildings*. A region with ambiguous identity usually has a low belief value, and may also have multiple labels. In this study, we use high-level features generated from three types of detectors: 1) output from *actual* object and material detectors; 2) output from *simulated* detectors; and 3) output from *best-case* detectors (hand-labeled regions).

A. Best-Case (Hand-Labeled) Detectors

Images in which the semantically critical regions have been hand-labeled form an integral part of this work. First, we use them for training. Specifically, we learn from them the distribution of which objects and materials appear in which scene and in what configurations. Second, we use them to test the performance of best-case material detectors. It suffices to assume that no actual detector can outperform a human on labeling typical materials in natural photographs, so we can use hand-labeled regions to determine an upper bound on performance for the classifiers we have designed. Third, we can perturb the region labels assigned by a human to simulate faulty detectors with various error characteristics (Section IV-D).

To label materials defined primarily by homogenous color and textures (like grass or sand), we start by automatically segmenting the image using a general segmentation algorithm (*mean-shift* [6]). Next, we manually label the semantically critical regions with their identities. The labels correspond to those ten high-interest materials for which we have detectors. Other regions are unmodeled and thus left unlabeled.

We use a different approach to label manmade structures such as buildings, houses, and boats. Because these regions tend to be greatly over-segmented by the segmentation algorithm, we instead use a block-based detector (similar to Bradshaw, *et al.*'s system [4] and described below) and modify the region map output by the utility described above as needed. A typical image, with the various labeling steps, is shown in Fig. 3.

B. Actual Detectors

Each of our actual detectors is based on color and texture features, similar to the common approach in the literature [13], [16], [20]. The following describes a typical material detector [15].

First, color (Luv) and texture (six high-frequency coefficients from a 2-level biorthogonal 3–5 wavelet transform of the luminance L band) features are computed for each pixel on the input image, and averaged locally. The features are fed to trained neural networks, which produce a probability or belief value in that material for each pixel in the image according to the color and texture characteristics [Fig. 4(b)]. The collection of pixel belief values forms a pixel belief map. After pixel classification,

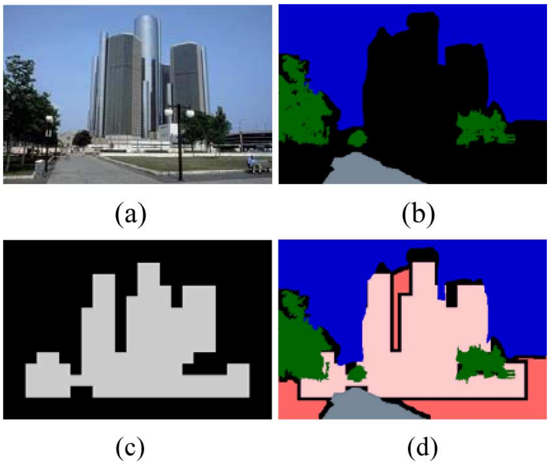


Fig. 3. Process of hand-labeling images. (a) Street scene. (b) Output from the segmentation-based labeling tool. (c) Output from a manmade object detector. (d) Combined output, used for learning spatial relation statistics.

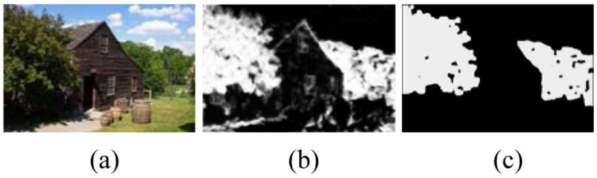


Fig. 4. Process of material detection, shown for the foliage detector. (a) Original image. (b) Pixel-level belief map. (c) Output of the detector. In (b) and (c), brightness corresponds to belief values.

spatially contiguous regions are obtained from the raw pixel belief map after thresholding the belief values. Next, each spatially contiguous region is post-processed according to unique region-based characteristics of the material type [output shown in Fig. 4(c)]. In blue sky detection, for example, the color gradient is calculated and is used to reject false positive sky regions. Because true blue sky becomes less saturated in color as it approaches the horizon, the detector can reject blue colored materials such as walls or clothing [11]. Finally, the belief value of each region is the average belief value of all pixels in the region.

Our manmade structure detector classifies blocks of pixels using color and wavelet texture features (similar to [4]): buildings tend to be grayish in color and have regular textures with strong vertical and horizontal components (e.g., windows, siding on houses) giving the belief that the block contains manmade content. Similar to the material detectors, the belief value for any group of manmade blocks is the average belief of the blocks in the group.

Note that these detectors are individually designed using bottom-up strategies with no spatial model initially imposed, and for general outdoor scenes. While some detectors have very good accuracies (e.g., sky, 95%), other material detectors have substantially lower performance (e.g., water and rock below 70%), primarily due to high false-positive detection rates.

All of the detectors are run independently. After this, the region maps are aggregated, inducing a segmentation map upon the image [Fig. 5]. Some regions are unambiguously detected as a single material. Commonly, however, some regions are detected as multiple materials (e.g., the snow detector and the cloudy sky detector often both fire on cloudy sky). In this case, we label that region with multiple labels, calculating beliefs of each material according to the process described in Section IV-C. For example, the region in the middle of the bottom half has a belief of 0.29 being manmade and 0.36 being unmodeled (the beliefs need not sum to 1.0). If the amount of overlap between any two regions is small (less than the size of the smallest region we care about, e.g., 0.1% of the image), we discard the overlapping region; otherwise, we create a new region with aggregated beliefs.

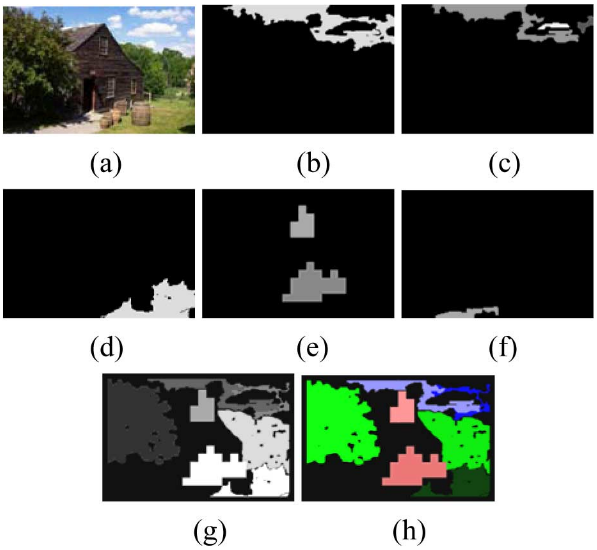


Fig. 5. Aggregating results of individual material detectors for an image (a) Original image. (b) Blue-sky. (c) Cloudy sky. (d) Grass. (e) Manmade. (f) Sand. The foliage detection result from Fig. 4 is also used. Other detectors gave no response. (g) The aggregate image with seven detected regions. (h) Pseudo-colored aggregate segmentation map.

This technique for material detection is a bottom-up strategy because no spatial model is initially imposed, and the detectors work for general outdoor scenes. While some of these individual material detectors have very good accuracies (e.g., sky, 95%), other material detectors have substantially lower performance (e.g., water and rock below 70%), primarily due to high false-positive detection rates.

C. *Combining Evidence for a Region From Multiple Detectors*

Each region is processed independently in the following manner. Let D be the set of material detectors, $D = \{D_1, D_2, \ldots D_{|M|}\}$. Each detector D_i outputs a belief value $\lambda_{Di}(B)$ in $[0, 1]$ for region B, where $\lambda_{Di}(B) = 0$ means the detector did not fire and $\lambda_{Di}(B) = 1$ means it fired with full confidence. The initial evidence for B is the vector $\lambda_D(B)$ of those belief values.

When combining the evidence, we need to incorporate two other factors. First, the detectors are faulty. For example, if the water detector fires, we want to allow for the possibility that it was a false detection on a sky region. Second, some detectors are more reliable than others. Materials with unambiguous characteristics and relatively little variability, like blue sky, are much more reliably detected than those with heterogeneous appearances and widely varying textures and colors, like rocks.

To incorporate this information in a principled way, we use a two-level Bayesian network (Fig. 6). Define the *characteristics*

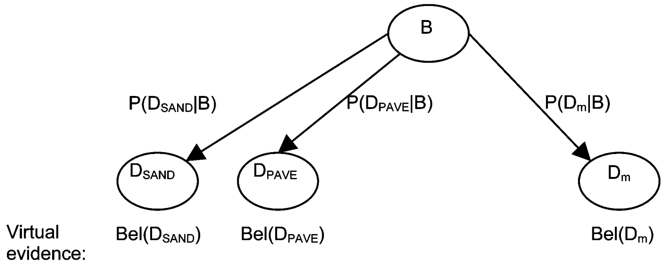


Fig. 6. Bayesian network showing relationship between regions and detectors.

TABLE I
CHARACTERISTICS OF SAND DETECTOR

$P(D_{\text{sand}}\|M)$	True material (M)
0.10	Background (unmodeled)
0.01	Bluesky
0.05	Cloudysky
0.01	Foliage
0.05	Grass
0.10	Manmade
0.05	Pavement
0.05	Rock
0.90	Sand
0.05	Snow
0.05	Water

of detector D on a set of materials M to be the set of conditional probabilities $\{P(D|m_i) : m_i \in M\}$. They include both *true positive rates* (recall) and *false positive rates*. Take the sand detector (Table I) as an example. The first column gives the probability that the sand detector fires, given the true material. In this example, the sand detector has a 90% recall of true sand, and detects sand falsely on 10% of manmade structures (made of concrete). Likewise, its false positive rate on water is 5% (because some water contains brown reflections or covers shallow sand). It also fires falsely on 10% of the unmodeled regions in the images because they have similar colors. Such characteristics can be learned by counting performance of detectors on a validation set or estimated when there is not enough data.

We have a detector for each material of interest. One node per detector is linked to the region, B, as shown in Fig. 6. Input to the Bayesian network consists of virtual evidence ($\lambda_D(B)$ described above) at the leaf nodes. Note that this graph's root node corresponds to a specific *region* in the image.

The beliefs generated by the individual material detectors are fed into the leaf nodes of the network and propagated to the root node, which outputs a composite material belief for that region. We follow Pearl's treatment and notation ([14, Sect. 2.2.2]), using λ for beliefs.

The combined belief of each material being the true material, given the set of detectors firing is given by

$$\begin{aligned}\lambda(B) &= \alpha \prod_{j=1}^{|M|} \lambda_{D_j}(B)\\ &= \alpha \prod_{j=1}^{|M|} A_{Dj|m} \lambda(D_j)\end{aligned} \tag{7}$$

where α is a normalizing constant and A is a probability matrix as defined by Pearl [14]: $A_{y|x} \equiv P(y|x)$. More specifically, the (i,j)th position in $A_{y|x} \equiv P(y_j|x_i)$.

These beliefs, $\lambda(B)$, are passed on to the remainder of the network, i.e., by attaching the subgraph to each material leaf in the factor graphs shown in Section II-D.

D. Simulating Faulty Detectors for a Region

While we have actual detectors, we are also interested in determining the usefulness of the scene models on a wider range of detector performance. Using the hand-labeled regions, we can simulate detector responses for each region. We start with a hand-labeled image, which is already segmented and labeled with the semantically critical regions. We then apply the following *material perturbation algorithm* for each region labeled with material M. We assume that the detector responses for each region are independent.

1) Determine which detectors fire by sampling the detector characteristics (the Bayesian network in Fig. 6); i.e., for each detector i, we generate a random number $x \in [0, 1]$. The detector fires if and only if $x < P(D_i|M)$.
2) For each detector that fires, sample the belief distribution to find the confidence in the detection. We assume that they are distributed with means μ_{TP} for true detections and μ_{FP} for false detections; using different parameters for each allows us to simulate detectors of various accuracy.
3) Propagate the beliefs in the Bayesian network to determine the overall belief of each material, as we did for actual detectors in Section IV-C.

In this process, the segmentation of the regions does not change, so the segmentation map generated by the simulator is identical to the corresponding hand-labeled map, except each region has different detected identities. As a corollary, the spatial relationships of the hand-labeled map and the simulated map are identical, i.e., the *identity* of each region was perturbed, not its *location*. Such simulation is somewhat limited as it avoids the over-segmentation often produced by actual detectors.

We change two parameters in this algorithm to simulate detectors with a wide range of accuracy. We consider 11 operating points, ranging from 0 (best) to 1 (worst) inclusive, in increments of 0.1. First, we allow each false detection rate $P(D|M) < 0.5$ to vary between 0 (best) and $2P(D|M)$ (worst) and allow each true detection rate $P(D|M) > 0.5$ to vary between 1 (best) and $1 - 2(1 - P(D|M))$ (worst). Second, when sampling the belief distributions, we fix $\mu_{\text{TP}} = 1$ (high beliefs in true positive regions are typical for our actual detectors) and vary μ_{FP}, allowing it to range from 1 (worst: when false positives are detected, it is usually with high belief, indistinguishable from that of true detections) to 0 (best: when the detector fires falsely, it tends to do so with low belief). This assumes that $\mu_{\text{FP}} = 0.5$ is realistic for the actual detectors. We anticipate that scene classification accuracy using "better" detectors (via simulation) will outperform the actual ones.

We conclude this section by elaborating on the fact that the Bayesian network used to combine detector results into a single set of likelihoods can be combined with the factor graph because factor graphs subsume Bayesian networks [10]. We separated

TABLE II
IMAGE SETS

Name	Size	Classes and comments
D1	937	Regions are hand-labeled; we also ran actual detectors on D1.
D2	165	Subset of D1 having three vertically arranged regions.
D3	1153	Drawn from same sources, but non-overlapping with D1. Used actual detectors.

TABLE III
SCENE CLASS DESCRIPTIONS

Class	Description
Beach	At least 5% each of water, sand, and sky
Field	No aerial view, not cluttered with trees ("open")
Mountain	Open, whole mountains, mid-range view
Street	Urban, plus must have pavement in the image
Suburban	Includes pictures of houses and yards and recreation areas (parks)
Open-water	Picture taken primarily of water from a boat (vs. those taken from land, which are beaches or coastal scenes).

TABLE IV
ACCURACY (%) COMPARISON BETWEEN TECHNIQUES

Model	Generative			Discriminative	
Test set	*Spatial Pairs*	*Mat'l. Pairs*	*Ind.*	High-level features	Low-level features
D1a	94.5	91.7	92.2	93.1	78.0 (doesn't use detectors)
D1b	80-93	70-89	67-90	74-91	
D1c	51.0	47.2	45.6	75.6	
D2	46.0	41.5	40.0	54.2	67.4

them in our experiments only because we must combine the semantic detector results into likelihoods for input to the discriminative model as well. Furthermore, our method of simulating detectors could be re-used in other contexts. While the combined model would allow for the detection likelihoods to be updated, that is not the goal of this work.

V. EXPERIMENTAL RESULTS

A. Experimental Setup

The three image databases we used (Table II) were chosen such that they could be described in terms of the objects or materials expected to be present in them. For example, an image with sky, water, and sand is a beach. Data set D2 included all of the images from D1 with a specific spatial arrangement: we ignored the *far* modifier and the enclosure relations (which occur rarely) and focused further on the single spatial arrangement occurring most often in training: the 165 images with a vertically stacked structure, X above Y, Y *above* Z, and X *above* Z.

Table III gives descriptions of various scene classes. In general, the images chosen did not have a dominant main subject. This is legitimate; for example, a close-up image containing a horse in a field would be considered a "horse" image, not a "field" image. Future work includes investigating how to adapt the framework for such images. The images are taken from a mix of personal images, Corel photos, and images from the Internet. Each set has approximately the same number of images from each of the six classes.

Data set D1 was automatically segmented and hand-labeled, as described in Section IV-A. To simulate actual detectors, which are faulty, we randomly perturbed the ground truth to create simulated detector responses, as described in Section IV-D.

Pairwise spatial relations in our model are encoded as probability density functions of the two regions and the scene for each of the seven spatial relations; these are learned from D1. The material detectors were trained on image sets not used in this study.

B. Spatial Pairs Model

To show the benefit of spatial relations, we compare them against the baseline models mentioned earlier: *Material Pairs* (using co-occurrence relations) and *Independent* regions. We then compare them against the discriminative approach. Finally, we compare the high-level features against a discriminative approach using low-level features (spatial color moment features [19] extracted using the same 7×7 grid and classified using a support vector machine [21]). Table IV shows classification accuracies for a number of scenarios. In each case, we learned from image set D1. We tested either on D1 using cross-validation (CV) to avoid biasing our results, or on D3. Fig. 7 is an expansion of row 2, comparing performance across the range of simulated detector accuracy.

In our experiments, classification using the *Spatial Pairs* model always outperformed those using the other generative models, *Material Pairs* and *Independent*, showing that spatial information does help distinguish scenes. The marginal difference between the two increased as the accuracy of the detectors decreased. This is not surprising, given that spatial relations are expected to mitigate the effects of faulty detectors, more so when they are more faulty.

Fig. 8 contains example images (and their corresponding segmentations) for which the *Spatial Pairs* model yielded an improvement over one of the baseline generative models. The first example shows a beach scene with a manmade structure (building) on the shore. In this case, because open-water scenes are more likely than beaches to contain manmade objects (boats), the *Material Pairs* model, using co-occurrence of materials only, classified it incorrectly as an open-water scene. However, the fact that the sand was beside the water and that the manmade structure was above the sand allowed the *Spatial Pairs* model to classify it correctly as a beach. In the second example (mountain), the sand and rock detectors both fired on the rock region. The presence of both water and sand caused the *Independent* model to misclassify it as a beach scene. However, the spatial model used the fact that the region was above both

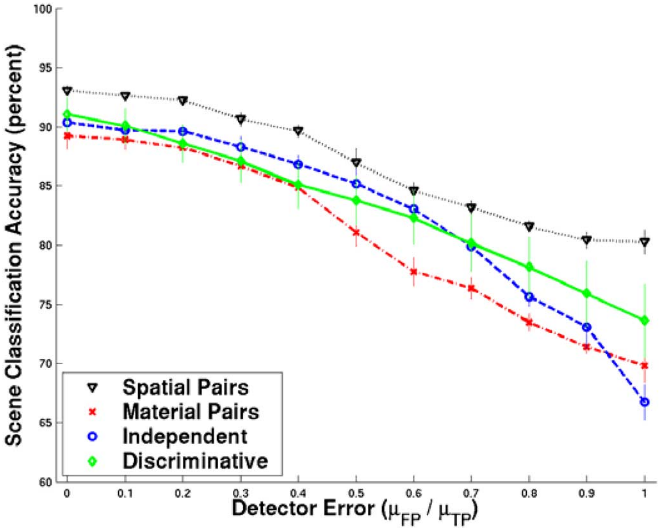


Fig. 7. Classification accuracy of the methods as a function of simulated detector error rate. We repeated each simulation ten times and report the mean accuracy. Error bars are for the standard error between test runs.

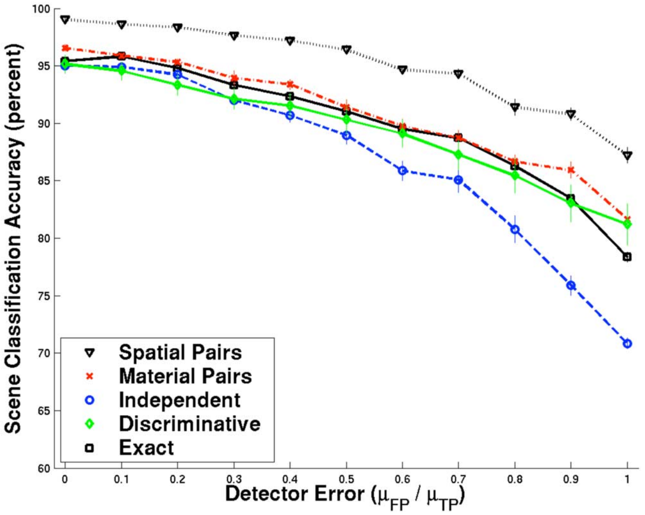


Fig. 9. Comparison between accuracy obtained using the *Spatial Pairs* model, the *Exact* model, and the discriminative model using high-level features for the range of simulated detector error rates.

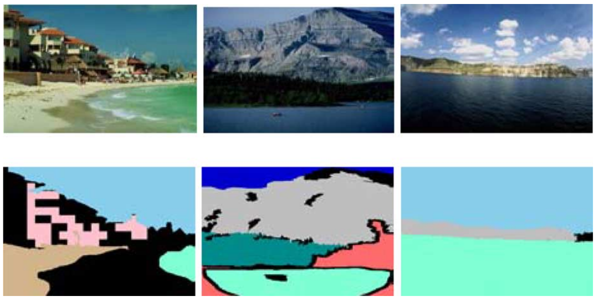


Fig. 8. Examples of images and segmentations for which the *Spatial Pairs* model gave correct results while the baseline model failed. See text for details.

foliage and water to rule out sand as an option (as sand is very unlikely to occur in the background beyond foliage). Finally, in the last image, the rock region was misdetected as sand. Without spatial information, sand, water, and sky appear most often in beach images, causing both the *MaterialPairs* and *Independent* models to classify this open-water image incorrectly. However, typical beach images have sand in the foreground (or next to the water if the photographer looks up the beach as in the first example), not the background, since the photographer usually stands on the beach; this spatial relationship caused *Spatial Pairs* to classify it correctly.

We also found that spatial relations outperformed the discriminative models when the simulated detectors were good enough or in the best-case (hand-labeled). Of course, if the detectors are inaccurate enough, then no amount of spatial information will help due to the abundance of noise. When using actual detectors, the discriminative approach outperformed the generative approach using the same feature set. This could be due to the following reasons. First, the discriminative models, by their nature, are trained to classify, whereas generative models are more general; they can perform classification, but only when the joint distribution between the class and the features is modeled. Second, while the generative model and the discriminative model both encode spatial information (explicitly and implicitly, respectively), only the discriminative model encodes the relative *size* of regions through the number of blocks occupied by the region and to a lesser degree the *shape* of the region. Therefore, if buildings are expected to occupy a large portion of a street scene, but only a smaller portion of a suburban scene, the discriminative model can differentiate the two, whereas the generative model, which has no "region area" evidence factor, cannot. While we have focused only on using spatial relations between regions to help performance, an interesting direction to explore in the future is to learning how to encode area or shape factors and determining whether or not using them would help classification.

A comparison between the two discriminative methods is certainly also of interest but beyond the scope of this paper; it is covered in depth in another paper [3].

C. Comparison Between All Methods

As stated earlier, the *Exact* model is limited in that it can only be applied when the spatial arrangement of the regions is fixed. Therefore, when comparing this model with the others on a single data set, we use data set D2, which includes the 165 vertically stacked images in D1. Furthermore, to ensure the model is populated densely enough, we must ignore the *far* modifier and the enclosing relations, as we did in [1]. We also compare it with the other generative models (*Material Pairs* and *Independent*) and the two discriminative models. We use leave-one-out cross-validation on set D2 and obtain the results shown in Fig. 9.

Fig. 9 shows that the approximate model *outperforms* the exact model. This is counter-intuitive because when we approximate, we are losing information. However, upon close examination of the data set, we see two types of images for which the model's output differs.

The first type, those images classified correctly by the *Exact* model, but not by *Spatial Pairs*, include images with full configurations that are repeated, though rarely, in the training set, but consist of pairwise relations seen more often in other scenes.

For example, three field images have a pond in the background, yielding "sky over water over grass". The relation "sky over water" (seen predominantly in beach and open-water scenes) dominates the relation "sky over grass" seen in Field scenes (while the third relation "water over grass" occurs rarely in any class), so the approximate model classified it as beach. However, because the full configurations of each of those three images were not unique, the exact model classified each correctly as field.

The second type, those images classified correctly by the approximate model, but not by the exact model, include those "singleton" images containing a unique configuration of materials. This is due to the fact that the model is undertrained, i.e., the distributions of scene configurations are too sparsely populated. For example, one open-water image has a water region broken into two parts, yielding "sky over water over water". The exact model classified it incorrectly because that configuration occurred nowhere else in training, whereas the approximate model, influenced by the relation "sky over water" occurring twice, classified it correctly as open water. Because there are more images of the second type, the accuracy of the exact model is lower on the training set.

D. Comparison With the Discriminative Models

We note that on this small data set, that when high-level features are used, the *Spatial Pairs* model outperforms the discriminative and the baseline generative models. The reduction in the discriminative model's accuracy is because the discriminative model is undertrained given its dimensionality and the available training data. High-level features outperform the low-level features (75.2% accuracy, not shown on graph) except when the simulated detectors are highly inaccurate, which is expected, given our previous results on the larger data set (Table IV).

We chose the discriminative approach as a baseline to compare to our approach because it is a mainstream approach that is also based on training. There is no known disadvantage in the discriminative approach when compared to logic/rule-based [12] and other "soft" methods [7]. While these soft methods may achieve a slightly higher accuracy in some cases, we expect rigorous training should ultimately provide the most robust and generalizable systems.

VI. Discussion and Conclusions

We presented a scheme for parsing the scene content in order to determine the semantic class of a scene, utilizing semantic object detectors (e.g., sky, foliage, and pavement) and generative scene-configuration models. With comprehensive experiments, we demonstrated the promise of this approach and compared it with existing discriminative approaches and those using low-level features.

Discriminative models have been used effectively in pattern recognition. However, generative models offer a number of advantages at the expense of slightly lower accuracies, even in the context of classification:

1) The systems can be highly modular, exemplified by the systems we have designed. In each case, the method of extracting local cues is independent of the model, allowing for independently developed cue extractors to be used. Furthermore, the cue extractors can be improved in the future and readily plugged in without having to retrain the model.
2) Generative models usually offer much insight to the relationship between and influence of various factors involved in the problem. This is often not the case with discriminative models such as neural networks. As another example, the conditional probabilities used in Bayesian networks are intuitive, whereas the linear models used in conditional random fields are not.
3) Generative models operate as well as discriminative models when there is a shortage of labeled training data. In semantic scene classification, this is often the case.
4) Generative models can be "surprised", that is, when confronted with data unlike any seen in training, they emit a small output probability. Discriminative models only offer a forced-choice solution.
5) Generative models can operate in the face of missing cues. By contrast, in a discriminative model, examples with missing cues cannot be used. In the context of scene classification, it may be desirable to turn off expensive semantic features to gain speed. Generative models would easily allow that.
6) Our generative model is flexible enough work backwards, using knowledge of the scene type to improve region labeling. To perform this, we instantiate the scene node, propagate messages, and take the highest marginal probabilities at the region nodes [2]. The model can also classify regions and the scene as a whole simultaneously.

Because our database already contains a large number of personal images from various sources, we believe that our approach will scale well to larger databases containing similar types of these images. We also believe our approach could be used for images of different environments (e.g., indoor images), given the appropriate bank of detectors for materials and objects in those environments (e.g., furniture [18]). In this case, the modularity of the approach is very important: to handle a wide variety of images, a large number of detectors can be created independently (and in fact, many have been created already) and integrated within this framework.

One limitation of our approach is that it does not handle closeup images of people, which are a common subject for photographs; however, our approach complements classification using face and people recognition.

In terms of future directions, improving the accuracy of the semantic detectors will clearly increase the performance on scene classification and should be monitored. As mentioned earlier, another interesting direction for future work includes encoding area and shape in the graphical model.

Acknowledgment

The authors thank Dr. A. Singhal and Dr. W. Hao for their work in developing a number of the semantic detectors used in this study.

References

[1] M. Boutell, J. Luo, and C. M. Brown, "Learning spatial configuration models using modified Dirichlet priors," in *Proc. Workshop on Statistical Relational Learning (in Conjunction With ICML)*, 2004.

[2] ——, “Improved semantic region labeling based on scene context,” in *Proc. Int. Conf. Multimedia and Expo.*, Amsterdam, The Netherlands, Jul. 2005.

[3] M. Boutell, A. Choudhury, J. Luo, and C. M. Brown, “Using semantic features for scene classification: How good do they need to be?,” in *Proc. IEEE Int. Conf. Multimedia and Expo.*, Jul. 2006.

[4] B. Bradshaw, J. Platt, and B. Scholkopf, Kernel Methods for Extracting Local Image Semantics 2001, Microsoft Research Technical Report MSR-TR-2001-99.

[5] P. Chou, The Theory and Practice of Bayesian Image Labeling. Rochester, NY, Univ. Rochester, 1988, Ph.D. dissertation.

[6] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.

[7] J. Fan, H. Luo, and Y. Gao, “Learning the semantics of images by using unlabeled samples,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Jun. 2005.

[8] W. Freeman, E. Pasztor, and O. Carmichael, “Learning low-level vision,” *Int. J. Comput. Vis.*, vol. 40, no. 1, pp. 24–57, Oct. 2000.

[9] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.

[10] F. R. Kschischang, B. H. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[11] J. Luo and S. P. Etz, “A physical model-based approach to detecting sky in photographic images,” *IEEE Trans. Image Process.*, vol. 11, no. 3, pp. 201–212, Mar. 2002.

[12] P. Mulhem, W. K. Leow, and Y. K. Lee, “Fuzzy conceptual graphs for matching images of natural scenes,” in *Proc. Int. Joint Conf. Artificial Intelligence*, 2001, pp. 1397–1404.

[13] M. R. Naphade and T. S. Huang, “A probabilistic framework for semantic video indexing, filtering, and retrieval,” *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 141–151, Mar. 2001.

[14] J. Pearl, Ed., *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA: Morgan Kaufmann, 1988.

[15] A. Singhal, J. Luo, and W. Zhu, “Probabilistic spatial context models for scene content understanding,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Madison, WI, Jun. 2003.

[16] J. R. Smith and C.-S. Li, “Image classification and querying using composite region templates,” *Comput. Vis. Image Understand.*, vol. 75, no. 1/2, pp. 165–174, 1999.

[17] J. Smith, C. Lin, M. Naphade, A. Natsev, and B. Tseng, “Multimedia semantic indexing using model vectors,” in *Proc. IEEE Int. Conf. Multimedia & Expo.*, Baltimore, MD, 2003.

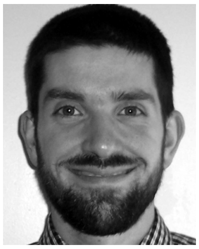
[18] A. Torralba, K. P. Murphy, and W. T. Freeman, “Sharing features: Efficient boosting procedures for multiclass object detection,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004, pp. 762–769.

[19] A. Vailaya, M. Figueiredo, A. Jain, and H.-J. Zhang, “Content-based hierarchical classification of vacation images,” in *Proc. IEEE Multimedia Systems ’99*, Florence, Italy, 1999.

[20] A. Vailaya and A. Jain, “Detecting sky and vegetation in outdoor images,” in *Proc. SPIE*, 2000, vol. 3972, pp. 411–420.

[21] N. Vapnik*, The Nature of Statistical Learning Theory*. New York: Springer, 1995.

[22] Y. Wang and H.-J. Zhang, “Detecting image orientation based on low-level visual content,” *Comput. Vis. Image Understand.*, vol. 93, no. 3, pp. 328–346, 2004.



Matthew Boutell (M’05) received the B.S. degree in mathematical science from Worcester Polytechnic Institute, Worcester, MA, in 1993, the M.Ed. degree from the University of Massachusetts at Amherst in 1994, and the Ph.D. degree in computer science from the University of Rochester, Rochester, NY, in 2005.

He served for several years as a mathematics and computer science Instructor at Norton High School and Stonehill College and as a Research Intern/Consultant at Eastman Kodak Company. Currently, he is an Assistant Professor of Computer Science and Software Engineering at Rose-Hulman Institute of Technology, Terre Haute, IN. His research interests include image understanding, machine learning, and probabilistic modeling.

Jiebo Luo (M’95–SM’99) received the B.S.E.E. degree from the University of Science and Technology of China in 1989 and the Ph.D. degree in electrical engineering from the University of Rochester, Rochester, NY, in 1995.

He is currently a Senior Principal Scientist at Kodak Research Laboratories, Rochester, and an Adjunct Professor at Rochester Institute of Technology. He has served as a co-advisor for many Ph.D. and M.S. students at various universities. He has authored over 100 technical papers and holds over 40 granted U.S. patents. His research interests include pattern recognition, computer vision, image processing, medical imaging, and multimedia communication.

Dr. Luo has actively participated in numerous international conferences and workshops as a Chair, an organizing committee member, or a technical program committee member. He was a Guest Coeditor for the *Journal of Wireless Communications and Mobile Computing* (Special Issue on Multimedia Over Mobile IP), the lead Guest Editor for the *Pattern Recognition* journal (Special Issue on Image Understanding for Digital Photos), and a member of the Kodak Research Scientific Council. Currently, he is on the editorial boards of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON MULTIMEDIA, *Pattern Recognition*, and the *Journal of Electronic Imaging*. He is a Kodak Distinguished Inventor.

Chris Brown received the B.A. degree from Oberlin College, Oberlin, OH, in 1967 and the Ph.D. degree from the University of Chicago, Chicago, IL, in 1972.

He is a Professor of Computer Science at the University of Rochester, Rochester, NY. He has published in many areas of computer vision and robotics. He co-authored *Computer Vision* with colleague D. Ballard (1992), and influential work on the “active vision” paradigm was reported in two special issues of the *International Journal of Computer Vision*. He edited the first two volumes of *Advances In Computer Vision*, with D. Terzopoulos, (Mahwah, NJ: Lawrence Erlbaum, 1995) and *Real-Time Computer Vision* (Cambridge, U.K.: Cambridge University Press). He is the co-editor of *VIDERE*, the first entirely on-line refereed computer vision journal (Cambridge, MA: MIT Press, 1996–2000). His most recent Ph.D. students have done research in infrared tracking and face recognition, features and strategies for image understanding, augmented reality, and three-dimensional reconstruction algorithms. He supervised the undergraduate team that twice won the AAAI Host Robot competition (and came third in the Robot Rescue competition in 2003).