frontiers in PSYCHOLOGY

ORIGINAL RESEARCH ARTICLE
published: 29 August 2013
doi: 10.3389/fpsyg.2013.00506

Basic level scene understanding: categories, attributes and structures

Jianxiong Xiao[1], James Hays[2]*, Bryan C. Russell[3], Genevieve Patterson[2], Krista A. Ehinger[4], Antonio Torralba[5] and Aude Oliva[6]

[1] *Computer Science, Princeton University, Princeton, NJ, USA*
[2] *Computer Science, Brown University, Providence, RI, USA*
[3] *Computer Science and Engineering, University of Washington, Seattle, WA, USA*
[4] *Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA, USA*
[5] *Department of EECS, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA*
[6] *Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA*

Edited by:
Tamara Berg, Stony Brook University, USA

Reviewed by:
Andrew M. Haun, Harvard Medical School, USA
Devi Parikh, Virginia Tech, USA

****Correspondence:***
James Hays, Computer Science Department, Brown University, 115 Waterman Street, Box 1910, Providence, RI 02912, USA
e-mail: hays@cs.brown.edu

A longstanding goal of computer vision is to build a system that can automatically understand a 3D scene from a single image. This requires extracting semantic concepts and 3D information from 2D images which can depict an enormous variety of environments that comprise our visual world. This paper summarizes our recent efforts toward these goals. First, we describe the richly annotated SUN database which is a collection of annotated images spanning 908 different scene categories with object, attribute, and geometric labels for many scenes. This database allows us to systematically study the space of scenes and to establish a benchmark for scene and object recognition. We augment the categorical SUN database with 102 scene attributes for every image and explore attribute recognition. Finally, we present an integrated system to extract the 3D structure of the scene and objects depicted in an image.

Keywords: SUN database, basic level scene understanding, scene recognition, scene attributes, geometry recognition, 3D context

1. INTRODUCTION

The ability to understand a 3D scene depicted in a static 2D image goes to the very heart of the computer vision problem. By "scene" we mean a place in which a human can act within or navigate. What does it mean to *understand a scene*? There is no universal answer as it heavily depends on the task involved, and this seemingly simple question hides a lot of complexity.

The dominant view in the current computer vision literature is to name the scene and objects present in an image. However, this level of understanding is rather superficial. If we can reason about a larger variety of semantic properties and structures of scenes it will enable richer applications. Furthermore, working on an over-simplified task may distract us from exploiting the natural structures of the problem (e.g., relationships between objects and 3d surfaces or the relationship between scene attributes and object presence), which may be critical for a complete scene understanding solution.

What is the ultimate goal of computational scene understanding? One goal might be to pass the **Turing test for scene understanding**: Given an image depicting a static scene, a human judge will ask a human or a machine questions about the picture. If the judge cannot reliably tell the machine from the human, the machine is said to have passed the test. This task is beyond the current state-of-the-art as humans could ask a huge variety of meaningful visual questions about an image, e.g., Is it safe to cross this road? Who ate the last cupcake? Is this a fun place to vacation? Are these people frustrated? Where can I set these groceries? etc.

Therefore, we propose a set of goals that are suitable for the current state of research in computer vision that are not too simplistic nor challenging and lead to a natural representation of scenes. Based on these considerations, we define the task of scene understanding as predicting the scene category, scene attributes, the 3D enclosure of the space, and all the objects in the images. For each object, we want to know its category and 3D bounding box, as well as its 3D orientation relative to the scene. As an image is a viewer-centric observation of the space, we also want to recover the camera parameters, such as observer viewpoint and field of view. We call this task **basic level scene understanding**, with analogy to basic level in cognitive categorization (Rosch, 1978). It has practical applications for providing sufficient information for simple interaction with the scene, such as navigation and object manipulation.

1.1. OUTLINE

In this paper we discuss several aspects of basic level scene understanding. First, we quickly review our recent work on categorical (section 2) and attribute-based scene representations (section 3). Finally, we go into greater detail about novel work in 3d scene understanding using structured learning to simultaneously reason about many aspects of scenes (section 4).

Supporting these research efforts is the **S**cene **UN**derstanding (SUN) database. By modern standards, the SUN database is not especially large, containing on the order of 100,000 scenes. But the SUN database is, instead, **richly annotated** with scene categories, scene attributes, geometric properties, "memorability" measurements (Isola et al., 2011), and object segmentations. There are 326,582 manually segmented objects for the 5650 object categories labeled (Barriuso and Torralba, 2012). Object

categories are visualized in **Figure 1** and annotated objects are shown in **Figures 2**, **3**, and **4**. We believe the SUN database is the largest database from which one can learn the relationship among these object and scene properties. This combination of scene diversity and rich annotation is important for scaling scene understanding algorithms to work in the real world.

2. SCENE CATEGORIES

One of the fundamental tasks of basic level scene understanding is to be able to classify a natural image into a limited number of semantic categories. What are the scene categories? From a human-centric perspective, the categories should capture the richness and diversity of environments that make up our daily experiences. Although the visual world is continuous, many environmental scenes are visual entities that can be organized in functional and semantic groups. A given scene or place may allow for specific actions, such as eating in a restaurant, drinking in a pub, reading in a library, or sleeping in a bedroom.

To capture this diversity, we have constructed a quasi-exhaustive taxonomy and dataset of visual scene categories that can be encountered in the world. We use WordNet, an electronic dictionary of the English language containing more than 100,000 words, and manually select all of the terms that describe scenes, places, and environments (any concrete noun that could reasonably complete the phrase "I am in a *place*", or "Let's go to the *place*"). This has yielded 908 scene categories, which are illustrated in **Figure 5**.

FIGURE 1 | Object categories in the SUN database. The area of each word is proportional to the frequency of that object category.

FIGURE 2 | Examples from the 19,503 fully annotated images in the SUN database.

Once we have a list of scenes, the next task is to collect images belonging to each scene category. Since one of our goals is to create a large collection of images with variability in visual appearance, we have collected Internet images using various image search engines for each scene category term. Then, a group of trained human participants manually prune the images that do not correspond to the definition of the scene category resulting in a database of 131,072 images. This collection of images is the core of the SUN database onto which all other annotations discussed are added. Using a variety standard image features (e.g., spatial pyramids of dense visual words) one can achieve roughly 40% accuracy in a 397-way scene categorization task (Xiao et al., 2010). Recent work has achieved 47% accuracy (Sanchez et al., 2013). We have also studied intra-category variations in the SUN database. Within the same scene category, human observers find some exemplars to be more typical than others and category membership is naturally graded, not discrete (Ehinger et al., 2011).

3. SCENE ATTRIBUTES

In this section we present the SUN attribute database—the first large-scale scene attribute database (Patterson and Hays, 2012). Recently, there has been interest in *attribute-based* representations of objects (Farhadi et al., 2009; Lampert et al., 2009; Berg et al., 2010; Endres et al., 2010; Farhadi et al., 2010; Russakovsky and Fei-Fei, 2010; Su et al., 2010), faces (Kumar et al., 2009), and actions (Liu et al., 2011; Yao et al., 2011) as an alternative or complement to category-based representations. However, there has been only limited exploration of attribute-based representations for scenes (Oliva and Torralba, 2001, 2002; Greene and Oliva, 2009; Parikh and Grauman, 2011), *even though scenes are uniquely poorly served by categorical representations*. For example, an object usually has unambiguous membership in one category. One rarely observes objects at the transition point between object categories (e.g., this object is on the boundary between "sheep" and "horse"), however, the analogous situation is common with scenes (e.g., this scene is on the boundary between "savanna" and "forest").

In the domain of scenes, an attribute-based representation might describe a image with "concrete," "shopping," "natural lighting," "glossy," and "stressful" in contrast to a categorical label such as "store". Note that attributes do not follow category boundaries. Indeed, that is one of the appeals of attributes—they can describe intra-class variation (e.g., a canyon might have water or it might not) and inter-class relationships (e.g., both a canyon and a beach could have water). We limit ourselves to *global, binary* attributes. but we average the binary labels from multiple annotators to produce real-valued confidences.

Our first task is to establish a taxonomy of scene attributes for further study. We use a simple, crowd-sourced "splitting task" (Oliva and Torralba, 2001) in which we show Amazon Mechanical Turk (AMT) workers two groups of scenes and ask them to list attributes that are present in one group but not the other. The images that make up these groups are "typical" (Ehinger et al., 2011) scenes from random categories of the SUN database. From the thousands of attributes reported by participants we manually

FIGURE 3 | Sample object segments from popular object categories in the SUN database.

FIGURE 4 | To demonstrate intra-category object variation within the SUN database, these are samples of the 12,839 chairs that were manually annotated in 3500 images.

collapse nearly synonymous responses (e.g., dirt and soil) into single attributes. We omit object presence attributes because the SUN database already has dense object labels for many scenes. In the end, we arrive at a taxonomy of 38 material attributes (e.g., cement, vegetation), 11 surface properties (e.g., rusty), 36 functions or affordances (e.g., playing, cooking), and 17 spatial envelope attributes (e.g., enclosed, symmetric). See **Figure 6** for the full list.

With our taxonomy of attributes finalized, we create the first large-scale database of attribute-labeled scenes. We build the SUN attribute database on top of the existing SUN categorical database (section 2) for two reasons: (1) to study the interplay between attribute-based and category-based representations and (2) to ensure a diversity of scenes. We annotate 20 scenes from each of 717 SUN categories totaling 14,340 images. We collect ground truth annotations for all of the 102 attributes for each

scene. In total we gather more than four million labels through crowdsourcing. After labeling the entire dataset once with the general AMT population, we identify a smaller group of 38 trusted workers out of the ∼800 who participated. We repeat the labeling process two more times using only these trusted workers.

3.1. BUILDING THE SUN ATTRIBUTE DATABASE

To quantitatively assess annotation reliability we manually grade random annotations in the database. Ninety-three percent positive annotations are reasonable (some are undoubtedly subjective). The negative annotations also have 93% accuracy, but this isn't as significant since negative labels make up 92% of the annotations. Like objects, it seems that scene attributes follow a heavy-tailed distribution with a few being very common (e.g., "natural") and most being rare (e.g., "wire"). If we instead evaluate the consensus annotation which two of the three annotators agree on for each scene attribute, the accuracy rises to 95%.

3.1.1. Exploring scenes in attribute space

Now that we have a database of attribute-labeled scenes we can attempt to visualize that space of attributes. In **Figure 7** we show all 14,340 of our scenes projected onto two dimensions by dimensionality reduction. We sample several points in this space to show the types of scenes present as well as the nearest neighbors to those scenes in attribute space. For this analysis the distance between scenes is simply the Euclidean distance between their real-valued, 102-dimensional attribute vectors. **Figure 8** shows the distribution of images from 15 scene categories in attribute space. The particular scene categories were chosen to be close to those categories in the 15 scene database (Lazebnik et al., 2006).

3.2. RECOGNIZING SCENE ATTRIBUTES

To recognize attributes in images, we create an independent classifier for each attribute using splits of the SUN Attribute dataset for training and testing data. We treat an attribute as present if it receives at least two of three possible votes from AMT annotators and absent if it receives zero votes. We represent each image

FIGURE 5 | List of 908 scene categories in our SUN database—the most exhaustive scene dataset to date. The height of each category name is proportional to the number of images belonging to the category.

FIGURE 6 | Attributes in the SUN Attribute database. The area of each word is proportional to the frequency of that attribute.

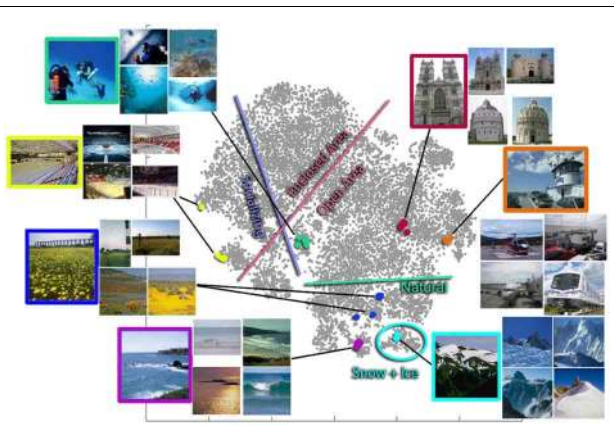

FIGURE 7 | 2D visualization of the SUN Attribute dataset. Each image in the dataset is represented by the projection of its 102-dimensional attribute feature vector onto two dimensions using t-Distributed Stochastic Neighbor Embedding. There are groups of nearest neighbors, each designated by a color. Interestingly, while the nearest-neighbor scenes in attribute space are semantically very similar, for most of these examples (underwater ocean, abbey, coast, ice skating rink, field wild, bistro, office) *none* of the nearest neighbors actually fall in the same SUN database category. The colored border lines delineate the approximate boundaries between images with and without the particular attributes.

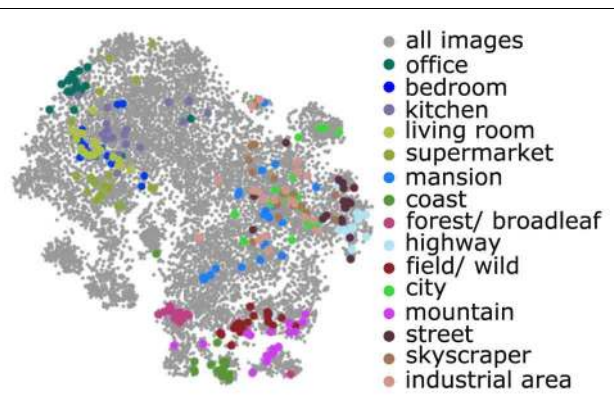

FIGURE 8 | The 15 scene category database images in attribute space. These 15 proxy categories occupy a relatively small fraction of attribute space spanned by the SUN database (all gray points).

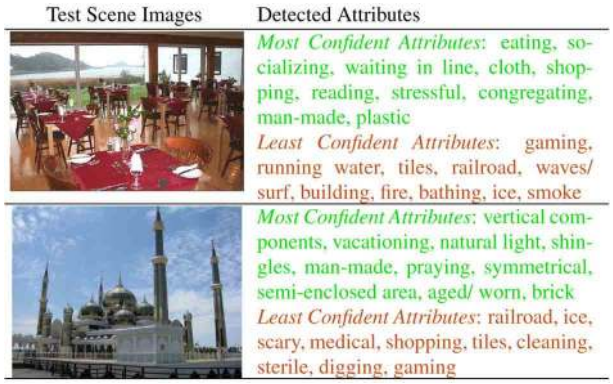

FIGURE 9 | Attribute detection. For each query, the most confidently recognized attributes (green) and the least confidently recognized attributes (red).

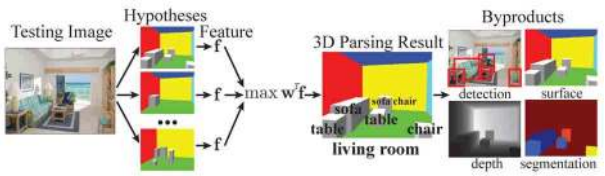

FIGURE 10 | Unified 3D scene parsing for basic level scene understanding. For each image, we generate a pool of hypotheses. For each hypothesis, we construct a feature vector **f** encoding both image features and scores from evaluating various contextual rules on the hypothesized scene structure (**Figure 11**). We choose the hypothesis that maximizes the objective function $\mathbf{w}^T\mathbf{f}$ as the result of 3D scene parsing. As by-products of our 3D parsing result we obtain information that have traditionally been considered in isolation, such as the 2D location of objects, their depth and 3D surface orientation.

with a subset of the features and kernels used for scene categorization in Xiao et al. (2010). We train Support Vector Machines on 90% of the SUN Attribute dataset and test on the remaining 10%. **Figure 9** shows the attributes detected in two query scenes. Attribute recognition accuracy varies considerably, e.g., average precision of 0.93 for "vegetation," 0.78 for "sailing," 0.60 for "moist," and 0.27 for "stressful." We show qualitative results of our attribute classifiers in **Figure 9**. Our classifiers and the code are publicly available[1].

4. SCENE STRUCTURES

Although an image is a 2D array, we live in a 3D world, where scenes have volume, affordances, and can be spatially arranged where one object can be occluded by another. The ability to reason about these 3D properties would be of benefit for tasks such as navigation and object manipulation.

We seek to build a unified framework for parsing the 3D structure of a scene. What does it mean to parse a scene? There is no universal answer, as it heavily depends on the tasks (e.g., the task of a house painter is to find all cracks on a wall). Here, we limit our scope to the basic 3D properties of the space, including the scene category, the 3D boundary of the space, and all the objects in the image. For each object, we want to know its category and 3D bounding box, including its orientation. As an image is a viewer-centric observation of the space, we also want to recover the camera intrinsic and extrinsic parameters. An example 3D parse result is depicted in **Figure 10** for a living room scene.

While it is possible to reason about these various scene properties independently, we desire an algorithm which considers them jointly. Thus an algorithm might suppress a false positive "bed" detection because it is sitting on a "table". There are numerous such scene layout "rules" which constrain the parsing of a scene and optimizing a scene parsing with respect to all such rules could lead to complicated inference procedures. The key idea of our algorithm is to generate a pool of possible output hypotheses and select the most likely one. We define a list of parsing rules and use structural Support Vector Machine (SVM) (Joachims et al., 2009) to learn the relative importance of these rules from the training data. For each hypothesis, we extract a vector which encodes how well each rule is satisfied by the hypothesis. The weight of each element in this vector in scoring the hypotheses is learned from training data. More specifically, given an image $\mathbf{x}$, we aim to predict a structured representation $\mathbf{y}$ for the 3D parsing result using a linear prediction rule: $\arg\max_{\mathbf{y}\in\mathcal{Y}} \mathbf{w}^T\mathbf{f}(\mathbf{x},\mathbf{y})$, where $\mathcal{Y}$ is the hypothesis space of all possible 3D parsing results for $\mathbf{x}$. The label $\mathbf{y}$ is a variable dimension data structure of a 3D scene parsing, which includes the scene category, camera parameters, space boundaries, and objects[2]. We encode image evidence and contextual constraints into the feature vector $\mathbf{f}(\mathbf{x},\mathbf{y})$. Therefore, a good scene parsing result $\mathbf{y}$ not only explains the image evidence well, but also satisfies the contextual constraints. The parsing rules are illustrated in **Figure 11**.

During training, given a training sample of input–output pairs $\big((\mathbf{x}_1,\mathbf{y}_1),\ldots,(\mathbf{x}_N,\mathbf{y}_N)\big)$ from manual annotation (we add annotations to the data set of Hedau et al., 2012), we seek to minimize the following convex optimization problem:

$$\min \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\xi_n, \tag{1}$$

such that $\mathbf{w}^T\mathbf{f}(\mathbf{x}_n,\mathbf{y}_n) - \mathbf{w}^T\mathbf{f}(\mathbf{x}_n,\hat{\mathbf{y}}) \geq \triangle(\mathbf{y}_n,\hat{\mathbf{y}}) - \xi_n$, for all $1 \leq n \leq N$ and for all possible output structures $\hat{\mathbf{y}} \in \mathcal{Y}_n$ in the

[1]SUN Attribute Classifiers along with the full SUN Attribute dataset and associated code are available at www.cs.brown.edu/~gen/sunattributes.html.

[2]We assign the origin of the world coordinate system to be at the camera location, and set the gravity direction as the negative y axis. The unknown camera parameters are focal length, principal point and the camera rotation. The space is parameterized as $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] \times [z_{\min}, z_{\max}]$, with the assumption that the space boundaries, e.g., walls, floor and ceiling, are perpendicular to each other. The objects in the scene are represented as a set of objects grounded on the floor or stacked on another object. Each object is represented as an object category, a 3D bounding box, including its center location (x_c, y_c, z_c), its size (x_s, y_s, z_s), and its yaw angle θ, with the assumption that the vertical axis of the bounding box must be parallel with the gravity direction.

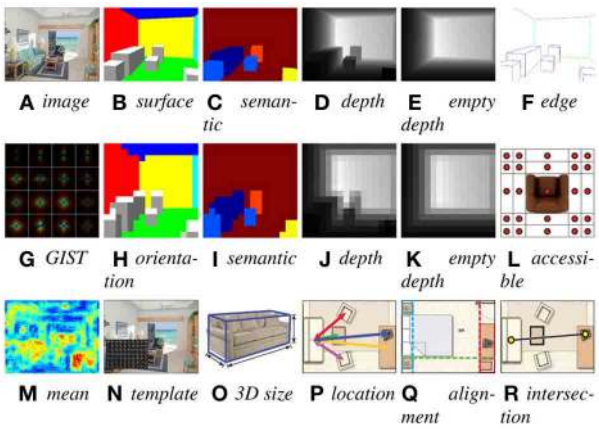

FIGURE 11 | (A–R) Illustration of various rules we design to describe both the image evidence and context compatibility. All these rules are encoded in the structural SVM feature function $\mathbf{f}(\mathbf{x}, \mathbf{y})$.

hypothesis space. $\triangle(\mathbf{y}_n, \hat{\mathbf{y}})$ is the loss function controlling the margin between correct label $\mathbf{y}_n$ and prediction $\hat{\mathbf{y}}$[3].

One of the major differences between structural SVM and standard SVM is that the feature $\mathbf{f}(\mathbf{x}, \mathbf{y})$ depends not only on $\mathbf{x}$, but also on $\mathbf{y}$. This enables us to encode many important image features and context rules (section 5) that were not possible in previous works. Moreover, the SVM discriminatively learns the relative importance of features and relations based on training data. For training, we use the cutting plane algorithm (Joachims et al., 2009; Desai et al., 2011). For each training image $\mathbf{x}_n$, we use simple heuristics to obtain a hypothesis pool $\mathcal{Y}_n$ (section 6), to be the initial working constraints to train the SVM. As the training goes on, we add more hypotheses with large $\mathbf{w}^T\mathbf{f}$ values as working constraints, based on the current $\mathbf{w}$. It can be seen as a generalization of hard negative mining in sliding window object detection (Dalal and Triggs, 2005; Felzenszwalb et al., 2010), and it significantly speeds up the computation and reduces memory consumption.

4.1. RELATED WORK

Coughlan and Yuille (1999); Delage et al. (2005); Hoiem (2007); Saxena et al. (2009): reconstruct surface orientation and depth from single view images. Han and Zhu (2005); Yu et al. (2008); Hedau et al. (2009, 2010); Lee et al. (2009, 2010); Wang et al. (2010); Gupta et al. (2011); Pero et al. (2011, 2012); Yu et al. (2011); Zhao and chun Zhu (2011); Hedau et al. (2012); Schwing et al. (2012): represent the state-of-the-art on room layout estimation and furniture arrangement. There are also many impressive techniques to model context and object relations (Hoiem, 2007; Rabinovich et al., 2007; Desai et al., 2011), and parse a scene (Han and Zhu, 2005; Heitz et al., 2008; Socher et al., 2011; Li et al., 2012) in a unified way for multiple tasks at the same time. Although they have some success on reasoning about 3D, their main focus is still on 2D. Meanwhile, structural SVMs (Joachims et al., 2009) have been successfully applied to many computer vision tasks (Hedau et al., 2009; Felzenszwalb et al., 2010; Wang et al., 2010; Desai et al., 2011; Gupta et al., 2011; Schwing et al., 2012). The main difference is that these approaches have not learned or predicted as rich and complex structures as ours.

[3]To measure the 2D observation difference, between $\mathbf{y}_n$ and $\hat{\mathbf{y}}$, we render both 3D meshes to obtain two surface orientation maps. We compare the two maps at each pixel and obtain the accuracy Ω_g over all pixels. We do the same thing for semantic segmentation masks to obtain the pixel-wise accuracy Ω_s. Because 2D errors are sometimes not very indicative of actual 3D errors, we also measure the object correctness in 3D and define Ω_o as the proportion of overlapping objects which share the same category. Finally, we measure the scene category correctness Ω_h. The weighted sum of all these measurement is our total loss function: $\triangle(\mathbf{y}_n, \hat{\mathbf{y}}) = w_g(1-\Omega_g) + w_s(1-\Omega_s) + w_o(1-\max(\Omega_o, 0)) + w_h(1-\Omega_h)$.

5. PARSING RULES

A good scene parsing result $\mathbf{y}$ not only explains the image evidence well, but also satisfies contextual constraints. Inspired by natural language parsing (NLP) using structural SVM (Joachims et al., 2009), we encode both types of rules—image likelihood and context constraints—into the feature vector $\mathbf{f}(\mathbf{x}, \mathbf{y})$. The structural SVM automatically figures out which rules are important discriminatively based on the training data. The image likelihood rule is evidence of the form "This set of pixels looks like a bed" while higher-order, contextual rules are of the form "a bed is in a bedroom". We accumulate all rules being used in a unified way into one fixed length $\mathbf{f}$, for both image and context rules defined below. The scoring function $\mathbf{w}^T\mathbf{f}(\mathbf{x}, \mathbf{y})$ for a hypothesis $\mathbf{y}$ is essentially a weighted sum of the scores from these rules.

5.1. REGION FEATURES

Given $\mathbf{y}$, the surface orientation is fully determined for each pixel, as shown in **Figure 11B**. We define several categories of surface orientation, including floor, ceiling, left, right, and frontal-parallel walls, as well as top, bottom, left, right, and frontal-parallel faces of a 3D object bounding box. For each image $\mathbf{x}_n$, we precompute a pixel-wise image feature map $\mathbf{F}_n$ (**Figure 11M**), using several state-of-the-art image features—SIFT (Lowe, 2004), 3×3 HOG template centered at each pixel (Dalal and Triggs, 2005; Felzenszwalb et al., 2010), Self-Similarity (Shechtman and Irani, 2007), and distance transformed Canny edge map, as well as the pixel locations, for a total of 440 dimensions. **Figure 11M** visualizes the mean feature values. To form a feature vector, for each surface orientation category, we aggregate $\mathbf{F}_n$ to sum the feature vectors over all pixels with this orientation category, based on the orientation map. For example, we sum over all feature vectors belonging to the floor by $\sum_{(x,y)\in\text{floor}} \mathbf{F}_n(x, y, :)$ to obtain the feature vector for the floor. Similar to surface orientation, we determine the semantic segmentation map using 3D bounding boxes and aggregate the image features to make another set of features based on object categories. In **Figure 11C**, different colors correspond to different object categories. Furthermore, inspired by scene classification, to encode the global structure of the scene, we extract the GIST feature (Oliva and Torralba, 2001) for the whole image, as shown in **Figure 11G**.

5.2. EDGE AND CORNER FEATURES

We extract edges of the space boundaries and object bounding boxes, and aggregate the image feature over edges. As shown in **Figure 11F**, we further separate the edges into six categories, including ground line, ceiling line, vertical wall edges as well as

the top, bottom, and vertical edges of an object bounding box. For each type, we aggregate the image feature in the same way as the region features. In a similar way, we also extract corner features, for four types of corners: floor corners, ceiling corners, as well as the top and bottom of objects.

5.3. HOLISTIC SEGMENTATION AND DEPTH STATISTICS

As shown in **Figure 11H**, to encode the global layout of surface orientation, we extract holistic statistics of the map. We downsample the full resolution surface orientation map in **Figure 11B** in a spatial pyramid of 16×16, 8×8, 4×4, 2×2, and 1×1 resolutions, and concatenate all of the values together to form a holistic feature of surface orientation. We do the same for semantic segmentation as well to encode the holistic statistics of the semantic map (**Figure 11I**). Furthermore, given **y**, the depth of the scene is fully determined, which is a strong contextual criterion to judge whether a parsing result is possible independent of the image observation. We can obtain the depth map (**Figure 11D**), and extract a holistic depth feature with spatial pyramid (**Figure 11J**).We also empty the room to extract the depth map of the room structure (**Figure 11D**), and build the spatial pyramid as well (**Figure 11K**). We compute the average depth map for each scene category, and use the difference between the current depth and the mean depth as part of features. The depth encodes some statistics of typical views, e.g., a camera looking directly at a nearby wall is not likely to be a good hypothesis.

5.4. OBJECT 2D TEMPLATES, COLOR AND TEXTURE CONSISTENCY

Inspired by sliding window object detection, for each object, we obtain the 2D bounding box from the 3D bounding box, and compute a HOG template (Dalal and Triggs, 2005; Felzenszwalb et al., 2010) of the 2D window (**Figure 11N**), as a view-dependent object template feature. For each object category, we obtain the average aspect ratio of the ground truth from the training data, and adjust the individual instances to have the same aspect ratio and template resolution during feature extraction.

5.5. OBJECT 3D SIZE AND LOCATION

Different object categories have very different absolute sizes and locations. Therefore, we extract the 3D size as features (**Figure 11O**). For the location of an object, because it is usually relative the space boundary, e.g., a bed flush to the wall, we find the closest vertical space boundary for an object, and use their distance and angle difference as features.

5.6. HUMAN ACCESSIBILITY

For the arrangement of objects, there is usually some space between objects for them to be accessible by humans. For example, we tend to have some space in front of a sofa to enable people to walk and sit there. To encode this information as a feature vector, as shown in **Figure 11L**, we put a 5×5 horizontal spatial grid that is 2 feet away around an object. We would like to record whether some space in each bin is occupied by other objects or outside the room. Because this computation is very expensive, we instead use only the center location of each bin, and check if it is inside some other objects or outside the room, to approximate the accessibility. We concatenate the occupancy information together to form a feature vector.

5.7. CO-OCCURRENCE AND RELATIVE LOCATION

Given **y**, we obtain the object co-occurrence count for each pair of object categories as our co-occurrence context feature. We also count how many times an object appears together with the scene category. For both types of occurrence relationship, we obtain the average statistics of the training data, and use the difference between the current statistics and the average one as features. Beyond co-occurrence, objects usually have certain position constraints with each other. For example, a sofa is usually parallel with a coffee table with a certain distance. Therefore, we encode the relative 3D distance and orientation difference of their bounding box as features (**Figure 11P**). Also, many objects tend to be arranged so that certain faces are aligned (**Figure 11Q**). Therefore, we encode the pairwise alignment relationship between objects, by checking whether certain facets are parallel or very close in the space. This is not as strong as imposing the Manhattan world assumption, but it encourages snapping of edges and corners.

5.8. HIGHER ORDER RELATIONSHIPS

Object relation is not just pairwise. For example, in a typical living room, between a sofa and a TV, there may be a coffee table but it is unlikely to be a tall object blocking the view between them. To encode this high order relationship, for each pair of objects, we draw a line segment connecting them, and check whether any other objects have bounding boxes intersecting this line segment (**Figure 11R**). If yes, we will record the object category and use this as a feature.

5.9. CAMERA AND VIEW CONSTRAINTS

When people take a picture, the camera is not randomly rotated and the intrinsic parameters must be in certain range constrained by the camera hardware. Therefore, we can encode the camera parameters as features as well. Also, we want to represent the volume of visual space in the field of view. To simplify the computation, we reconstruct the floor in 3D and calculate the 3D area of that part that is visible from the camera. We use the 3D area, and the difference of the area to the average floor area across the training set as a feature.

For all these rules, some of them are unary features that encode rules about one instance of the object, and some of them are pairwise or higher order relationship among objects and scenes. Some rules are hypothesis independent features in the sense that their values do not change because of **y**, while others are hypothesis dependent features with values depending on **y**. Some features are view-independent and encode the absolute 3D statistics independent of the camera, while others are view dependent features that heavily depend on the camera.

6. HYPOTHESIS GENERATION

We propose a two step algorithm for generating hypotheses and performing fast inference. For any image, either during training or testing, we first generate a large pool of initial hypotheses, without considering the objective function. Then, we do several

iterations of heuristic search, based on the initial hypothesis pool and $\mathbf{w}$, and simply pick the one with the highest objective value $\mathbf{w}^T\mathbf{f}$ as the solution. **Figure 12** shows some examples of hypotheses generated.

For the first step of constructing an initial hypothesis set, we use two approaches. First, we use a bottom–up approach to generate some hypotheses based on image evidence. We detect edges and fit line segments, to obtain camera parameters from vanishing points at orthogonal directions. To look for reliable line segments and vanishing points, we use many state-of-the-art line detection algorithms (Han and Zhu, 2005; Toldo and Fusiello, 2008; Hedau et al., 2009; Tardif, 2009; Feng et al., 2010; Lee et al., 2010; von Gioi et al., 2012). For each set of camera parameters estimated from all these methods, if the focal length is outside the range of a normal digital camera (we use 24–105 mm), or the principal point is far away from the center of the image, we remove the estimated camera from further considerations. In this way, we usually obtain about seven good camera models per image, out of hundreds estimated by all methods. For each of the cameras, we randomly sample some edges to form a hypothesis for the space boundary and group line segments into cuboids (Lee et al., 2010). We also exhaustively slide cuboids in 3D for each object category using the average size of objects in the category from the training set. We explicitly enforce the support relationship to make sure each objects are grounded on the floor or on top of other objects. The second way to generate hypotheses is to copy ground truth hypotheses from the training set and add them into the initial hypothesis space. We also generate hypotheses by using the estimated cameras from line detection and replace them in the copied training set labels. With all these hypotheses, we randomly sample a subset of all hypotheses as our initial hypothesis pool. For the training data, we also randomly change the ground truth to generate more training hypotheses nearby the truth to learn a better $\mathbf{w}$.

The second step is to obtain some better hypotheses based on $\mathbf{w}$ by modifying the initial hypotheses. This is used to find the most violated constraints during training, and a best solution during testing. We use $\mathbf{w}$ to pick 200 top hypotheses and randomly choose another 200 hypothesis as our working set. For each of these 400 hypotheses, we adjust them in various ways to generate new hypothesis, in a randomized fashion. We allow the algorithm to change the space boundary positions, scene category, all object properties including category, location, rotation and size. We also allow removing some objects, or adding some objects. We iteratively run this step for several times, until we cannot find any better solutions in two consecutive iterations.

We show example outputs in **Figure 13**. Notice that our algorithm correctly detects many objects and recovers the 3D layout of a variety of scenes. In the first column, although our model fails to recognize the drawers, it manages to recognize a chair that is not labeled in the ground truth due to labeling error. In column 5, row 3 of **Figure 13** we see that our model mistakes the white carpet in front of the sofa as a table, which is a typical configuration for a living room. We also visualize the actual learned weights in **Figure 14**. This shows that the contextual cues are found to be very important for evaluating hypotheses.

7. CONCLUSION

We have proposed basic level scene understanding as a tractable research goal, and have summarized our recent effort to probe the

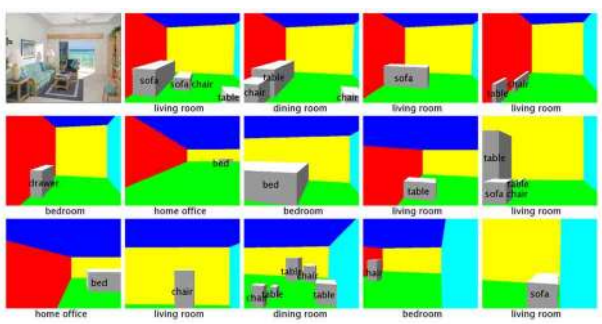

FIGURE 12 | Random subset of hypotheses generated for one image, ordered increasingly by their values of loss function $\Delta(\mathbf{y}_n, \hat{\mathbf{y}})$.

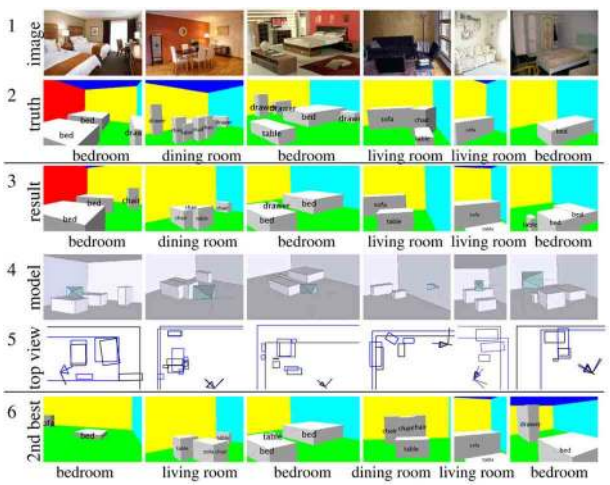

FIGURE 13 | Example results of our algorithm. The 1st row contains the input test image. The 2nd row contains the ground truth. The 3rd row contains a 2D rendering of our 3D parsing results. The 4th row is the rendering of the 3D mesh model and the camera frustum from a different viewpoint. The 5th row is the top view of the 3D parsing result: the *blue* lines are the ground *truth*, and the *black* lines are the *result*. The last row is the 2nd best prediction, which gives us an alternative parsing result.

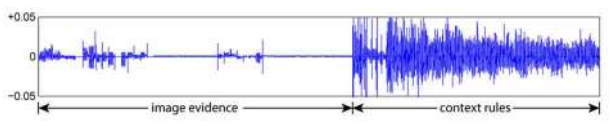

FIGURE 14 | Visualization of the w learned by the structural SVM. We group the weights into two separate categories roughly corresponding to image evidence and context constraints. We can see that the model puts larger weights on context features, which indicatives that they are discriminatively more informative. Note that we handle feature normalization carefully so that different features are at similar order of magnitude.

state of the art of several domains and questions related to visual scene understanding. First, we describe the richly annotated SUN database with object, attribute, and geometric labels for many scenes. This database allows us to systematically study the space of scenes and to establish a benchmark for scene and object recognition. Furthermore, we augment the categorical SUN database with millions of scene attribute labels and explore attribute recognition. Finally, we propose a unified framework for parsing a 3D scene to generate a basic level 3D representation. By modeling the task as a structural SVM problem, we train a complete end-to-end system in a single step by optimizing a unified objective function. With our proposed image and context rules, the SVM automatically weighs the relative importance of the rules based on training data. Current and future investigations are concerned with applications of the work to domains, such as image-based modeling (Xiao et al., 2008, 2009; Xiao and Quan, 2009; Xiao and Furukawa, 2012), viewpoint extrapolation (Xiao et al., 2012; Zhang et al., 2013), and assessment of subjective visual scene properties (Isola et al., 2011; Khosla et al., 2012a,b).

ACKNOWLEDGMENTS

Jianxiong Xiao was supported by Google U.S./Canada Ph.D. Fellowship in Computer Vision. Genevieve Patterson is supported by the Department of Defense (DoD) through the National Defense Science and Engineering Graduate Fellowship (NDSEG) Program. This work is funded by NSF grant (1016862) to Aude Oliva, Google research awards to Aude Oliva and Antonio Torralba, NSF Career Award (1149853) to James Hays, NSF Career Award (0747120) and ONR MURI N000141010933 and to Antonio Torralba.

REFERENCES

Barriuso, A., and Torralba, A. (2012). Notes on image annotation. *CoRR*. e-print: arXiv:1210.3448

Berg, T., Berg, A., and Shih, J. (2010). Automatic attribute discovery and characterization from Noisy Web data. *ECCV* 6311, 663–676. doi: 10.1007/978-3-642-15549-9_48

Coughlan, J. M., and Yuille, A. (1999). "Manhattan world: compass direction from a single image by bayesian inference," in *ICCV* (Kerkyra). doi: 10.1109/ICCV.1999.790349

Dalal, N., and Triggs, B. (2005). "Histograms of oriented gradients for human detection," in *CVPR* (San Diego, CA). doi: 10.1109/CVPR.2005.177

Delage, E., Lee, H., and Ng, A. Y. (2005). "Automatic single-image 3d reconstructions of indoor manhattan world scenes," in *ISRR* (San Francisco, CA).

Desai, C., Ramanan, D., and Fowlkes, C. (2011). Discriminative models for multi-class object layout. *IJCV*. 95, 1–12. doi: 10.1007/s11263-011-0439-x

Ehinger, K. A., Xiao, J., Torralba, A., and Oliva, A. (2011). "Estimating scene typicality from human ratings and image features," in *Proceedings of the 33rd Annual Conference of the Cognitive Science Society*, eds L. Carlson, C. Holscher, and T. Shipley (Austin, TX: Cognitive Science Society), 2114–2119.

Endres, I., Farhadi, A., Hoiem, D., and Forsyth, D. (2010). "The benefits and challenges of collecting Richer object annotations," in *ACVHL 2010 (in conjunction with CVPR)* (San Francisco, CA).

Farhadi, A., Endres, I., and Hoiem, D. (2010). "Attribute-centric recognition for cross-category generalization," in *CVPR* (San Francisco, CA). doi: 10.1109/CVPR.2010.5539924

Farhadi, A., Endres, I., Hoiem, D., and Forsyth, D. (2009). "Describing objects by their attributes," in *CVPR* (Miami, FL). doi: 10.1109/CVPR.2009.5206772

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *PAMI* 32, 1627–1645. doi: 10.1109/TPAMI.2009.167

Feng, C., Deng, F., and Kamat, V. R. (2010). "Semi-automatic 3d reconstruction of piecewise planar building models from single image," in *CONVR* (Sendai).

Greene, M., and Oliva, A. (2009). Recognition of natural scenes from global properties: seeing the forest without representing the trees. *Cogn. Psychol.* 58, 137–176. doi: 10.1016/j.cogpsych.2008.06.001

Gupta, A., Satkin, S., Efros, A. A., and Hebert, M. (2011). "From 3D scene geometry to human workspace," in *CVPR* (Colorado, CO).

Han, F., and Zhu, S.-C. (2005). "Bottom-up/top-down image parsing by attribute graph grammar," in *ICCV* (Beijing). doi: 10.1109/ICCV.2005.50

Hedau, V., Hoiem, D., and Forsyth, D. (2009). "Recovering the spatial layout of cluttered rooms," in *ICCV* (Kyoto). doi: 10.1109/ICCV.2009.5459411

Hedau, V., Hoiem, D., and Forsyth, D. (2010). "Thinking inside the box: using appearance models and context based on room geometry," in *ECCV* (Crete). doi: 10.1007/978-3-642-15567-3_17

Hedau, V., Hoiem, D., and Forsyth, D. (2012). "Recovering free space of indoor scenes from a single image," in *CVPR* (Providence, RI). doi: 10.1109/CVPR.2012.6248005

Heitz, G., Gould, S., Saxena, A., and Koller, D. (2008). "Cascaded classification models: combining models for holistic scene understanding," in *NIPS* (Vancouver, BC).

Hoiem, D. (2007). *Seeing the World Behind the Image: Spatial Layout for 3D Scene Understanding*. PhD thesis, Carnegie Mellon University.

Isola, P., Xiao, J., Torralba, A., and Oliva, A. (2011). "What makes an image memorable?" in *CVPR* (Colorado Springs, CO). doi: 10.1109/CVPR.2011.5995721

Joachims, T., Finley, T., and Yu, C.-N. J. (2009). Cutting-plane training of structural svms. *Mach. Learn.* 77, 27–59. doi: 10.1007/s10994-009-5108-8

Khosla, A., Xiao, J., Isola, P., Torralba, A., and Oliva, A. (2012a). "Image memorability and visual inception," in *SIGGRAPH Asia* (New York, NY). doi: 10.1145/2407746.2407781

Khosla, A., Xiao, J., Torralba, A., and Oliva, A. (2012b). "Memorability of image regions," *NIPS* 25, 305–313.

Kumar, N., Berg, A., Belhumeur, P., and Nayar, S. (2009). "Attribute and simile classifiers for face verification," in *ICCV* (Kyoto).

Lampert, C. H., Nickisch, H., and Harmeling, S. (2009). "Learning To detect unseen object classes by between-class attribute transfer," in *CVPR* (Miami, FL). doi: 10.1109/CVPR.2009.5206594

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. *CVPR* 2, 2169–2178. doi: 10.1109/CVPR.2006.68

Lee, D. C., Gupta, A., Hebert, M., and Kanade, T. (2010). Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. in *NIPS* (Vancouver, BC).

Lee, D. C., Hebert, M., and Kanade., T. (2009). "Geometric reasoning for single image structure recovery," in *CVPR* (Miami, FL). doi: 10.1109/CVPRW.2009.5206872

Li, C., Kowdle, A., Saxena, A., and Chen, T. (2012). Towards holistic scene understanding: feedback enabled cascaded classification models. *IEEE Trans. Pattern Anal. Mach.* 34, 1394–1408. doi: 10.1109/TPAMI.2011.232

Liu, J., Kuipers, B., and Savarese, S. (2011). "Recognizing human actions by attributes," in *CVPR* (Providence, RI). doi: 10.1109/CVPR.2011.5995353

Lowe, D. G. (2004). "Distinctive image features from scale-invariant keypoints," *IJCV* 60, 91–110. doi: 10.1023/B:VISI.0000029664.99615.94

Oliva, A., and Torralba, A. (2001). Modeling the shape of the scene: a holistic representation of the spatial envelope. *IJCV* 42, 145–175. doi: 10.1023/A:1011139631724

Oliva, A., and Torralba, A. (2002). "Scene-centered description from spatial envelope properties," in *2nd Workshop on Biologically Motivated Computer Vision (BMCV)* (Tübingen, Germany). doi: 10.1007/3-540-36181-2_26

Parikh, D., and Grauman, K. (2011). "Interactively building a discriminative vocabulary of nameable attributes," in *CVPR* (Providence, RI). doi: 10.1109/CVPR.2011.5995451

Patterson, G., and Hays, J. (2012). "SUN attribute database: discovering, annotating, and recognizing

scene attributes," in *CVPR* (Providence, RI). doi: 10.1109/CVPR.2012.6247998

Pero, L. D., Bowdish, J. C., Fried, D., Kermgard, B. D., Hartley, E. L., and Barnard, K. (2012). "Bayesian geometric modelling of indoor scenes," in *CVPR* (Providence, RI). doi: 10.1109/CVPR.2012.6247994

Pero, L. D., Guan, J., Brau, E., Schlecht, J., and Barnard, K. (2011). "Sampling bedrooms," in *CVPR* (Providence, RI). doi: 10.1109/CVPR.2011.5995737

Rabinovich, A., Vedaldi, A., Galleguillos, C., Wiewiora, E., and Belongie, S. (2007). "Objects in context," in *ICCV* (Rio de Janeiro). doi: 10.1109/ICCV.2007.4408986

Rosch, E. (1978). "Principles of categorization," in *Cognition and Categorization*, eds E. Rosch and B. B. Lloyd (Hillsdale, NJ: Erlbaum). Reprinted in: Margolis, E., and Laurence, S. (Eds.). (1999). *Concepts: Core Readings*. Cambridge, MA: MIT Press.

Russakovsky, O., and Fei-Fei, L. (2010). "Attribute learning in largescale datasets," in *ECCV 2010 Workshop on Parts and Attributes* (Crete). doi: 10.1007/978-3-642-35749-7_1

Sanchez, J., Perronnin, F., Mensink, T., and Verbeek, J. (2013). Image classification with the fisher vector: theory and practice. *Int. J. Comput. Vis.* doi: 10.1007/s11263-013-0636-x. [Epub ahead of print].

Saxena, A., Sun, M., and Ng, A. (2009). Make3d: learning 3d dscene structure from a single still image. *PAMI* 31, 824–840. doi: 10.1109/TPAMI.2008.132

Schwing, A. G., Hazan, T., Pollefeys, M., and Urtasun, R. (2012). "Efficient structured prediction for 3d indoor scene understanding," in *CVPR* (Providence, RI). doi: 10.1109/CVPR.2012.6248006

Shechtman, E., and Irani, M. (2007). "Matching local self-similarities across images and videos," in *CVPR* (Minneapolis, MN). doi: 10.1109/CVPR.2007.383198

Socher, R., Lin, C. C., Ng, A. Y., and Manning, C. D. (2011). "Parsing natural scenes and natural language with recursive neural networks," in *ICML* (Bellevue, WA).

Su, Y., Allan, M., and Jurie, F. (2010). "Improving object classification using semantic attributes," in *BMVC*. doi: 10.5244/C.24.26

Tardif, J.-P. (2009). "Non-iterative approach for fast and accurate vanishing point detection," in *ICCV* (Kyoto). doi: 10.1109/ICCV.2009.5459328

Toldo, R., and Fusiello, A. (2008). "Robust multiple structures estimation with j-linkage," in *ECCV* (Marseille, France). doi: 10.1007/978-3-540-88682-2_41

von Gioi, R. G., Jakubowicz, J., Morel, J.-M., and Randall, G. (2012). LSD: a line segment detector. *Image Process. Line* 2012. Available online at: http://www.ipol.im/pub/art/2012/gjmr-lsd/

Wang, H., Gould, S., and Koller, D. (2010). "Discriminative learning with latent variables for cluttered indoor scene understanding," in *ECCV* (Heraklion, Crete). doi: 10.1007/978-3-642-15561-1_36

Xiao, J., Ehinger, K., Oliva, A., and Torralba, A. (2012). "Recognizing scene viewpoint using panoramic place representation," in *CVPR* (Providence, RI). doi: 10.1109/CVPR.2012.6247991

Xiao, J., Fang, T., Tan, P., Zhao, P., Ofek, E., and Quan, L. (2008). "Image-based façade modeling," in *SIGGRAPH Asia* (New York, NY). doi: 10.1145/1457515.1409114

Xiao, J., Fang, T., Zhao, P., Lhuillier, M., and Quan, L. (2009). Image-based street-side city modeling. *ACM TOG* 28, 1. doi: 10.1145/1618452.1618460

Xiao, J., and Furukawa, Y. (2012). "Reconstructing the world's museums," in *ECCV* (Florence, Italy). doi: 10.1007/978-3-642-33718-5_48

Xiao, J., Hays, J., Ehinger, K., Oliva, A., and Torralba, A. (2010). "SUN database: large-scale scene recognition from abbey to zoo," in *CVPR* (San Francisco, CA). doi: 10.1109/CVPR.2010.5539970

Xiao, J., and Quan, L. (2009). "Multiple view semantic segmentation for street view images," in *ICCV* (Kyoto). doi: 10.1109/ICCV.2009.5459249

Yao, B., Jiang, X., Khosla, A., Lin, A. L., Guibas, L., and Fei-Fei, L. (2011). "Human action recognition by learning bases of action attributes and parts," in *ICCV* (Barcelona, Spain). doi: ieeecomputersociety.org/10.1109/ICCV.2011.6126386

Yu, L.-F., Yeung, S. K., Tang, C.-K., Terzopoulos, D., Chan, T. F., and Osher, S. (2011). Make it home: automatic optimization of furniture arrangement. *ACM Trans. Grap.* 30, 86. doi: 10.1145/2010324.1964981

Yu, S., Zhang, H., and Malik, J. (2008). "Inferring spatial layout from a single image via depth-ordered grouping," in *IEEE Workshop on Perceptual Organization in Computer Vision* (Anchorage, Alaska).

Zhang, Y., Xiao, J., Hays, J., and Tan, P. (2013). "Framebreak: dramatic image extrapolation by guided shift-maps," in *CVPR* (Portland, OR).

Zhao, Y., and Chun Zhu, S. (2011). "Image parsing with stochastic scene grammar," in *NIPS* (Vancouver, BC).

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 24 April 2013; accepted: 17 July 2013; published online: 29 August 2013.
Citation: Xiao J, Hays J, Russell BC, Patterson G, Ehinger KA, Torralba A and Oliva A (2013) Basic level scene understanding: categories, attributes and structures. Front. Psychol. ***4****:506. doi: 10.3389/fpsyg.2013.00506*
This article was submitted to Perception Science, a section of the journal Frontiers in Psychology.