Learning Reconfigurable Scene Representation by Tangram Model

Jun Zhu [*,†], Tianfu Wu [†,‡], Song-Chun Zhu [†,‡], Xiaokang Yang [*], Wenjun Zhang [*]
[*]Institute of Image Communication and Information Processing, Shanghai Jiao Tong University
[†]Lotus Hill Institute for Computer Vision and Information Science
[‡]Department of Statistics, University of California, Los Angeles
zhujun.lhi@gmail.com, {tfwu,sczhu}@stat.ucla.edu, {xkyang,zhangwenjun}@sjtu.edu.cn

Abstract

This paper proposes a method to learn reconfigurable and sparse scene representation in the joint space of spatial configuration and appearance in a principled way. We call it the tangram model, which has three properties: (1) Unlike fixed structure of the spatial pyramid widely used in the literature, we propose a compositional shape dictionary organized in an And-Or directed acyclic graph (AOG) to quantize the space of spatial configurations. (2) The shape primitives (called tans) in the dictionary can be described by using any "off-the-shelf" appearance features according to different tasks. (3) A dynamic programming (DP) algorithm is utilized to learn the globally optimal parse tree in the joint space of spatial configuration and appearance. We demonstrate the tangram model in both a generative learning formulation and a discriminative matching kernel. In experiments, we show that the tangram model is capable of capturing meaningful spatial configurations as well as appearance for various scene categories, and achieves state-of-the-art classification performance on the LSP 15-class scene dataset and the MIT 67-class indoor scene dataset.

1. Introduction

1.1. Motivation

Learning a good representation is essential in object and scene recognition. In the recent literature, compositional hierarchical models [4] have made progress in the object level, e.g. the deformable part-based model [3] and the image grammar [16]. The success lies in that they are capable of learning reconfigurable representation to account for structural and appearance variations.

By contrast, on scene categorization task, most work [5, 8, 14, 6] use a fixed spatial pyramid which is a quadtree representation for images (see Fig. 1 (a)), and then rely on rich appearance features for improving performance. To go beyond the fixed spatial pyramid structure, we propose a *tangram model* to address the two issues as follows:

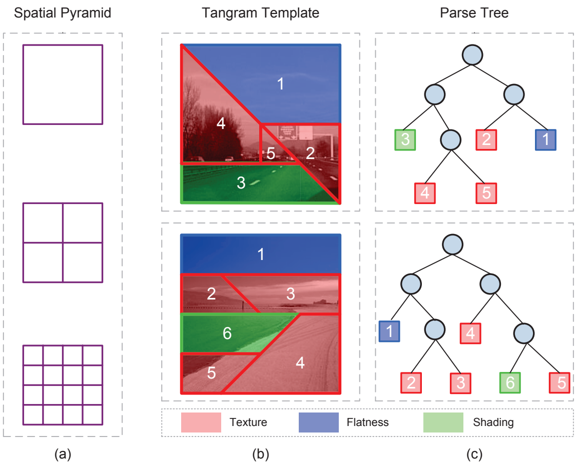


Figure 1. Illustration on the tangram model. (a) The spatial pyramid is a fixed quadtree of partitions on image lattice. (b) The tangram templates for two different scene categories (highway and coast), by jointly capturing meaningful spatial configuration and appearance prototypes (e.g. texture, flatness and shading visual patterns). (c) The reconfigurable representation of parse trees adaptive to different tangram templates shown in (b).

(i) *Compactness*. It entails a sparse representation on spatial configurations. As illustrated in the first row of Fig. 1 (b) and (c), the tangram model for a highway scene consists of five instances of different types of shape primitives, which are enough to capture the spatial configuration in a compact yet meaningful way (see the parse tree).

(ii) *Adaptivity*. It accounts for the reconfigurability of models capable of being adapted to various scene categories or large intra-class variations (i.e., different subcategories), as the tangram model for a coast scene shown in the second row of Fig. 1 (b) and (c).

In this paper, the term of "tangram" is borrowed from an ancient invention from China, which has seven pieces packed in a square, and can express thousands of object configurations. The contributions of this paper are summarized as follows: (1) A dictionary of compositional shape primitives (called *tans*) organized into an *And-Or directed acyclic Graph* (AOG), with expressive power to quantize the configuration space; (2) A reconfigurable *parse tree*, which is an instantiation of AOG, to explicitly represent the spatial configuration and appearance for a scene category; (3) A dynamic programming (DP) algorithm capable of searching the globally optimal tangram parse tree in learning; (4) A tangram matching kernel for facilitating discriminative learning in a kernel-based framework.

The experimental results show the advantage of proposed tangram model with respect to the fixed spatial pyramid: (1) It learns explicit and meaningful tangram templates for scene configurations through parse trees, which can be used in constructing compact and informative feature representation to facilitate discrimination. (2) It outperforms the spatial pyramid counterpart on scene categorization, and achieves state-of-the-art classification performance on two benchmark scene datasets (i.e. LSP 15-class scene dataset [5] and MIT 67-class indoor scene dataset [8]).

1.2. Related Works

In the literature of scene recognition, related methods can be divided into two categories: (1) The holistic methods use low-level global color and texture histograms [11] or the ”gist” features [7] to classify coarse level scene categories, e.g. outdoor versus indoor, and city versus countryside. (2) The methods based on local image features (e.g. SIFT) build an intermediate representation to perform scene categorization, such as probabilistic Latent Semantic Analysis (pLSA) [9], latent Dirichlet analysis (LDA) [2], spatial pyramid matching [5] and sparse coding [14]. Besides, high-level semantic information is also investigated in scene representation [12, 6].

Our model is related to the hybrid image template (HiT) [10], which learns explicit templates for object recognition, but differs from it in two aspects: (1) The primitives. Instead of using the sketch features (i.e. Gabor wavelets) for representing object shape, we adopt region-based hybrid features for describing appearance of tans; (2) The learning algorithm. In [10], the HiT is learned by a greedy shared matching pursuit algorithm, while the tangram model adopts DP algorithm to achieve the globally optimal configuration. Moreover, the proposed tangram matching k-ernel is more flexible than spatial pyramid matching [5], which uses fixed pyramid as the spatial representation rather than reconfigurable structure of parse trees in our method and then becomes a special case of the tangram matching kernel.

The remainder of this paper is organized as follows: In Sec. 2, we introduce a compositional tan dictionary organized into an AOG. Then, a reconfigurable parse tree as well as the tangram template are presented in Sec. 3. We present the generative learning formulation and the tangram matching kernel in Sec. 4 and Sec. 5 respectively, followed by a series of experiments in Sec. 6. Finally, we conclude this paper in Sec. 7.

2. Quantizing Spatial Configuration Space

2.1. Tiling Image Lattice to Form Shape Primitives

Denote by Λ the image lattice. We first partition Λ into a grid of $n_c = w_c \times h_c$ cells. For each cell in the grid, it is further decomposed into two triangular tiles in two alternative ways (in diagonal or back-diagonal direction). Fig. 2 (a) illustrates the tiling of image lattice into a 4×4 grid.

To quantize spatial configurations, it asks for a potentially over-complete dictionary of tans with various shape types, scales and locations on Λ. In this paper, a *tan* is defined as a polygon composed of several non-overlapping triangular tiles, and its size is defined by the number of its triangular constituents (i.e., how many triangular tiles it is composed of, and the maximum value the size can take is $2n_c$). There are only four types of triangles as primitives which are at the bottom layer in the AOG, shown in Fig. 2 (b) and (d), and the number of tans increases exponentially with their size if there were not constraints on their shape, which is avoided by introducing compositional rules in the AOG (see Sec. 2.2.1).

2.2. A Layered Tan Dictionary

The tan dictionary is a layered collection of tan primitives with various sizes. The layer index, denoted by l, of a tan is defined by its size. In this paper, ”layer” is used only to imply the relative size of a tan with respect to that of the smallest triangular primitives, not the actual layer (or depth) of a tan in the AOG built later on.

Given the image lattice Λ with n_c cells, a tan dictionary, denoted by Δ, is defined as the union of L ($L = 2n_c$ in our case) subsets: $\Delta = \bigcup_{l=1}^{L} \Delta^{(l)}$, where $\Delta^{(l)}$ denotes the subset of tans at the l^{th} layer. For $\Delta^{(l)}$, it consists of N_l tans $\{B_{(l,i)} \mid i = 1, 2, \cdots, N_l\}$.

Besides, one tan can produce a series of different copies (called *tan instances*) through placing it onto different positions on the $w_c \times h_c$ grid of Λ. For each tan $B_{(l,i)}$, we denote its instances by $\{B_{(l,i,j)} \mid j = 1, 2, \cdots, J_{(l,i)}\}$, where each tan instance $B_{(l,i,j)}$ is associated with domain $\Lambda_{(l,i,j)} \subseteq \Lambda$. The tans define conceptual shape of polygonal ones, and the instances, linking to the image data, are their instantiations when placed on Λ.

For example, Fig. 2 (c) illustrates a 32-layer tan dictionary. We can observe that there are four types of triangle primitives as the tans in the 1^{st} layer, and the most top

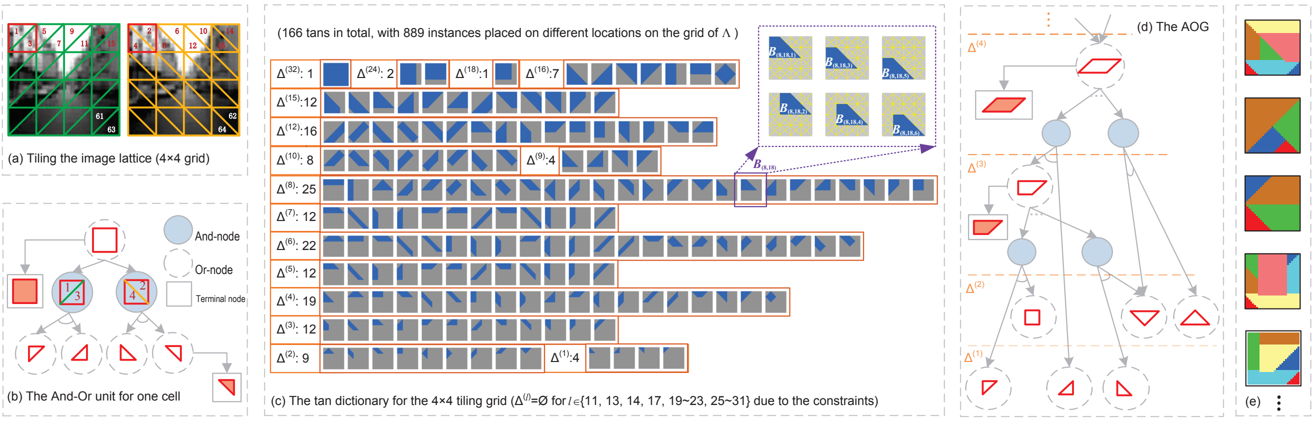

(a) Tiling the image lattice (4×4 grid)
(b) The And-Or unit for one cell
And-node
Or-node
Terminal node
(166 tans in total, with 889 instances placed on different locations on the grid of Λ)
Δ(32): 1
Δ(24): 2
Δ(18):1
Δ(16):7
Δ(15):12
Δ(12):16
Δ(10): 8
Δ(9):4
Δ(8): 25
Δ(7): 12
Δ(6): 22
Δ(5): 12
Δ(4): 19
Δ(3): 12
Δ(2): 9
Δ(1):4
B(8,18)
B(8,18,1)
B(8,18,2)
B(8,18,3)
B(8,18,4)
B(8,18,5)
B(8,18,6)
(c) The tan dictionary for the 4×4 tiling grid (Δ(l)=Ø for l∈{11, 13, 14, 17, 19~23, 25~31} due to the constraints)
(d) The AOG
Δ(4)
Δ(3)
Δ(2)
Δ(1)
(e)

Figure 2. (a) Tiling the image lattice by triangles. (b) Illustration of an And-Or unit, where a rectangular cell at the 2^{nd} layer can be decomposed into two triangular primitives in two alternative ways. (c) Illustration on a 32-layer tan dictionary. It only shows one instance for each tan for clarity. (d) Illustration on organizing the tan dictionary into the AOG, where only a portion of the AOG is shown for clarity. (e) Examples on tangram configurations sampled from the AOG of 32-layer tan dictionary. (Best viewed in color with magnification)

(32^{th}) layer has only one tan (also the instance) such that $\Lambda_{(32,1,1)} = \Lambda$. In addition, it is shown in the right top corner of Fig. 2 (c) that the tan $B_{(8,18)}$ has 6 instances with different positions on the cell grid of image lattice.

2.2.1 Organizing The Tan Dictionary into AOG

Motivated by the image grammar model [16], we propose a way of recursive shape composition to bottom-up construct the tan dictionary Δ, which is organized into an associated AOG, denoted by Υ_Δ. The AOG is defined as a directed acyclic graph, where *And*-nodes and *Or*-nodes are introduced to respectively represent the composition from a set of tans to a larger one (e.g. composing two triangular tiles to a square shown in Fig. 2 (b)), and the alternative ways for composition (e.g. the two different ways of composing two triangular tiles to a square in Fig. 2 (b)).

Starting from the 1^{st} layer (i.e. $\Delta^{(1)}$ shown in Fig. 2 (c)), a valid tan is generated by composing the ones at layers below with all of the following three rules satisfied:

(i) we relax the valid tans to be one of three shape types: triangle, trapezoid and parallelogram. It accounts for non-rectangular shape of regions appeared in complex scene configurations, while avoiding combinatorial explosion at higher layers.

(ii) The size of each tan in the AOG should not be larger than that of Λ (i.e. $2n_c$).

(iii) By allowing deep hierarchical structure in building Υ_Δ, we only apply the binary production rule to keep the graph structure tractable.

As illustrated in Fig. 2 (b) and (d), one tan may be alternatively generated by different ways of composing two ones at layers below. Consequently, it results in an *And-Or unit* for each tan $B_{(l,i)}$: $\{v^T_{(l,i)}, v^{Or}_{(l,i)}, \{v^{And}_{(l,i),o}\}_{o=1}^{O_{(l,i)}}\}$, where $v^T_{(l,i)}$, $v^{Or}_{(l,i)}$ and $v^{And}_{(l,i),o}$ are terminal node, Or-node and And-node, respectively. Concretely, the terminal node $v^T_{(l,i)}$ is namely $B_{(l,i)}$. The And-node $v^{And}_{(l,i),o}$ represents that it can be composed by two child tans at layers below. The Or-node $v^{Or}_{(l,i)}$ represents that it can either directly terminate into $v^T_{(l,i)}$ or further alternatively decompose into two child tans, in one of $O_{(l,i)}$ different ways. It is associated with a state variable $\omega_{(l,i)} \in \{0, 1, 2, \cdots, O_{(l,i)}\}$ to indicate the selection of its child node. $\omega_{(l,i)}$ taking the value of $1 \leq o \leq O_{(l,i)}$ indicates that $B_{(l,i)}$ is decomposed into two child tans according to the And-node $v^{And}_{(l,i),o}$, while $\omega_{(l,i)} = 0$ implies that it selects $v^T_{(l,i)}$ and the partition is ended.

Thus, the AOG Υ_Δ is constituted by the And-Or units, to organize the tans generated in Δ, as illustrated in Fig. 2 (d). Besides, as the correspondence between a tan and its instances, there is also an And-Or unit built for each tan instance, which inherits all the And-Or compositionality for corresponding tan from Υ_Δ. Similarly, it produces an isomorphic AOG, denoted by Υ'_Δ, to organize all the tan instances for Δ.

Actually, the top-layer tan $B_{(L,1)}$ in $\Delta^{(L)}$ defines the root node for Υ_Δ. This suggests a post-processing operation to prune the tans which are not involved in the path of composing it to the end. Moreover, there could be no valid tans available at some layers in the dictionary, due to that it cannot find any two tans at layers below to compose a valid one according to the compositional rules. E.g., for a 32-layer tan dictionary, there is no tans available obtained at the layers of $l \in \{\{11, 13, 14, 17\} \cup [19, 23] \cup [25, 31]\}$, which are ignored to be not shown in Fig. 2 (c).

3. The Tangram Model

3.1. The Reconfigurable Parse Tree

In this paper, the tangram model is defined through a reconfigurable *parse tree* of the AOG $\Upsilon^{'}_{\Delta}$. The *parse tree*, denoted by *Pt*, is defined as a binary tree, involving a set of non-terminal *pass-by nodes* V_N^{Pt} and a set of terminal *leaf nodes* V_T^{Pt}.

We first define a recursive operation, denoted by $\mathbf{PARSE}(B_{(l,i,j)}; \Upsilon^{'}_{\Delta})$, on parsing a tan instance $B_{(l,i,j)}$ as follows: (1) Starting from its Or-node, select one of the child nodes according to its state $\omega_{(l,i,j)}$; (2) If an And-node is selected, join $B_{(l,i,j)}$ into V_N^{Pt} and call $\mathbf{PARSE}()$ to each of its child tans; (3) If reaching the terminal node, join $B_{(l,i,j)}$ into V_T^{Pt} and stop traveling in $\Upsilon^{'}_{\Delta}$.

Then, *Pt* is built through this recursive dynamic starting from the top-layer tan in Δ. It is a kind of instantiation on the AOG $\Upsilon^{'}_{\Delta}$, according to the assignments of its state variables at its encountered Or-nodes. In *Pt*, the pass-by nodes specify intermediate split process in the hierarchy, while the leaf nodes partition image lattice to form a spatial configuration. Fig. 1 (c) illustrates two examples of parse trees for different scene configurations.

3.2. Representing Scene by Tangram Template

The tangram template, denoted by *Tgm*, is defined as a set of non-overlapping attributed tan instances specified by the leaf nodes in *Pt*. That is

$$\mathit{Tgm} = \{(B_k, \Lambda_k, \rho_k) \mid k = 1, 2, \cdots, |V_T^{Pt}|\}, \qquad (1)$$
$$\Lambda_{\mathit{Tgm}} = \cup_{k=1}^{|V_T^{Pt}|} \Lambda_k \subseteq \Lambda \text{ and } \Lambda_i \cap \Lambda_j = \emptyset \ (\forall i \neq j),$$

where each selected tan B_k [1], associated with domain Λ_k and appearance model ρ_k, corresponds to a leaf node of parse tree *Pt*. $|V_T^{Pt}|$ denotes the total number of leaf nodes selected in *Pt*. As shown in Fig. 1 (b) and (c), the tangram template explicitly represents scene configuration as well as the appearance for each tan through the collapse of a parse tree. The reconfigurability of parse trees accounts for being adaptive to encode diverse configurations compactly.

3.3. Appearance Model for A Tan

Besides the parse tree specifies spatial configuration, we specify an appearance model for each tan selected in *Tgm*.

For a tan B_k, the appearance model ρ_k is represented by a two-tuple as $\rho_k = (\psi_k, h_k)$, with appearance type ψ_k and prototype parameter h_k. Let I_{Λ_k} and $H(I_{\Lambda_k}; \psi_k)$ denote the image patch on Λ_k and the mapping function on feature extraction for type ψ_k, respectively. Generally, H can use any "off-the-shelf" visual descriptor, e.g. HOG [1], Gist [7] or 'bag of visual words' (BOV) features [2]. In this paper, similar as [10], three types of visual patterns (i.e. *flatness*, *shading* and *texture*, respectively denoted by $'flat'$, $'shad'$ and $'txtr'$) are adopted, competing each other to explain the image appearance for a tan. Besides, the BOV descriptor is also used as appearance feature in experiments.

For $\psi_k \in \{flat, shad, txtr\}$, we use very simple intensity gradients to compute the appearance features. The flatness and shading features are computed by the average magnitude of the 1^{st} and 2^{nd} gradients, respectively. The texture feature is computed by the statistics on gradient mass over various orientations, as the HOG descriptor [1].

Moreover, we define the feature response $r(I_{\Lambda_k}; \rho_k)$ for appearance model ρ_k. As in [10], for scalar feature type $\psi_k \in \{flat, shad\}$, the response is computed by $r(I_{\Lambda_k}) = \mathcal{S}(H(I_{\Lambda_k}; \psi_k); \tau)$, where $\mathcal{S}()$ is a sigmoid function

$$\mathcal{S}(x; \tau) = 2 - 2[1 + exp(\frac{-2x}{\tau})]^{-1}, \qquad (2)$$

with saturation parameter τ. It transforms original feature H into $[0, 1]$, and obtains a large value for flat or shading patches (i.e. with small 1^{st} or 2^{nd} gradient magnitude). For the vector-wise features such as HOG and BOV, we can compute responses by any valid similarity measurement between H and the prototype parameter h_k. h_k is a vector with the same dimension as H. In this paper, we adopt the histogram intersection kernel (HIK) [13], which is an effective but simple measurement for histogram features.

$$r(I_{\Lambda_k}) = \sum_{b=1}^{\mathcal{B}} \min(H^{(b)}(I_{\Lambda_k}), h_k^{(b)}), \qquad (3)$$

where $H^{(b)}$ and $\mathcal{B}$ refer to the value of the b^{th} bin and the dimension for H respectively.

4. Generative Learning of The Tangram Model

In this section, we present a generative formulation, generalized from the *information projection principle* [10], to learn the tangram model as explicit template.

4.1. Statistical Formulation and Log-Linear Model

Let $D^+ = \{I_1^+, I_2^+, \cdots, I_N^+\}$ denote a set of N positive images, which are assumed sampled from an underlying probability model $f(I)$ to be learned, for a scene category. Besides, we introduce a background model $q(I)$ to represent general natural image statistics, which is characterized by an image set $D^- = \{I_1^-, I_2^-, \cdots, I_M^-\}$ consisting of the training images collected from various scene categories. Our objective is to learn a model probability $p(I; \mathit{Tgm})$ of *Tgm* from D^+, to approach $f(I)$ starting from $q(I)$. To simplify notation in following discussion, let $r_k = r(I_{\Lambda_k}; \rho_k)$ for B_k. Similarly, we denote its feature responses on D^+ and D^- by $\{r_{k,n}^+\}_{n=1}^N$ and $\{r_{k,m}^-\}_{m=1}^M$, respectively.

[1] The subscript k is a linear index of tan instance, which is interchangeably used with the triple-tuple index (l, i, j) for notation simplicity.

For the *Tgm* of a scene category, the model space $\Omega_p(\mathit{Tgm})$ is defined as:

$$\Omega_p(\mathit{Tgm}) = \{\, p(I;\mathit{Tgm}) \,|\, E_p[r_k] = E_f[r_k], \forall k \,\}, \quad (4)$$

where $E_p[r_k] = E_f[r_k]$ accounts for that the model expectation of feature response for each selected tan is expected to match the empirical statistics. According to the maximum entropy principle that $\hat{p} = \arg\min_{p \in \Omega_p(\mathit{Tgm})} K(p \,||\, q)$, a factorized log-linear model [10] is obtained, due to non-overlapping tan instances in *Tgm*:

$$\hat{p}(I;\mathit{Tgm}) = q(I) \prod_{k=1}^{|V_T^{Pt}|} [\frac{1}{z_k} \exp\{\lambda_k r_k\}], \quad (5)$$

where λ_k and z_k refer to the parameters of weight and normalizing factor for B_k in *Tgm*, respectively.

4.2. Learning by Maximizing Information Gain

Similar to [10], we define regularized information gain of *Tgm* as the learning objective. That is

$$\mathcal{IG}(\mathit{Tgm}) = KL(f \,||\, q) - KL(f \,||\, \hat{p}) - \mathcal{M}(\mathit{Tgm}) \quad (6)$$
$$= \textstyle\sum_{k=1}^{|V_T^{Pt}|} \{\lambda_k E_f[r_k] - \log z_k - \frac{1}{2}\beta {\lambda_k}^2\} - \alpha |V_T^{Pt}|.$$

where $[KL(f \,||\, q) - KL(f \,||\, \hat{p})]$ is an information-theoretical measurement on the improvement of the learned model $\hat{p}(I;\mathit{Tgm})$ approaching $f(I)$ relative to $q(I)$. $\mathcal{M}(\mathit{Tgm})$ refers to the regularization term on model complexity, in which β and α denote the trade-off parameters on shrinking the weight λ_k and punishing large number of tans selected, respectively. Thus, learning the optimal tangram template *Tgm** (as well as the model parameters λ_k^* and z_k^*) for D^+ is achieved by maximizing its information gain $\mathcal{IG}(\mathit{Tgm})$ over the solution space of parse trees.

As in [10], for each candidate tan instance B_k, we estimate the prototype parameter h_k for its appearance model by averaging the feature descriptors $H(I_{\Lambda_k})$ over positive images from D^+. Then, we estimate λ_k and z_k for B_k. Through solving $\frac{\partial \mathcal{IG}}{\partial \lambda_k} = 0$, the optimum values are given by

$$(\lambda_k^*, z_k^*) :\; E_f[r_k] - E_{\hat{p}}[r_k] = \beta \lambda_k. \quad (7)$$

We simply calculate the empirical expectation $E_f[r_k]$ by the mean response value on positive images. That is $E_f[r_k] \approx \frac{1}{N}\sum_{n=1}^{N} r_{k,n}^{+}$. The term of $E_{\hat{p}}[r_k]$ is approximately estimated using the feature responses on D^-: $E_{\hat{p}}[r_k] = E_q[\frac{1}{z_k}\exp(\lambda_k r_k)] \approx \frac{1}{M}\sum_{m=1}^{M}[\frac{1}{z_k}\exp(\lambda_k r_{k,m}^{-})]$, where $z_k \approx \frac{1}{M}\sum_{m=1}^{M}\exp(\lambda_k r_{k,m}^{-})$. On computation, we can solve Equ. (7) by Newton method or line search [10].

Then, we obtain the information gain for a tan B_k:

$$g_k = \max(\lambda_k^* E_f[r_k] - \log z_k^* - \frac{1}{2}\beta {\lambda_k^*}^2 - \alpha, 0) \quad (8)$$

where $\max(\cdot, 0)$ implies that the tans giving negative information gain values will be not involved in *Tgm**. After that, a DP algorithm is called to find *Tgm**.

4.3. The DP Algorithm

The recursive And-Or structure with deep hierarchy is capable of representing a huge space of spatial configurations, each of which is specified by a parse tree instantiated from the AOG. Although exponential number of parse trees (as well as tangram templates) need to be considered in the solution space, the direct acyclic characteristic of AOG makes the globally optimal solution can be efficiently searched through a DP algorithm.

For a node v in Υ'_Δ, let g_v and $Ch(v)$ denote its information gain and the set of child nodes, respectively. Before starting the DP algorithm, we assume the gain of each terminal node is computed by Equ. (8). Then, in this DP algorithm, it propagates their gains to And-nodes (by the *SUM* operation: $\mathbf{SUM}(v^{And}) = \sum_{u \in Ch(v^{And})} g_u$) and Or-nodes (by the *MAX* operation: $\mathbf{MAX}(v^{Or}) = \max_{u \in Ch(v^{Or})} g_u$, with recording the optimal state ω^* at the same time) through a bottom-up step. After that, the globally optimal parse tree *Pt**, which is defined as the one with maximum gain value at the root node, can be top-down retrieved by calling the parsing operation $\mathbf{PARSE}()$ on the top-layer tan instance. The DP algorithm is summarized in Alg. 1.

Algorithm 1: The DP Algorithm for Searching Globally Optimal Parse Tree

```
Input: AOG $\Upsilon'_\Delta$, information gain on terminal nodes:
       $\{g_{v^T_{(l,i,j)}} \,|\, \forall l, i, j\}$
Output: the optimal parse tree Pt*
Step I: bottom-up propagating information gain:
foreach level $l = 1$ to $L$ do
    foreach tan $i = 1$ to $N_l$ and $j = 1$ to $J_{(l,i)}$ do
        foreach And node $o = 1$ to $O_{(l,i)}$ do
            Let $g_{v^{And}_{(l,i,j),o}} = \mathbf{SUM}(v^{And}_{(l,i,j),o})$;
        end
        Let $g_{v^{Or}_{(l,i,j)}} = \mathbf{MAX}(v^{Or}_{(l,i,j)}), \omega^*_{(l,i,j)} \rightarrow \Upsilon'_\Delta$;
    end
end
Step II: top-down parsing by depth first search:
$\mathbf{PARSE}(B_{(L,1,1)}; \Upsilon'_\Delta)$.
```

4.4. Image Representation by Tangram Model

On the scene categorization task, the learned tangram templates can be used to construct a new feature representation for an image I. Given $\mathcal{C}$ learned tangrams[2], denoted

[2] In this paper, we simply learn one tangram for each scene category.

by $\{Tgm^{(c)} \mid c = 1, 2, \cdots, \mathcal{C}\}$, we can use one of the following two ways to construct image representation: (1) The set of all the tangrams defines a $\mathcal{C}$-dimensional feature space, each dimension of which corresponds to a *tangram score* $\phi_c(I) = \sum_{k=1}^{K^c}\{\lambda_k^c r_k^c - \log z_k^c\}$. λ_k^c, z_k^c and r_k^c are respectively the learned model parameters and appearance feature response for the k^{th} tan instance (K^c in total) selected in *Tgm*$^{(c)}$. (2) The responses of all the tan instances are pooled from each tangram, and concatenated into a N_{tan}-dimensional ($N_{tan} = \sum_{c=1}^{\mathcal{C}} K_c$) vector. Based on the new representation, we train a linear classifier (e.g., linear SVM or logistic regression) for scene classification.

Thus, the learned tangram templates are actually utilized in a non-linear dimension reduction method, specified by either a few of meaningful scene configurations or informative tans. It can achieve a compact low-dimensional representation to facilitate discriminative classification.

5. The Tangram Matching Kernel

In this section, another learning framework based on the tangram model is presented. Motivated by spatial pyramid matching (SPM) [5] in scene and object recognition, we propose a matching kernel based on the tangram AOG, called by the tangram matching kernel (TMK), for discriminative learning. In [5], a fixed spatial pyramid is adopted to subdivide the image lattice into increasingly finer cells (shown in Fig.1 (a)), and a weighted histogram intersection is accumulated from the finest layer to coarser ones. It deduces an effective similarity measurement to approximate the spatial correspondences between two images, by imposing the spatial constraints over orderless image feature representation (e.g., BOV). However, the subdivision scheme of fixed pyramid may prevent it from exploiting more flexible spatial configurations, which are adaptive to individual images, to further boost discrimination. Taking advantage of the AOG tan dictionary with extremely high expressive power on spatial configurations, we hypothesize it can benefit scene recognition from considering more sophisticated configuration by our tangram model.

Given a pair of images, we first compute the matching score s_{v^T} for each terminal node in Υ'_{Δ} as the intersection between their histogram features (i.e. the matched features on this tan instance) by using Equ. (3). Then, the matched features are bottom-up accumulated from the 1^{st} layer to the top one: the matching score of an And-node v^{And} is computed by accumulating the ones of its child tans, plus the weighted increment of the intersection value which corresponds to the matched features newly found when relaxing the spatial constraint imposed by the And-node. That is

$$s_{v^{And}} = \sum_{u^{Or}} s_{u^{Or}} + \frac{1}{l}\left(s_{v^T} - \sum_{u^T} s_{u^T}\right) \qquad (9)$$

where $u^{Or} \in Ch(v^{And})$ and $u^T \in Ch'(v^{And})$ respectively denote the Or-node and the terminal one of a child tan instance for v^{And}. similar as [5], we simply set the weight of matched features newly found as $\frac{1}{l}$, which is inverse to the layer index of v^{And}, implying that the features matched in larger tans are more penalized due to the relaxation of spatial constraints. For an Or-node v^{Or}, the matching score $s_{v^{Or}}$ is obtained as follows: if it lies in the 1^{st} layer, we directly set it by the one of terminal node $s_{v^{Or}} = s_{v^T}$, otherwise we use either the *MAX_OR* operation

$$\mathbf{MAX\_OR}(v^{Or}) = \max_{u^{And} \in Ch(v^{Or})} s_{u^{And}} \qquad (10)$$

or the *MEAN_OR* operation

$$\mathbf{MEAN\_OR}(v^{Or}) = \frac{1}{|Ch(v^{Or})|}\sum_{u^{And}} s_{u^{And}} \qquad (11)$$

to calculate $s_{v^{Or}}$, where $u^{And} \in Ch(v^{Or})$ is a child And-node for v^{Or}. Finally, the value of TMK for these two images is returned by the matching score of root Or-node in Υ'_{Δ}. On implementation, it can be efficiently computed by the DP algorithm in Alg. 1, with slight modification[3].

Based on the *MAX_OR* or *MEAN_OR* TMK, a kernel-based SVM classifier is trained to perform scene categorization. Actually, the support vectors obtained are regarded as a series of reconfigurable scene templates to score the testing image through TMK. Intuitively, the *MAX_OR* TMK adaptively searches a tangram parse tree with the largest accumulated histogram intersection value between the query image and a scene template, and the *MEAN_OR* TMK fuses the matching values found w.r.t. different spatial constraints at the Or-nodes by means of ”marginalizing” them.

6. Experiments

In experiments, we investigate the proposed tangram model in following two aspects: (1) We show results of explicit scene templates learned by the generative formulation introduced in Sec. 4, and demonstrate the advantage of the DP algorithm in learning. (2) We apply it on the task of supervised scene categorization, and evaluate classification performance on two widely-used scene datasets in the literature: the 15-class scene dataset (LSP_15) [5] and the MIT 67-class indoor scene dataset (MIT_Indoor) [8]. The experimental results validate the advantage of the tangram model compared to fixed spatial pyramid representation, by using both linear classification based on the learned tangram templates in Sec. 4 and the kernel-based method in Sec. 5.

6.1. Learning Tangram Templates by DP Algorithm

This experiment shows that the tangram model is capable of jointly discovering spatial configurations and appear-

[3] It requires to displace $\mathbf{SUM}$ by Equ. (9), and to substitute $\mathbf{MAX}$ by $\mathbf{MAX\_OR}$ or $\mathbf{MEAN\_OR}$.

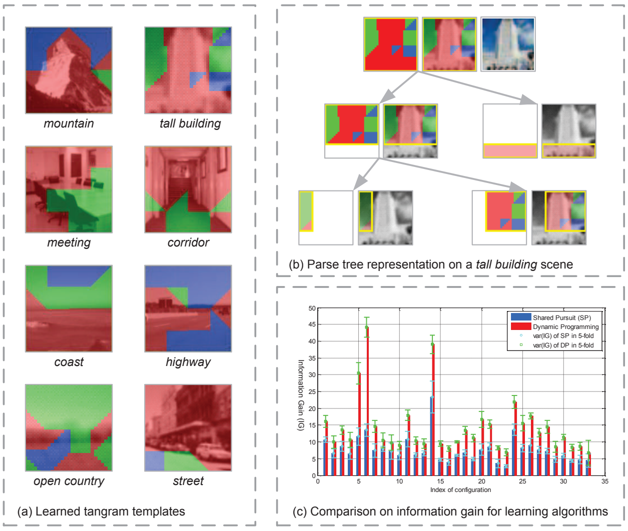
ance prototypes, which are represented by tangram templates adaptive to different scene categories.

The data: To demonstrate that the tangram model can be learned as explicit templates, we use roughly aligned images in this experiment, similar as [10]. We select images of 10 scene categories, consisting of 6 outdoor scene categories (*coast*, *highway*, *mountain*, *opencountry*, *street*, *tall building*) and 4 indoor ones (*bedroom*, *store*, *meeting*, *corridor*), collected from the MIT 8-class scene dataset [7], MIT_Indoor dataset and the LHI scene dataset [15]. For each category, there are 120 to 250 images, manually divided into 3 to 4 different configurations (33 in total).

Appearance features: The hybrid features (i.e. flatness, shading and texture) in Sec. 3.3 are adopted to describe the appearance for terminal tan instances. When determining the information gain for a candidate tan discussed in Sec. 4.2, we first compute the information gain for each of the three appearance types respectively, and then choose the one giving maximum information gain value (i.e. best explaining the image patch by certain visual pattern).

For each scene configuration, we randomly select 20 images to learn a tangram parse tree. The experimental results of learned tangram templates are illustrated in Fig. 3. We observe that the tangram model can capture meaningful appearance prototypes as well as the spatial layout of scene configuration in a compact parse tree representation, which makes for the scene categorization task to be demonstrated in Sec. 6.2.

To demonstrate the advantage of the DP algorithm w.r.t. greedy shared matching pursuit algorithm [10] in learning, we compare the values of information gain (Equ. (6)) obtained by these two algorithms. Fig. 3 (c) shows that the DP algorithm, capable of searching globally optimal parse trees, consistently outperforms greedy pursuit in learning tangram templates for various scene configurations.



(a) Learned tangram templates

(b) Parse tree representation on a *tall building* scene

(c) Comparison on information gain for learning algorithms

Figure 3. (a) shows the tangram templates (superposed on a randomly selected training image) learned from 8 different categories, based on the generative formulation introduced in Sec. 4. The red color represents the texture appearance prototypes, blue for flatness and green for shading. (b) The parse tree of a scene configuration from *tall building*. (Only a portion of the parse tree is shown for clarity.) (c) The comparison of the DP algorithm and greedy shared matching pursuit algorithm based on the information gain of learned tangram templates on 33 scene configurations. (Best viewed in color with magnification)

6.2. Evaluation on Scene Categorization

This experiment shows that the classification performance on scene categorization can be improved by utilizing the proposed tangram model, compared to widely used spatial pyramid. Our method achieves state-of-the-art performance on two benchmark scene datasets [5, 8].

The data and protocol: For LSP_15 dataset (It involves 15 natural scene classes, each of which has 200 to 400 images.), we use 100 images for training and the rest for testing following [5]. For MIT_Indoor dataset, we use the same training images (80 samples per class) and testing ones (20 samples per class) experimented in [8]. Following the literature on scene categorization [5, 8, 6], classification performance is evaluated by the average of per-class classification rates, which is calculated as the mean over the diagonal values of confusion matrix.

Appearance features: For fair comparison, We adopt the same appearance features used in [5], i.e. the dense SIFT BOV features with 200 visual words.

First, we evaluate the representation constructed by the learned tangram templates as described in Sec. 4.4. In following discussion, we denote the methods of using the tangram scores and tan responses by *fTgm_s* and *fTgm_r*, respectively. Besides, the tangram template defined in Equ. (1) is actually no more than a flat (single-layered) structure, which only involves the tans at the terminal nodes of a parse tree. According to the observation that it is preferable to use a multi-layered representation [5], we can use a multi-layered tangram template, from which the methods of constructing image representation by tangram scores and tan responses (denoted by *mTgm_s* and *mTgm_r*, respectively) are consistent with the flat version, by including the tans of non-terminal pass-by nodes besides the leaf ones.

Table 1 and 2 compare our methods with fixed spatial pyramid (SP) [5] on LSP_15 dataset, by using linear classifiers such as logistic regression (LR) or linear SVM (SVM). We observe that the representation constructed by *mTgm_r* consistently outperforms the fixed-structure spatial pyramid in each combination of granularity level (i.e. 2 × 2 or 4 × 4 grid) and classifier type, while it requires much lower dimensionality of the feature representation. To be noted, even with extremely low dimensional representation (i.e. only a 15-dimensional vector), the method of *fTgm_s* can

Table 1. Classification rates (%) for Linear classifiers (LSP_15)

	SP	Our method			
		fTgm_s	*mTgm_s*	*fTgm_r*	*mTgm_r*
LR(2×2)	75.0	74.1	73.5	75.4	**76.5**
LR(4×4)	75.8	74.6	73.3	73.4	**76.9**
SVM(2×2)	73.5	72.6	71.8	74.4	**76.5**
SVM(4×4)	74.5	73.0	71.1	73.7	**76.3**

Table 2. Comparison on the dimension of representation

	SP	Our method			
		fTgm_s	*mTgm_s*	*fTgm_r*	*mTgm_r*
2×2	1000	15	15	118	225
4×4	4200	15	15	472	945

Table 3. Classification rates (%) for the TMKs (LSP_15)

	SPM[5]	*OB*[6]	[14]	Our method	
				MAX_OR	*MEAN_OR*
2×2	79.0	-	-	80.5	**81.3**
4×4	81.1	80.9	80.3	**81.6**	81.4

Table 4. Classification rates (%) for the TMKs (MIT_Indoor)

	SPM[5]	*OB*[6]	[8]	Our method	
				MAX_OR	*MEAN_OR*
2×2	37.0	-	-	39.5	**39.7**
4×4	38.3	37.6	26	**42.1**	41.8

obtain comparable classification rate w.r.t. SP (the difference $\leq 1.5\%$). It implies the compactness of our model capable of retaining significant information by using a few number of tangram templates or tan instances.

Besides the methods using the generatively learned tangrams as non-linear feature dimension reduction, we evaluate the tangram model in a kernel-based classification framework as introduced in Sec. 5, and compare it with state-of-the-art methods. From table 3 and 4, we observe that the proposed TMKs achieve better classification performance than SPM in both 2×2 and 4×4 grid, which supports our motivation that more flexible configuration as well as inducing the Or-nodes would facilitate scene recognition. Particularly, our methods outperform state of the art in a large margin (about 4%) on $\mathrm{MIT\_Indoor}$ dataset. It may be caused by the fact that the indoor scene categories involve more complicated configuration variations than natural outdoor scenes, asking for a more sophisticated way to explore scene configurations as the tangram does.

Computational time: The average runtime for DP algorithm and computing the information gain of candidate tan instances is about 1.3 ms and 2.7 s respectively, with matlab and C-mex implementation on a standard PC (Intel Core2 2.0 GHz CPU and 4G Byte RAM), after all the appearance features are pre-computed. The average time on evaluating TMK is about 0.4 ms (2×2 grid) and 6.0 ms (4×4 grid).

7. Conclusion

Learning reconfigurable representation is of importance in scene modeling and recognition. This paper presents a tangram model for jointly discovering spatial configuration and appearance by using the AOG and DP algorithm. It goes beyond the widely used spatial pyramid, and obtains state-of-the-art performance on scene categorization.

Acknowledgement This work is mainly done when Jun Zhu is a visiting student at LHI. We thank the support of NSFC (90920009, 61071155, 60932006) and 973 Program (2010CB731401, 2010CB731406).

References

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[2] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005.

[3] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9), 2010.

[4] S. Geman, D. Potter, and Z. Y. Chi. Composition systems. *Quart. Appl. Math*, 60(4):707–736, 2002.

[5] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[6] L.-J. Li, H. Su, E. P. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *NIPS*, 2010.

[7] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[8] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *CVPR*, 2009.

[9] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. V. Gool. Modeling scenes with local descriptors and latent aspects. In *ICCV*, 2005.

[10] Z. Si and S.-C. Zhu. Learning hybrid image template (hit) by information projection. *TPAMI*, Accepted to appear, 2011.

[11] M. Szummer and R. W. Picard. Indoor-outdoor image classification. In *CAIVD*, 1998.

[12] J. Vogel and B. Schiele. Semantic modeling of natural scenes for content-based image retrieval. *IJCV*, 72(2):133–157, 2007.

[13] J. Wu and J. Rehg. Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel. In *ICCV*, 2009.

[14] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

[15] B. Yao, X. Yang, and S.-C. Zhu. Introduction to a large scale general purpose ground truth dataset: methodology, annotation tool, and benchmarks. In *EMMCVPR*, 2007.

[16] S.-C. Zhu and D. Mumford. A stochastic grammar of images. *Found. Trends. Comput. Graph. Vis.*, 2(4):259–362, 2006.