Supplementary material of "Learning Hierarchical Space Tiling for Scene Modeling, Parsing and Attribute Tagging"

Shuo Wang[1], Yizhou Wang[1], and Song-Chun Zhu[2]

[1]Department of EECS, Peking University, Beijing, China, 100871
{shuowang, Yizhou.Wang}@pku.edu.cn
[2]Departments of Statistics and Computer Science, University of California, Los Angeles, USA, 90095
sczhu@ucla.edu

1 Derivation of the parameter estimation

In Section 3.2, we estimate the parameter Θ by MLE, which takes the derivative of $L(\Theta)$ (Eq.15) *w.r.t.* $\theta(v|v_i)$ and sets it to zero.

$$\begin{aligned}\frac{\partial L(\Theta)}{\partial \theta(v_i|v)} &= \sum_{m=1}^{M} \frac{1}{p(I_m;\Theta)} \sum_{pt_m,k} \frac{\partial p(I_m, pt_m, C_m^k;\Theta)}{\partial \theta(v_i|v)} - \alpha_v \\ &= \sum_{m=1}^{M} \sum_{pt_m,k} \frac{p(I_m, pt_m, C_m^k;\Theta)}{p(I_m;\Theta)} \frac{\partial \log p(I_m, pt_m, C_m^k;\Theta)}{\partial \theta(v_i|v)} - \alpha_v \\ &= \sum_{m=1}^{M} \sum_{pt_m,k} p(pt_m|C_m^k;\Theta) p(C_m^k|I_m) \left(\frac{\partial \log p(I_m|C_m^k)}{\partial \theta(v_i|v)} + \frac{\partial \log p(C_m^k|pt_m)}{\partial \theta(v_i|v)} + \frac{\partial \log p(pt_m;\Theta)}{\partial \theta(v_i|v)} \right) - \alpha_v\end{aligned} \tag{1}$$

According to Eq.9, Eq.12 and Eq.11, we have $\frac{\partial \log p(I_m|C_m^k)}{\partial \theta(v_i|v)} = 0$, $\frac{\partial \log p(C_m^k|pt_m)}{\partial \theta(v_i|v)} = 0$ and $\frac{\partial \log p(pt_m;\Theta)}{\partial \theta(v_i|v)} = \frac{1}{\theta(v_i|v)}$. Thus the derivation becomes

$$\frac{\partial L(\Theta)}{\partial \theta(v_i|v)} = \frac{1}{\theta(v_i|v)} \sum_{m=1}^{M} \sum_{pt_m,k} p(pt_m|C_m^k;\Theta) p(C_m^k|I_m) - \alpha_v \tag{2}$$

Set the derivative to zero, we will get

$$\begin{aligned}\theta(v_i|v) =& \frac{1}{\alpha_v} \sum_{m=1}^{M} \sum_{pt_m,k} p(pt_m|C_m^k;\Theta) p(C_m^k|I_m) \\ \approx& \frac{1}{\alpha_v} \sum_{m=1}^{M} \max_{pt_m,k} p(pt_m|C_m^k;\Theta) p(C_m^k|I_m)\end{aligned} \tag{3}$$

We assume $p(I_m)$ is uniform, then $p(C_m^k|I_m) = p(I_m|C_m^k)p(C_m^k)$. Set $p(C_m^k) = \frac{1}{|k|}$, the above formulation becomes

$$\theta(v_i|v) = \frac{1}{\alpha_v} \sum_{m=1}^{M} p(pt_m|C_m^k;\Theta) p(I_m|C_m^k) \tag{4}$$

where (pt_m, C_m^k) are the optimal parse tree and segmented layer inferred from HST-geo (the E-step in Section 3.2) and a_v is the normalization term.

Fig. 1. Evaluating the contribution of each energy term in scene attribute localization. The 1st column shows the attribute localization results using all the terms. The 2nd to 5th columns show the results of disabling each of the term one by one. The last column shows the ground-truth.

2 Model Analysis

2.1 Energy term contribution

In this section we evaluate the contribution of each energy term in our model. In Eq.19 in the paper, we define the HST energy function as

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

where the first term $E^{OR} = -\ln \theta(v_i|v)$ (Eq.11) encodes the compositional prior of scene structures learned from the HST-geo. The second term $E^n = -\ln \Phi(\hat{a}^n, v)$ (Eq.20) accounts for the association between the noun attributes and the terminal nodes (scene parts). The third term $E^{adj} = -\ln p(\hat{a}^{adj}|\hat{a}^n) = -\ln \frac{\sum_{m=1}^{M} \mathbb{1}[a^n \in A_m^n] \mathbb{1}\left[a^{adj} \in A_m^{adj}\right]}{\sum_{m=1}^{M} \mathbb{1}[a^n \in A_m^n]}$ (Eq.21) models the compatibility between the nouns and the adjectives. The fourth term $E^T = -\ln p(\hat{a}|I(v))$ (Eq.22) is an attribute specific data term, where $p(\hat{a}|I(v)) = \frac{\exp\{F(I(v),\hat{a})\}}{\sum_{a'} \exp\{F(I(v),a')\}}$ and $F(\cdot, a)$ returns a score of classifying terminal node v as attribute a. The last term E^A enforces an inferred attributes to be within the set of attributes from the image text description A_m. It is 0, if an inferred attributes $\hat{a} \in A_m$; and ∞, otherwise. Notice that in testing, we only take images as input (without text description), thus the constrained term of E^A is simply abandoned during testing.

To analysis the contribution of each energy term, we take SceneAtt dataset as the test bed, and compare the attribute recognition and localization performance when disabling the energy terms one by one. As shown in Fig.1 and Table 1, we get the following observation.

(i) About E^{OR}: This compositional prior of scene structures regulates the model complexity in explaining scene structures. As shown in the second column of Fig.1, without considering such prior, the learning and inference result in more

TABLE 1
Analysis of energy term.

	with all terms	disable E^{OR}	disable E^{n}	disable E^{adj}	disable E^{T}
Recognition MAP (%)	72.17	**76.14**	66.82	72.16	11.16
Localization MAP (%)	**51.20**	49.24	47.39	51.09	3.57

fragmental scene parts. Consequently, these parts derive more attributes for an image, and this can increase the chance to hit the image text description, hence, improve the attribute recognition precision (76.14% when disabling E^{OR} vs 72.17% when using it). However, it can break the integrity of the compositional scene structures and harm the attribute localization accuracy (49.24% versus 51.20%).

(ii) About E^{n}: This association between noun attributes and scene parts encodes the spatial prior of objects, *e.g.,* grassland has high probability appearing at the lower part of an image. As shown in the third column of Fig.1, without using this term, those attributes with similar appearances, such as "snowy field" and "dust-haze sky", "tower and waterfall", may get confused, hence, degrade both the attribute recognition rate (66.82% versus 72.17%) and attribute localization accuracy (47.39% versus 51.20%).

(iii) About E^{adj}: Considering the compatibility between nouns and adjectives has marginal impact on the attribute recognition (0.01% increase) and localization (0.11% increase). This is because some scene parts and appearance can be loosely coupled due to the object appearance variations or occlusions, etc. Notice that the computation of E^{adj} is trivial.

(iv) About E^{T}: When turning off this attribute specific data term, the inference is independent of input images. Thus for any image, as shown in the fifth column in Fig.1, the HST will output the same configuration with same attribute assignment only according to the prior without considering the data.

2.2 Different initializations

When learning of HST, we initialize HST-geo and HST-att individually. In both initializations, we set the parameters to follow the uniform distribution.

(i) We clarified the initialization of HST-geo in Section 3.2. *"Θ is set to be uniform as initialization"*. Formally,

$$\theta(v_i|v) = \frac{1}{|Ch(v)|}, \forall v \in V^{OR}, v_i \in Ch(v)$$

(ii) We clarified the initialization of HST-att in Section 3.3. *"So we initialize ϕ_m by turning on all possible assignments,* i.e., $\phi_m(a, v) = 1, \forall (a, v) \in A_m^n \times V^T(pt_m^*), m = 1, ..., M$." *"At initialization, $p(a|I(v))$ is set to be uniform."*.

In Section 3, we adopt an EM-like method to learn the HST, which includes the E-step of inferring an optimal parse tree for each training sample and the M-step of updating the HST parameters. In the M-step, we use a Viterbi algorithm to approximate the HST parameters based on optimal parse trees instead of all parse trees. Thus our learning method cannot guarantee the global optimality and will be affected by different initializations.

For comparison, we (i) randomly initialize the HST-geo and uniformly initialize HST-att, (ii) randomly initialize the HST-att and uniformly initialize HST-geo, and (iii) initialize both the HST-geo and HST-att randomly. We show the performance changes in attribute recognition as well as in attribute localization in Table 2 and Fig.2.

(i) The random initialization of HST-geo increases the attribute recognition precision (75.80% versus 72.17%). We observe that the random initialization of HST-geo always learns a larger scene part dictionary than the uniform initialization. Thus, as shown in the second column of Fig.2, it usually proposes more attributes candidates at inference. However, this damages the attribute localization accuracy (50.73% versus 51.20%). (ii) As is shown in the third column of Fig.2, although the random HST-att initialization decreases both the performance of attribute recognition and localization, the changes are not very big. (iii) As is shown in the fourth column of Fig.2, when randomly initializing both the HST-geo and HST-att, we can get even more attribute proposals, hence, a better attribute recognition precision, but the attribute localization accuracy is decreased.

TABLE 2
Analysis of different intializations.

	Uniform Init. Both	Random Init. HST-geo	Random Init. HST-att	Random Init. Both
Recognition MAP (%)	72.17	75.80	68.91	**77.37**
Localization MAP (%)	**51.20**	50.73	47.29	48.48

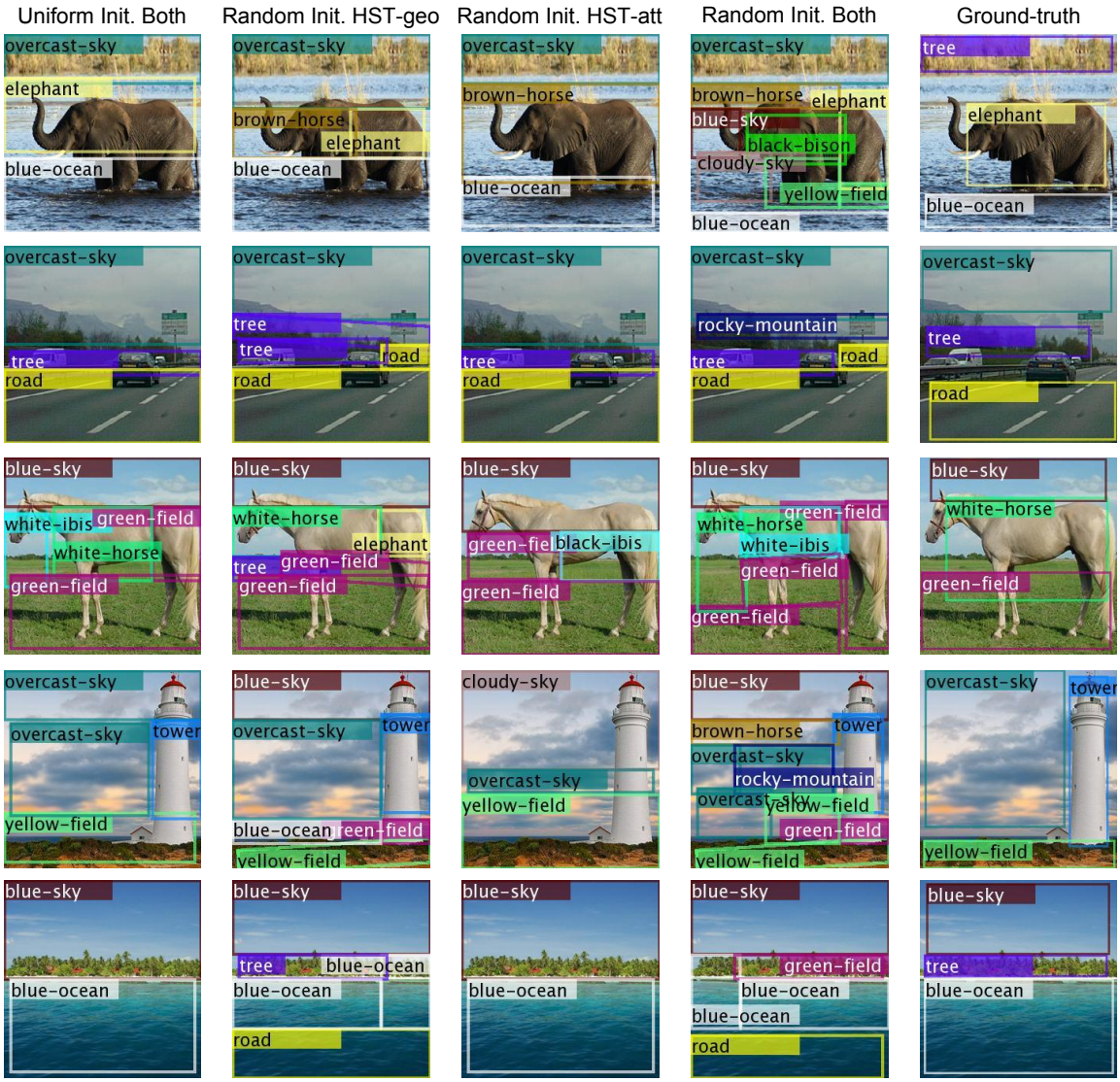


Fig. 2. Comparisons of different initializations. The 1st column shows the attribute localization results with uniform initialization. The 2nd to 4th columns show the results when randomly initializing (i) the HST-geo, (ii) the HST-att and (iii) both the HST-geo and HST-att. The last column shows the ground-truth.

2.3 Weakly supervise vs. fully supervise

In practice, the annotation task of localizing objects may not be a heavy burden for a small dataset. However, when dealing with large data on the Internet, say millions of images, whose descriptive text might be easy to get (*e.g.,* from surrounding text of an image on a webpage), whereas, the annotation task can be very intensive.

Compared with the weakly supervised training, which has **72.17**% and **51.20**% for attribute recognition and localization respectively, we run the experiments with known regions, *i.e.,* fully supervised learning, and get **74.04**% for attribute recognition and **56.19**% for attribute localization. This shows that the proposed weakly supervised learning method does not hurt learning the model much.

3 Supplementary Experiment Results

3.1 Analysis of representation efficiency

In section 4.2, we quantitatively compare the representation efficiency of HST-geo by the rate distortion curve which is defined as the coding error *w.r.t.* the coding length. Because of the space limit, Fig.8 in the paper only shows the rate distortion curves for 4 scene categories in LMO dataset. Here, as a supplement, we show the entire comparison results in Fig.3.

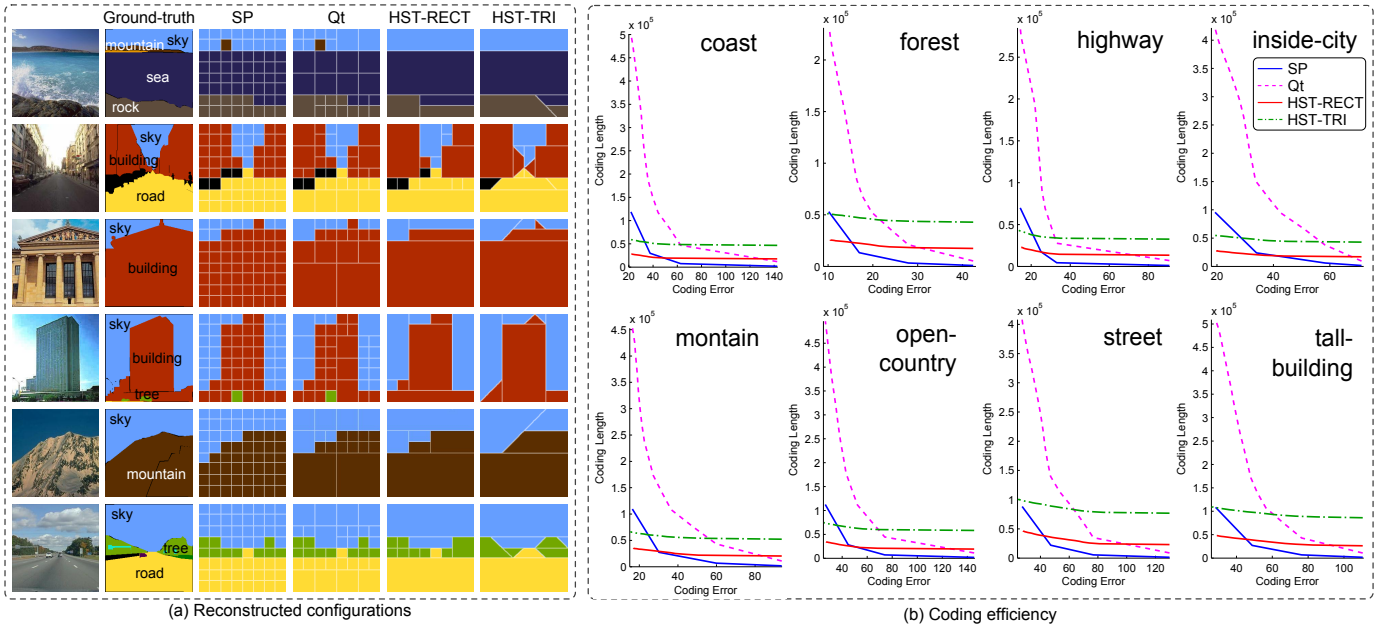


Fig. 3. Efficiency of representation. (a) Given the annotated label maps in the 2nd column, we reconstruct the label maps by Spatial Pyramid (SP), Quadtree(Qt), HST-geo with squares and rectangles (HST-RECT) and HST-geo with triangles and trapezoids (HST-TRI) methods in the 3rd - 6th columns respectively. (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.

3.2 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 (Fig.4), according to the posterior probability. We utilize these templates for feature extraction and discriminative training. As shown in the left of Fig.5, firstly, the appearance descriptors such as "SIFT" and "color moment" are extracted from all training images. Secondly, K-means clustering is applied to quantize the appearance descriptors into codewords with $W_{\text{txt}} = 200$ and $W_{\text{clr}} = 50$ for texture and color respectively. Thirdly, for each template, we align it to the training images by scaling, and then collect a histogram of detected appearance codewords inside each region of the template and concatenate them into a *template histogram vector*. The template histogram vectors of each template are then connected together into a long feature vector named *scene configuration appearance descriptor (SCAD)*. Finally, we train multi-class scene classifies using the SCADs as input data. The classifier is support vector machines (SVMs) using one-versus-all mode. In testing, we extract SCADs from an input image, and the scene category label is the one whose classifier has the highest response.

With the same data split, we compare with five methods: (i) a holistic "Gist" feature based method [1], (ii) a BoW based method [2], (iii) spatial pyramid matching (SPM) [3], (iv) an extension of SPM named locality-constrained linear coding (LLC) [4], and (iv) the Tangram model [5]. We run their released source codes and report the results in the right table in Fig.5. The average precision (AP) of the proposed model is 91.71% which outperforms the others. We believe that it is the scene configurations that improve the classification performance. The BoW method ignores the spatial layout information, while the Gist, SPM and LLC extract features from rigid image partitions to account for scene structures. Although the Tangram model utilizes more flexible image partitions, their pre-defined dictionary is not able to account for the compositional priors of the local structure of the scenes. Compared with those methods, our model allows more flexible configurations, and it also incorporates the scene spatial layout so as to improve the classification performance.

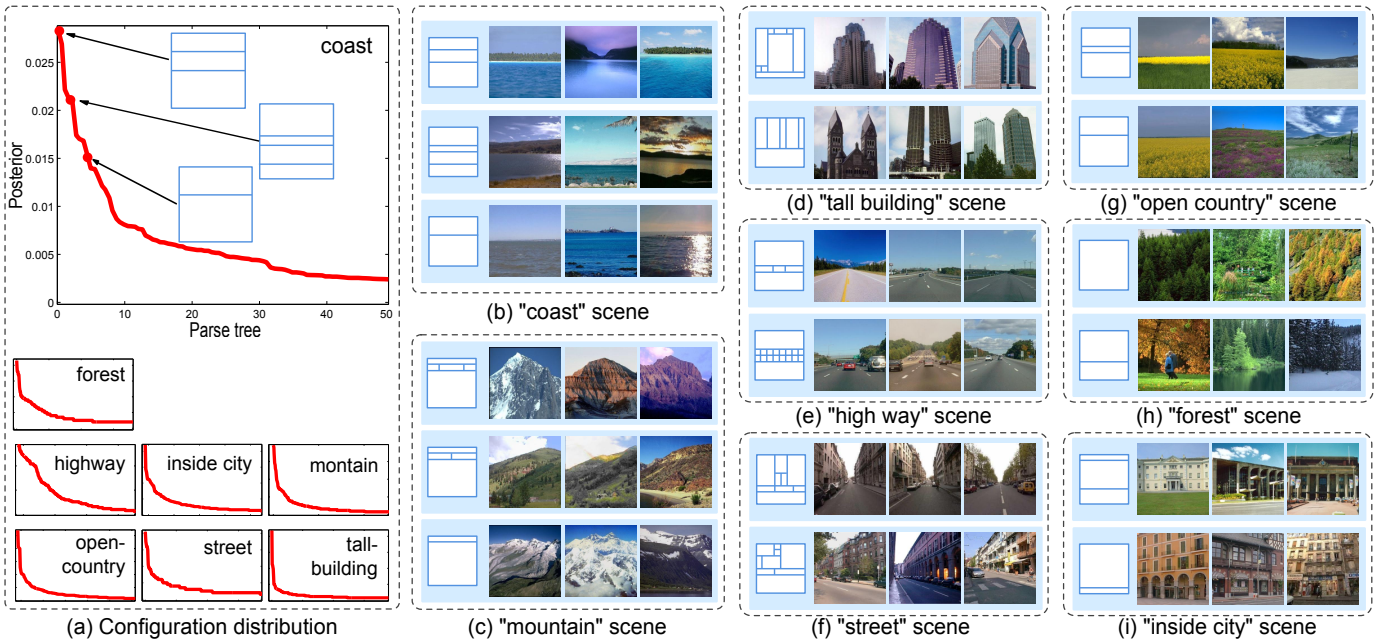
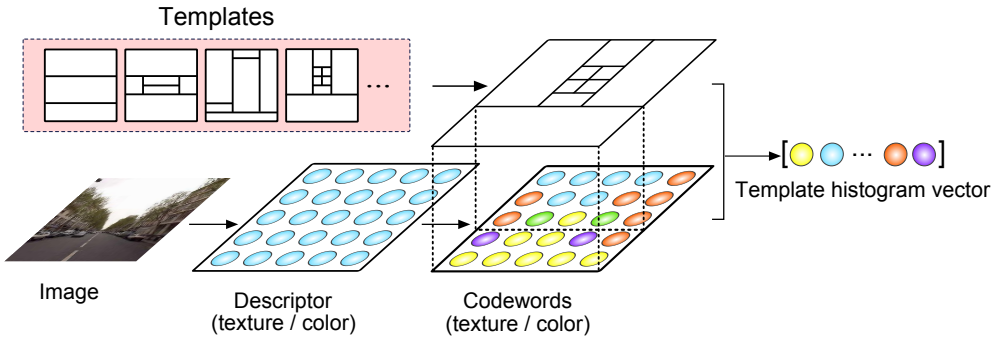


Fig. 4. Most frequent configurations for scene categories. (a) The posterior probability distribution ranked in decreasing order for each scene category, where the horizontal axis is the index of parse tree and vertical axis is the posterior probability. (b)-(i) The categorical typical configurations for each scene category.



Methods	AP(%)
Gist [1]	72.15
BoW [2]	84.57
SPM [3]	84.92
LLC [4]	87.97
Tangram [5]	86.07
Ours	**91.71**

Fig. 5. The pipeline of feature extraction using typical configurations as templates (left) and the scene classification performance (right)

3.3 Attribute recognition and localization

In Section 4.4 and 4.5, we show the mean average precision (MAP) of attribute recognition and attribute localization for scene images. As a supplement, here we show the average precision (AP) of each attribute in Fig.6 and Fig.7.

References

[1] A. Oliva and A. Torralba, "Modeling the shape of the scene: a holistic representation of the spatial envelope," *IJCV*, 2001.
[2] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," *CVPR*, 2005.
[3] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," *CVPR*, 2006.
[4] J. Wang, J. Yang, F. Lv, T. Huang, and Y. Gong, "Locality-constraind linear coding for image classification," *CVPR*, 2010.
[5] J. Zhu, T. Wu, S. Zhu, X. Yang, and W. Zhang, "Learning reconfigurable scene representation by tangram model," *WACV*, 2012.

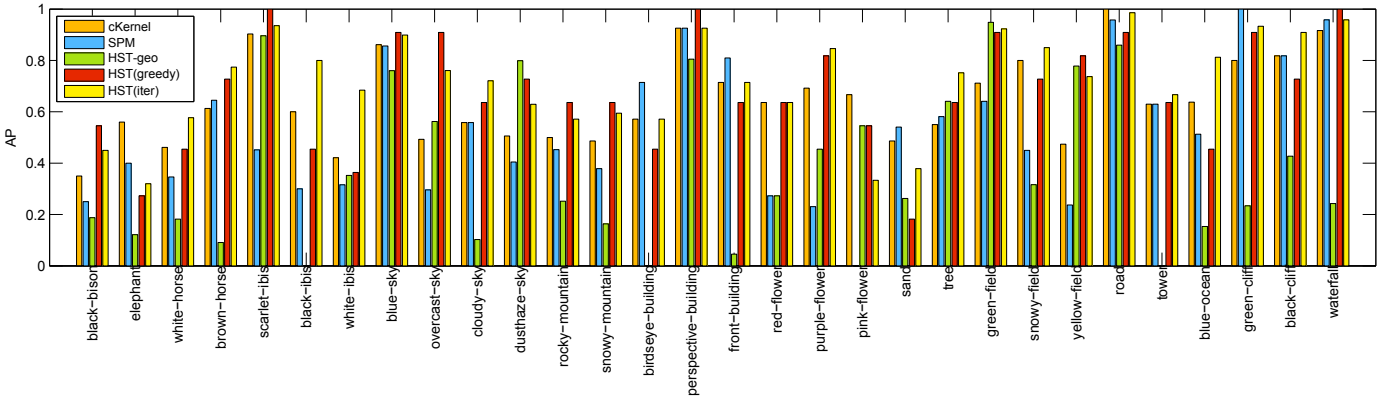


Fig. 6. Average precision of attribute recognition.

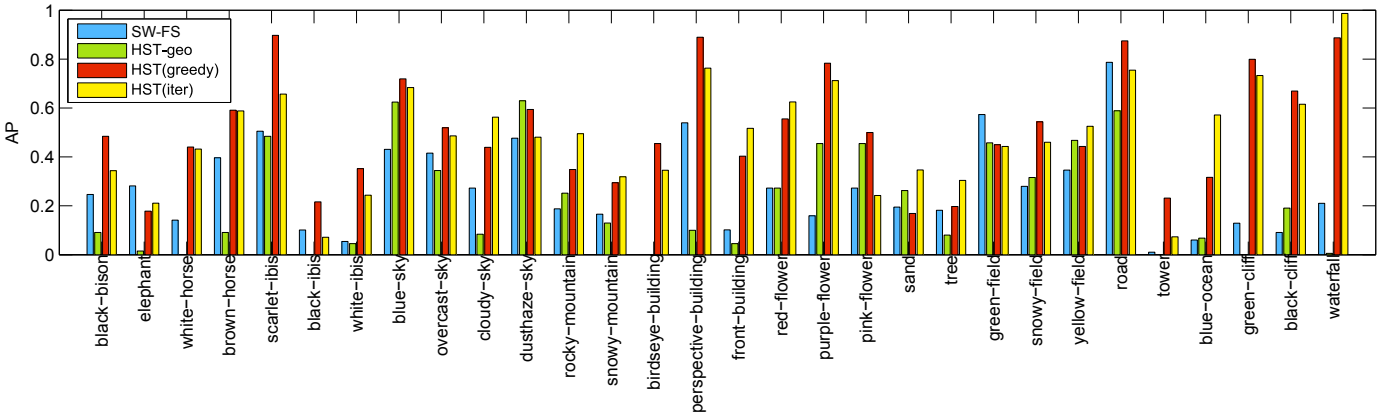


Fig. 7. Average precision of attribute localization.