A Coarse-to-Fine Model for 3D Pose Estimation and Sub-category Recognition

Roozbeh Mottaghi[1]*, Yu Xiang[2,3], and Silvio Savarese[3]
[1]Allen Institute for AI, [2]University of Michigan-Ann Arbor, [3]Stanford University

Abstract

Despite the fact that object detection, 3D pose estimation, and sub-category recognition are highly correlated tasks, they are usually addressed independently from each other because of the huge space of parameters. To jointly model all of these tasks, we propose a coarse-to-fine hierarchical representation, where each level of the hierarchy represents objects at a different level of granularity. The hierarchical representation prevents performance loss, which is often caused by the increase in the number of parameters (as we consider more tasks to model), and the joint modeling enables resolving ambiguities that exist in independent modeling of these tasks. We augment PASCAL3D+ [34] dataset with annotations for these tasks and show that our hierarchical model is effective in joint modeling of object detection, 3D pose estimation, and sub-category recognition.

1. Introduction

Traditional object detectors [33, 32, 7] usually estimate a 2D bounding box for the objects of interest. Although the 2D bounding box representation is useful, it is not sufficient. In several applications (e.g., autonomous driving or robotics manipulation), we need to reason about objects' 3D pose or viewpoint in addition to their bounding box location. Therefore, pose estimation methods [29, 25, 1] have been developed to provide a richer description for objects in terms of their viewpoint/pose. Fine-grained recognition methods [6, 36, 3] are another class of methods that also aim to provide richer descriptions since they enable more accurate reasoning about the detailed geometry and appearance of objects. Ideally, an object detector should estimate an object's location, its 3D pose and sub-category.

Note that these three tasks, namely object detection, 3D pose estimation, and sub-category recognition, are correlated tasks. For instance, learning an object model for *sedans* seen from a particular viewpoint is 'easier' than learning a model for general *cars* as the former forms a tighter cluster in the appearance space. On the other hand, more accurate localization of the object helps to better estimate its sub-category and viewpoint. Although these tasks are highly correlated, they are usually solved independently. One of the main issues in joint modeling of these tasks is that the number of parameters increases as we consider more tasks to model. This typically leads to requiring a larger number of images for training in order to avoid overfitting and performance loss compared to independent modeling. For instance, images of a particular type of *truck* taken from a certain viewpoint might be rare in the training set, hence learning a robust model for that might be difficult. This issue has been addressed in the literature by different techniques (for example, part sharing between different viewpoints [13, 35]). In this work, we take an alternative approach and leverage coarse-to-fine modeling.

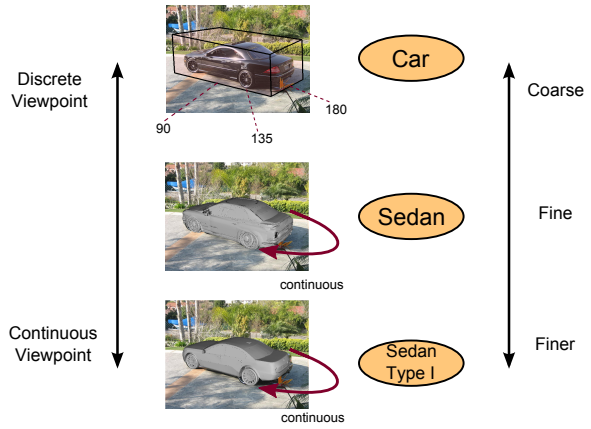


Figure 1. A coarse-to-fine hierarchical representation of an object. The top-layer captures high-level information such as a discrete viewpoint and a rough object location, while the layers below represent the object more accurately using continuous viewpoint, sub-category, and finer-sub-category information.

We propose a novel coarse-to-fine hierarchical model to represent objects, where each layer of the hierarchy represents objects at a different level of granularity. As shown in Figure 1, the coarsest level of the hierarchy reasons about the basic-level categories (e.g., *cars* vs. other categories) and provides a rough discrete estimate for the viewpoint. As we go down the hierarchy, the level of granularity changes, and more details are added to the model. For instance, for *car* recognition, at one level we reason about sub-categories

*The work was done while the first author was at Stanford University.

such as *SUV*, *sedan*, *truck*, etc., while at a finer level we distinguish different types of *SUVs* from each other. Also, we have a more detailed viewpoint representation (continuous viewpoint) in the layers below.

There are advantages of this coarse-to-fine hierarchical representation. First, tasks at different levels of granularity can benefit from each other. For instance, if there is ambiguity about the viewpoint of the object, knowing the sub-category might help resolving the ambiguity or reduce the uncertainty in viewpoint estimation. Second, different types of features are required for these three tasks. For instance, a feature that is most discriminative for distinguishing *cars* from other categories is not necessarily useful for distinguishing different types of *SUVs*. The hierarchical representation provides a principled framework to learn feature weights for different tasks jointly. Finally, we can better leverage the structure of the parameters so the performance does not drop as we increase the complexity of the model (or equivalently, the layers of the hierarchy).

Our hierarchical model is a hybrid random field as it contains discrete (e.g., sub-category) and continuous (e.g., continuous viewpoint) random variables. We employ a particle-based method to handle the mixture of continuous and discrete variables in the model. During learning, the parameters of the model in all layers of the hierarchy are estimated jointly. Inference is also a joint estimation of the object location, and its continuous viewpoint, sub-category and finer-sub-category.

For our experiments, we use PASCAL3D+ [34] dataset, which provides viewpoint annotations for rigid categories of PASCAL VOC 2012 dataset. To evaluate and train our model, for a subset of categories, we augment PASCAL3D+ with sub-category and finer-sub-category annotations. Our results show that our hierarchical model is effective in joint estimation of object location, 3D pose and (finer-)sub-category information. Also, the performance typically does not drop significantly or even improves as we increase the complexity of the model. Moreover, the hierarchical model provides significant improvement over a flat model that uses the same set of features.

2. Related Work

Hierarchical Models. Hierarchical models have been used extensively for object detection and recognition. [9] and [37] use hierarchies of object parts for object detection, where the parts in each layer are a composition of the parts in layers below. [26] discover a hierarchical structure to group objects based on common visual elements. [24] uses a hierarchy to share features between categories so they boost the recognition performance for categories with few training examples. We use a hierarchy as a unified model for 3D pose estimation, sub-category recognition, and object detection. The motivation, representation and the details of our model are different from the mentioned methods.

3D Pose Estimation. Several methods address the problem of object detection and pose estimation by incorporating 3D cues. Here we mention a few examples. Some of these methods, such as [28, 19], link parts across views, which allows a continuous viewpoint representation. [15, 13] treat 2D appearance and 3D geometry separately and combine them in a later stage. Hedau et al. [12] represent object appearance by a rigid template in 3D. Fidler et al. [8] extend that work by considering deformable faces. The methods mentioned above are limited to basic-level categorization, while we reason about sub-category information as well.

Sub-category Recognition. There is a considerable body of work on fine-grained categorization in the 2D recognition literature [6, 36, 3, 5, 18], which typically ignore reasoning about the 3D information. Recently, the 3D recognition community has shown that 3D object representation is beneficial for fine-grained categorization and vice versa. The work by [38] infers sub-categories in addition to the 3D pose. However, their sub-category recognition is performed as a post-processing step, while we perform that in a joint fashion. [16] also address the problem of viewpoint and sub-category estimation. However, they solve a binary classification problem (a particular sub-category vs. background), while we solve a multi-class problem, which is more challenging. [27] uses fine-grained category information to better understand a scene in 3D. [14] extends Spatial Pyramid Matching and Bubble Bank to 3D to perform fine-grained categorization and viewpoint estimation. [17] optimize fine-grained recognition and 3D model fitting jointly. [23] propose a transfer learning method for simultaneous object localization and viewpoint estimation and show that this transfer is beneficial for sub-category estimation. These methods suffer from one or more of the following issues. They assume the object bounding box is given, work only on clean images that do not contain any occlusion, cannot estimate continuous viewpoint or cannot estimate elevation of the camera or its distance from the object.

3. Coarse-to-fine Hierarchical Object Model

In this section, we describe our hierarchical model, which jointly performs object detection, 3D pose estimation, and sub-category recognition. The key intuition is that an object can be represented at different levels of granularity in different layers, where some constraints impose consistency across layers. We formulate the problem as learning and inference in a hybrid random field, which contains a mixture of discrete and continuous random variables. The hierarchy that we consider has three layers. The top layer (coarsest layer) captures coarse information, i.e., the object label (e.g., *aeroplane* or not) and also a coarse (dis-

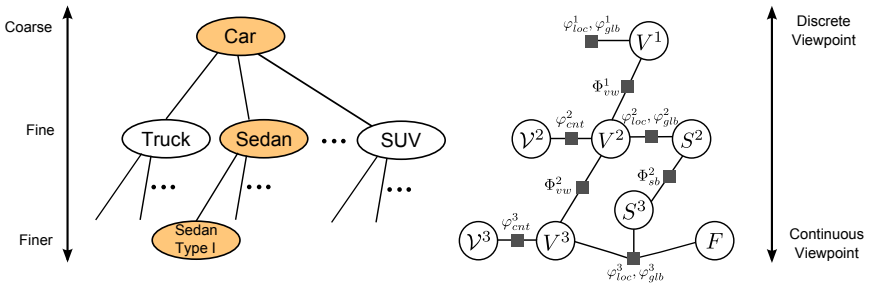


Figure 2. The graphical model of the hierarchy. For clarity, we have removed object node O. On the squares we have shown the potential functions defined on the nodes connecting to them. See text for the details.

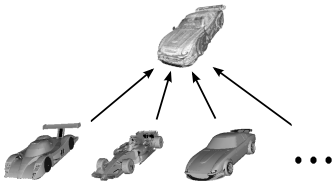


Figure 3. A coarse CAD model is made from the more detailed CAD models in the layers below. See text for more details.

cretized) viewpoint. This information is represented by a set of discrete random variables. The layer below in the hierarchy adds information about sub-category (e.g., *airline aeroplane*, *fighter aeroplane*, *etc.*) and also continuous viewpoint. Sub-category is represented by a discrete variable, while a continuous random variable represents the continuous viewpoint information. The bottom layer (or the finest layer) adds detailed information about the sub-categories that we refer to as finer-sub-category (e.g., a certain type of *airline aeroplane*). Viewpoint information is represented using a continuous random variable at this layer as well.

More formally, the binary random variable O represents the object label, where it will be equal to 1 if it is the object of interest and 0 otherwise. The coarse viewpoint is denoted by V^l, which takes values in the following discrete set of coarse viewpoints $\mathcal{A} = \{a_1, a_2, \ldots, a_m, b\}$, where m specifies the number of azimuth sections, and b represents background (no viewpoint should be associated to a background region). Therefore, each section covers $360/m$ degrees. The superscript l indexes the level in the hierarchy. The continuous viewpoint is denoted by $\mathcal{V}^l = (a, e, d, occ)$, which is decomposed into azimuth a, elevation e, distance (depth) d, and occlusion occ. We will describe these variables in more detail when we describe the potential functions defined on them. Another variable in the model is the sub-category variable S^l, which chooses a value from the set $\mathcal{S} = \{s_1, s_2, \ldots, s_n, b\}$, where n is determined according to the number of sub-categories we consider for an object category. Similarly, the random variable F represents the finer-sub-category in the model and selects a label in the set $\mathcal{F}_s = \{f_{s1}, f_{s2}, \ldots, f_{sp}, b\}$, where s indexes the subcategories and p indexes the finer-sub-categories of sub-category s.

3.1. Potential functions

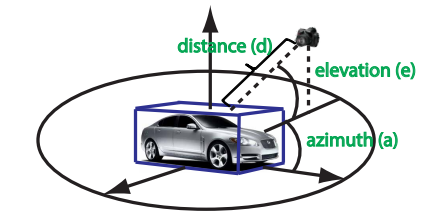
We now describe the potential functions defined for our three layer hierarchy. The level of the potential function is specified by the superscript l, e.g., φ^l. We have illustrated the graphical model for object O in Figure 2.

Global shape. We capture the global shape of the objects with HOG templates. We denote these potential functions as $\varphi^1_{glb}(V^1; \mathcal{R})$, $\varphi^2_{glb}(V^2, S^2; \mathcal{R})$, and $\varphi^3_{glb}(V^3, S^3, F; \mathcal{R})$. As mentioned above, V^l corresponds to the viewpoint and S^l and F denote the (finer-)sub-category information. Note that the term in the first layer of the hierarchy is a function of the viewpoint only, while in the layers below, it becomes a function of viewpoint and sub-category. These terms basically represent the HOG feature that we compute for region $\mathcal{R}$. Region $\mathcal{R}$ is a proposal bounding box in the image, which can be generated by methods such as [31].

Local appearance. We introduce these terms to capture local appearance information. For this purpose, we train a convolutional neural network (CNN) to compute the features used in the potential functions. We refer to them as 'local', because typically CNN units respond on portions of the objects and implicitly act as a 'part detector'. We use the CNN implementation of [10], but use only five convolutional layers to compute the features. We denote these terms by $\varphi^1_{loc}(V^1; \mathcal{R})$, $\varphi^2_{loc}(V^2, S^2; \mathcal{R})$, and $\varphi^3_{loc}(V^3, S^3, F; \mathcal{R})$ for the three layers of the hierarchy. Similar to above, the CNN features are computed on region $\mathcal{R}$.

Continuous viewpoint. The terms defined so far are based on a discretized viewpoint (discrete azimuth angle only). The azimuth angle alone is not sufficient to accurately represent the 3D pose of an object. This term in the energy function is computed based on the alignment of image data with the projection of a 3D CAD model. An advantage of using the 3D CAD models is that we can search for viewpoints not observed during training since the CAD models can be rendered from any viewpoint and also we can better reason about occlusions with 3D CAD models.

The potential function that we now define makes the connection between the continuous variable $\mathcal{V}^l$, which denotes the continuous viewpoint, and the discretized viewpoint V^l. The continuous viewpoint is a 4-tuple $\mathcal{V}^l = (a, e, d, occ)$. The range of azimuth angle a is $[0, 2\pi)$, while the elevation angle e is in the range $[0, \pi/2]$. The distance (depth) d corresponds to the distance of the camera from the object. The 3D pose of an object can be determined by these three parameters. For clarification, we show these parameters in



(a)

(b)

Figure 4. Parameters of the continuous viewpoint.

Figure 4(a). The last variable occ is for better handling of truncation and occlusion and it is described below.

The idea for using the occlusion variable occ is that we translate the projected CAD model in a neighborhood around the original point of projection (center of the bounding box), so it better fits the observation in the image. For instance, in Figure 4(b), if we translate the projection of the CAD model to the right, it will be better aligned with the truncated car. Basically, occ is a translation vector that moves the projection from the center of the bounding box (blue point) to a new location (green point).

The alignment between the projection of the CAD model and the observation in the image is computed as follows. We render the 3D CAD model onto the image according to $\mathcal{V}^l$. Then we compute HOG features on the contour (outline) of the projection and compare it with the HOG feature computed on region $\mathcal{R}$. We consider only the portion of projection that falls into $\mathcal{R}$.

The potential function is defined as:

$$\varphi^l_{cnt}(V^l, \mathcal{V}^l, C^l; \mathcal{R}) = \frac{1}{|\mathcal{R}|} \max_{\nu^l} \phi(P_{\nu^l, C^l})^T \phi(\mathcal{R}), \quad (1)$$

where $\phi(.)$ denotes the HOG feature and P_{ν^l, C^l} is the projection of the CAD model, C^l, according to ν^l. We perform normalization so this term does not depend on the scale of $\mathcal{R}$. ν^l is a set of samples that are generated according to the discrete viewpoint, and the one that maximizes the alignment between $\phi(P_{\nu^l, C^l})$ and $\phi(\mathcal{R})$ (described above) is chosen to compute the potential function. The samples of the continuous viewpoint variable are generated as follows: $\nu^l_a \sim \mathcal{N}(v^l; \sigma_a)$, $\nu^l_e \sim \mathcal{N}(\mu_e; \sigma_e)$, $\nu^l_{occ} \sim \mathcal{N}(\mathcal{R}_c; \sigma_{rx}, \sigma_{ry})$, where ν^l_a, ν^l_e, and ν^l_{occ} represent azimuth, elevation and the occlusion variable in the continuous viewpoint, respectively. v^l is one of the m discrete values in $\mathcal{A}$ (recall that the discrete viewpoint is only defined on the azimuth angle), μ_e is the average of elevations in training data, and $\mathcal{R}_c$ is the center of the proposal bounding box. We empirically set σ_a and $\sigma_{r.}$, and σ_e is computed from training data.

This sampling strategy allows us to make a connection between the continuous and discrete viewpoints. Note that solving for unconstrained continuous variables directly is difficult. The discrete variables somewhat constrain the values that the continuous variables can take. Furthermore, computing the right hand side of Equation 1 requires maximization over a continuous domain, which is not practical. Sampling makes this problem tractable as well.

The distance d is sampled differently from the other parameters. We use the following simple procedure for sampling the distance, but more sophisticated methods can be adopted instead. As shown in Figure 5, there is a correlation between distance d and size of the proposal box $\mathcal{R}$. During training, we know both distance and box size. During test, we have to estimate the distances given the proposal box size. We assign a weight to each training instance based on the difference in width and height of the training instances and the test instance (higher weight to smaller differences). We sample training instances according to these weights and use their distance d to form the set of distance samples.

A small proposal bounding box can correspond to a far away object or it can correspond to a nearby but truncated/occluded object. The distance sampling enables us to explore both of these possibilities.

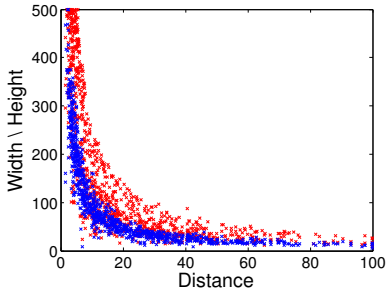


Figure 5. Correlation of object distance with the height and width (in pixels) of its 2D bounding box for *car* training instances. Width is shown in red and height in blue.

Now, the question is which 3D CAD model, C^l, should be selected for computing this term. For the bottommost layer of the hierarchy, we collect different CAD models to represent intra-class variation in a sub-category. For the mid layer, we combine the fine-grained CAD models in the lower layer to make a new CAD model, which captures generic shape properties of the object sub-category. For in-

stance, we combine all different types of race cars to make a coarse race car model (Figure 3). To combine the CAD models we scale them to the same size and orient them to a common direction. Then, we superimpose the CAD models and voxelize them. We keep only the voxels that vertices from a certain fraction of the CAD models fall into them.

Across layer consistency. To impose consistency between different layers we define a set of pairwise potentials. The discrete viewpoint should be the same across all layers. Also, the sub-category should be consistent across layers. So,

$$\Phi^l_{vw}(V^l, V^{l+1}) = \begin{cases} 1 & v^l = v^{l+1} \\ -\infty & \text{otherwise} \end{cases} \quad l = 1, 2 \qquad (2)$$

$$\Phi^l_{sb}(S^l, S^{l+1}) = \begin{cases} 1 & s^l = s^{l+1} \\ -\infty & \text{otherwise.} \end{cases} \quad l = 2 \qquad (3)$$

Note that we do not enforce direct consistency between continuous viewpoints, as they might be different depending on the level of granularity of the CAD model.

Top-level Detector. We use a pre-trained binary classifier that is applied to the proposal boxes and determines the confidence of a box belonging to the basic-level category of interest. In particular, we use the classifier of [10]. We denote this potential function by $\varphi_{det}(O; \mathcal{R})$.

3.2. Full energy function

The energy function is written as the sum of the energy functions in the three layers of the hierarchy:

$$E = \sum_{l=1}^{3} E^l = w_1 \varphi_{det} + \sum_{l=1}^{3} \left({\mathbf{w}_2^l}^T \varphi^l_{glb} + {\mathbf{w}_3^l}^T \varphi^l_{loc} \right) + \sum_{l=2}^{3} {\mathbf{w}_4^l}^T \varphi^l_{cnt} + \sum_{l=1}^{2} {\mathbf{w}_5^l}^T \Phi^l_{vw} + {\mathbf{w}_6^l}^T \Phi^2_{sb}, \qquad (4)$$

where $\mathbf{w}$'s are the parameters of the model that are estimated by the learning method described below.

4. Learning & Inference

As the result of inference on our model we can determine if a proposal box belongs to the category of interest and we also estimate its 3D viewpoint, sub-category, and finer-sub-category. Therefore, we find the configuration that maximizes $E(O, \{V^l\}, \{\mathcal{V}^l\}, \{S^l\}, F; \mathcal{R})$ given the weights $\mathbf{w}$ that are estimated during learning:

$$\left(O^*, \{V^{*l}\}, \{\mathcal{V}^{*l}\}, \{S^{*l}\}, F^*\right) = \underset{O, \{V^l\}, \{\mathcal{V}^l\}, \{S^l\}, F}{\operatorname{argmax}} E(O, \{V^l\}, \{\mathcal{V}^l\}, \{S^l\}, F; \mathcal{R}), \qquad (5)$$

where $l = 1, 2, 3$ for V^l, and $l = 2, 3$ for $\mathcal{V}^l$ and S^l.

Our inference method should estimate continuous and discrete variables in the model so we adopt an inference procedure that shares similarities with particle convex belief propagation (PCBP) [20]. The continuous variable in the model corresponds to the continuous viewpoint. First, we draw multiple samples around each discrete viewpoint. Basically, these samples can be considered as labels in a discretized MRF and allow us to compute the potential function defined in Eq. 1. After this step, the model can be considered as a fully discrete MRF and we can apply inference techniques for discrete MRFs. The advantage of particle methods is that they prevent committing to a fixed quantization of the state space. We can perform exact inference using exhaustive search since the number of possibilities is not too huge.

We use a structured SVM framework [30] to learn the weights in the model. Our positive training examples are a set of bounding boxes for the category of interest. In addition, we provide viewpoint as well as sub-category and finer-sub-category annotations for each example. The loss function Δ^l depends on the level of the hierarchy as well. We use Δ^1 to penalize mis-prediction of the viewpoints. Δ^2 penalizes sub-category mis-predictions and Δ^3 assigns a penalty to the incorrect predictions of the finer-sub-category. We perform loss augmented inference to find the most violating constraint. Note that each layer contributes its corresponding loss to the total loss. We use the 1-slack cutting plane implementation of [4] for the optimization. The details of the learning procedures are summarized in Algorithm 1.

input : Training examples: $\mathbf{x}_i = (o, v, \nu, s, f; \mathcal{R}) \quad i = 1, \ldots, N$
output: Estimated weights $\mathbf{w}_j$

Initialize weights $\mathbf{w}_j$ randomly;
for $t \leftarrow 1$ **to** *# of iterations* **do**
  foreach *training sample* $\mathbf{x}_i$ **do**
    foreach *layer* l **do**
      Compute the potentials defined based on the discrete variables: $\varphi_{det}, \varphi^l_{glb}, \varphi^l_{loc}, \Phi^l_{vw}, \Phi^l_{sb}$;
      foreach *possible discrete viewpoint* $v \in \mathcal{A}$ **do**
        Sample K continuous viewpoints ν (according to the sampling strategy in Section 3.1);
        foreach *sub-category or finer-sub-category (depending on the layer)* **do**
          Project the corresponding CAD model according to the sampled viewpoints;
          Compute the corresponding entry in φ^l_{cnt};
        end
      end
      Compute the loss function Δ^l (defined in Section 4);
    end
    Perform loss augmented inference to find the most violating constraint;
    Solve for $\mathbf{w}_j$ similar to the discrete SSVM;
  end
end

Algorithm 1: SSVM for our MRF, which is a mixture of continuous and discrete random variables.

	Bounding Box	All	Sub-category & Viewpoint	Sub-category	Viewpoint (8 views)
RCNN [10]	51.4	✘	✘	✘	✘
DPM-VOC+VP [22]	29.5	✘	✘	✘	21.8
V-DPM [7]	27.6	✘	✘	✘	16.2
SV-DPM [7]	**27.8**	✘	**8.4**	**13.8**	**18.2**
FSV-DPM [7]	25.8	0.35	7.9	12.7	16.1

Table 1. Results of variation of DPM [7], DPM-VOC+VP [22] and RCNN [10] on PASCAL3D+ [34] for all three or a subset of tasks.The result of DPM-VOC+VP [22] is adopted from [34]. The first column ('Bounding Box') is equivalent to the standard detection AP of PASCAL VOC. The meaning of ✘ is that the method is not capable of doing that task. We have shown the results averaged over classes.

	Bounding Box	All	Sub-category & Viewpoint	Sub-category	Viewpoint (8 views)
1-layer hierarchy (ours)	49.5	✘	✘	✘	28.9
2-layer hierarchy (ours)	51.0	✘	16.0	27.5	**29.5**
3-layer hierarchy (ours)	**51.6**	**3.2**	**17.6**	30.6	**29.5**
Flat model (ours)	51.6†	2.6	14.8	27.8	26.3
Separate (ours)	51.6†	1.9	16.1	**31.0**	28.7

Table 2. Results of variations our hierarchical model, a flat model that uses the same set of features as those of the 3-layer hierarchy, and also separate classifiers on PASCAL3D+ [34]. † We consider the same confidence values as those of the 3-layer model. So the bounding box detection results are identical.

5. Experiments

In this section, we demonstrate the result of our method for object detection, 3D pose estimation, and (finer-)sub-category recognition.

Dataset. For our experiments, we use PASCAL3D+ [34] dataset, which provides continuous viewpoint annotations for 12 rigid categories in PASCAL VOC 2012. We augment three categories (*aeroplane*, *boat*, *car*) of PASCAL3D+ with sub-category and finer-sub-category annotations. We consider 12, 12, and 60 finer-sub-categories for *aeroplane*, *boat*, and *car* categories, respectively. We group finer-sub-categories into 4, 4, and 8 sub-categories, respectively. For instance, the sub-categories we consider for *cars* are *sedan, SUV, truck, race*, etc., and the finer-sub-categories represent different types of *sedans* or *SUVs*. For the full list, refer to the supplementary material. For each finer-sub-category, we have a corresponding 3D CAD model, and for annotation we assign the instance in the image to the most similar CAD model. We use the `train` subset of PASCAL VOC 2012 for training, and the `val` subset for evaluation.

Implementation details. For generating proposal bounding boxes ($\mathcal{R}$) we use the method of [31], but any other method that produces object hypotheses can be used. The losses for the top layer (Δ^1) and the finest layer (Δ^3) are set to 0.1, and the mid-layer loss (Δ^2) is set to $0.3/K$, where K is the frequency of the sub-category in training data. The standard deviations used for sampling in Eq. 1 is computed as follows. σ_a is 1/3 of each azimuth section, σ_e is computed from training data, and σ_r is set to $0.15 \times L$, where L is the maximum of height and width of the proposal bounding box. We compute 5, 3, 2, 2 samples for azimuth, elevation, distance, and *occ*, respectively so we have 60 viewpoint samples in total. We set the C parameter of the structured SVM to 1. The inference takes about a minute per image on a single 3.0 GHz CPU. Most time is used to compute φ^l_{cnt} that requires rendering CAD models.

Results. We evaluate the three tasks using an evaluation method similar to average viewpoint precision (AVP) of [34]: we consider a box to be correct if the bounding box has more than 50% overlap with ground truth (the standard PASCAL detection criteria), and its viewpoint, sub-category, and finer-sub-category are estimated correctly as well. Therefore, the tasks are much more difficult than the standard bounding box localization. In the tables we show results for all tasks (referred to as 'All') as well as a subset of tasks. For example, for evaluating 'Sub-category & Viewpoint', we ignore if the finer-sub-category has been estimated correctly or not.

We report results for the tasks using various baseline methods. The first is RCNN [10] (refer to Table 1). For per-class results, refer to the supplementary material. Next we show the results of variations of DPM [7] in Table 1. V-DPM refers to the case that DPM mixture components correspond to different viewpoints (8 azimuth angles in this case). SV-DPM is the scenario that the mixture components represent both viewpoint and sub-categories (e.g., for *cars*, we consider 8 (viewpoints) $\times$ 8 (sub-categories) $=$ 64 components). Similarly, FSV-DPM considers finer-sub-categories as well (e.g., 60 finer-sub-categories for cars). Our purpose for providing these results is to illustrate the performance drop in all tasks when we compare the results of SV-DPM and FSV-DPM, which is due to the increase in the number of parameters or lack of training instances per component.

The result of our hierarchical model is shown in Table 2. We consider three scenarios, a one-layer hierarchy, which is only the coarse viewpoint layer, a two-layer hierarchy, and a three-layer hierarchy, which is our full model. Unlike the DPM case, we typically do not observe a performance

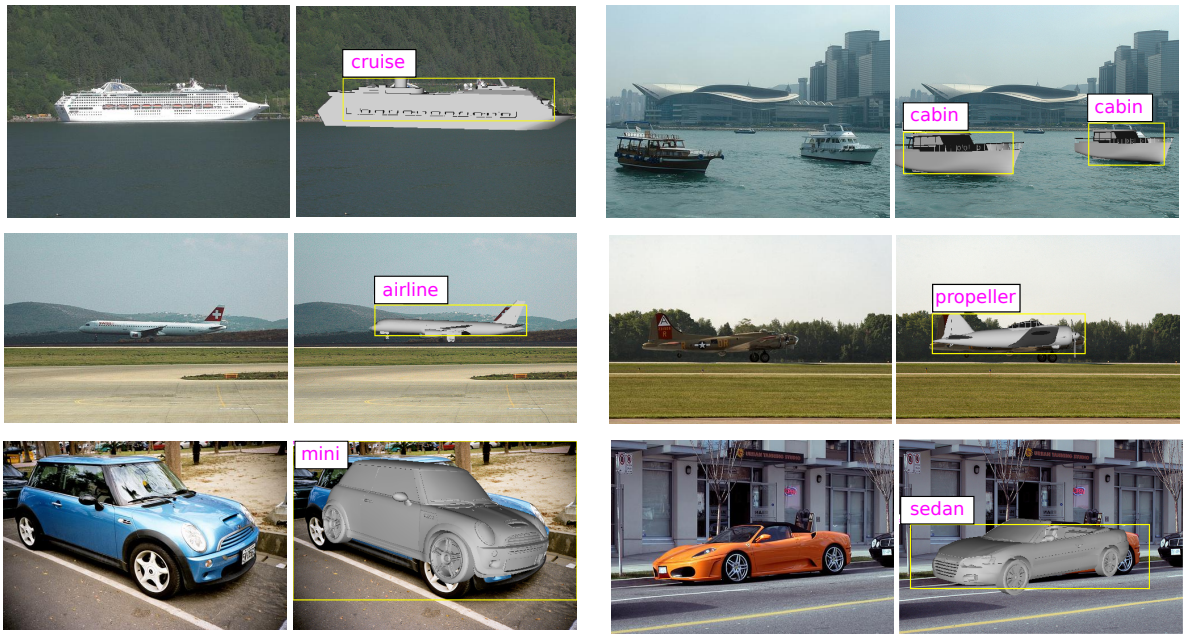


Figure 6. The result of object detection, 3D pose estimation, and (finer-)sub-category recognition. We show the projection of the 3D CAD model corresponding to the estimated finer-sub-categories according to the estimated continuous viewpoint. The magenta text is the estimated sub-category. Note that the 3D CAD model might not be the exact model for objects in PASCAL images.

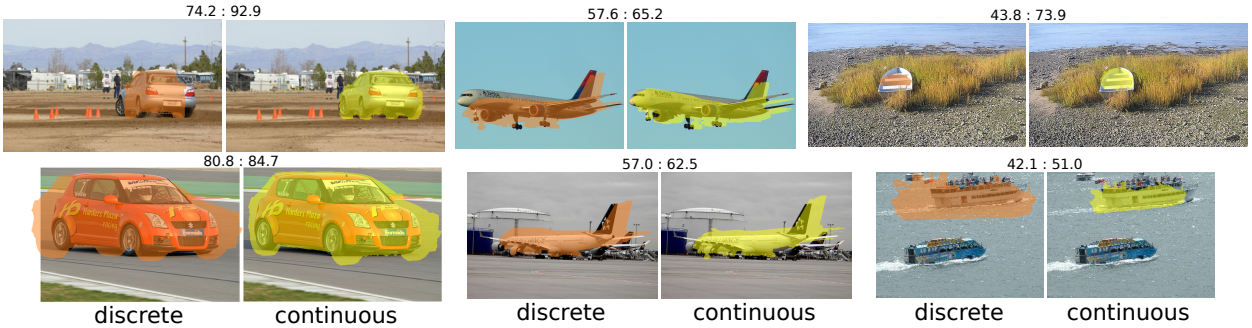


Figure 7. The left and the right image show the results of segmentation with the discrete and continuous versions of our model, respectively. The numbers on top are the corresponding intersection over union measures. Groundtruth segmentation mask is used to compute the overlap accuracy.

drop as we add more layers to the model. In some cases we see significant improvement. For instance, the result of sub-category recognition, and joint sub-category and viewpoint estimation improves by 3.1 and 1.6, respectively, for the 3-layer hierarchy compared to the 2-layer hierarchy. For detailed per-class results, refer to the supplementary material.

For the sake of comparison of viewpoint evaluations, we discretize the estimated continuous viewpoint into 8 azimuth angles. Note that the 1-layer hierarchy is already better than the current state-of-the-art (compare its results to DPM-VOC+VP [22] in Table 1, which is the state-of-the-art in viewpoint estimation) partially because of the powerful CNN features. Therefore, providing improvement over the first layer is not an easy task. Also, note that the performance for 'All' is quite low, which indicates the difficulty of modeling all tasks together. For instance, for *cars*, in addition to object detection, we should correctly infer one of the 8 azimuth angles, one of the 8 sub-categories, and one of the ~ 8 finer-sub-categories corresponding to the estimated sub-category. Figure 6 illustrates detection results for the 3-layer hierarchy.

Note that more supervision should not necessarily result in better accuracy. The reason is that we consider more tasks (viewpoint, subcategory, etc.) to model as we increase supervision. As the number of tasks increases, the space of parameters becomes huge, and learning the optimal parameters becomes much harder than the case where we model only a single task. Mainly due to this issue, most works on joint object detection and 3D pose estimation (e.g., [2] or [21]) are outperformed by DPM that uses less supervision for the single task of 'bounding box detection'. Note however that DPM is not capable of 3D pose estimation.

In Table 2, we also compare our hierarchical model to a flat model that uses the same set of features as those of the 3-layer hierarchy. The flat model is basically a lin-

CAD Alignment	3-layer discrete	3-layer continuous
aeroplane	50.5	**51.5**
boat	35.7	**40.3**
car	60.4	**64.4**

2D Segmentation	3-layer discrete	3-layer continuous
aeroplane	36.5	**37.4**
boat	35.6	**39.9**
car	61.4	**64.3**

Table 3. Segmentation results obtained by discrete and continuous versions of our model.

ear classifier whose output labels are joint viewpoint and (finer-)sub-categories, and it is applied to the proposal regions. The confidence values we obtain by the flat model are different from those of the hierarchy, which results in large performance difference (the flat model is significantly lower). To compare viewpoint and subcategory estimation irrespective of the confidence, for the flat case, we consider the same confidence (energy) as that of the 3-layer hierarchy. As shown in the table, the 3-layer hierarchy provides significant improvement over the flat model. Even for the difficult 'All' task we observe around 23% improvement. Table 2 also includes the results for separate classifiers i.e., we have a classifier for viewpoint, a separate classifier for sub-category and another set of classifiers for finer-sub-categories (unlike the flat model that is a joint classifier).

We computed the RMSE for estimating azimuth, elevation and distance. The results are shown in Table 4. Unfortunately, we cannot compare the results with other methods as other methods do not provide results for distance and elevation. We compare our method with [22] for different discretizations of the azimuth in Table 5. Note that our method is trained with 8 views. The confusion matrix for sub-category recognition for the *car* category is shown in Figure 8. The confusion matrices for other categories can be found in the supplementary material. Note that the AVP measure favors dominant categories and we chose the parameters such that we maximize AVP. Hence, the confusion matrix is biased towards *Sedan*, which is the dominant category.

Note that DPM [7], DPM-VOC-VP [22], or the flat model are classifiers for azimuth and it is impractical to incorporate other parameters of the continuous viewpoint into them since the output label space becomes huge. To show the advantage of our method that estimates continuous viewpoints over the discrete classifiers, we perform the following experiment. We project the CAD model corresponding to the estimated finer-sub-category according to the estimated continuous viewpoint and measure the intersection over union (IOU) of the projection mask with the groundtruth object mask. We consider two cases: 1) We use the projection of the groundtruth CAD given the groundtruth viewpoint as the groundtruth mask (referred to as 'CAD Alignment' in Table 3). 2) We use the groundtruth segmentation mask of [11] for evaluation (referred to as '2D Segmentation'). Unlike case (1), this case considers occlusion by external objects as well. The result is shown in the right hand side of Table 3.

In both cases, using continuous viewpoint provides a significant improvement over the discrete case of our model (evaluated based on the standard PASCAL segmentation criteria), which means our continuous viewpoint provides better alignment with the objects. Note that for this evaluation we consider only the true positive bounding boxes. By 'discrete version of our model', we mean the case that we ignore φ_{cnt} in the model. For the discrete case, we assume the elevation is equal to the mean of the elevations in training data and the distance is equal to the distance of the sample with the highest weight (refer to the distance sampling procedure in Sec. 3.1). Figure 7 shows some qualitative results.

RMSE	Azimuth (degree)	Elevation (degree)	Distance
Aeroplane	73.15	19.21	8.19
Boat	100.48	12.71	13.4
Car	73.16	6.59	11.25

Table 4. Continuous viewpoint estimation error.

AVP	4 views	8 views	16 views	24 views
3-layer hierarchy trained with 8 views	**32.7**	**29.5**	15.2	10.2
DPM-VOC+VP [22]	24.9	21.8	**15.3**	**12.2**

Table 5. Results for different discretization of azimuth.

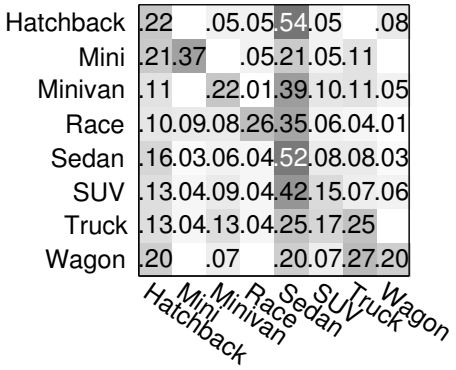


Figure 8. Confusion matrix for the sub-categories of the *cars*.

6. Conclusion

We proposed a novel coarse-to-fine hierarchy as a unified framework for object detection, 3D pose estimation, and sub-category recognition. We showed that our hierarchical model is effective in modeling these tasks jointly. Additionally, we showed that continuous viewpoint estimation (which is not practical for discrete classifiers) provides better alignment with the groundtruth object and significantly improves segmentation accuracy. We provided a new dataset that provides sub-category and finer-sub-category annotations for a subset of categories in PASCAL3D+ and used it to train and evaluate our model.

Acknowledgments We acknowledge the support of ONR grant N00014-13-1-0761 and NSF CAREER 1054127.

References

[1] M. Arie-Nachimson and R. Basri. Constructing implicit 3d shape models for pose estimation. In *ICCV*, 2009.

[2] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *CVPR*, 2014.

[3] T. Berg and P. N. Belhumeur. Poof: Part-based one-vs-one features for fine-grained categorization, face verification, and attribute estimation. In *CVPR*, 2013.

[4] C. Desai, D. Ramanan, and C. C. Fowlkes. Discriminative models for multi-class object layout. *IJCV*, 2011.

[5] K. Duan, D. Parikh, D. Crandall, and K. Grauman. Discovering localized attributes for fine-grained recognition. In *CVPR*, 2012.

[6] R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell, and L. S. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*, 2011.

[7] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2010.

[8] S. Fidler, S. Dickinson, and R. Urtasun. 3d object detection and viewpoint estimation with a deformable 3d cuboid model. In *NIPS*, 2012.

[9] S. Fidler and A. Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. In *CVPR*, 2007.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[11] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *ICCV*, 2011.

[12] V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *ECCV*, 2010.

[13] M. Hejrati and D. Ramanan. Analyzing 3d objects in cluttered images. In *NIPS*, 2012.

[14] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *3dRR Workshop*, 2013.

[15] J. Liebelt and C. Schmid. Multi-view object class detection with a 3d geometric model. In *CVPR*, 2010.

[16] J. Lim, H. Pirsiavash, and A. Torralba. Parsing ikea objects: Fine pose estimation. In *ICCV*, 2013.

[17] Y.-L. Lin, V. I. Morariu, W. Hsu, and L. S. Davis. Jointly optimizing 3d model fitting and fine-grained classification. In *ECCV*, 2014.

[18] O. M. Parkhi, A. Vedaldi, C. V. Jawahar, and A. Zisserman. Cats and dogs. In *CVPR*, 2012.

[19] N. Payet and S. Todorovic. From contours to 3d object detection and pose estimation. In *ICCV*, 2011.

[20] J. Peng, T. Hazan, D. McAllester, and R. Urtasun. Convex max-product algorithms for continuous mrfs with applications to protein folding. In *ICML*, 2011.

[21] B. Pepik, P. Gehler, M. Stark, and B. Schiele. 3d2pm - 3d deformable part models. In *ECCV*, 2012.

[22] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3d geometry to deformable part models. In *CVPR*, 2012.

[23] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Multi-view priors for learning detectors from sparse viewpoint data. In *ICLR*, 2014.

[24] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *CVPR*, 2011.

[25] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. In *ICCV*, 2007.

[26] J. Sivic, B. C. Russell, A. Zisserman, W. T. Freeman, and A. A. Efros. Unsupervised discovery of visual object class hierarchies. In *CVPR*, 2008.

[27] M. Stark, J. Krause, B. Pepik, D. Meger, J. J. Little, B. Schiele, and D. Koller. Fine-grained categorization for 3d scene understanding. In *BMVC*, 2012.

[28] H. Su, M. Sun, L. Fei-Fei, and S. Savarese. Learning a dense multiview representation for detection, viewpoint classification and synthesis. In *ICCV*, 2009.

[29] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. V. Gool. Towards multi-view object class detection. In *CVPR*, 2006.

[30] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004.

[31] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulder. Selective search for object recognition. *IJCV*, 2013.

[32] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009.

[33] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.

[34] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *WACV*, 2014.

[35] Y. Xiang and S. Savarese. Estimating the aspect layout of object categories. In *CVPR*, 2012.

[36] B. Yao, G. Bradski, and L. Fei-Fei. A codebook-free and annotation-free approach for fine-grained image categorization. In *CVPR*, 2012.

[37] L. Zhu, C. Lin, H. Huang, Y. Chen, and A. Yuille. Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion. In *ECCV*, 2008.

[38] Z. Zia, M. Stark, B. Schiele, and K. Schindler. Detailed 3d representations for object recognition and modeling. *PAMI*, 2013.