This ICCV2013 paper is the Open Access version, provided by the Computer Vision Foundation.
The authoritative version of this paper is available in IEEE Xplore.

# Building Part-based Object Detectors via 3D Geometry

Abhinav Shrivastava Abhinav Gupta
The Robotics Institute, Carnegie Mellon University
http://graphics.cs.cmu.edu/projects/gdpm

## Abstract

*This paper proposes a novel part-based representation for modeling object categories. Our representation combines the effectiveness of deformable part-based models with the richness of geometric representation by defining parts based on consistent underlying 3D geometry. Our key hypothesis is that while the appearance and the arrangement of parts might vary across the instances of object categories, the constituent parts will still have consistent underlying 3D geometry. We propose to learn this geometry-driven deformable part-based model (gDPM) from a set of labeled RGBD images. We also demonstrate how the geometric representation of gDPM can help us leverage depth data during training and constrain the latent model learning problem. But most importantly, a joint geometric and appearance based representation not only allows us to achieve state-of-the-art results on object detection but also allows us to tackle the grand challenge of understanding 3D objects from 2D images.*

## 1. Introduction

While object detection remains one of the most stubbornly difficult problems in computer vision, substantial progress has been made in the last decade, as evidenced by steadily improving detection rates for common categories such as faces, cars, etc. However, there are reasons to worry that current advancements might be reaching a plateau. Consider the popular PASCAL object detection benchmark [12]: after rapid gains early on, detection performance has stagnated for most object classes at levels still too low for practical use (e.g., bird, sofa and chair categories are all below 20% AP). Interestingly, the standard trick of boosting performance by increasing the size of the training set does not seem to be working any longer: Zhu et al. [33] report training a standard detector on 10 times more data without seeing any improvement in performance. This indicates the need for better models and learning approaches to handle the intra-class variability of object categories.

At the forefront of detection research has been the deformable part-based model (DPM) [13] which has consistently achieved state-of-the-art performance in object detection tasks. It models objects as a constellation of parts where the parts are defined in an unsupervised manner based on heuristics such as high gradient energy. This part-based model is trained discriminatively; however, learning this model is a complex task as it involves optimization of a non-convex function over a set of latent variables (part locations and mixture memberships). In some cases, the parts in the DPM have shown little or no improvement [10]. Due to these reasons, recent work has focused on using strongly-supervised part models [1] where semantically meaningful part annotations are used to initialize the parts and improve the learning process. However, using semantically meaningful parts has two major problems: (1) manually labeling these semantic parts can be quite cumbersome and requires

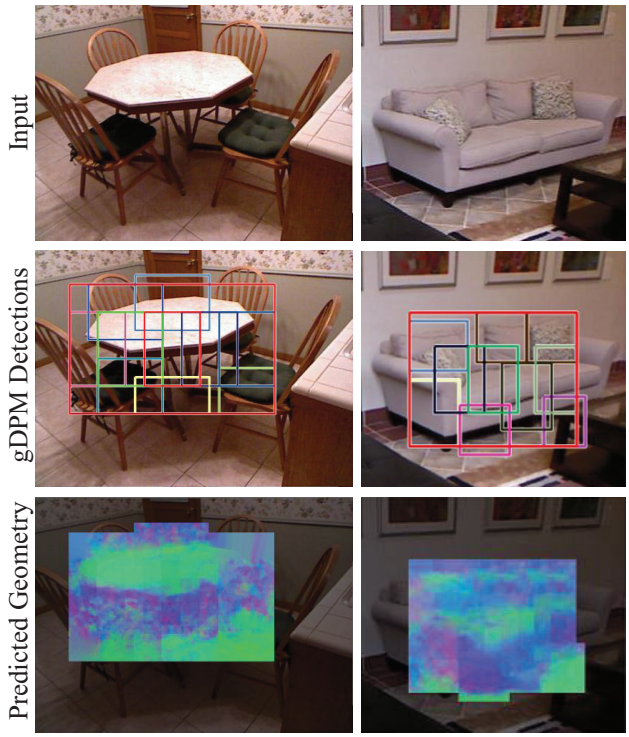



Figure 1. Examples of object detection and surface normal prediction using the proposed gDPM model. Our gDPM not only improves the state of the art performance in object detection but it also predicts the surface normals with the detection. Legend for normals: blue: X; green: Y; red: Z.

a lot of human effort; (2) more importantly, unlike articulated objects, for many object categories, such as beds, it is not even clear what a semantic part corresponds to.

We propose a geometry-driven deformable part-based model (gDPM) that can be learned from a set of labeled RGBD images. In a gDPM, object parts are defined based on their physical properties (i.e., their geometry) rather than just their appearance. Our key hypothesis is that while the arrangement of parts might vary across the instances of object categories, the constituent parts will still have consistent underlying 3D geometry. For example, every sofa has a L-shaped part that is the intersection of a vertical surface and a horizontal surface for sitting. Therefore, the underlying 3D geometry can provide weak supervision to define and initialize the parts. While the learning objective in case of gDPM is still non-convex (similar to [13]), we show how the depth data can be used as weak supervision to impose geometric constraints and guide latent updates at each step. Empirically this leads to faster convergence, and a better model in terms of detection performance. But more importantly, because our parts have a 3D geometrical representation they can be used to jointly detect objects and infer 3D properties from a single 2D image. Figure 1 shows two examples of objects detected by our gDPM model and the predicted surface normal geometry by the gDPM. Notice how our approach predicts nicely aligned flat horizontal surface of the table within the bounding box and how the approach predicts the horizontal and vertical surfaces of the couch.

**Contributions:** Our key contributions include: (1) We propose to marry deformable part-based model with the geometric representation of objects by defining parts based on consistent underlying 3D geometry. (2) We demonstrate how the geometric representation can help us leverage depth data during training and constrain the latent model learning problem. The underlying 3D geometry during training helps us guide the latent steps in the right direction. (3) Most importantly, a joint geometric and appearance based representation not only allows us to achieve state-of-the results on object detection but also allows us to tackle the grand challenge of understanding 3D objects from 2D images.

## 2. Related Work

The idea of using geometric and physical representation for objects and their categories has a rich history in computer vision [5, 23, 24]. While these approaches resulted in some impressive demos such as ACRONYM [6], these systems failed to generalize. That led us to the modern era in computer vision where instead of representing objects in 3D, the focus changed to representing objects using low-level image features such as HOG [9] and using machine learning to learn an appearance model of the object. The most successful approaches in this line of work are the deformable part-based models [13] that extend the rigid template from [9] to a latent part-based model that is trained discriminatively. While there has been a lot of progress made over the last decade, the performance of these appearance based approaches seems to have been stagnated.

Therefore, recent research has now focused on developing richer representations for objects and effective ways of learning these representations. Most of the recent work on improving deformable part models can be broadly divided in two main categories:

**(a) Better 2D Representations and Learning:** The first and the most common way is to design better representations using 2D image features. In this area, researchers have looked into using strongly-supervised models for parts [1, 4, 11, 32], using key point annotations to search for parts [3] or discovering mid-level parts in a completely unsupervised manner [28]. Other directions include using sharing to increase data-size across categories [22] or finding visual subcategories based on appearances before learning a part-based model [7, 10].

**(b) Using 3D Geometry:** The second direction that has been explored is to bring back the flavor of the past and develop rich models by representing 3D geometry explicitly. One of the most common ways to encode viewpoint information is to train a mixture of templates for different viewpoints [17]. An alternative approach is to explicitly consider the 3D nature of the problem and model objects as a collection of local parts that are connected across views [16, 27, 29, 30]. Another way to account for 3D representation is to explicitly model the 3D object in terms of planes [8, 14, 27, 31] or parts [26], and use a rigid template [18], spring model [14] or a CRF [8].

Our approach lies at the intersection of two these directions. Unlike other approaches which incorporate geometry in DPM via CAD models [26] or manually-labeled 3D cuboids [14, 18], our approach uses noisy depth data for training (similar to [15]). This allows us to access more and diverse data (hundreds of images compared to 40 or so CAD models). The scale at which we build 3D priors and do geometric reasoning during latent learning allows us to obtain improvements of as much as 11% in some categories (previous approaches performed at-par or below DPM). We would also like to point out that even though our approach uses depth information during training, it is used as a weak supervisory signal (and not as an extra input feature) to guide the training in the right direction. The discriminative model is only learned in the appearance space. Therefore, we do not require depth at test time and can use gDPM to detect objects in RGB images. Most other work in object detection/recognition using RGBD [2, 20, 21] uses depth as an extra input feature to learn an object model and therefore, also requires depth information at test time.

## 3. Overview

As input to the system, at training, we use RGB images of object instances along with their underlying geometry in terms of depth data. We convert the depth data into surface normals using the standard procedure from [25]. Our goal is to learn a deformable part-based model where the parts are defined based on their appearance and underlying geometry. We argue that using a geometric representation in conjunction with appearance based deformable parts model not only allows us to have a better initialization but also provides additional constraints during the latent update steps. Specifically, our learning procedure ensures not only that the latent updates are consistent in the appearance space but also that the geometry predicted by underlying parts is consistent with the ground truth geometry. Hence, the depth data is not used as an extra feature, but instead provides weak supervision during the latent update steps.

In this paper, we present a proof-of-concept system for building gDPM. We limit our focus on man-made indoor rigid objects, such as bed, sofa etc., for three reasons: (1) These classes are primarily defined based on their physical properties, and therefore learning a geometric model for these categories makes intuitive sense; (2) These classes have high intra-class variation and are challenging for any deformable parts model. We would like to demonstrate that a joint geometric and appearance based representation gives us a powerful tool to model intra-class variations; (3) Finally, due to the availability of Kinect, data collection for these categories has become simpler and efficient. In our case, we use the NYU v2 dataset [25], which has 1449 RGBD images.

## 4. Technical Approach

Given a large set of training object instances in the form of RGBD data, our goal is to discover a set of candidate parts based on consistent underlying geometry, and use these parts to learn a geometry-driven deformable part-based model (gDPM). To obtain such a set of candidate parts, we first discover a dictionary of geometric elements based on their depth information (section 4.1) in a category-free manner (pooling the data from all categories). A category-free dictionary allows us to share the elements across multiple object categories.

We use this dictionary to choose a set of parts for every object category based on frequency of occurrence and consistency in the relative location with respect to the object bounding-boxes. Finally, we use these parts to initialize and learn our gDPM using latent updates and hard mining. We exploit the geometric nature of our parts and use them to enforce additional geometrical constraints at the latent update steps (section 4.3).

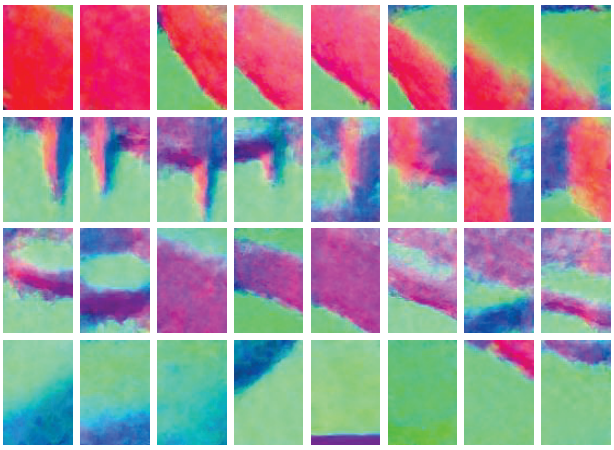


Figure 2. A few elements from the dictionary after the initialization step. They are ordered to highlight the over-completeness of our initial dictionary.

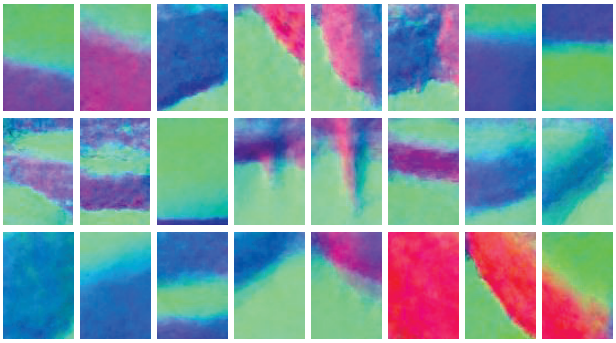


Figure 3. A few examples of resulting elements in dictionary after the refinement procedure.

### 4.1. Geometry-driven Dictionary of 3D Elements

Given a set of labeled training images and their corresponding surface normal data, our goal is to discover a dictionary of elements capturing 3D information that can act as parts in DPM. Our elements should be: 1) representative: frequent among the object categories in question; 2) spatially consistent with respect to the object. (e.g., a horizontal surface always occurs on the top of a table and bed, while it occurs at center of a chair and a sofa). To obtain a dictionary of candidate elements which satisfy these properties, we use a two step process: first we initialize our dictionary by an over-complete set of elements, each satisfying the representativeness property; and then we refine the dictionary elements based on their relative spatial location with respect to the object.

**Initializing the dictionary**: We sample hundreds of thousands of patches, in 3 different aspect-ratios (AR), from the object bounding boxes in the training images ($100 - 500$ patches per object bounding box). We represent these patches in terms of their raw surface normal maps. To extract a representative set of elements for each AR, we perform clustering using simple $k$-means (with $k \sim 1000$), in raw surface normal space. This clustering process leads to

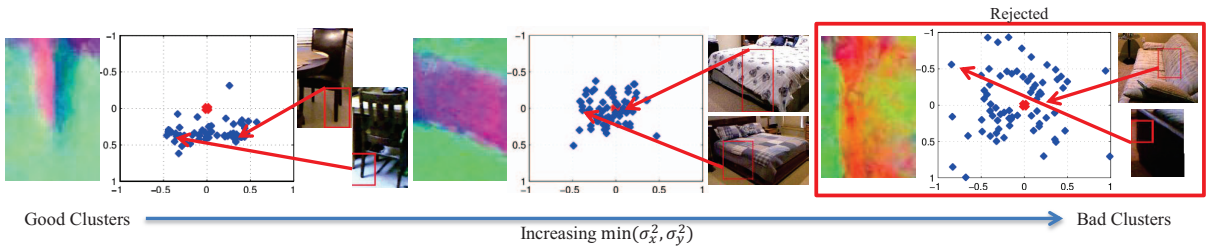



Figure 4. Refinement: After creating an initial dictionary we do the refinement procedure where we find the set of elements that occur at a consistently occur at same spatial location with respect to the object center.

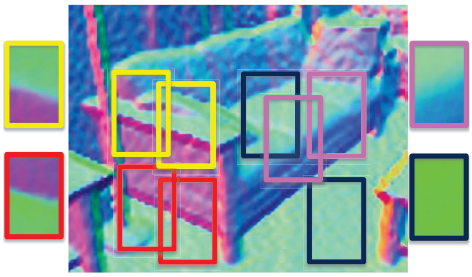


Figure 5. An example of detection/localization of discovered dictionary elements in surface normal space.

an over-complete set of geometric elements. We remove any cluster with less than $N$ members, for not satisfying the frequency property. We represent every remaining cluster by an element which is the pixel-wise median of the nearest $N$ patches to the cluster center. A few examples of this set of elements is shown in Figure 2. (See the website for more elements and other AR clusters.) In practice, we use $N = 25$. As one can notice from the figure, the dictionary is over-complete. To reject the clusters with bad (non-homogenous) members and to remove redundancy we follow this clustering step with a refinement procedure.

**Refinement**: Given the clusters from the initialization step, we first check each cluster for spatial consistency, i.e., how consistent the cluster is with respect to the center of the object. For this, we record the location of each member in the cluster relative to the object center as: $(dx^i, dy^i) = \left(\frac{(p_x^i - p_x^o)}{w^o}, \frac{(p_y^i - p_y^o)}{h^o}\right)$, where $p^o$, $w^o$ and $h^o$ are the object center, width and height respectively, and $p^i$ is the center of element $i$. Examples of this voting scheme are given in Figure 4, where each blue dot represents a vote from the cluster's member, and red dot represents object center. To capture consistency in relative locations, we sort the clusters based on $\min(\sigma_x^2(dx), \sigma_y^2(dy))$ (minimum variance of their relative $x$, $y$ locations). Clusters like the legs of furniture (consistently below the object and closer to the center) and sides of a bed (consistently near the center of object) rank much higher than noisy cluster shown at the right. After pruning bad clusters by thresholding, we perform a step of agglomerative clustering to merge good clusters which are close in feature space (raw surface normals) as well as have consistent distribution of $(dx, dy)$. This gives us a dictionary $\mathcal{D}$ of 3D elements. A few examples of resulting elements are shown in Figure 3.

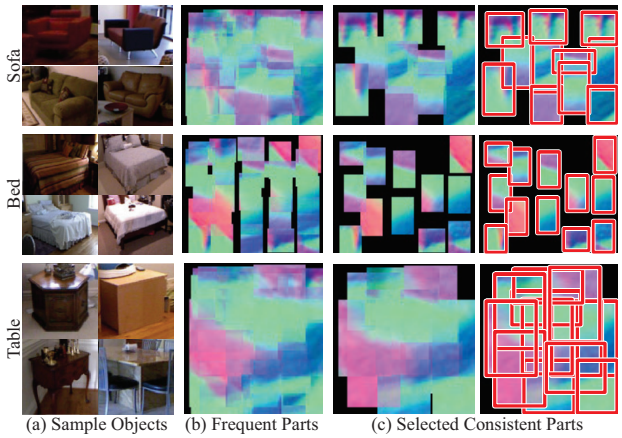



Figure 6. From 3D Parts to object hypothesis: (a) few examples images in the cluster; (b) all the geometrically consistent candidate parts selected (before greedy selection); (c) final part hypothesis for initializing gDPM (after greedy selection)

### 4.2. From 3D Parts to Object Hypothesis

Given a dictionary of geometric elements $\mathcal{D}$, we would like to discover which geometric elements can act as parts for which object categories. Since our categories share the geometric elements, every element in the dictionary can act as a part for any number of object categories. We represent a part $p^j$ for an object category with three aspects: (a) the geometric element $e_i \in \mathcal{D}$; (b) the relative location $l^j : (dx^j, dy^j)$ of the part with respect to object center (in normalized coordinates); (c) the spring model (or variance in $(dx, dy)$) for the part, which defines how spatially consistent a part is with respect to the object. Note that an object part is different from the geometric element and a geometric element can act as different parts based on the location (e.g., two armrests for the chair; an armrest is a geometric element but two different parts). The goal is to find set of parts (or an object hypothesis) $\mathbf{p} = [p^1, \ldots, p^N]$, where $p^j : (e_i, l^j)$, that occur consistently in the labeled images.

Similar to DPM [13], we represent each object category as a mixture of components and each component is loosely treated as a category of its own. Therefore, our goal is to find a set of parts for each component of all object categories. Given a set of training images for a component, we first localize each element $e$ in the surface normal map. For

example, Figure 5 shows the elements detected in the case of a sofa. We then pool the element localizations from all images and find the most frequent elements at different locations in an object. These frequent elements act as candidate parts for representing an object. Figure 6(b) shows the candidate parts for one component of three categories: bed, sofa and table.

We now use a greedy approach to select the final parts with the constraints that we have 6-12 parts per object component and that these parts cover at least 60% of the object area. At each step, we select the top-most part hypothesis based on the frequency of occurrence and consistency in the relative location with respect to the object. Therefore, if a geometric element occurs quite frequently at a particular location, then it is selected as a part for the object. Once we have selected a part, the next part is selected based on frequency and consistency of occurrence, and its overlap with the already selected parts (a part that overlaps a lot with already selected parts is rejected).

## 4.3. Learning gDPM

Once we have obtained a set of parts for a given object category, we can now use it to initialize the learning of our proposed gDPM model. Following the general framework of deformable part models [1, 11, 13, 32], we model an object by a mixture of $M$ components, each of which is a non-rigid star-shaped constellation of parts. The key difference between learning the gDPM and the original DPM lies in the scoring function. Unlike the original model which only captures appearance and location of parts, we explicitly include a geometric consistency term in the scoring function used at the latent update step. This allows us to enforce geometric consistency across the latent update steps and guide the latent updates in the right direction. We will now first discuss a few preliminaries about DPM and then discuss how we add the geometric consistency term to the scoring function.

**DPM Preliminaries** For each mixture component, indexed by $c \in \{1, \ldots, M\}$, the object hypothesis is specified by $z = (l_0, l_1, \ldots, l_{n_c})$, where $l_i = (u_i, v_i, s_i)$ denotes the $(u, v)$-position of i-th filter (every part acts a filter) at level $s_i$ in the feature pyramid (root is indexed at 0, and $l_0$ corresponds to its bounding-box) and $n_c$ is number of parts in component $c$. Following [13], we enforce that each part is at twice the resolution of the root.

The score of a mixture component $c$, with model parameter $\beta_c$, at any given $z$ (root and part locations) in an image $I$ is given by

$$S(I, z, \beta_c) = \sum_{i=0}^{n_c} F_i \cdot \phi(I, l_i) - \sum_{i=1}^{n_c} d_i \cdot \psi(l_i - l_0) + b \quad (1)$$

where the first term scores appearance using image features $\phi(I, l_i)$ (HOG features in this case) and model's appearance parameters $(F_0, \ldots, F_{n_c})$. The second term enforces the deformation penalty using $\psi(l_i - l_0) = \{dx, dy, dx^2, dy^2\}$ where $(dx, dy) = (l_i^x, l_i^y) - (2(l_0^x, l_0^y) - v_i)$ and $v_i$ is the anchor position of the part. Thus, each component's model parameter is $\beta_c = \{F_0, \ldots, F_{n_c}, d_1, \ldots, d_{n_c}, b\}$.

The final score of a DPM model for an object category on an image $I$ at any $z$ is given by

$$S(I, z) = \max_{c \in \{1 \ldots M\}} S(I, z, \beta_c), \quad (2)$$

which is the maximum over scores of all the components. Thus, the final object model parameter is $\beta = (\beta_1, \ldots, \beta_M)$ which encapsulates all $M$ mixture components.

### 4.3.1 Enforcing Geometric Constraints & Learning

Given the training data $\{(x_i, y_i)\}_{1,\ldots,N}$, we aim to learn a discriminative gDPM. In our case, $x = \{I, I^G, l\}$, where $I$ denotes an RGB image, $I^G$ denotes the surface normal map and $l$ is location of the bounding box, and $y \in \{-1, 1\}$. Similar to [1, 11, 13, 32], we minimize the objective function:

$$L_D(\beta) = \frac{1}{2}\|\beta\|^2 + C \sum_{i=1}^{N} \max(0, 1 - y_i f_\beta(x_i)), \quad (3)$$

$$f_\beta(x) = \max_z S(I, z) = \max_{z,c} S(I, z, \beta_c). \quad (4)$$

The latent variables, $z$ (root and part locations) and $c$ (mixture memberships), make (3) non-convex. [13] solves this optimization problem using a coordinate-descent based approach, which iterates between a latent update step and a parameter learning step. In the latent update step, they estimate the latent variables, $z$ and $c$, by relabeling each positive example. In the parameter learning step, they fix the latent variables and estimate the model parameter $\beta$ using stochastic gradient descent (SGD).

The latent updates in [13] are made based on image appearance only. However, in our case, we also have a geometric representation of our parts and the underlying depth data for training images. We exploit this and constrain the latent update step such that the part geometry should match the underlying depth data. Intuitively, depth data provides part-level geometric supervision to the latent update step. Thus, enforcing this constraint only affects the latent update step in the above optimization. This is achieved by augmenting the scoring function $S(I, z, \beta_c)$ with a geometric consistency term:

$$f_\beta(x) = \max_{c \in \{1 \ldots M\}, z} \left[ S(I, z, \beta_c) + \lambda \sum_{i=1}^{n_c} S_G(e^i, \omega(I^G, l_i)) \right] \quad (5)$$

where $e^i$ is the geometric element (raw surface normal) corresponding to i-th part, $\omega(I^G, l_i)$ is the raw surface normal map extracted at location $l_i$, $S_G(\cdot)$ is the geometric similarity function between two raw surface normal maps and $\lambda$ is the trade-off parameter, controlling how much we want the optimization to focus on geometric consistency. We train our gDPM models using a modified version of the Latent SVM solver from [13]. In our coordinate-descent approach,

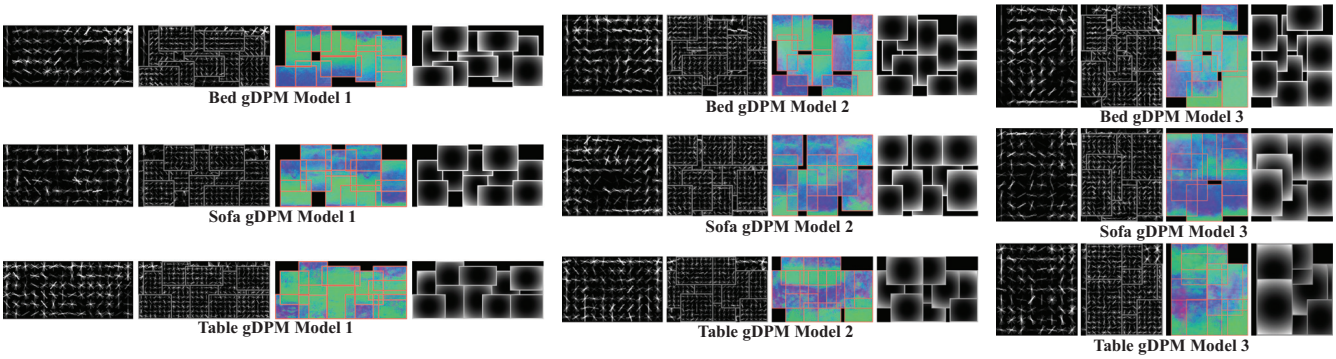



Figure 7. Learned gDPM models for classes bed, sofa and table. The first visualization in each template represents the learned appearance root filter, the second visualization contains learned part filters super-imposed on the root filter, the third visualization is the surface normal map corresponding to each part and the fourth visualization is of the learned deformation penalty.

the latent update step on positives uses $f_\beta$ from (5) to estimate the latent variables; then we apply SGD to solve for $\beta$ by using standard $f_\beta$ (4) and hard-negative mining. At test time, we only use the standard scoring function (2) (which is also equivalent to setting $\lambda = 0$ in (5)).

## 5. Experiments

We now present experimental results to demonstrate the effectiveness of adding geometric representation and constraints to a deformable part-based model. We will show how adding 3D parts and geometric constraints not only help improve the performance of our object detector but also help us to develop 3D understanding of the object (in terms of surface normals). We perform our experimental evaluation on the NYU Depth v2 dataset [25]. We learn a gDPM model for five object categories: bed, chair, monitor+TV (M.+TV), sofa and table. We use 3 components for each object category and some of the learned models are shown in Figure 7. This dataset has 1,449 images; we use the train-test splits from [25] (795 training and 654 test images). We convert the object instance segmentation masks (provided by [25]) to bounding boxes for training and testing object detectors. For surface normal prediction for the object, we superimpose the surface normals corresponding to each part and take the pixel-wise median. We also use colorization from [25] to in-paint missing regions in the object for visualization.

**Qualitative:** Figure 8 shows the performance of gDPM detector on a few examples. Our gDPM model not only localizes the object better but is also able to predict the surface normals for the detected objects. For example, in the first row, gDPM not only predicts the flat sittable surface of the couch but it also predicts the vertical backrest and the horizontal surface on the top of it. Similarly, in the second row, our approach is able to predict the horizontal surface of the small table. Figure 9 shows one of the false positives of our approach. In this case, a chair is predicted as a sofa by gDPM but notice the predicted surface normals by gDPM. Even in the case of wrong category prediction, gDPM does a reasonable job on the task of predicting surface normals including the horizontal support surface of the chair.

Table 1. AP performance on the task of object detection.

| | Bed | Chair | M.+TV | Sofa | Table |
|---|---|---|---|---|---|
| DPM (No Parts) | 20.94 | 10.69 | 6.38 | 5.51 | 2.73 |
| DPM | 22.39 | **14.44** | 8.10 | 7.16 | 3.53 |
| DPM (Our Parts, No Latent) | 26.59 | 5.71 | 2.35 | 6.82 | 3.41 |
| DPM (Our Parts) | 29.15 | 11.43 | 4.17 | 8.30 | 1.76 |
| gDPM | **33.39** | 13.72 | **9.28** | **11.04** | **4.05** |

**Quantitative:** We now evaluate gDPM quantitatively on the task of 2D object detection. As a baseline, we compare our approach against the standard DPM model with and without parts. We also evaluate the performance of DPM by treating our initial part hypothesis as strong supervision (ground truth parts) and not doing any latent updates. Finally, we also evaluate the performance of our parts with the standard latent updates which do not consider the geometric constraint based on depth data. Table 1 shows the average precision (AP). Our approach improves over the standard DPM by approximately 3.2% mean AP over 5 categories; and for categories like bed and sofa, the improvement is as much as 11% and 4% respectively. We also evaluate our surface normal prediction accuracy in a small quantitative experiment. Against Geometric Context [19], our surface normal prediction is 2° better, in terms of median per-pixel error.

## 6. Conclusions

We proposed a novel part-based representation, geometry-driven deformable part-based model (gDPM), where the parts are defined based on their 3D properties. gDPM effectively leverages depth data to combine the power of DPMs with the richness of geometric representation. We demonstrate how depth data can be used to define parts and provide weak supervision during the latent update steps. This leads to a better model in terms of detection performance. But more importantly, a joint geometric and

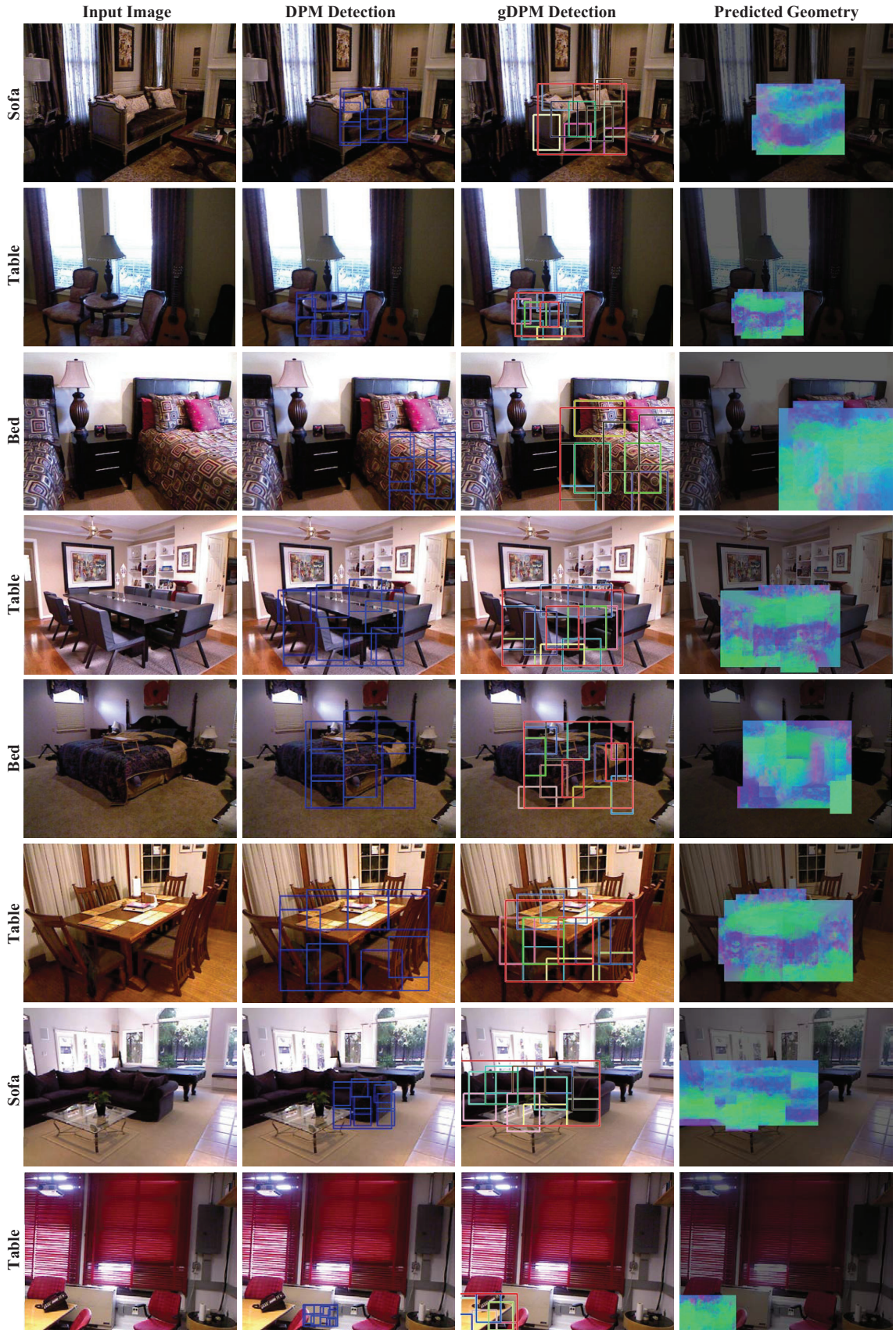
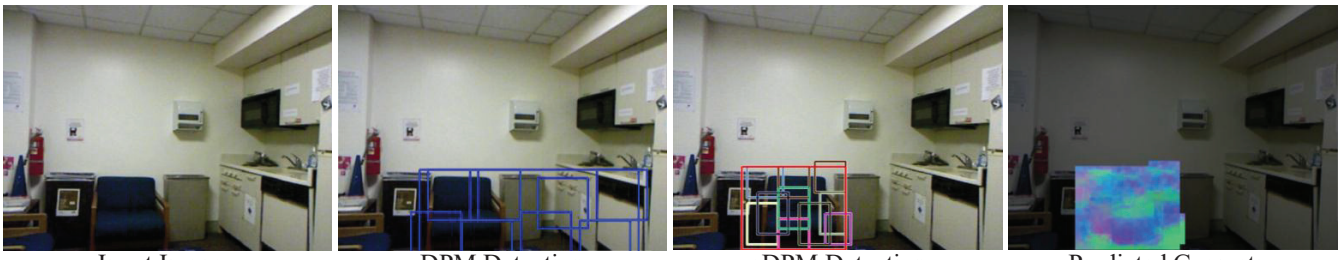



Figure 8. Qualitative Results: Our gDPM not only localizes the object but also predicts the surface normals of the objects.



Input Image DPM Detection gDPM Detection Predicted Geometry

Figure 9. False Positives: Our sofa detector detecting chair. Notice that the geometry still looks plausible.

appearance based representation allows us to jointly tackle the grand challenge of object detection and understanding 3D objects from 2D images.

**Acknowledgments:** This work was supported by NSF IIS-1320083 and ONR-MURI N000141010934. The authors would like to thank David Fouhey and Varun Ramakrishna for many helpful discussions.

## References

[1] H. Azizpour and I. Laptev. Object detection using strongly-supervised deformable part models. In *ECCV*, 2012. 1, 2, 5

[2] L. Bo, K. Lai, X. Ren, and D. Fox. Object recognition with hierarchical kernel descriptors. In *CVPR*, 2011. 2

[3] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3D human pose annotations. In *ICCV*, 2009. 2

[4] S. Branson, S. Belongie, and P. Perona. Strong supervision from weak annotation: Interactive training of deformable part models. In *ICCV*, 2011. 2

[5] R. Brooks. Symbolic reasoning among 3D models and 2D images. *Artificial Intelligence*, 1981. 2

[6] R. Brooks, R. Creiner, and T. Binford. The acronym model-based vision system. *IJCAI*, 1978. 2

[7] X. Chen, A. Shrivastava, and A. Gupta. NEIL: Extracting visual knowledge from web data. In *ICCV*, 2013. 2

[8] H. Chiu, L. Kaelbling, and T. Lozano-Perez. Virtual training for multi-view object class recognition. In *CVPR*, 2007. 2

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 2

[10] S. Divvala, A. Efros, and M. Hebert. How important are 'deformable parts' in the deformable parts model? In *ECCV Parts and Attributes Workshop*, 2012. 1, 2

[11] I. Endres, V. Srikumar, M.-W. Chang, and D. Hoiem. Learning shared body-plans. In *CVPR*, 2012. 2, 5

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, 2005. 1

[13] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. In *TPAMI*, 2010. 1, 2, 4, 5

[14] S. Fidler, S. Dickinson, and R. Urtasun. 3D object detection and viewpoint estimation with a deformable 3D cuboid model. In *NIPS*, 2012. 2

[15] D. Fouhey, A. Gupta, and M. Hebert. Data-driven 3D primitives for single image understanding. In *ICCV*, 2013. 2

[16] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *ICCV*, 2011. 2

[17] C. Gu and X. Ren. Discriminative mixture-of-templates for view-point classification. In *ECCV*, 2010. 2

[18] V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *ECCV*, 2010. 2

[19] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 2007. 6

[20] K. Lai, L. Bo, X. Ren, and D. Fox. A scalable tree-based approach for joint object and pose recognition. In *AAAI*, 2011. 2

[21] K. Lai, L. Bo, X. Ren, and D. Fox. Sparse distance learning for object recognition combining RGB and depth information. In *ICRA*, 2011. 2

[22] J. Lim, R. Salakhutdinov, and A. Torralba. Transfer learning by borrowing examples for multiclass object detection. In *NIPS*, 2011. 2

[23] D. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 1987. 2

[24] D. Marr and H. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proc. Roy. Soc.*, 1978. 2

[25] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, 2012. 3, 6

[26] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3D geometry to deformable part models. In *CVPR*, 2012. 2

[27] S. Savarese and L. Fei-Fei. 3D generic object categorization, localization and pose estimation. In *ICCV*, 2007. 2

[28] S. Singh, A. Gupta, and A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012. 2

[29] M. Stark, M. Goesele, and B. Schiele. Back to the future: Learning shape models from 3D CAD data. In *BMVC*, 2010. 2

[30] M. Sun, H. Su, S. Savarese, and L. Fei-Fei. A multi-view probabilistic model for 3D object classes. In *CVPR*, 2009. 2

[31] Y. Xiang and S. Savarese. Estimating the aspect layout of object categories. In *CVPR*, 2012. 2

[32] Y. Yang and D. Ramanan. Articulated pose estimation with exible mixtures-of-parts. In *CVPR*, 2011. 2, 5

[33] X. Zhu, C. Vondrick, D. Ramanan, and C. Fowlkes. Do we need more training data or better models for object detection? In *BMVC*, 2012. 1