Understanding Image Structure via Hierarchical Shape Parsing

Xianming Liu[†] Rongrong Ji[‡] Changhu Wang[§] Wei Liu[♮] Bineng Zhong[♯] Thomas S. Huang[†]
[†]University of Illinois at Urbana-Champaign [‡]Xiamen University
[§]Microsoft Research [♮]IBM T. J. Watson Research Center [♯]Huaqiao University
{xliu102,t-huang1}@illinois.edu rrji@xmu.edu.cn chw@microsoft.com
weiliu@us.ibm.com bnzhong@hqu.edu.cn

Abstract

Exploring image structure is a long-standing yet important research subject in the computer vision community. In this paper, we focus on understanding image structure inspired by the "simple-to-complex" biological evidence. A hierarchical shape parsing strategy is proposed to partition and organize image components into a hierarchical structure in the scale space. To improve the robustness and flexibility of image representation, we further bundle the image appearances into hierarchical parsing trees. Image descriptions are subsequently constructed by performing a structural pooling, facilitating efficient matching between the parsing trees. We leverage the proposed hierarchical shape parsing to study two exemplar applications including edge scale refinement and unsupervised "objectness" detection. We show competitive parsing performance comparing to the state-of-the-arts in above scenarios with far less proposals, which thus demonstrates the advantage of the proposed parsing scheme.

1. Introduction

Understanding structure of images is one of fundamental challenges in the computer vision community and beyond [26][15][32]. It is commonly agreed in the cognitive research [15] that such structure is hierarchically organized in general, and visual appearances along the structure range from coarse to fine configurations. These evidences result in a multi-scale image representation [16][34].

In this paper, we target at exploring such structure for images, by proposing a novel *hierarchical shape parsing*. By "shape parsing", we mean to detect visual components (such as parts of objects) indicated by shapes, which will be organized into hierarchical structure according to the coarse-to-fine cognitive rule (such as "part of", and "outline-and-details" relations) to generate a multi-scale representation. Different from the image parsing [32][19] and scene parsing [31][9], which aim at finding semantically meaningful labels for pixels, the problem we address in this paper focuses on exploring the structure of visual scenes in an unsupervised manner, instead of inference on particular semantic labels using supervised learning.

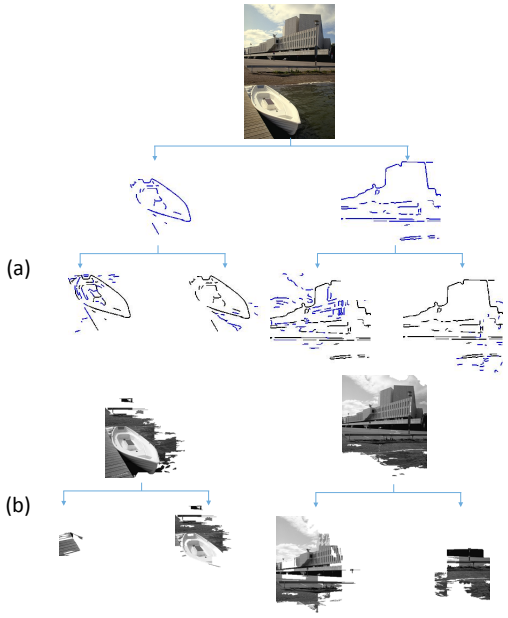


Figure 1. Example of the hierarchical shape parsing. (a) the original image and the hierarchical edge parsing tree. For better visualization, the edge segments of each node are in blue, while the ones of its ancestor nodes are in black. (b) the results after appearance bundling. To illustrate the coarse-to-fine phenomenon, the appearances of child nodes are integrated to the parent node.

Parsing such a structure roles as a novel coarse-to-fine feature representation to simulate the human vision system, *i.e.*, taking the scene outlines in the coarse scale (at higher levels in the tree) and the appearance details in the fine scales (lower levels) [26][15]. To a certain degree, such a simulation is also similar to the "simple-to-complex" biologically inspired feature [29].

Figure 1 shows an example about how we parse the struc-

ture of objects in the hierarchical shape parsing. In Figure 1 (a) the input image is parsed into two parts (*e.g.*, the boat and building) and further into more detailed components recursively, *in a top-down manner*. The parent-children linkage reflects certain "visual ontology" relationships (such as "part of", "co-occurrence"). We also demonstrate results appending regions (in term of superpixels [10]) onto each node, which leads to Figure 1 (b). This type of structure coincides with the coarse-to-fine cognition: the edges of larger scale tend to be the out boundaries while vice versa [23].

Successfully parsing the image structure at low level can benefit a wide variety of computer vision tasks, such as feature designing, scene understanding, object recognition and detection. Three of the benefits are listed here but not limited. First, it is possible to design multiple-level image description, and partial descriptors of objects from the hierarchical structure. Experimental results in this paper also show that the hierarchical shape parsing tree captures the organization in parts-objects-scene. Secondly, instead of searching ROIs (*e.g.*, objects) through sliding window, the hierarchical structure of objects makes it more efficient in object detection, recognition *et al.*, similarly to Selective Search [33] and BING [5]. In our experiment, by searching along the hierarchical parsing tree, we reduce the candidate region number from thousands of in [33, 5] to less than 20, in the meanwhile retain competitive recall in object region proposals. This is critical in object detection algorithms, such as R-CNN [12]. Finally, it provides a way to structural inference of the visual content, by integrating the recently developed structural learning methods.

Related Work. To accomplish this goal, considerable efforts have been made in the past years, among which the structure of image is typically expressed as spatial splitting [20, 2], statistical co-occurrence [30] [39], and more recently convolutional neural network [9].

To this end, approaches such as Spatial Pyramid Matching (SPM) [20], and multiple level wavelets [25] can be categorized as the fixed spatial splitting in the global sense. Although being efficient in practice, it lacks in providing an explanation about the intrinsic scene-object-component relationships. UCM [2] implements a more flexible region integration algorithm based on edges / boundaries, in a bottom-up manner. With the popularity of visual pattern mining and bundling techniques like [38][28][35], bottom-up data-driven structure parsing has also been widely investigated. Their main drawback lies in the lack of global insights at the scene level, while being computationally intensive when extending to higher-order phrases.

The most recent advance in convolutional network [18] also suggests efficient approaches that utilize unsupervised or supervised feature learning to discover the structure between different level of neurons, by incorperating the convlution and max pooling operations on different layers. However, understanding and extracting these hierarcal structures remains a problem.

Uijlings *et al.* propose the method "selective search", by using hierarchical image segmentation to build the image structure, and achieves improvements on objectness proposal, in both recall and efficiency. It is further integrated into object detection algorithms based on Deep Convolutional Nerual Networks, *e.g.*, R-CNN [12], and reports to be significant efficient compared with traditional sliding window approaches. Despite positive, Selective Search uses a heuristic bottom-up integration of super-pixels, which omits the cognitive principle in human vision system. In the meanwhile, thousands of object proposals produced by the algorithm impose great burden on convolutional neural networks and make the object detection computational expensive. Unfortunately, efficiently exploring the image structure, if not impossible, remains a challenging problem.

Inspiration. Studies in scale space [34][16][22] reveal that the hierarchical organization of visual information widely exists in image structure. And different scales will exhibit representation on various level of details (*i.e.*, "coarse-to-fine"). On the other hand, shape (*e.g.*, boundaries, and edges) provides a good indication of objects and components, which plays an important role in human cognition [26]. It has been widely recognized as a fundamental cue towards scene understanding and structural description [17][40][23]. Bruna and Mallat also point out that instead of learning the visual representation, geometric properties such as scale and spatial layout provide plenty of meaningful information on feature representation [4].

Inspired by our previous work on the scale of edges [23], we are motivated to parse hierarchical structure of image components according to their scale distributions and shapes, as the example shown in Figure 1. To further improve the discrimination of parsed visual components, appearances of image regions are embed correspondingly in a statistical way.

As for visual search and matching, human brains recognize objects based on not only visual appearances, but also heavily relying on structure, according to recent advance in cognitive study [7]. With such a structure, we simulate the human cognitive mechanism in visual search as a *conditional matching process*: matchings are formulated as Markov Process, with dependencies defined by the structural "visual ontology" along the hierarchical parsing tree. By simple statistical inference, we derive a *hierarchical structural pooling strategy* to approximate the above process when building region descriptions.

Approach. Our approach starts with building a *Hierarchical Edge Tree* in a top-down manner. Given such a coarse-to-fine shape structure, local regions are further appended onto corresponding tree nodes to increase the discriminative ability. When matching two parsing trees /

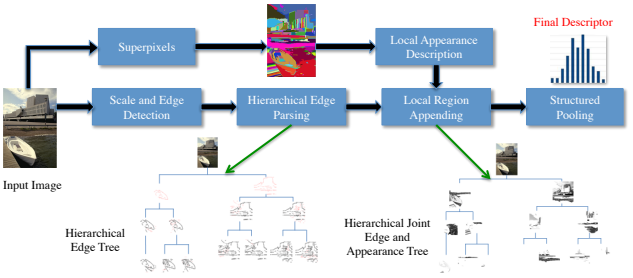


Figure 2. Workflow of the proposed hierarchical shape parsing with appearance pooling scheme.

subtrees, an structural appearance pooling operation is performed based on a *Markov Process*, in which the parent-children dependency is forced. By this pooling operation, the tree hierarchy is encoded together with the appended appearance information, and it avoids the time-consuming recursive subtree alignment schemes in existing works. The whole work flow is shown in Figure 2. In this paper, we show two exemplar applications about our scheme, including unsupervised objectness detection [1, 33]. Quantitative experiments with comparisons to the state-of-the-arts show advantages of our algorithm.

The rest of the paper is organized as following: Section 2 introduces the hierarchical shape parsing. In Section 3 we perform the structured appearance pooling, which is further used in description and matching of object components. We show two exemplar applications in Section 4, and finally conclude the whole paper in Section 5.

2. Hierarchical Shape Parsing

To parse the structure of images, we first introduce the building of the *Hierarchical Edge Tree* in Section 2.1, which decomposes the image structure based on a edge scale detection. Local appearances, *e.g.* superpixels, are subsequently appended to enrich its discriminability in Section 2.2. Both phases are fully unsupervised.

2.1. Hierarchical Edge Tree

Inspired by our results in the edge scale detection [23], as well as previous researches in scale space [16][34], we propose to parse the hierarchical structure of an image by organizing a shape hierarchy. The conclusion that scales of edges are capable to distinguish different levels of details [23], leads to our approach clustering edges varied in both spatial and scales and organizing into a hierarchy.

Edge Scale Detector. The algorithm starts with building the edge scale detector, following the approach in [23]. It detects edges and their scales simultaneously by building an Anisotropic Scale space, upon which a 3D-Harris detector $R(x, y, t)$ is performed to find the maximal edge response over the scale factor t:

$$s(x, y) = \arg\max_t |R(x, y, t)|, R(x, y, t) < 0, \quad (1)$$

where $s(x, y)$ is the detected scale for pixel (x, y). This indicates the edge being salient on the corresponding image resolution. Applying Equation 1 over the Anisotropic Scale space converts the target image into hierarchical organization of shapes with different scales, as shown in Figure 3 (a). We then distill and partition the image based on the "coarse-to-fine" principle. In such a case, the edges of larger scales correspond to the object boundaries, while those of smaller scales are the texture details.

We then denote the boundary of each object as a convex surface $C_i, \forall i \in N$ in the scale space where N is the total number of objects in the scene, as shown in Figure 3 (b). By quantizing along scales in $s_j, j = 1, 2, \ldots, K$, a hierarchical shape representation of the image is derived into a tree structure, as shown in Figure 3 (c), in which the root performs as the global representation, while the subsequent siblings correspond to objects or their components at a finer level.

Parsing as Shape Clustering. To efficiently and effectively distinguish the objects / components C_i from each other as shown in Figure 3 (b), we utilize the spatial information as a separable rule, based on the hypothesis that the edges from the same object or component are more likely to be connected. The adjacency between edge strokes is measured by the smallest Manhattan distance between all the pixels from the two edge segments, *i.e.*,

$$d(s, s') = \min_{i \in s, j \in s'} d(i, j)$$

where s and s' are two edge segments while i and j are pixels on s and s' respectively. This distance works well for various edge / boundary applications to connect adjacent segments [11]. We use clustering to re-group all the edge segments into "connected" shape contours.

More specifically, as shown in Figure 3 (b) and (c), given the detected set of edges $\mathcal{C}$ with scale factors S, we first split S into the K-partition $\mathbb{S} = \{s_1, s_2, \ldots, s_K\}$, and then use spectral clustering [27] to cluster shapes falling in the scale interval $\mathcal{C}|s_k, \forall k \in [1, K]$ into spatially independent components, *i.e.*,

$$\mathcal{C}|s_k = \bigcup_i C_{k,i}, \text{ and } \forall i, j, C_{k,i} \bigcap C_{k,j} = \phi. \quad (2)$$

The above operation can be digested as separating different objects from a given scene, with the consideration of determining the component numbers robustly. In our experiments, to ensure balance and regularize the tree structure, we enforce to use the largest two clusters in all $\mathcal{C}|s_k$, which leads to a binary tree structure.

This operation is conducted in a top-down manner as detailed in Algorithm 1, which results in the *Edge Tree* by connecting each component according to the "belongingness." That is, the node $C_{k,i}$ is assigned to the spatially nearest

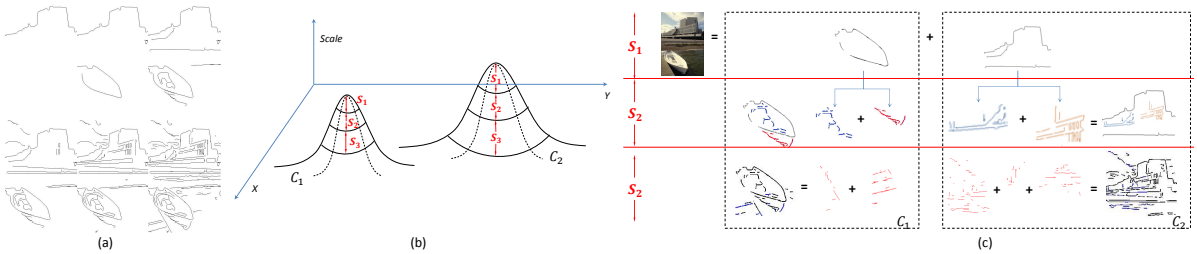


Figure 3. Illustration of building the hierarchical edge tree with spatial clustering: (a) the edges of different scales vary in descriptions of visual contents, while larger scales tend to be coarse and smallers to be the fine details (b) Represent the scale space by convex surfaces, and quantize scales into different tree levels; (c) Motivation and the basic process of spatial clustering.

Algorithm 1: Hierarchical Shape Parsing

Input: Shape collection $\mathcal{C}$ and scale indicator S for image I

Output: Edge Tree T

$T = \Phi$;
Split S into $\{s_1, s_2, ..., s_K\}$;
for k = *1,...K* **do**
  Find $\mathcal{C}|s_k$;
  Perform spectral clustering on $\mathcal{C}|s_k$, into N_k clusters $C_k = \{C_{k,1}, ..., C_{k,N_k}\}$;
  for $C_{k,i} \in C_k$ **do**
    Find the nearest node $C_{k-1,j} \in C_{k-1}$ in T;
    Add $C_{k,i}$ into T as the child node of $C_{k-1,j}$;
  end
end
return T

node $C_{k-1,\cdot}$ as a child. Intuitively, a scene would be decomposed into objects while an object would be decomposed into components. Figure 3 (c) shows an example of hierarchical edge parsing into three levels, where the whole scene is first parsed into objects, *e.g.* "building" and "boat". In the second layer, each object is further parsed into components or parts, *e.g.*, "building" into 'windows" plus "platform". Such a parsing provides a much higher flexibility compared to the fixed splitting such as Spatial Pyramid Matching [20].

2.2. Region Bundling

In the second step, we append the local region appearances onto corresponding nodes in the edge parsing tree to enrich its discriminability. Typically, only the geometric properties of edge segments are ambiguous in object recognition, such as length, curvature and etc. With the appearance of regions enclosed in the edge contours, more robust visual features could be extracted and utilized to describe objects. However, the difficulty lies on determining the most suitable region scale and appending corresponding regions to our edge tree, in the case multiple subtrees are hit.

We adopt a heuristic solution to resolve the challenge. First our algorithm segments the image into superpixels [10]. Then for each node on the edge tree, a convex hull of its edge segments is calculated to claim its "belongings". Finally superpixels are bundled to the node which encloses the superpixel, in a bottom-up direction. This heuristic method works well because: 1) superpixels are those uniform small regions in an image, which are considered being enclosed by edges; and 2) smaller superpixels are registered onto lower level nodes in a bottom-up manner, which solves the "belonging" ambiguity.

3. Structured Appearance Pooling

Bundling the structural information in visual matching provides meaningful discriminations to object detection, texture classification etc [36], especially in complex scenes. This agrees with the hypothesis in human cognitive procedure that not only the appearance but also the spatial and scale structure play important roles in object matching and recognition [7]. The hierarchical shape parsing tree provides an alternative of the coarse-to-fine search strategy.

To fully deploy the structure information in hierarchical shape parsing trees to object matching, we utilize a joint probability formulation to model the matching process. The basic principle is that *two objects or components are matched with higher probability if their appearances are similar and their structures coincide*. In our scenario, it means two tree nodes are matched if they are close in visual feature space and their sub-tree structures agree with each other.

In this section, we first introduce the notations of our formulation, then a simplified conditional probability model for object matching based on Markov Process is illustrated to model the above principle. Finally, this model is relaxed and leads to a structural pooling algorithm to generate tree node representation with such structural information embedded, to further improve computational efficiency and facilitate feature vector indexing.

Notation and Preliminary. For each region bundled on the hierarchical parsing tree, we extract SIFT features [24] that quantilized using Bag-of-Word model (with dictionary size $1,000$), to discribe its appearance. All the notations are

listed in Table 1:

Table 1. Symbols and notations in *Structured Pooling* inference

Variable	Notations
R_i	Node i in the parsing tree
x_i	Feature vector for R_i
T_i	The sub-tree rooted at R_i
X_i	Representation for T_i
$T_{i,k}$ and $R_{i,k}$	Sub-trees / Children of R_i

Tree Matching as Markov Process. The above principle indicates a subtree matching problem. However, direct subtree matching with exhaustive scanning is computational expensive, in this paper we propose a conditional probability inference along tree structures top-down. Since we are aiming at measuring the similarity between objects considering both the appearance and structure, the goal of matching two objects or components is to calculate the joint probability of two sub-tree structures T_q and T_i, where the former is the query object with its structure while the latter is the sample in the target collection. More specifically, given the representation X_q and X_i as well as the structure T_i, the problem can be formualted as:

$$\begin{aligned} p(X_q, X_i|T_i) &= p(X_q, X_i|R_i, \cup T_{i,k}) \\ &= p(X_q, x_i|R_i) \cdot \sum_k [p(X_q, X_{i,k}) p(R_i|T_{i,k})], \end{aligned} \tag{3}$$

where $T_{i,k} \rightarrow R_i \rightarrow T_i$ is a *Markov process*, which indicates the independence in statistical inference. By assuming equal contribution among individual sub-tree, *i.e.*, $p(R_i|T_{i,k}) = \frac{1}{K}$, we have:

$$\begin{aligned} p(X_q, X_i|T_i) &= p(X_q, x_i|R_i) \frac{1}{K} \sum_k p(X_q, X_{i,k}) \\ &= p(X_q, x_i|R_i) \mathbb{E}[p(X_q, X_{i,k})], \end{aligned} \tag{4}$$

which is denoted as *Markov-Structured Matching Rule* thereafter.

Structured Appearance Pooling. The above *Markov Matching Rule* derives a structured appearance pooling, which facilitates the fast representation and matching between hierarchical parsing structure of images. To do this, we assume the current node R_i and the successors $T_{i,k}$ are independent, the Equation 4 can be relaxed as:

$$p(X_q, X_i|T_i) = p(X_q, (x_i \cdot \mathbb{E}[X_{i,k}])), \tag{5}$$

where $X_i = x_i \cdot \mathbb{E}[X_{i,k}]$ indicates the pooling of X_i, which comes from feature x_i of the current node and the average pooling [3] of all its subtrees $\mathbb{E}[X_{i,k}]$[1]. To further simplify the computation, we use the intersection operation instead of the multiplication, with some tolerance of quantization error.

[1]Note that our structured pooling differs from the local descriptor pooling in building spatial pyramid [37][13], which involves more complex hierarchical structure inference and embedding, as well as Markov Process based approximation.

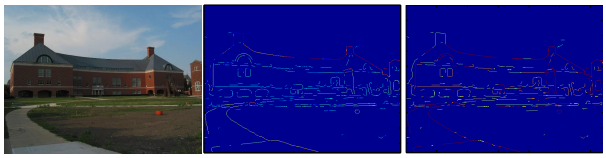


Figure 4. Examples of edge scale detection before and after our edge scale refinement based upon the state-of-the-art edge scale detection scheme in [23]. (a) is the original image, (b) is the detection result using [23], (c) is the proposed scale refinement. Red indicates larger scales while blue means smaller. The refined results are more reasonable by treating the out boundary of the building as uniform and gets the largest scales.

In practice, we build the above appearance representation in a bottom-up manner, and for the structure collection $\mathbb{T} = \{T_i; i = 0, 1, 2, ..., N\}$, a series of descriptors are obtained as $\mathbb{X} = \{X_i; T_i \in \mathbb{T}\}$. The $\mathbb{X}$ describes the visual features of all the parsed scenes, objects and components, based on which the matching can be done for both the whole image, or objects or parts of this image.

Usage of Tree Matching and Structural Pooling: The "*Structural Appearance Pooling*" could be utilized as a visual feature constructor for various computer vision problems, and is adopted in our "objectness" experiment in Section 4.2. We use the structural pooled feature to build index and construct queries.

4. Exemplar Applications

Exploring the hierarchical structure of images has the potential for a wide variety of computer vision applications. Two representative applications are investigated with quantitative evaluations, *i.e.* structural edge scale refinement and unsupervised objectness detection in this paper.

4.1. Structural Edge Scale Refinement

The scale of visual elements, defined as on what level the image details could be perceived by human eyes [16], is a fundamental concept in computer vision. It is important for feature designing, boundary detection and object recognition. In this example, we are trying to see how the proposed hierarchical structure of images can help to refine individual edge scale detections. Moreover, considering the fact that the visual elements of larger scales tend to be object outlines while the smaller ones tend to be inside details, it also provides a validation on how the constructed hierarchical structure coincides with human perception.

The basic assumption is, the edges of the same component should share similar scales. Under this circumstance, the initial edge scale is determined using the method proposed in [23], by searching the local extreme in anisotropic

Table 2. Comparison on the edge scale detection between the proposed method and [21] and [23]

Method	$Accu$	$Accu^{ordered}$
Lindeberg[21]	0.300	0.103
3D-Harris [23]	0.370	0.469
SegPropagation [23]	0.375	0.478
Hierarchical Refinement	0.397	0.485

scale space. Then, our task is to refine these initial detected scales based on the levels of their corresponding nodes in the hierarchical edge tree. More specifically, the edges registered to the same node should be of similar scales. To that effect, given the node N_i in the shape parsing tree T, for each edge $e_j \in N_i$, we perform a *median-filter* to filter out the outlier scales.

We tested the refinement of edge scales on the *Scale of Edge Dataset* [23], where there are 36 natural images with manually labeled edge scales by human beings on 5 different resolutions. The groundtruth of this dataset reflects the levels of scales on which edge segments can be perceived by human beings. We compare our scheme with both the Isotropic approach [21] and 3D-Harris alternatives [23], as shown in Table 2.

In Table 2, two measurements are evaluated: the orderless pair-wise order accuracy $Accu$ and the ordered on $Accu^{ordered}$, same with those used in [23]. From the comparison, it is obvious that the proposed method improves the performance significantly, by adding the object / component scale consistency constraint. The intuition behind is, the structural information (*i.e.*, the Hierarchical Parsing Tree) imposes the consistency of scales within the same visual components, which leads to less wrong detection compared with the original algorithm.

Moreover, the improved performance suggests that the organization of image components in the hierarchical shape parsing tree coincides with human perception: the edges of larger scales labeled by human are arranged on higher levels of the tree. Figure 4 further shows exemplar comparisons before and after our hierarchical tree based refinement. The boundaries of the building (especially the roof) are of larger scales which means they are assigned to high level in the hierarchical shape tree.

4.2. Objectness

Objectness Detection. The second scenario to validate the proposed shape and appearance parsing approach comes from the so-called "objectness" detection task. Objectness [1], as proposed in [1], aims at judging whether a given region contains a generic object, which poses great potential to improve the object detection efficiency, for example Selective Search [33] and BING [5]. The proposed hierarchical parsing tree well fits the objectness detection by looking over the object "structure", that is, objects are more likely to have a common shape, appearance, and structure, as frequently appeared in the reference image corpus, where the "common" is defined by its density in the joint feature space of both the appearance and the structures encoded in shape features.

We use the feature derived from *structural pooling* in Section 3 to describe region descriptions composed by both appearances and structure. More formally speaking, given a collection of images $\mathbb{I} = \{I_n\}_{n \in N}$, where each image I_n, $\mathbb{T}$ denotes the collection of parsed hierarchical structures, and $\mathcal{X}_n = \{X_{n,i}\}$ denotes the descriptors build in Section 3 for image n. Let

$$\mathbb{X} = \cup_{n \in N} \mathcal{X}_n \tag{6}$$

be the collection of structure descriptors for all structures extracted from $\mathbb{I}$. The objectness can be expressed as the density of $\mathbb{X}$ in the feature space, because $\mathcal{X}_n$ exhausts all the existences of possible objects / components.

In practice, instead of estimating the density of $\mathbb{X}$, we perform a K-Means clustering on the feature collection $\mathbb{X}$. Thereafter, the cluster centers $O_k \in \mathcal{O}$ are the potential common structure and appearance template for some object, which in means of a discrete "objectness".

Subsequently, for each node $X_i \in \mathcal{X}$ in the parsing tree of the input image I, where $\mathcal{X}$ is the collection of all the parsed structures, its objectness $o(X_i)$ can be calculated as

$$o(X_i) = p(\mathcal{O}|X_i) \propto 1/\min_{O_k \in \mathcal{O}} \|X_i - O_k\|_2, \tag{7}$$

by which we evaluate the likelihood being an object in terms of its distance to the nearest cluster center (object template).

Experiments and Evaluation. For validation, we use the Pascal VOC 2007. for training and testing, which contains 20 object classes over $4,592$ images. The number of clusters is set to be $K = 100$, considering the number of classes in VOC Pascal 2007 dataset and the total number of nodes generated by our algorithm. To perform the hierarchical parsing, we fixed the depth of binary-tree to be $D = 4$, which results 14 nodes for each image (the root node is not considered as a candidate).

To evaluate the performance of the proposed "Objectness", we use the similar numeric criteria as [1]: the single value evaluation measurement Area Left of the Curve (ALC), which is the area bounded by the Detection Ratio / Signal-To-Noise (DR/STN) curve.

The results of *ALC* are shown in Table 3, compared with several baseline methods, including the saliency based methods [14], feature detection based method using HoG [6], random chance, and variances of the Objectness in [1][2]. We also compare with the all the single-cue methods in [1]: **MS** (Multi-scale Saliency), **CC** (Color Contrast), **ED** (Edge

[2]As the structural objectness by means of hierarchical parsing tree is treated as a low level feature representation, we only compare with the single feature objectness detection.

Table 3. The ALC measurements comparison between random choose (RC), the proposed method, the parsing tree without structural pooling, and salience based method [14], HoG [6], and the methods in [1] (MS, CC, ED, SS)

Method	ALC Score
Saliency [14]	0.004
Random Chance	0.025
HoG [6]	0.035
CC [1]	0.064
ED [1]	0.083
MS [1]	0.137
SS [1]	0.193
Objectness w/o SPooling	0.187
Structured Objectness	0.215

Density), and **SS** (Superpixels Straddling). The comparisons show that:

1) *Saliency*: Due to the intrinsic that our method could be viewed as a structured saliency measurement, it performs much better than all the salience detection methods, *i.e.*, Saliency [14] and MS [1].

2) *Feature*: It outperforms feature-based methods (5 - 8 times more accurate), *i.e.*, HoG [6] (shape and texture) and CC [1] (color), in both of which the structural is missing.

3) *Segmentation and Edges*: Our structural objectness also performs much better than ED (edge features) [1], and achieves competitive performance as SS (Superpixels Straddling) [1]. However, our proposed method is totally unsupervised, while SS is supervised and trained as a two class classification problem (object and non-object).

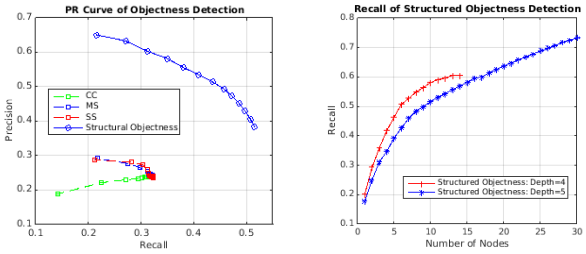
4) *Structural Pooling*: we also tested the performance of the proposed method without using structural pooling to derive features, which performs worse as shown in Table 3. The main reason is that without structural constraints, the non-object regions will have chance to be miss-ranked in the top of returned results and reduce the precision.

Table 4. The number of windows (candidate regions) used in [1] and the structural objectness

	HoG[6]	Objectness [1]
#Win	1,000	1,000
	Selective Search[33]	Proposed
#Win	395	14

Another advantage of our structural objectness is the efficiency. Traditional methods [1] [6] rely on the sliding window, which evaluate everywhere on the test image, which is set 1,000 in [1]. On the contrast, our algorithm reduces the number of candidate windows dramatically. In our experiments, we build a 3-level binary tree for each image, leading to 14 nodes. While Selective Search [33] utilizes 395 windows to achieve similar recall. Detailed comparison is shown in Table 4.

In more details, we show the efficiency of the proposed hierarchical structure in Figure 5(a), for the task of objectness detection. From this figure, we have several advantages compared with state-of-the-arts: 1) Due to the more efficient candidate region selection introduced by the structural prior knowledge in our algorithm, we dramatically reduce the number of false alarm which leads to a fairly high precision. 2) Most of the windows detected by MS, CC and SS [1] are overlapped because of sliding windows in their methods, and the recall is very limited though large number of candidate regions are returned. However, our method can get rid of this problem. Figure 6 further shows several objectness detection results of our proposed methods. It can be seen that most of our bounding boxes are within the object areas.



(a) (b)

Figure 5. (a) Performance comparison: PR-curves of different methods on the task of objectness detection, by varying the thresholds. (b) Recall of (*Structural Objectness*) on object proposals. Tested on VOC 2007 test dataset.

Architecture of Parsing Tree: Different architectures of the parsing trees may derive different results. In Figure 5(b) we show Recalls of objectness detection for binary parsing trees with depth $D = 4$ and $D = 5$ respectively. Note that the recall is only determined by the total number of nodes in the parsing tree; and for a complex scene, a deeper binary tree is capable to represent its structure. Using deeper parsing trees (*e.g.*, depth 5) could generate regions for objects of fine scales, but will also produce more non-meaningful tree nodes, which increases the chance of false alarm. As seen in Figure 5(b), parsing tree of $D = 4$ performs much better in top 10 detections than $D = 5$. An statistic shows that most object regions occurs in level 2 and level 3 nodes in our parsing tree. To balance the performance and efficiency, we adopt depth $D = 4$.

Objectness on the Hierarchy To further understand how the shape parsing tree helps the objectness detection, several samples are shown in Figure 7, which tracks the objectness detection results in different levels. As looked in bottom-up, the detection of low-levels focuses on object parts, such as the wheels of the motors, and it gradually integrates into object-level detections as towards to the root. It is critically important for partial matching and detection, and even designing powerful image features.

Recall on Object Proposals. To further validate our analysis of the proposed algorithm, we test the recall of ob-

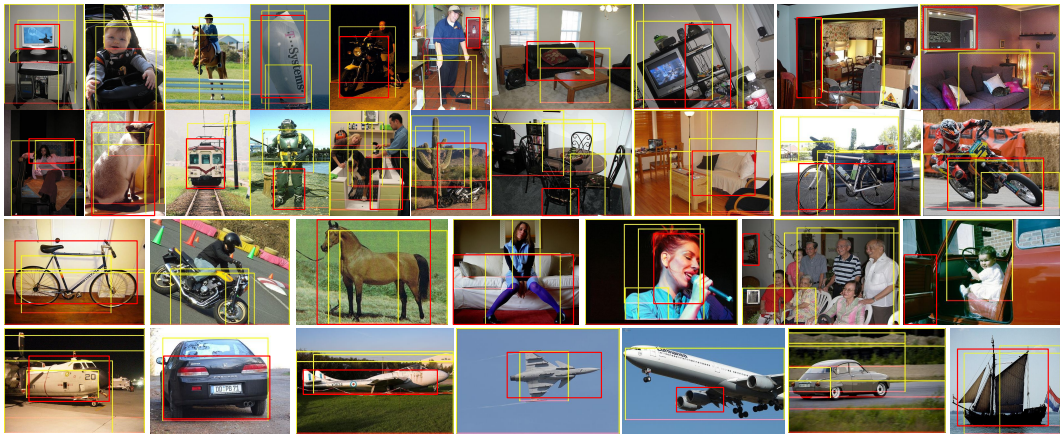


Figure 6. The detection results using the proposed "structured objectness" based on the *shape parsing tree* compared with objectness [1]. The yellow rectangles are our detection results using threshold = 75, and the ones of MS method in [1] are shown in red. Generally, we achieve competitive results. Especially, for the complex scenes such crowds, we perform acceptably good.

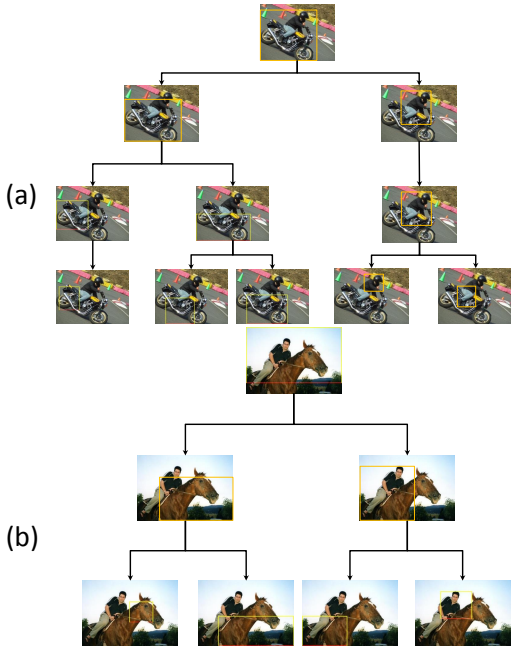


Figure 7. The learned objectness in a hierarchical organization. It shows how the structural information affects the detection process.

ject window proposals for the structural objectness, on VOC 2007 test dataset. We also compare our algorithm with Selective Search [33]. For the selective search, we use the same settings as [33] on VOC 2007 test dataset. Figure 5(b) shows the recall of our algorithm. Though our algorithm only achieves a recall at 0.612 using 14 proposals, compared with 0.829 for selective search using $3,574$ proposals, we perform much better in the top ones, which suggests the high hit rate in the top results. In the meanwhile, selective search can only achieves a recall of 53.5 using top 100 proposals. When we increase the level of parsing tree from 3 to 4, and the proposal number increases from 14 to 30, the recall can reach 0.732 in the top 30 proposals. Selective search can only achieve such a performance using around 300 proposals.

5. Conclusion and Future Work

In this paper, we study the problem of exploring image structure by organizing visual components into a hierarchical parsing tree. The main technical contributions are twofold: first, we parse an image into a hierarchical structure tree according to scales and shapes (in term of edges), and link the parsing result with local appearances in an unsupervised way towards a more discriminative scene understanding; second, we propose an efficient yet robust structural feature representation by means of a structural pooling strategy, which facilitates fast tree matching between two hierarchical parsing trees. We demonstrate that the hierarchical parsing scheme can be applied to various computer vision problems by providing more discriminative features such as object region proposal. We also show how the structural information helps improve feature discrimination through two exemplar applications.

For future work, we will extend the shape parsing scheme proposed in this paper to work under a supervised manner, such as semantic image segmentation and parsing. Though positive, the obtained parsing tree still needs further refinements due to the complicated scale distribution in the scale space for complex scenes. In order to improve the expressive power for complex scenes, it is feasible to employ randomized algorithms to add perturbations in the scale splitting similar to the approach in [8], by which forests are built instead of a tree for each image.

Acknowledgment: Xianming Liu and Thomas S. Huang are supported by National Science Foundation (No. 1318971). Rongrong Ji is supported by the Excellent Youth Science Foundation of NSFC (No. 61422210), and the Nature Science Foundation of China (No. 61373076).

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 73–80. IEEE, 2010.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):898–916, 2011.

[3] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2559–2566. IEEE, 2010.

[4] J. Bruna and S. Mallat. Invariant scattering convolution networks. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1872–1886, 2013.

[5] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *IEEE CVPR*, 2014.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[7] J. J. DiCarlo, D. Zoccolan, and N. C. Rust. How does the brain solve visual object recognition? *Neuron*, 2012.

[8] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1841–1848. IEEE, 2013.

[9] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning hierarchical features for scene labeling. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(8):1915–1929, 2013.

[10] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.

[11] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(1):36–51, 2008.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2013.

[13] K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *The Journal of Machine Learning Research*, 8:725–760, 2007.

[14] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[15] D. H. Hubel. *Eye, brain, and vision*. Scientific American Library New York, 1988.

[16] J. J. Koenderink. The structure of images. *Biological cybernetics*, 50(5):363–370, 1984.

[17] I. Kokkinos and A. Yuille. Hop: Hierarchical object parsing. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 802–809. IEEE, 2009.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[19] M. P. Kumar, H. Turki, D. Preston, and D. Koller. Learning specific-class segmentation from diverse data. In *IEEE International Conference on Computer Vision*, pages 1800–1807. IEEE, 2011.

[20] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE conference on Computer Vision and Pattern Recognition*, pages 2169–2178. IEEE, 2006.

[21] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):117–156, 1998.

[22] T. Lindeberg. Feature detection with automatic scale selection. *International journal of computer vision*, 30(2):79–116, 1998.

[23] X.-M. Liu, C. Wang, H. Yao, and L. Zhang. The scale of edges. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 462–469. IEEE, 2012.

[24] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[25] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(7):674–693, 1989.

[26] D. Marr and A. Vision. A computational investigation into the human representation and processing of visual information. *WH San Francisco: Freeman and Company*, 1982.

[27] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

[28] T. Quack, V. Ferrari, B. Leibe, and L. Van Gool. Efficient mining of frequent and distinctive feature configurations. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[29] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 994–1000. IEEE, 2005.

[30] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE, 2003.

[31] J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *Computer Vision–ECCV 2010*, pages 352–365. Springer, 2010.

[32] Z. Tu, X. Chen, A. L. Yuille, and S.-C. Zhu. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of Computer Vision*, 63(2):113–140, 2005.

[33] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.

[34] A. Witkin. Scale-space filtering: A new approach to multi-scale description. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84.*, volume 9, pages 150–153. IEEE, 1984.

[35] Z. Wu, Q. Ke, M. Isard, and J. Sun. Bundling features for large scale partial-duplicate web image search. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 25–32. IEEE, 2009.

[36] P. Xu, X. Liu, H. Yao, Y. Zhang, and S. Tang. Structured textons for texture representation. In *ICIP*, pages 240–244, 2013.

[37] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE, 2009.

[38] J. Yuan, Y. Wu, and M. Yang. Discovery of collocation patterns: from visual words to visual phrases. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[39] Y. Zhang, Z. Jia, and T. Chen. Image retrieval with geometry-preserving visual phrases. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 809–816. IEEE, 2011.

[40] L. Zhu, Y. Chen, and A. Yuille. Learning a hierarchical deformable template for rapid deformable object parsing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(6):1029–1043, 2010.