Learning Hierarchical Space Tiling for Scene Modeling, Parsing and Attribute Tagging

Shuo Wang, Yizhou Wang, and Song-Chun Zhu

Abstract—A typical scene category contains an enormous number of distinct scene configurations that are composed of objects and regions of varying shapes in different layouts. In this paper, we first propose a representation named *Hierarchical Space Tiling (HST)* to quantize the huge and continuous scene configuration space. Then, we augment the HST with attributes (nouns and adjectives) to describe the semantics of the objects and regions inside a scene. We present a *weakly supervised* method for simultaneously learning the scene configurations and attributes from a collection of natural images associated with descriptive text. The precise locations of attributes are unknown in the input and are mapped to the HST nodes through learning. Starting with a full HST, we iteratively estimate the HST model under a learning-by-parsing framework. Given a test image, we compute the most probable parse tree with the associated attributes by dynamic programming. We quantitatively analyze the representative efficiency of HST, show the learned representation is less ambiguous and has semantically meaningful inner concepts. In applications, we apply our model to four tasks: scene classification, attribute recognition, attribute localization, and pixel-wise scene labeling, and show the performance improvements as well as higher efficiency.

Index Terms—Scene Representation, Hierarchical Space Tiling, Scene Attributes

1 Introduction

1.1 Motivations

A Typical natural scene category, *e.g.,* street and beach, contains an enormous number of distinct scene configurations depending on different viewpoints, resolutions and shape variations of the objects (*e.g.,* buildings, cars) and regions (*e.g.,* sky, water). A well-known representation that can explicitly address such representational complexity effectively is the family of hierarchical compositional models, which are reconfigurable and can generate a combinatorial number of configurations through a small dictionary of shape elements. In the past few years, learning the structures of such models has become a hot topic in two communities: learning stochastic image grammar [1] and deep learning [2]. However, this structure learning remains a challenge in computer vision due to two main difficulties.

(i) The searching space of the hierarchical compositional model is huge or essentially continuous.

(ii) The representations are often ambiguous, *e.g.,* a configuration may have more than one way of parsing. Hence, the learned model partially loses its power in parsing as it diffuses the probability over multiple possible interpretations.

- *S. Wang and Y. Wang are affiliated with Nat'l Eng. Lab. for Video Technology, Cooperative Medianet Innovation Center, Key Lab. of Machine Perception (MoE), Department of EECS, Peking University, Beijing, China, 100871.*
  E-mail: {shuowang, Yizhou.Wang}@pku.edu.cn
- *S.C. Zhu is affiliated with the Departments of Statistics and Computer Science, University of California, Los Angeles, USA, 90095.*
  E-mail: sczhu@stat.ucla.edu

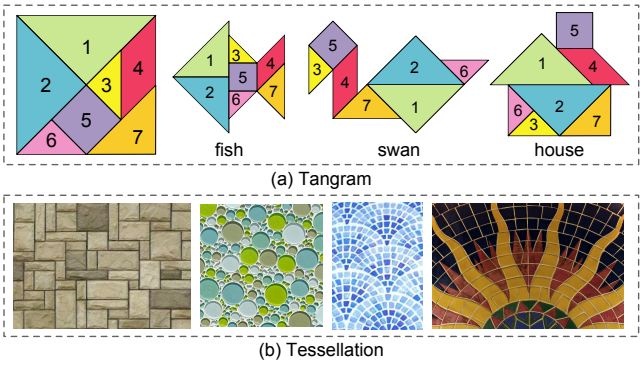


Fig. 1: Tiling examples. (a) A tangram consists of seven pieces (left) and can generate complex shapes. (b) Tessellation for decorating floor and wall.

In this paper, we propose a scene representation namely *Hierarchical Space Tiling (HST)*. The underlying intuition of HST is that the complex shapes can be composed of smaller and simpler shape elements. Fig.1(a) shows a Chinese tiling puzzle called "tangram", which consists of seven flat shape elements. By assembling all the seven pieces without overlapping, the tangram can generate thousands of specific shapes, *e.g.,* fish, swan and house. People manufacture tiles of a few shapes (triangles, squares, rectangles) and in a few sizes (2 $\times$ 2 inches to 20 $\times$ 20 inches), and compose any patterns according to customer needs under an economic budget. Fig.1(b) shows the examples of tessellated pavement and tiled wall. HST is aimed to quantize the huge and continuous scene configuration space so as to transfer the structure learning problem to a manageable solution space.

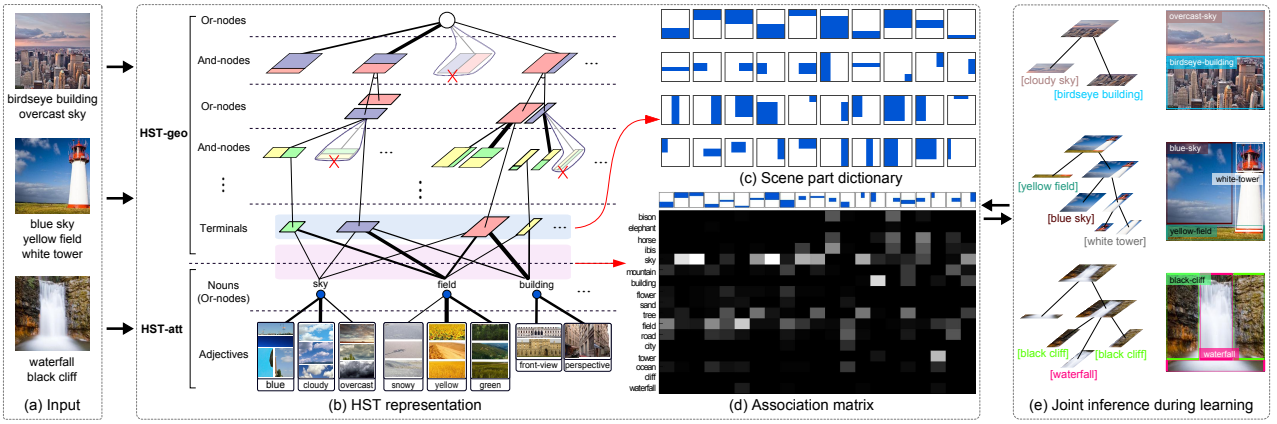


Fig. 2: Flowchart of HST learning. (a) The input images and text descriptions. (b) The HST representation consists of two components: the HST-geo and HST-att, in an And-Or hierarchy. The thickness of the edges under each Or-node indicates the value of branching probabilities. The branches are pruned (see the red crosses) if their probabilities are near zero. (c) Scene part dictionary formed by the terminal nodes (blue panel in (b)). (d) Association matrix measures the assignment probabilities between scene parts and noun attributes (pink panel in (b)). (e) The inferred parse trees (scene configurations) augmented with attributes.

Recently, attributes become popular because they provide rich mid-level semantics and are shared across categories. However, most previous work treated attributes as flat classification features. For example, Patterson *et al.* [3] identify 102 scene attributes and train 102 classifiers to recognize if an image has certain attributes or not. In this paper, we associate the attributes to the hierarchical scene model to harness the synergy between the semantics and hierarchy.

1.2 Overview

Fig.2 shows the flowchart of the weakly supervised method for learning the HST. The model contains two components: HST-geo and HST-att, modeling the 2D configurations and scene attributes respectively.

(i) HST-geo. As shown in the top part of Fig.2(b), the HST-geo quantizes the huge space of scene configurations in an And-Or Tree (AoT) structure [4]. The And-nodes correspond to the decomposition rules, *e.g.*, a coast scene is decomposed as sky on the top and ocean underneath. The Or-nodes correspond to the alternative sub-structures, *e.g.*, buildings appear on both sides of an image when a camera faces along the street, or only on one side in other views. The terminal nodes are some shape elements, *e.g.*, squares and rectangles, corresponding to the scene parts. They are of different sizes, locations and shapes, and compose a scene part dictionary (Fig.2(c)). With the And-Or structure, the HST-geo can generate a full space of possible parsing.

(ii) HST-att. Scene attributes are given by text descriptions (Fig.2(a)), consisting of nouns (*e.g.*, sky, mountain) and adjectives (*e.g.*, cloudy, rocky). They define the objects and regions inside a scene and their appearance. As shown in the bottom part of Fig.2(b), attributes are represented as a two-level AoT, where each noun attribute is an appearance-Or node having a mixture of adjectives, *e.g.*, sky can be blue, cloudy or overcast. Furthermore, each terminal node in the HST-geo links to the noun attributes according to an association matrix (Fig.2(d)) which measures the co-occurrence between local regions and objects, *e.g.*, road always appears at image bottom. Therefore, the scene configurations and attributes are integrated under a unified framework.

(iii) Learning and inference. The And-Or structure defines a set of grammar rules, and the HST embodies a probabilistic context free grammar (PCFG). In learning, we first learn the HST-geo and HST-att separately as an initial HST model. Then, we jointly learn the two sub-models through an learning-by-parsing manner. In this way, the challenging *structure learning* problem is transferred into a tractable *parameter learning* problem. Finally, given a test image, an optimal parse tree (scene configuration) augmented with attributes can be inferred by dynamic programming (Fig.2(e)).

(iv) Evaluation. We quantitatively show the proposed HST is clearly more effective than other popular representations, such as spatial pyramid [5] and Quadtree [6], by analyzing the rate-distortion curve as in coding theory. We also show the learned representation is less ambiguous in parsing and has semantically meaningful inner concepts. We demonstrate the practical value of HST through four applications: scene classification, attribute recognition, attribute localization and pixel-wise scene labeling. We show the better performance and higher efficiency of our model over the previous methods.

1.3 Related work

Scene representations We summarize the existing scene representations into five typical classes. (i) *Bag-*

of-Words (BoW) representations [7, 8] treat a scene as a collection of visual words and ignore the spatial layout information. (ii) *Grid structure representations*, such as gist-based representation [9, 10], local semantic model [11], spatial pyramid matching (SPM) [5] and a "reconfigurable" model [12], implicitly adopt squares as elements in different sizes and locations, and divide the image into grids. (iii) *Region based representations* [13, 14] segment an image into semantic regions, then model the contextual relations between adjacent regions. (iv) *Non-parametric representations*, such as label transfer [15], SuperParsing [16, 17] and scene collage [18], memorize all the observed scene images and interpret a new scene through its nearest neighbor. [19] explores the image correspondences among nearest neighbor images, and propagates annotations from a partially annotated dataset. (v) The most related work is the Tangram model [20], which introduces the scene hierarchy by a pre-defined dictionary and infers a single configuration for each scene category. We extend the Tangram by proposing a learning-by-parsing method to learn the dictionary and introduce the HST for modeling scenes. Then we augment the HST by adding scene attributes. The earlier versions of our work appeared at [21, 22], in this paper, we will explore the HST model in depth with more experiments and insights.

Scene attributes Visual attributes are demonstrated as valuable semantic cues in various problems such as scene classification [23, 11], generating descriptions of unfamiliar objects [24, 25] and image annotation [26, 27, 28, 29]. Li *et al.* [23] take objects as attributes and propose an "Object Bank" representation containing 200 object detectors. Patterson *et al.* [3] select 102 binary attributes to describe intra-class scene variations and inter-class scene relations. Parikh *et al.* [30] introduce the relative attributes to provide a semantically richer way in describing and comparing scenes. These attributes are learned and inferred at image level, without localization. In contrast, we jointly parse natural images into spatial configurations and localize the attributes, which allows us to provide high precision descriptions to the images.

Attribute localization We categorize the related work into three types: (i) The methods which assume the independence of image regions. Lampert *et al.* [31] propose an efficient sub-window search (ESS) evaluating a classifier function at different sub-windows of an image and then predicting one with the highest score. Berg [32] use Multiple Instance Learning (MIL) to discover attributes. MIL views images as bags of segments and trains a binary classifier to predict the class of segments, under the assumption that each positive training image contains at least one true-positive segment. However, these methods only detect/locate one object at a time, while we aim at parsing an image into multiple objects and attributes simultaneously. (ii) The methods which utilize the geometric context of neighboring image regions. Datta *et al.* [33] classify image regions by discriminative classifiers then use the spatial links between regions to annotate the image. Gupta and Davis [27] exploit the object labels together with prepositions (*e.g.*, on, beside) and comparative adjectives (*e.g.*, larger, smaller). These methods do not consider the long-range relations so that they may confuse the objects with similar appearances (*e.g.* "blue ocean" and "blue sky"), while our model will not. (iii) The methods which search hierarchical relations between categories. Li *et al.* [29] suggest a hierarchical generative model to segment images, annotate regions and categorize scenes. These methods have powerful representation, however, their computations are expensive because the image regions are continuous and the combinations of region-attribute assignment are innumerable. Thus we propose to quantize the scene configuration space and transfer the structure learning problem to parameter learning.

The remainder of this paper is organized as follows: Section 2 defines the HST model for scene representation; Section 3 introduces the weakly supervised learning method of HST; Section 4 shows the experiment results; and finally, in Section 5, a summary is made and some future work is discussed.

2 HST Representation

2.1 Definition of HST-geo

We divide the image lattice into an $n_w \times n_h$ grid and treat each cell as an atomic shape element, then organize these atomic shape elements in an And-Or Tree (AoT) structure. In experiments, we set $n_w = n_h = 8$. For clarity, Fig.3 shows a HST-geo example with $n_w = n_h = 2$. There are three types of nodes in the HST-geo:

(i) Or-nodes V^{OR}, shown as the hollow circles in Fig.3(a), correspond to the grammar rules like

$$r^{OR} : S \rightarrow s|EF|GH$$

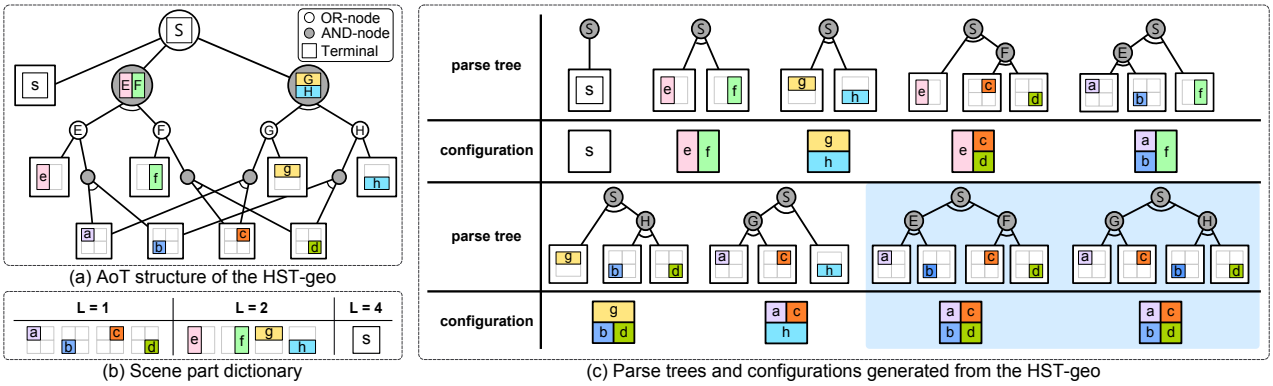
which act as "switches" between the possible decompositions. The branching probabilities $p(s|S)$, $p(EF|S)$, $p(GH|S)$ account for the preference for each decomposition and can be learned as in Section 3.2.

(ii) And-nodes V^{AND}, shown as the solid circles in Fig.3(a), correspond to the grammar rules like

$$r^{AND} : E \rightarrow a \cdot b$$

which represent a fixed decomposition from a parent node E to its child nodes a and b. For simplicity, we only allow two-way decompositions in this paper.

(iii) Terminal nodes V^T, shown as the hollow squares in Fig.3(a). The nodes in HST-geo can terminate at all levels to represent the visual concepts at multiple resolutions. We see a terminal node as a "scene part", and the terminal nodes from all levels form a scene part dictionary $\Delta = V^T$ in Fig.3(b).



(a) AoT structure of the HST-geo

(b) Scene part dictionary

(c) Parse trees and configurations generated from the HST-geo

Fig. 3: Scene configuration representation by HST-geo. (a) The AoT structure of HST-geo on a 2×2 image grid. (b) The scene part dictionary (the empty level, *i.e.*, L=3, is not shown). (c) Parse trees and configurations generated from the HST-geo. The parse trees highlighted in the blue panel show the ambiguity in the HST-geo.

The atomic shape elements are at the bottom of the hierarchy ($L = 1$). According to the grammar rules described above, a number of atomic shape elements compose higher-level nodes at different scales, locations and shapes. The "level" L here means the number of atomic shape elements being used. To avoid the combinatorial explosion, only regular shapes, *i.e.*, squares and rectangles, are allowed. The HST-geo can also allow non-regular shape elements, such as triangles, parallelograms and trapezoids, which make the representation more flexible but complex. We will analyze these choices in the experiment in Section 4.2.

Formally, we define the HST-geo as a 4-tuple

$$\text{HST-geo} = (S, V^N, V^T; \Theta), \tag{1}$$

where S is a start symbol at root and $V^N = V^{AND} \cup V^{OR}$ is a set of non-terminal nodes. Let v index the node and $Ch(v)$ denote its child node set. Θ is a set of branching probabilities at Or-nodes.

$$\begin{aligned} &\Theta = \{\theta(v_i|v); v \in V^{OR}, v_i \in Ch(v)\} \\ &\text{s.t.} \sum_{i=1}^{|Ch(v)|} \theta(v_i|v) = 1; \forall v \in V^{OR}. \end{aligned} \tag{2}$$

The HST-geo is recursively defined with homogeneous structures. Starting from a root which is an Or-node, HST-geo generates alternating levels of And-nodes and Or-nodes, and stops at terminal nodes. The And-Or structure defines a full space of possible parsing with probabilistic context free grammar (PCFG), which we call it a "full HST". By selecting the branches at Or-nodes, a parse tree pt can be derived. Intuitively, when a parse tree collapses, it produces a planar configuration. We utilize this configuration to represent the layout/configuration of a scene.

Fig.3(c) enumerates all the parse trees and configurations generated from a 2×2 HST-geo using only squares and rectangles. The parse trees highlighted in the blue panel in Fig.3(c) show the ambiguity in the HST-geo. The ambiguity arises from the shape elements shared by more than one parent node, which will admit two or more reasonable parse trees for one configuration. The ambiguity can be reduced during learning. Through depth first search, we count the number of nodes ($|V^{OR}|, |V^{AND}|, |V^T|$), parse trees ($|pt|$) and configurations ($|cfg|$) that can be generated from the HST-geo in different granularities. As is shown in Table 1, the parse tree space of HST-geo expands exponentially as the granularity of image grid increases, which brings the potential to account for the complexity of scene configurations.

TABLE 1: Number of nodes, parse trees and configurations generated from the HST-geo

Grid	$\lvert V^{OR}\rvert$	$\lvert V^{AND}\rvert$	$\lvert V^{T}\rvert$	$\lvert pt\rvert$	$\lvert cfg\rvert$
2×2	9	6	9	9	8
4×4	100	200	100	2.87×10^5	6.85×10^4
8×8	1296	6048	1296	1.99×10^{24}	2.00×10^{21}

2.2 Definition of HST-att

Terminal nodes at HST-geo may have semantic labels, called scene attributes in this paper. Thus we extend the HST-geo by HST-att. As is shown at the bottom of Fig.2(b), the HST-att is modeled as a two-level AoT. There are two types of attributes.

(i) Adjective attributes $\mathcal{A}^{adj}$, such as "green" and "cloudy", describe the characteristics of a scene.

(ii) Noun attributes $\mathcal{A}^{n}$, such as "field" and "sky", denote the objects and regions inside a scene. Each noun attribute, acting as an appearance-Or node, has a mixture of adjective attributes, *e.g.*, sky can be blue, cloudy or overcast.

We link each terminal node $v \in \Delta$ in the HST-geo to a noun attribute $a^n \in \mathcal{A}^n$, and further to an adjective attribute $a^{adj} \in \mathcal{A}^{adj}$ according to an association matrix Φ (Fig.2(d)).

$$\Phi : \mathcal{A}^n \times \Delta \mapsto [0, 1], \text{s.t.} \sum_{a \in \mathcal{A}^n} \Phi(a, v) = 1, \forall v \in \Delta, \tag{3}$$

TABLE 2: The terminology used in the HST

Notation	Meaning
V^{OR}, V^{AND}, V^T	Or-nodes, And-nodes and terminal nodes
Δ	scene part dictionary in the HST-geo
v	index of a node
$Ch(v)$	child node set of v
$\Theta = \{\theta(v_i\|v)\}$	branching probabilities at Or-nodes
pt	parse tree
$\mathcal{A}^n, \mathcal{A}^{adj}$	noun and adjective attribute sets
$\Phi : \mathcal{A}^n \times \Delta \mapsto [0,1]$	association matrix
$\phi : A \times V^T \mapsto \{0,1\}$	mapping from attributes to terminal nodes
$pt^+ = (pt, \phi)$	parse tree with attribute assignment

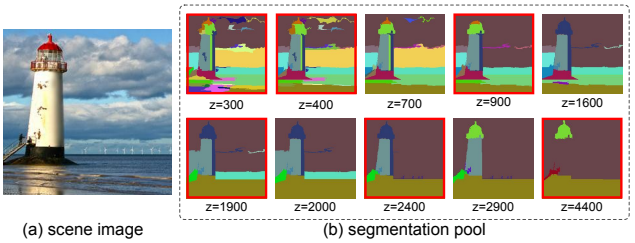


Fig. 4: Multi-scale segmentation. (a) Input image I_m. (b) Segmentations in different layers. The segmented layers in the red frames compose a multi-scale segmentation set $C_m = \{(C_m^k, z_m^k)\}_{k=1}^6$.

where the rows in Φ are the noun attributes and the columns are the scene parts, and we normalize the sum of each column to 1.

Φ measures the probabilities of assigning noun attributes to certain scene parts, *e.g.*, "road" has high probability appearing at the bottom of an image, and the learning method is presented in Section 3.3.

Formally, we define the HST-att as a 3-tuple

$$\text{HST-att} = (\mathcal{A}^n, \mathcal{A}^{adj}; \Phi). \tag{4}$$

In Tabel 2, we summarize the main notations used to describe the HST model.

3 Learning of HST

3.1 Weakly supervised learning

As Fig.2(a) shows, the input of learning is a set of natural images $\mathbf{I} = \{I_m\}_{m=1}^M$ and their text descriptions $\mathbf{A} = \{A_m = (A_m^n, A_m^{adj})\}_{m=1}^M$, where $A_m^n \subseteq \mathcal{A}^n$ and $A_m^{adj} \subseteq \mathcal{A}^{adj}$ denote the noun and adjective attribute sets for the image I_m, respectively.

The hidden variables of HST are

$$\{pt_m^+ = (pt_m, \phi_m)\}_{m=1}^M, \tag{5}$$

where pt_m is the inferred parse tree and ϕ_m is the attribute assignment, as shown in Fig.2(e). Formally, ϕ_m is a mapping from the the inferred attribute set $\hat{A}_m$ to the terminal node set of pt_m, *i.e.*, $V^T(pt_m)$.

$$\phi_m : \hat{A}_m \times V^T(pt_m) \mapsto \{0,1\}, \tag{6}$$

where $\phi_m(a,v) = 1$ if an attribute a is assigned to a terminal node v and $\phi_m(a,v) = 0$ otherwise.

Because the precise locations of attributes are unknown in the input, the learning is weakly supervised. The objective of learning is to estimate the HST parameters, *i.e.*, the branching probabilities Θ and the association matrix Φ, by maximizing a log-likelihood.

$$\begin{aligned}(\Theta^*, \Phi^*) &= \arg\max_{\Theta,\Phi} \log p(\mathbf{I}, \mathbf{A}; \Theta, \Phi) \\ &= \arg\max_{\Theta,\Phi} \sum_{m=1}^M \log \sum_{pt_m^+} p(I_m, A_m, pt_m^+; \Theta, \Phi).\end{aligned} \tag{7}$$

We first separately learn the HST-geo (Section 3.2) and HST-att (Section 3.3) as an initial HST model. Then we jointly learn (Θ, Φ) and infer the hidden parse trees $\{pt_m^+\}_{m=1}^M$ in Section 3.4. Starting from a full HST, we learn the branching probabilities and association matrix, prune the redundant branches, and finally get a compact model. Therefore, we transfer the structure learning of AoT to a tractable parameter learning problem.

3.2 Learning HST-geo

This section presents the learning of HST-geo without text input. Since we do not have ground-truth scene configurations, we use multi-scale segmentations (Fig.4), corresponding to the coarse-to-fine scene configurations, to propose candidate terminal nodes or scene parts in the HST-geo. Given a training image I_m, we first adopt [34] to obtain a multi-scale segmentation by tuning $z \in \{300, 400, \cdots, 5000\}$, where z controls the granularity of segmentation. Then we select six distinct segmented layers (red frames in Fig.4(b)) by comparing the adjacent layers in pixels, and compose a multi-scale segmentation set $C_m = \{(C_m^k, z_m^k)\}_{k=1}^6$, where C_m^k is one segmented layer with the control variable z_m^k.

To learn the HST-geo, we estimate the branching probabilities Θ by maximizing a log-likelihood. We sum out the hidden variables C_m^k and pt_m.

$$\begin{aligned}\Theta^* &= \arg\max_{\Theta} \log p(\mathbf{I}; \Theta) \\ &\propto \arg\max_{\Theta} \sum_{m=1}^M \log \sum_{pt_m,k} p(I_m|C_m^k) p(C_m^k, pt_m; \Theta),\end{aligned} \tag{8}$$

where $p(I|C^k)$ is the likelihood given one segmented layer and we omit the index m hereafter in the derivation for simplicity when there is no confusion.

$$\log p(I|C^k) \propto -\sum_{c\in\{R,G,B\}} \sum_{r\in C^k} ||I^c(r) - \bar{I}^c(r)||^2 - z^k, \tag{9}$$

where c is a color channel; r is a segmented region in C^k; $I(r)$ is the image patch covered by r and $\bar{I}^c(r)$ is the average intensity of r at color channel c. This

term measures the segmentation homogeneity of pixel intensity and penalizes the large z^k.

$p(C^k, pt; \Theta) \propto \exp\{-E(C^k, pt; \Theta)\}$, following the Gibbs distribution, is the joint probability with Θ being the parameters to be learned. Since the HST-geo embodies a PCFG, the contextual relations among the And-nodes are not considered. Thus the energy is defined on two potential terms corresponding to the Or-nodes and terminal nodes of a parse tree.

$$\begin{aligned} & E(C^k, pt; \Theta) \\ &= \sum_{v \in V^{OR}(pt)} E^{OR}(v_i|v) + \lambda \sum_{v \in V^T(pt)} E^T(C^k(v)|v) \end{aligned} \tag{10}$$

where λ is the parameter balancing the two terms (in this paper $\lambda = 0.25$ is set empirically through cross validation). $V^{OR}(pt)$ and $V^T(pt)$ denote the sets of Or-nodes and terminal nodes in pt.

The energy of an Or-node is defined on its branching probability.

$$E^{OR}(v_i|v) = -\ln \theta(v_i|v) = -\ln \frac{\#(v \to v_i)}{\sum_{i=1}^{|Ch(v)|} \#(v \to v_i)}, \tag{11}$$

where $\#(v \to v_i)$ is the number of times that v selects the i-th node/branch $v_i \in Ch(v)$. We learn the branching probabilities in the following subsection.

The energy for a terminal node is defined on the homogeneity of the terminal node in terms of the segmentation label, *i.e.*, how well the configuration of pt fits to the segmented layer C^k.

$$E^T(C^k(v)|v) = -\ln \frac{\sum_{i \in C^k(v)} \mathbb{1}\left[l_i^k = l_v^k\right]}{|C^k(v)|}, \tag{12}$$

where $\mathbb{1}[\cdot]$ is the indicator function and $C^k(v)$ denotes the segmented patch covered by the terminal node v. In the k-th layer, l_i^k is the segmentation label of pixel i and l_v^k is the dominant label of terminal node v.

In addition, the terminal nodes are allowed to be locally adjustable to fit the scene boundaries. We introduce 12 node activities including perturbations in location ($\delta(x) = [\pm 8, \pm 16]$), scale ($\delta(s) = [1 \pm \frac{1}{32}, 1 \pm \frac{1}{16}]$) and orientation ($\delta(a) = [\pm \frac{\pi}{48}, \pm \frac{\pi}{24}]$).

Taking the multi-scale segmentations as input, we solve Eq.8 by a learning-by-parsing method which is an EM-like strategy. The E-step infers the optimal parse trees pt^* which approximate the scene configurations with small error and low complexity by dynamic programming. The M-step estimates the parameters Θ by maximum likelihood estimation (MLE).

(i) E-step: parse tree inference. Keeping the current branching probabilities Θ fixed and assuming $p(I)$ is uniform, an optimal parse tree pt^* can be inferred from the HST-geo for each training sample.

$$\begin{aligned} pt^* &= \arg\max_{pt} \log p(pt|I; \Theta) = \arg\max_{pt} \log p(I, pt; \Theta) \\ &\approx \arg\max_{pt,k} \log p(I|C^k) p(C^k, pt; \Theta), \end{aligned} \tag{13}$$

Here, we use the best segmented layer to approximate the summation of all layers. In practice, given a training image, we infer an optimal parse tree for each segmented layer by minimizing Eq.10. Then we choose the best parse tree and segmented layer according to Eq.13.

Because of the tree-structure of HST-geo and the independence assumption in PCFG, the optimal parse tree can be obtained by Dynamic Programming (DP). Specifically, we start with calculating the data term (Eq.12) for each terminal node, then for the upper level Or-node v, DP evaluates all its possible branches $v_i \in Ch(v)$ and selects the best one such that

$$v_i^* = \arg\min_{v_i \in Ch(v)} (E^{OR}(v_i|v) + E(C^k(v_i), pt(v_i); \Theta)), \tag{14}$$

where $pt(v_i)$ is the sub-tree with v_i as the root and the sub-tree energy $E(C^k(v_i), pt(v_i); \Theta)$ is defined in Eq.10.

(ii) M-step: update branching probabilities. We rewrite the objective function in Eq.8 as

$$\begin{aligned} L(\Theta) = & \sum_{m=1}^{M} \log \sum_{pt_m, k} p(I_m|C_m^k) p(C_m^k, pt_m; \Theta) \\ & + \sum_{v=1}^{|V^{OR}|} \alpha_v (1 - \sum_{i=1}^{|Ch(v)|} \theta(v_i|v)), \end{aligned} \tag{15}$$

where α_v is the Lagrange multiplier for the branching probabilities at each Or-node to be normalized.

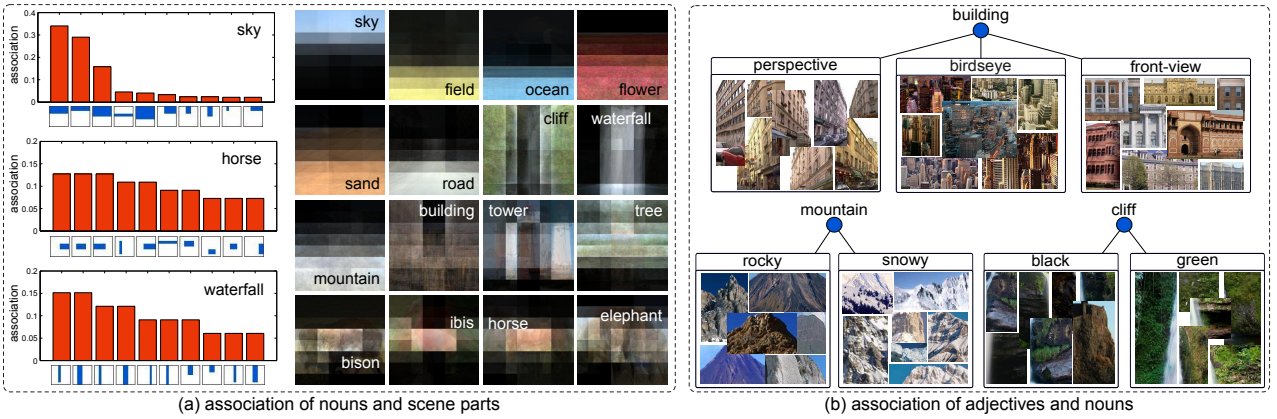
We estimate Θ by MLE, which takes the derivative of $L(\Theta)$ *w.r.t.* $\theta(v_i|v)$ and sets it to zero. We adopt the Viterbi algorithm [35], which is an approximated method using the optimal parse tree instead of all parse trees, and update the branching probabilities (see supplementary material for detailed derivation).

$$\begin{aligned} & \theta^{(t+1)}(v_i|v) \\ &= \frac{1}{\alpha_v} \sum_{m=1}^{M} \mathbb{1}[v_i \in pt_m^*] \cdot p(pt_m^*|C_m^k; \Theta^{(t)}) p(I_m|C_m^k), \end{aligned} \tag{16}$$

where pt_m^* and C_m^k are the parse tree and segmented layer inferred in the E-step, and $\mathbb{1}[v_i \in pt_m^*]$ indicates if the branch v_i is selected in pt_m^*. Θ is set to be uniform as initialization. We repeat inferring parse trees (E-step) and updating parameters (M-step) until convergence. Finally, those branches whose probabilities are below a certain threshold (say 0.01) are pruned. We collect the terminal nodes from all levels, and they compose the scene part dictionary Δ.

3.3 Learning HST-att

Recall $\mathcal{A}^n$ is the noun attribute set, $\mathcal{A}^{adj}$ is the adjective attribute set and Φ: $\mathcal{A}^n \times \Delta \mapsto [0, 1]$ is the association matrix measuring the probabilities of assigning noun attributes to scene parts. For an image I_m, $\phi_m : A_m \times V^T(pt_m) \mapsto \{0, 1\}$ is the attribute



(a) association of nouns and scene parts (b) association of adjectives and nouns

Fig. 5: Analysis of HST-att learning. (a) The histogram on the left shows the association probability between a noun attribute and scene parts, where the horizontal axis indexes different scene parts and the vertical axis is the association probability from Φ. The right figure shows the learned spatial prior for each noun attribute, which is the average image of attribute patches. (b) The adjective clusters belonging to the noun attributes.

assignment from the attribute set $A_m = (A_m^n, A_m^{adj})$ to the terminal node set $V^T(pt_m)$.

We compute Φ by counting the co-occurrence of nouns $a \in A^n$ and terminal nodes $v \in V^T(pt)$ in the training images.

$$\Phi(a,v) = \frac{\sum_{m=1}^{M} \mathbb{1}[a \in A_m^n] \cdot \mathbb{1}\big[v \in V^T(pt_m)\big] \cdot \phi_m(a,v)}{\sum_{a \in \mathcal{A}^n} \Phi(a,v)}, \tag{17}$$

where $\phi_m(a,v) \in \{0,1\}$ indicates whether a noun attribute a is assigned to a terminal node v.

In learning the HST-geo in Section 3.2, an optimal parse tree pt^* can be inferred for each training sample. However, the correspondence between scene parts (terminal nodes in pt^*) and attributes is still unknown because the attributes are annotated at image level rather than on image regions. So we initialize ϕ_m by turning on all possible assignments, *i.e.*, $\phi_m(a,v) = 1, \forall (a,v) \in A_m^n \times V^T(pt_m^*), m = 1, ..., M$.

Then, we learn the HST-att through an iterative procedure, including two steps.

(i) Update association matrix Φ through noun attribute localization. Given the current $\Phi^{(t)}$, we establish a bipartite graph $G(V^T(pt_m^*), A_m^n, \xi_m)$ for each training image, where $V^T(pt_m^*)$ denotes the terminal nodes in pt_m^* and A_m^n denotes the noun attributes from text. If $|V^T(pt_m^*)| \neq |A_m^n|$, add dummy nodes for balance. ξ_m is the edge set connecting $V^T(pt_m^*)$ and A_m^n, whose weight is defined as:

$$w(a,v) = \Phi^{(t)}(a,v) \cdot p(a|I(v)). \tag{18}$$

Let $F(I(v),a)$ be the score of classifying the image patch $I(v)$ as an attribute a, thus $p(a|I(v)) = \max_c \frac{\exp\{F^c(I(v),a)\}}{\sum_{a'} \exp\{F^c(I(v),a')\}}$. As defined in Section 2.2, one noun attribute has a mixture of adjectives, c is the number of adjectives and F^c denotes the adjective classifier. At initialization, $p(a|I(v))$ is set to be uniform, since the appearance models are empty.

We adopt the Hungarian algorithm [36] to solve the bipartite graph and get a one-to-one matching $\phi_m^{(t+1)}$, *i.e.*, for each image we localize the noun attributes to scene parts (terminal nodes). Then we update the association matrix $\Phi^{(t+1)}$ by recalculating Eq.17.

(ii) Update attribute appearance models. For each noun attribute, as shown in Fig.5(b), we crop the image patches covered by the assigned terminal nodes and do clustering according to the given adjectives, such as "rocky mountain" and "snowy mountain". Then for each cluster (*i.e.*, a noun and adjective pair), we train a kernel SVM classifier with one-versus-all mode as its appearance model. In this paper, we adopt bag-of-words (BoW) features on color histogram and SIFT, and utilize the histogram intersection kernel.

Repeating the above steps till the change of $\{\phi_m\}_{m=1}^M$ below a threshold, finally, we get the association matrix Φ and the attribute appearance models.

Fig.5(a)(left) shows the association of noun attributes and scene parts/terminal nodes, where the horizontal axis indexes the scene parts and the vertical axis indexes the association probability from Φ. For example, "sky" has a higher probability to cover the top of an image and "horse" has a higher probability to cover the middle part. To qualitatively evaluate the association, for each noun attribute, we average the image patches assigned to it. Interestingly, as illustrated in Fig.5(a)(right), although learning in a weakly supervised way, our association shows the similar spatial priors of the object categories to [15] (see Fig.3 in [15]). Fig.5(b) shows that the image patches assigned to each noun are split into multiple clusters according to the given adjectives.

3.4 Joint inference and learning

Using the learned HST-geo and HST-att as an initial HST model, we can infer

$$pt_m^+ = (pt_m, \phi_m), \quad \phi_m : \hat{A}_m \times V^T(pt_m) \mapsto \{0,1\}$$

directly from the image I_m and text description A_m, rather than multi-scale segmentation, by maximizing $p(I_m, A_m, pt_m^+; \Theta, \Phi) \propto \exp\{-E(I_m, A_m, pt_m^+; \Theta, \Phi)\}$. Let $\hat{a} = (\hat{a}^n, \hat{a}^{adj}) \in \hat{A}_m$ denote the inferred attribute. The energy is rewritten from Eq.10.

$$\begin{aligned} & E(I_m, A_m, pt_m^+; \Theta, \Phi) \\ &= \sum_{v \in V^{OR}(pt_m)} E^{OR}(v_i|v) + \lambda_1 \sum_{v \in V^T(pt_m)} E^n(\hat{a}^n|v) \\ &+ \lambda_2 \sum_{\hat{a}^n \in \hat{A}_m^n} E^{adj}(\hat{a}^{adj}|\hat{a}^n) + \lambda_3 \sum_{v \in V^T(pt_m)} E^T(\hat{a}|I(v)) \\ &+ \sum_{\hat{a} \in \hat{A}_m} E^A(\hat{a}, A_m), \end{aligned} \tag{19}$$

where $(\lambda_1, \lambda_2, \lambda_3)$ are the parameters balancing the energy terms (in this paper $(\lambda_1, \lambda_2, \lambda_3) = (0.7, 0.1, 2)$ are set empirically through cross validation). The first term measures the scene configuration prior which is the same as Eq.11. The second term measures the noun attribute association with the terminal node:

$$E^n(\hat{a}^n|v) = -\ln \Phi(\hat{a}^n, v). \tag{20}$$

The third term is designed to model the compatibility between a noun and an adjective

$$E^{adj}(\hat{a}^{adj}|\hat{a}^n) = -\ln p(\hat{a}^{adj}|\hat{a}^n), \tag{21}$$

where $p(a^{adj}|a^n) = \frac{\sum_{m=1}^M \mathbb{1}[a^n \in A_m^n] \mathbb{1}[a^{adj} \in A_m^{adj}]}{\sum_{m=1}^M \mathbb{1}[a^n \in A_m^n]}$ can be counted from the given text phrases. The fourth term is an attribute specific data term

$$E^T(\hat{a}|I(v)) = -\ln p(\hat{a}|I(v)), \tag{22}$$

where $I(v)$ is the image patch occupied by v and $p(\hat{a}|I(v)) = \frac{\exp\{F(I(v), \hat{a})\}}{\sum_{a'} \exp\{F(I(v), a')\}}$. $F(I(v), a)$ is the score of classifying $I(v)$ as attribute a. The last term $E^A(\hat{a}, A_m)$ assumes ∞ on attributes outside A_m and 0 otherwise, making sure $\hat{a} \in A_m$.

Because both the HST-geo and HST-att are tree-structured, DP algorithm can be applied to jointly infer the parse tree pt_m and attribute assignment ϕ_m. In the joint inference, we start with calculating E^T for the terminal nodes. Then, for every terminal node, DP evaluates all possible attributes according to the sum of E^T, E^n and E^{adj}, and assigns a best attribute (a noun and adjective pair) to it. Next, DP iteratively proceeds to the upper level Or-nodes and selects best branches until finds the optimal parse tree with associated attributes:

$$(pt_m^+)^* = \arg\max_{pt_m^+} p(I_m, A_m, pt_m^+; \Theta, \Phi). \tag{23}$$

In training, we use E^A to constrain the consistency between inferred attributes and given text descriptions. While it cannot be done for testing images as their attributes are unknown. Thus in testing, we simply abandon this energy term.

TABLE 3: The learning algorithm

Initialization	
1	Learn HST-geo with parameter $\Theta^{(0)}$ based on the multi-scale segmentation. (Section 3.2)
2	Learn HST-att with parameter $\Phi^{(0)}$ and train appearance models. (Section 3.3)
Iteratively learn HST (Section.3.4)	
3	Jointly infer pt^+ for each training sample. (*Eq.19*)
4	Update $\Theta^{(t+1)}$ in HST-geo. (*Eq.16*)
5	Update $\Phi^{(t+1)}$ in HST-att and retrain appearance models. (*Eq.17*)
6	Repeat 3 - 5 until convergence.

Follow the learning-by-parsing framework introduced in Section 3.2, we can re-estimate the HST-geo and HST-att based on $\{pt_m^+\}_{m=1}^M$. We summarize the entire learning procedure in Table 3, which contains two steps. (i) Separately learn HST-geo parameters Θ and association matrix Φ based on the multi-scale segmentations as an initialization; and (ii) Iteratively update the HST-geo and the HST-att based on the joint inference.

4 Experiments

4.1 Datasets

We test our method on two datasets: the LabelMe Outdoor (LMO) dataset [15] and an outdoor scene attribute (SceneAtt) dataset [22] collected by ourselves.

The LMO includes 2,688 images of 256×256 pixels in size from 8 outdoor scene categories, *i.e.*, coast, forest, highway, inside city, mountain, open country, street and tall building. These images are annotated with label maps containing 33 semantic labels, such as sky and road, plus a void label. To better use the LMO dataset, we do the following pre-processes. (i) Merging synonyms. Because the LMO dataset was annotated through Amazon Mechanical Turk, the annotations are not always consistent. As shown in Fig.6(a), some people labeled grassland as grass, while others labeled it as field. Thus we merge the synonymous labels, such as grass and field, water and rivers, to resolve the annotation ambiguities. (ii) Filling holes. We fill the non-labeled regions in LMO, *e.g.*, the "void" regions in Fig.6(b). (iii) Ignore tiny areas of small objects. The goal of this work is focus on learning the semantic structures of outdoor scenes at a grand scale. We observe that those tiny objects, such as birds, poles, street lights and so on, do not affect the semantic meaning of scenes neither the global scene structures very much. Hence we ignore tiny objects in this work. Specifically, we either simply ignore them by assigning void label instead (top in Fig.6(c)) or merge them to the surrounding regions, *e.g.*, merge sun or moon to sky (bottom in Fig.6(c)). Finally, we got 12 labels: $\mathcal{L}$ = {bridge, building, desert,

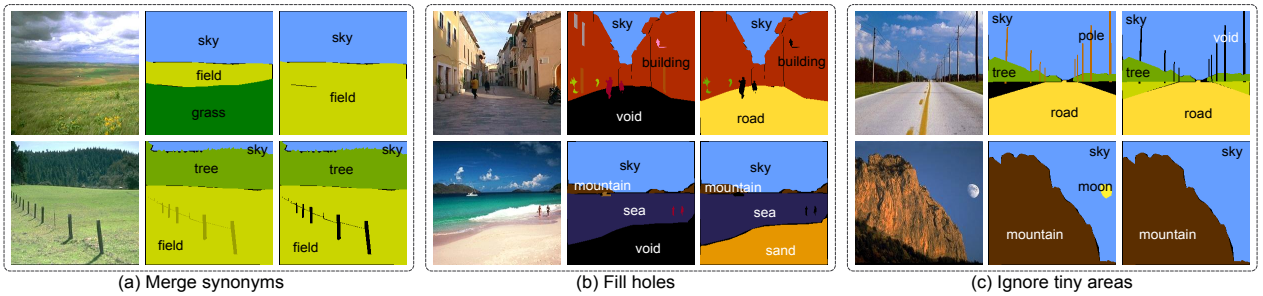


Fig. 6: Three pre-processes on LMO dataset. For each pre-process, the 1st column is the original image; the 2nd column is the label map in [15] and the 3rd column is the pre-processed one. (a) Synonymous labels, such as "grass" and "field", are merged. (b) "void" regions are filled. (c) Tiny objects, such as "pole" and "moon" are ignored, either by assigning "void" label instead (top) or merging to the surrounding regions (bottom).

Fig. 7: Examples of SceneAtt dataset and the ground-truth for evaluation.

field, mountain, river, road, rock, sand, sea, sky, tree}, which are coarse but essential parts for a scene[1].

We also created a new dataset (Fig.7), namely outdoor scene attribute (SceneAtt) dataset[2], to test the attribute recognition and localization. The previous datasets containing images and text descriptions, such as [3, 37, 38, 39], usually designed attributes for foreground objects or human activities, which are beyond our scope, as our focus is to model background scene configurations. We collected 1,225 outdoor images (256×256 pixels in size) from LMO [15], SUN Attribute dataset [3] and image engines, such as Google images and Flickr. We created the text descriptions and got the attribute set of SceneAtt $\mathcal{A}^n$={*sky, flower, mountain, ibis, horse, ...*}, $\mathcal{A}^{adj}$={*blue, cloudy, rocky, snowy, brown, ...*} containing 17 noun attributes and 30 noun and adjective attribute pairs in total. For evaluating the localization accuracy, SceneAtt provides the ground-truth bounding box for each attribute (right panel in Fig.7).

4.2 Analysis of learning

Efficiency of representation We evaluate the efficiency of HST-geo for representing scene configurations on LMO dataset. Let C denote the label map corresponding to a scene configuration, and Ω^* denote an unknown set of valid configurations for a scene category. The HST-geo is actually a grammar, whose language is the set of all valid configurations,

$$\Omega(\text{HST-geo}) = \{C : C = g(pt; \Delta)\}.$$

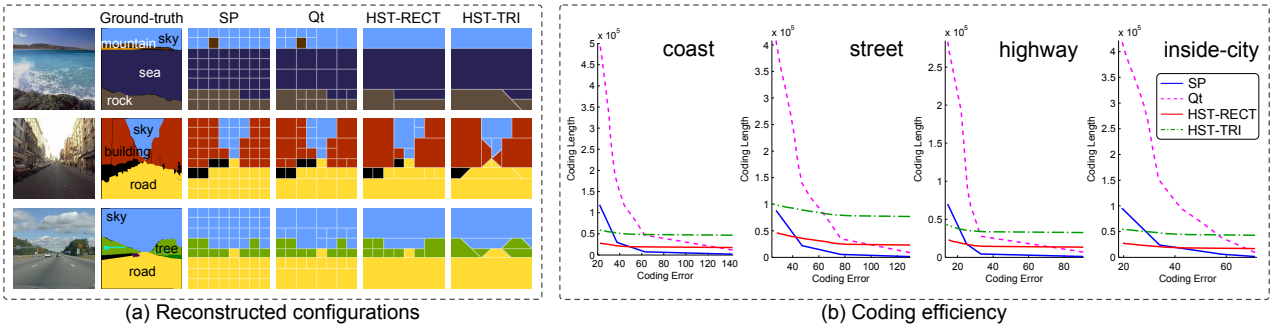
1. *http://vcla.stat.ucla.edu/people/∼shuo.wang/HST_att.html*
2. *http://vcla.stat.ucla.edu/people/∼shuo.wang/SceneAtt.html*

Here pt is the parse tree for C, Δ is the scene part dictionary and $g()$ is the generation function. The representation efficiency means that given any scene configuration $C \in \Omega^*$, we can generate a configuration $\hat{C} \in \Omega(\text{HST-geo})$ by a parse tree pt so that $\hat{C}$ approximates C with less than ϵ error and pt is small.

For the images of size 256×256, we divide them to an 8×8 grid at the bottom level. For each category, we randomly select 100 samples for training and the rest for testing to learn the HST-geo model.

We compare our model against three scene representations. (i) Spatial Pyramid (SP) [5]: The SP model generates a S-level representation for a scene, of which each level divides the scene images into $2^s \times 2^s (s = 0, ..., S-1)$ grid. (ii) Quadtree(Qt) [6]: Given an image, the Qt recursively divides it into four quadrants of equal size until reaching a threshold of homogeneity or size. (iii) HST-TRI: As we discussed in Section 2.1, the proposed HST-geo only allows squares and rectangles as valid shapes of terminal nodes. By allowing more shapes, such as triangles, parallelograms and trapezoids, the model will become more flexible but complex. For distinction, in this section we use HST-RECT to denote the HST-geo model with only squares and rectangles, and HST-TRI to denote the model that also allows triangles, parallelograms and trapezoids.

Given an annotated label map, we reconstruct it by inferring a parse tree and filling each terminal node with the dominant label. Fig.8(a) shows the reconstructed results of SP, Qt, HST-RECT and HST-TRI. Intuitively, SP uses the most terminal nodes and gets the finest reconstructed label maps, while



(a) Reconstructed configurations (b) Coding efficiency

Fig. 8: Efficiency of representation. (a) Given the annotated label maps in the 2nd column, we reconstruct the label maps by Spatial Pyramid (SP), Quadtree(Qt), HST-geo with squares and rectangles (HST-RECT) and HST-geo with triangles, parallelograms and trapezoids (HST-TRI) methods in the 3rd - 6th columns respectively. (b) The rate-distortion curve of SP, Qt, HST-RECT and HST-TRI, where the horizontal axis denotes the coding error and the vertical axis denotes the coding length. (see more results in supplementary material)

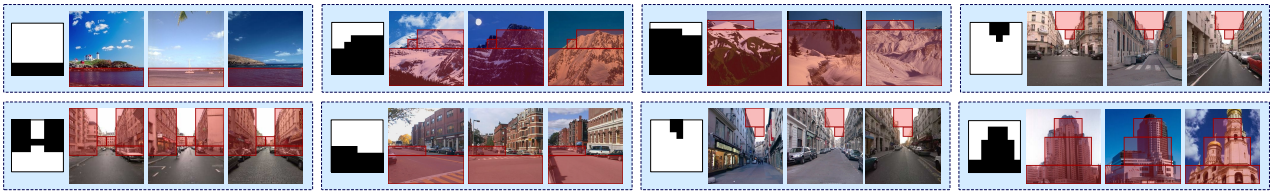


Fig. 9: Terminal node groups which are often observed in different scenes.

HST-RECT uses much less terminal nodes and gets coarser but still satisfying results. Quantitatively, we evaluate the representation efficiency by the rate-distortion curve defined by the coding error *w.r.t.* the coding length. The coding error and coding length are measurements balancing the model complexity and representative ability. With small coding error and low coding length, a model can effectively capture the main configurations of scenes without overfitting.

The *coding error* counts the per-pixel error ratio in the reconstructed scene label maps, defined as

$$CE = \frac{1}{|\Lambda|}\sum_{m=1}^{M} ||\hat{C}_m - C_m||, \tag{24}$$

where $|\Lambda| = 256 \times 256$ is the image lattice, $\hat{C}$ is the reconstructed label map and C is the ground-truth.

The *coding length* is defined as the total bits stored in a reconstructed binary file which varies with different methods. (i) For SP, $CL_{\text{SP}} = \sum_{m=1}^{M}\sum_{s=0}^{S-1} 2^{2s} \times B_l$, where $S = 3$ and $B_l = \lceil \log_2(|\mathcal{L}|) \rceil$ is the coding bits for the semantic label. (ii) For Qt, $CL_{\text{Qt}} = \sum_{m=1}^{M} n_m \times B_{sq}$, where n_m is the number of terminal nodes in the reconstructed configuration and $B_{sq} = B_l + B_p + B_s$ is the coding bits for each node. $B_p = \lceil \log_2(w_{C_m}) + \log_2(h_{C_m}) \rceil$ is the coding bits for the node position, where w_C and h_C are the width and height of C_m, and $B_s = \lceil \log_2 S \rceil$ is the coding bits for scale, here $S = 3$ is the maximum level allowed in Qt. (iii) For HST-geo, considering the scene part dictionary, the coding length is defined as

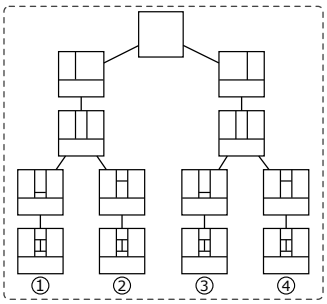
$$CL_{\text{HST-geo}} = CL(\Delta) + \sum_{m=1}^{M}\sum_{v \in V^T(pt_m)} (B_l - \log p(v)), \tag{25}$$

where $p(v)$ is the frequency of the terminal node v appearing in the dictionary, and $CL(\Delta)$ is the coding length of learned scene part dictionary defined as:

$$CL(\Delta) = \sum_{s=1}^{S} |\Delta_s| \cdot (s-1) \cdot 2\log_2 s, \tag{26}$$

where $|\Delta_s|, s = 1...S$ denotes the number of terminal nodes in the s-level of the dictionary. An s-level terminal node consists s atomic shape elements (Fig.3(b)). $s-1$ means we code the rest of atomic elements *w.r.t.* the first one. $2\log_2 s$ is the sum of coding bits to localize an atomic element *w.r.t.* the first one in horizontal and vertical by assuming that a terminal node is a connected composition of atomic elements.

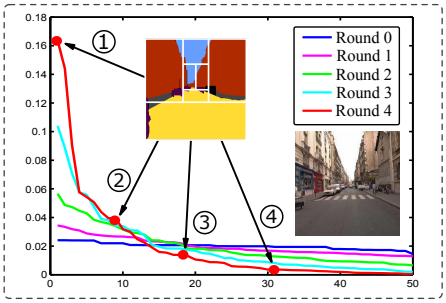
Fig.8(b) shows the changes of coding error (horizontal axis) *w.r.t.* the coding length (vertical axis). We observed: (i) When the coding error is high, less terminal nodes are selected. The coding lengths of HST-RECT and HST-TRI are always above SP and Qt due to the coding bits for the scene part dictionary. (ii) When the coding error decreases, the coding lengths of SP and Qt increase exponentially because of the exponential growth of the number of additional terminal nodes. However, due to the over-complete dictionary and adaptive variable-length coding strategy (Shannon entropy coding), the coding lengths of HST-RECT and



(a) Ambiguous parse trees

Round	①	②	③	④
0	0.024	0.024	0.020	0.020
1	0.034	0.024	0.023	0.016
2	0.048	0.032	0.024	0.012
3	0.058	0.034	0.018	0.010
4	0.164	0.040	0.014	0.000

(b) Posterior probability



(c) Posterior probability distribution of parse tree

Fig. 10: Ambiguity reduction. (a) An illustration of ambiguous parse trees. (b) The posterior probability in each learning round of the ambiguous parse trees. (c) The posterior distribution of the inferred parse trees, where the horizontal axis denotes different parse trees and the vertical axis denotes their posterior probabilities.

HST-TRI are much more stable and finally go below SP and Qt when the coding error is small. (iii) Compared with HST-RECT, although HST-TRI contains richer shapes, it costs more coding bits consistently for every category. So we adopt HST-RECT to model the scenes, which can always get reasonable coding accuracy with compact coding length.

Meaningful terminal groups Fig.9 shows some learned terminal node groups which are often observed in different scenes. They are composed of several terminal nodes of same labels and form meaningful sub-configurations of semantic regions in the scenes, such as "ocean", "mountain" and "building".

Ambiguity reduction The compositional ambiguity of HST-geo is reduced during learning. Fig.10(a) shows four different parse trees for the same configuration of a street scene . The ambiguity of parse trees is measured by their posterior probabilities (Eq.13). Fig.10(b)&(c) show the posterior probabilities are similar at initialization (Round=0) but become increasingly polarized after each round of learning.

4.3 Application I. Scene classification

For the task of scene classification, as it does not involve parsing and attributes, we select 2-5 most frequent configurations from the HST-geo as templates for each scene category according to the posterior probability. Then we use these templates for feature extraction and discriminative training (see details in supplementary material). Table 4 compares the average precision (AP) of scene classification, and it shows the improvement of our model over existing methods.

4.4 Application II. Scene attribute recognition

We train a complete HST model, *i.e.,* HST-geo and HST-att, across scene categories. We use the SceneAtt dataset as the test bed, whose data are outdoor scene images with attributes in text, where the precise locations of attributes are unknown. We randomly split the dataset into 645 images for training (50 images per noun and adjective attribute pair on average) and the rest for testing. We show the attribute recognition in this section, followed by attribute localization in the next section.

TABLE 4: The scene classification performance

	Gist [9]	BoW [7]	SPM [5]	LLC [8]	Tangram [20]	Ours
AP (%)	72.15	84.57	84.92	87.97	86.07	**91.71**

TABLE 5: The attribute recognition performance

	cKernel [40]	SPM [5]	HST-geo [22]	HST(greedy) [22]	HST(iter)
MAP(%)	64.48	53.11	51.67	67.58	**72.17**

Attribute recognition evaluates the accuracy of whether an attribute presence in an image or not. We compare our model with four methods: (i) cKernel: [40] shows the combined feature kernels are more powerful than any individual feature. For comparison, we use a combined kernel of gist, dense SIFT, HOG 2×2, self-similarity, *etc.* (see [40] for details) and train a binary SVM classifier for each attribute. (ii) Spatial Pyramid Matching (SPM) [5]: SPM partitions the image into increasingly finer spatial sub-regions and computes BoW features on SIFT from each sub-region. (iii) HST-geo: To evaluate the contribution of attribute association, we compare our method with HST-geo. Specifically, given an image, we parse it based on its multi-scale segmentation then classify each terminal node by the classifiers trained in (i). (iv) HST(greedy): Our previous work [22] adopts a greedy method to learn the HST-att. While in this paper, we extend it to an iterative algorithm based on bipartite graph matching (Section 3.3). We compare these two learning methods, and use HST(greedy) and HST(iter) to denote the two methods.

Table 5 shows the mean average precision (MAP) of attribute recognition. SPM has low performance because of the lack of color features, which are strong

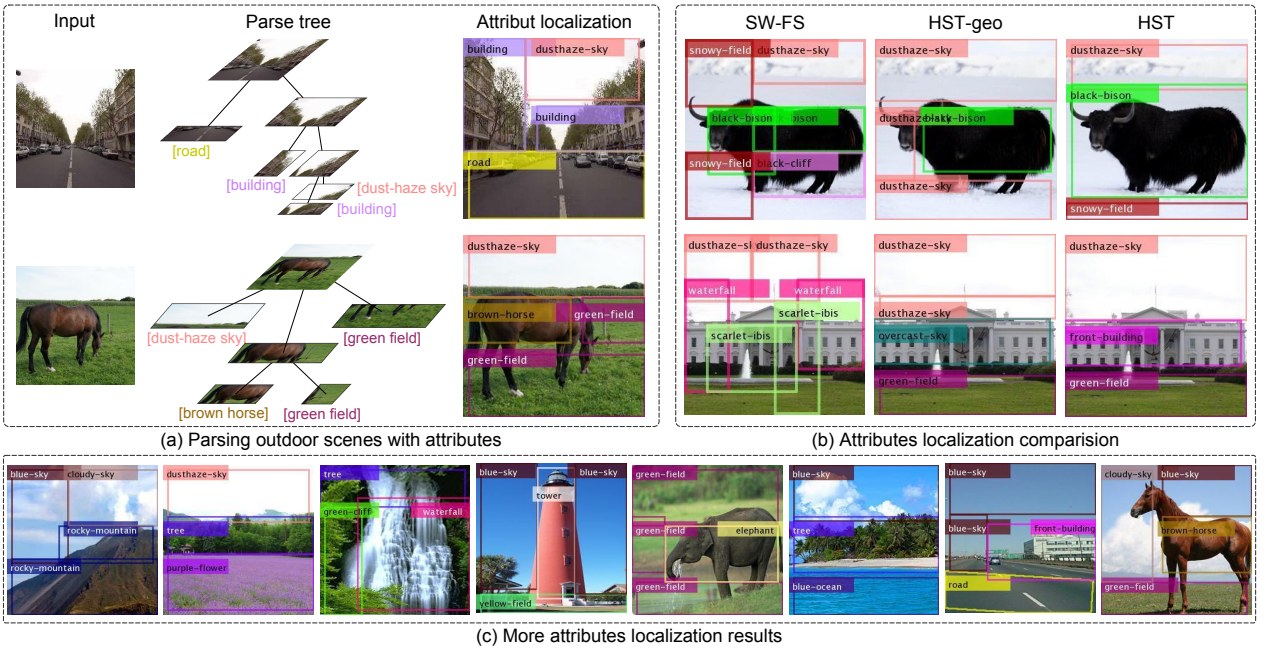


Fig. 11: Attribute localization results. (a) Parse trees with associated attributes. (b) Comparison of baseline methods with ours. (c) More attribute localization results from our method.

cues in scene attribute recognition. Although HST-geo and cKernel share classifiers, cKernel performs better because those classifiers are trained on image level while the testing inputs in HST-geo are image patches. Benefiting from integrating scene layouts with attributes, HST outperforms all others. Moreover, the iterative learning method makes our model more reliable and gets the best performance.

4.5 Application III. Scene Attribute localization

We evaluate our method in attribute localization against three baseline methods. (i) A fully supervised sliding window method (SW-FS) [31]: SW-FS first trains attribute classifiers based on the ground-truth bounding boxes, then applies those classifiers to sub-images at different locations and scales. The sub-images are ordered by classification scores and taken as detected attribute regions by non-maximum suppression with 0.3 overlap threshold. (ii) HST-geo: As mentioned in the last section, given an image, we first parse it according to its multi-scale segmentation then classify each terminal node by the cKernel SVM classifiers. (iii) HST(greedy): We compare the greedy learning method in [22] with our iterative learning method in attribute localization.

Fig.11(b) shows the comparison of the benchmark methods with ours. Without considering the association between scene parts and attributes, SW-FS may divide a semantic region into fragments, and HST-geo may confuse certain attributes with similar appearances, such as "dust-hazed sky" and "snowy field." Fig.11(a) shows the parse trees generated from the HST and Fig.11(c) shows more localization results.

TABLE 6: The attribute localization performance

	SW-FS [31]	HST-geo [22]	HST(greedy) [22]	HST(iter)
MAP (%)	33.88	32.55	50.22	**51.20**

We quantitatively evaluate the attribute localization by following the procedure in [40]. An inferred bounding box B_v is a correct localization if $\frac{area(B_v \cap B_{\text{gdth}})}{area(B_v)} >= \mathcal{T}$, where B_{gdth} is the ground-truth bounding box. We do not care if the ground-truth window is larger than the localization, *e.g.*, a "blue sky" patch is correctly localized even if the ground-truth "blue sky" has much greater spatial occupation. As in [40], we set $\mathcal{T} = 50\%$ to tolerate the inaccurate bounding box of highly non-convex objects, *e.g.*, steep mountain. We use 11-point interpolated average precision [41] to evaluate the localization accuracy. The mean average precision (MAP) reported in Table 6 shows a large improvement of our method.

Some typical failure examples are shown in Fig.12. (i) Some attributes with similar spatial prior and appearances may confuse our model, such as "black ibis" and "black bison" in Fig.12(a). (ii) Some attributes can only be partially localized, such as "white horse" in Fig.12(b). (iii) Some attributes are missing, such as "tower" in Fig.12(c).

4.6 Application IV. Scene labeling

In this section, we extend the attribute localization to pixel-wise scene labeling. We test our model on

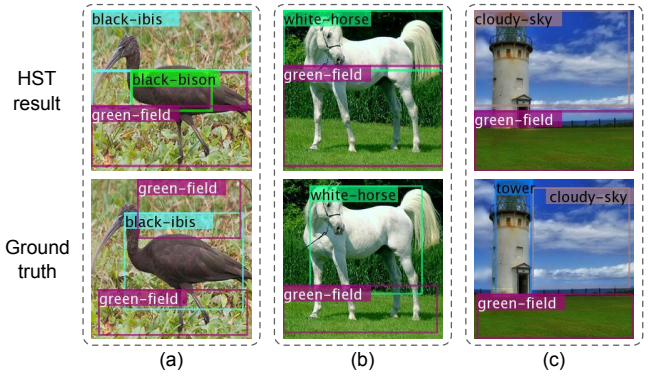


Fig. 12: Some typical failures. (a) "Black ibis" is confused with "black bison" because their similar spatial prior and appearance. (b) "White horse" is partially localized. (c) "Tower" is missing.

the pre-processed LMO dataset with 800 images (100 per category) for training and the rest for testing, and show the improved scene labeling with reduced running time against the previous methods.

Since the LMO dataset only has object labels, such as "sky" and "mountain", we take them as noun attributes and set adjective attributes to void in the HST training. The labeling procedure is described in Fig.13(a). Given a test image, we first infer a parse tree $pt^+ = (pt, \phi) = \{(v_i, a_i), v_i \in V^T(pt), a_i \in \hat{A}\}$ to associate the labels a_i (noun attributes) to scene parts v_i. Then we use $z = 300$ to get an overly segmented image $C = \{r_j, j = 1...N_r\}$ [34], where r_j is a segmented region and N_r is the total number of regions depending on the images. For each r_j, annotate its inside pixels as a_i, if r_j is occupied most by v_i, *i.e.*, $r_j \rightarrow a_i$, if $(v_i, a_i) = \arg\max_{v_i \in V^T(pt)} \frac{area(r_j \cap v_i)}{area(r_j)}$.

We compare our method with two fully supervised methods. (i) Label transfer (LT) [15]: Given an image, the LT retrieves K=20 nearest neighbors from the training data based on gist and HoG, then warps the annotations of retrieved images to fit the target image by SIFT flow. (ii) SuperParsing [16]: For a given image, SuperParsing first retrieves K=200 images based on the global features (SIFT, gist, *etc.*), then it divides the given image into superpixels and gets the nearest neighbor superpixel matches from the retrieval set. With the same data split and annotations, we run the released source codes of LT and SuperParsing, then compare the labeling results in Fig.13(b) and report per-class accuracy on LMO dataset in Table 7. On average, our method outperforms LT and SuperParing.

Since HST quantizes the scene configuration space and the model becomes compact through learning, our method is much faster than LT and SuperParsing. On an Intel 64 bit i7-3770 CPU, 3.40GHz, 20.0GB RAM, Window 7 system, for 1,888 testing images of 256×256 in size, our running time is 3.39 hours (6.45s per image) in total, while LT takes 64.61 hours (123.20s per image) and SuperParsing takes 14.42 hours (27.50s per image). Our method is nearly 19 times faster than LT and 4 times faster than SuperParsing.

5 Summary

This paper presents a novel scene representation namely Hierarchical Space Tiling (HST) and a weakly supervised learning method. The contributions of the paper are threefold. (i) We quantize the space of scene configurations in an And-Or Tree (AoT) structure, and transfer the challenging structure learning problem to a tractable parameter learning problem. (ii) We connect the attributes to a hierarchical scene model, which provides rich semantics to scenes. (iii) We propose a weakly supervised method to learn the HST model when the precise locations of attributes are unknown in training. We show the HST can learn a parsimonious scene representation. In applications, we demonstrate our method in scene classification, attribute recognition, localization, and pixel-wise scene labeling. The attributes used in this paper are related to local objects and regions. Global attributes, such as indoor and outdoor, will be studied in the ongoing research by extending our model to attribute grammar.

This work is supported by 973-2015CB351800, NSFC-61272027, NSFC-61231010, NSF CNS-1028381, MURI ONR N00014-10-1-0933 and China Scholarship Council.

The supplementary material can be downloaded at *http://vcla.stat.ucla.edu/people/∼shuo.wang/HST_att.html*.

References

[1] Z. Si and S. C. Zhu, "Learning and-or templates for object modeling and recognition," *PAMI*, 2013.
[2] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, 2006.
[3] G.Patterson and J.Hays, "Sun attribute database: discovering, annotating, and recognizing scene attributes," *CVPR*, 2012.
[4] S. C. Zhu and D. Mumford, "A stochastic grammar of images," *Found. Trends. Comput. Graph. Vis.*, 2006.
[5] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," *CVPR*, 2006.
[6] M. Berg, O. Cheong, M. Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications (3rd edition)*. Springer-Verlag, 2008.
[7] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," *CVPR*, 2005.
[8] J. Wang, J. Yang, F. Lv, T. Huang, and Y. Gong, "Locality-constraind linear coding for image classification," *CVPR*, 2010.
[9] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *IJCV*, 2001.
[10] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *PAMI*, 2007.
[11] J. Vogel and B. Schiele, "Semantic modeling of natural scenes for content based image retrieval," *IJCV*, vol. 72, no. 2, pp. 133–157, 2007.
[12] S. N. Parizi, J. Oberlin, and P. Felzenszwalb, "Reconfigurable models for scene recognition," *CVPR*, 2012.
[13] D. Gokalp and S. Akso, "Scene classication using bag-of-regions representationss," *CVPR*, 2007.
[14] R. Socher, C. Lin, A. Ng, and C. Manning, "Parsing natural scenes and natural language with recursive neural networks," *ICML*, 2011.
[15] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *PAMI*, 2011.
[16] J.Tighe and S. Lazebnik, "Superparsing: scalable nonparametric image parsing with superpixels," *IJCV*, 2013.
[17] J. Tighe and S. Lazebnik, "Finding things: image parsing with regions and per-exemplar detectors," *CVPR*, 2013.
[18] P. Isola and C. Liu, "Scene collaging: analysis and synthesis of natural images with semantic layers," *ICCV*, 2013.

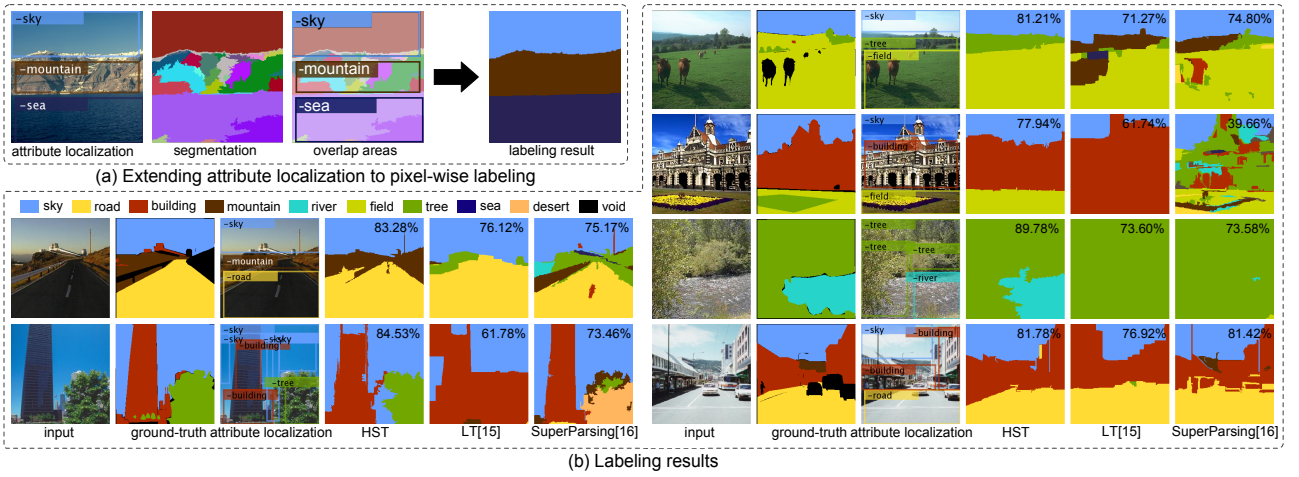


Fig. 13: Labeling results. (a) The procedure of extending the HST attribute localization to pixel-wise labeling. (b) Labeling results comparison. The six columns are source images, ground-truths, attribute localizations and labeling results of our method, the labeling results of label transfer (LT) [15] and SuperParsing [16], respectively. The numbers on the top right corner indicate the per-pixel labeling accuracy for each image.

TABLE 7: The scene labeling performance

	bridge	building	desert	field	mountain	river	road	rock	sand	sea	sky	tree	MAP
LT [15] (%)	9.71	91.54	5.03	44.25	58.54	15.94	87.93	1.02	5.41	75.83	88.35	72.05	46.30
SuperParsing [16] (%)	12.19	87.90	35.31	29.34	52.71	16.58	78.05	5.76	7.92	55.21	90.30	80.61	45.99
HST (%)	16.78	89.05	5.66	55.06	72.56	11.45	87.30	2.23	11.91	70.79	90.86	70.43	**48.67**

[19] M. Rubinstein, C. Liu, and W. T. Freeman, "Annotation propagation in large image databases via dense image correspondence," *ECCV*, 2012.
[20] J. Zhu, T. Wu, S. Zhu, X. Yang, and W. Zhang, "Learning reconfigurable scene representation by tangram model," *WACV*, 2012.
[21] S. Wang, Y. Wang, and S. C. Zhu, "Hierarchical space tiling in scene modeling," *ACCV*, 2012.
[22] S. Wang, J. Joo, Y. Wang, and S. C. Zhu, "Weakly supervised learning for attribute localization in outdoor scenes," *CVPR*, 2013.
[23] L. Li, E. P. X. S. Hao, and L. Fei-Fei, "Object bank: a high-level image representation for scene classification and semantic feature sparsification," *NIPS*, 2010.
[24] A. Farhadi, I. Endres, D. Hoiem, and D. A. Forsyth, "Describing objects by their attributes," *CVPR*, 2009.
[25] X. Yu and Y. Aloimonos, "Attribute-based transfer learning for object categorization with zero or one training example," *ECCV*, 2010.
[26] G. Wang and D. Forsyth, "Joint learning of visual attributes, object classes and visual saliency," *ICCV*, 2009.
[27] A. Gupta and L. S. Davis, "Beyond nouns: exploiting prepositions and comparative adjectives for learning visual classifiers," *ECCV*, 2008.
[28] P. Duygulu, K. Barnard, J. Freitas, and D. Forsyth, "Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary," *ECCV*, 2002.
[29] J. Li, R. Socher, and L. Fei-Fei, "Towards total scene understanding: classication, annotation and segmentation in an automatic framework," *CVPR*, 2009.
[30] D. Parikh and K. Grauman, "Relative attributes," *ICCV*, 2011.
[31] C. Lampert, M. Blaschko, and T. Hofmann, "Beyond sliding windows: object localization by efficient subwindow search," *CVPR*, 2008.
[32] T. Berg and A. Berg, "Automatic attribute discovery and characterization from noisy web images," *ECCV*, 2010.
[33] R. Datta, W. Ge, J. Li, and J. Wang, "Toward bridging the annotation retrieval gap in image search by a generative modeling approach," *ACM Multimedia*, 2006.
[34] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *IJCV*, 2004.
[35] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, 1967.
[36] R. E. Burkard, M. D. Amico, and S. Martello, *Assignment Problems*. SIAM Philadelphia, 2009.
[37] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg, "Babytalk: understanding and generating simple image descriptions," *CVPR*, 2011.
[38] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky, "Describing visual scenes using transformed objects and parts," *IJCV*, 2008.
[39] V. Ordonez, G. Kulkarni, and T. Berg, "Im2text: describing images using 1 million captioned photographs," *NIPS*, 2011.
[40] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," *CVPR*, 2010.
[41] G. Salton and M. J. McGill, *Introduction to modern information retrieval*. McGraw-Hill, 1986.

Shuo Wang received her Ph.D. degree in computer science from Peking University (China) in 2015 and received her bachelor degree in digital media technology from Zhejiang University (China) in 2009. Her research interests are computer vision and statistical modeling.

Yizhou Wang is a professor of the Computer Science Department at Peking University (PKU), China. He received his Ph.D. in computer science from University of California at Los Angeles (UCLA) in 2005. He was a Research Staff of the Palo Alto Research Center (Xerox-PARC) from 2005 to 2008. His research interests include computer vision, statistical modeling and learning.

Song-Chun Zhu received a Ph.D. degree from Harvard University in 1996. He is currently a professor of Statistics and Computer Science at UCLA. He received a number of honors, including the J.K.Aggarwal prize from the Int'l Association of Pattern Recognition in 2008, the David Marr Prize in 2003, twice Marr Prize honorary nominations in 1999 and 2007, He received the Sloan Fellowship in 2001, a US NSF Career Award in 2001, and an US ONR Young Investigator Award in 2001. He received the Helmholtz Test-of-time award in ICCV 2013, and he is a Fellow of IEEE since 2011.