Data-Driven 3D Primitives for Single Image Understanding

David F. Fouhey, Abhinav Gupta, Martial Hebert
The Robotics Institute, Carnegie Mellon University
{dfouhey,abhinavg,hebert}@cs.cmu.edu

Abstract

What primitives should we use to infer the rich 3D world behind an image? We argue that these primitives should be both visually discriminative and geometrically informative and we present a technique for discovering such primitives. We demonstrate the utility of our primitives by using them to infer 3D surface normals given a single image. Our technique substantially outperforms the state-of-the-art and shows improved cross-dataset performance.

1. Introduction

How do you infer the 3D properties of the world from a 2D image? This question has intrigued researchers in psychology and computer vision for decades. Over the years, researchers have proposed many theories to explain how the brain can recover rich information about the 3D world from a single 2D projection. While there is agreement on many of the cues and constraints involved (e.g., texture gradient and planarity), recovering the 3D structure of the world from a single image is still an enormously difficult and unsolved problem.

At the heart of the 3D inference problem is the question: What are the right primitives for inferring the 3D world from a 2D image? It is not clear what kind of 3D primitives can be directly detected in images and be used for subsequent 3D reasoning. There is a rich literature proposing a myriad of 3D primitives ranging from edges and surfaces to volumetric primitives such as generalized cylinders, geons and cuboids. While these 3D primitives make sense intuitively, they are often hard to detect because they are not discriminative in appearance. On the other hand, primitives based on appearance might be easy to detect but can be geometrically uninformative.

In this paper, we propose data-driven geometric primitives which are **visually-discriminative**, or easily recognized in a scene, and **geometrically-informative**, or conveying information about the 3D world when recognized. Our primitives can correspond to geometric surfaces, corners of cuboids, intersection of planes, object parts or even whole objects. What defines them is their **discriminative** and **informative** properties. We formulate an objective
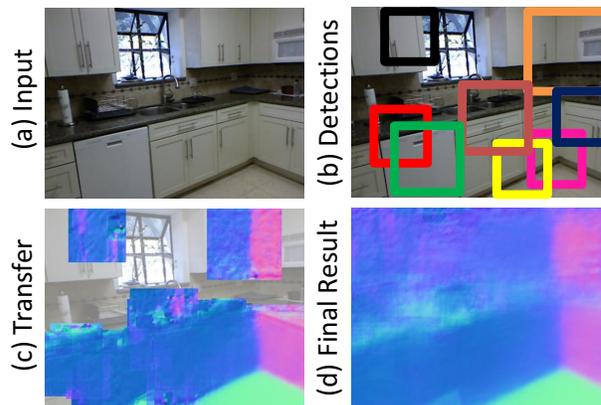

Figure 1. We propose an approach to discover discriminative and geometrically informative primitives. These can be recognized with high precision in RGB images (b) and convey the underlying 3D geometry (c). We can use sparse detections of these primitives to find dense surface normals via simple label transfer (d).

function which encodes these two criteria and learn these 3D primitives from indoor RGBD data (see Fig. 2 for some examples of discovered primitives). We then demonstrate that our primitives can be recognized with high precision in RGB images (Fig. 1(b)) and convey a great deal of information about the underlying 3D world (Fig. 1(c)). We use these primitives to densely recover the surface normals of a scene from a single image via simple transfer (Fig. 1(d)). Our 3D primitives significantly outperform the state-of-the-art as well as a number of other credible baselines. We also demonstrate that our primitives generalize well by showing improved cross-dataset performance.

1.1. Historical Background

The problem of inferring the 3D layout of a scene from a single image is a long-studied problem in computer vision. Early work focused on geometric primitives, for instance [6, 15, 30] which used 3D contours as primitives. Contours were first detected and the 3D world was inferred via a consistent line-labeling over the contours. While successful on line-drawings, these approaches failed to work on natural images: 3D contour detection is unquestionably difficult and remains, decades later, an active area of research. Other research focused on volumetric 3D primitives such

Figure 2. Example primitives discovered by our approach. (Left: canonical normals and detectors; right: patch instances). Our discovery method finds a wide range of primitives: **Row 1:** objects and parts, **Rows 2-4:** 3 and more plane primitives, **Row 5:** 1 plane, **Row 6-7:** 2 planes. Our primitives contain visual synonyms, (e.g., row 6), in which different appearance corresponds to the same underlying geometry. More examples can be found in the supplemental material. **Legend for normals: blue: X; green: Y; red: Z.**

as generalized cylinders [3] and geons [2]. However, although these primitives produced impressive demos such as ACRONYM [5], they failed to generalize well and the field moved towards appearance-based approaches (e.g., [4, 10]).

Recently, there has been a renewed push toward more geometric approaches where the appearance of primitives is learned using large amounts of labeled [14] or depth data [23]. The most commonly used primitives include oriented 3D surfaces [14, 19, 23, 31] represented as segments in the image, or volumetric primitives such as blocks [11] and cuboids [18, 33]. However, since these primitives are not discriminative, a global consistency must be enforced, e.g., by a learned model such as in [23], a hierarchical segmentation [14], physical and volumetric relationships [18], recognizing primitives as parts of semantic objects [31], or assuming a Manhattan world and low-parameter room layout model [13, 35, 26]. Non-parametric approaches provide an alternative approach to incorporating global constraints, and instead use patch-to-patch [12], scene [17], or 2D-3D [22] matching to obtain nearest neighbors followed by label transfer. While all of these constraint-based approaches have improved 3D scene understanding, accurate detection of primitives still remains a major challenge.

On the other hand, there have been recent advances in the field of detection enabled by discriminative appearance-based approaches. Specifically, instead of using manually defined and semantically meaningful primitives or parts, these approaches discover primitives in labeled [4, 9], weakly-labeled [8] or unlabeled data [29]. While these primitives have high detection accuracy, they might not have consistent underlying geometry. Building upon these advances, our work discovers primitives that are both **discriminative** and **informative**. We use depth data from Kinect to provide the supervisory signal for automatically discovering these primitives. Note that depth is not used as a source of features (e.g., as in [21]), but instead as a form of training-time-only supervision (e.g., as in [27]).

2. Overview

Our goal is to discover a vocabulary of 3D primitives that are visually discriminative and geometrically informative; in other words, primitives need to be easily recognized in unseen images and convey information about 3D proper-

ties of the scene when recognized. Our primitive representation incorporates three aspects: (a) Appearance: a well-calibrated discriminative linear *detector* over HOG [7] (denoted $\mathbf{w}$), which can be used to find the primitive in new images; (b) Geometry: a *canonical form* (denoted $\mathbf{N}$), which represents the underlying geometry in terms of surface normals, akin to a cluster prototype; (c) Instances: particular examples of the primitive, which are regions of training scenes in which the primitive appears. Note that this is an over-complete representation: e.g., instances can be used to obtain a detector and vice-versa.

We build our primitives using an iterative procedure detailed in Section 3. After using an initialization that ensures both visual discriminativity and geometric informativity, we optimize the objective function by alternating between finding instances, learning the detector, and computing the canonical form. Once the primitives have been discovered, we use them to interpret new images and demonstrate that our detectors can trade off between sparsity and accuracy of predictions. Finally, in Section 4 we show that our sparse detections can be used to predict dense surface normals from a single image using a simple transfer approach: we align copies of the training set with the test image with the correspondence between primitive detections and primitive instances, and estimate the test image's surface normals as a weighted sum of the training images.

3. Discovering 3D Primitives

Given a set of training images and their corresponding surface normals, our goal is to discover geometric primitives that are both discriminative and informative. The challenge is that the space of geometric primitives is enormous, and we must sift through all the data to find geometrically-consistent and visually discriminative concepts. Similar to object discovery approaches, we pose it as a clustering problem: given millions of image patches, we group them so each cluster is discriminative (we can learn an accurate detector) and geometrically consistent (all patches in the cluster have consistent surface normals).

Mathematically, we formulate the problem as follows. As input, we have a collection of image patches in the training dataset, $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$. Each patch has a geometric component $\mathbf{x}_i^G$ (a 2D array of surface normals scaled to a canonical scale) and appearance representation $\mathbf{x}_i^A$ (HOG). Our goal is to cluster the data and learn geometric primitives. Each primitive is represented as $\langle \mathbf{w}, \mathbf{N}, \mathbf{y} \rangle$ where $\mathbf{w}$ is the weight vector of a linear support vector machine (SVM) learned in appearance space, $\mathbf{N}$ represents the underlying geometry of the primitive and $\mathbf{y} \in \{0, 1\}^m$ is an instance indicator vector with $y_i = 1$ for instances of the primitive and zero otherwise. Ideally, we would like to minimize the following objective function:

$$\min_{\mathbf{y}, \mathbf{w}, \mathbf{N}} R(\mathbf{w}) + \sum_{i=1}^{m} c_1 y_i \Delta(\mathbf{N}, \mathbf{x}_i^G) + c_2 L(\mathbf{w}, \mathbf{x}_i^A, y_i), \quad (1)$$

where R is a regularizer on the classifier, each c_i trades off between terms, Δ is a distance measure over 3D geometry, and L is a loss function. To avoid trivial solutions, we also constrain the membership so each cluster has at least s members.

The first term regularizes the primitive's detector, the second enforces consistent geometry across instances so that the primitive is geometrically informative, and the third ensures that the clusters' instances can be distinguished from other patches (that is, the loss function over the correct classification of training patches). In our case, we use an SVM-based detector and represent geometry with surface normals. Therefore, we set $R(\mathbf{w})$ to $||\mathbf{w}||_2^2$, Δ to the mean per-pixel cosine distance between patches' surface normals, and L to hinge-loss on each $\mathbf{x}_i^A$ with respect to $\mathbf{w}$ and $\mathbf{y}$. Note that there will be many local minima, with each minimum $\mathbf{w}, \mathbf{y}, \mathbf{N}$ corresponding to a single primitive: as shown in Fig. 2, many 3D primitives are discriminative and informative. We obtain a collection of primitives by finding the many minima, which we do via multiple initialization.

Exactly minimizing this objective function is an extremely difficult problem, but its optimization is a chicken-and-egg problem: if we knew a good set of geometrically-consistent and visually discriminative patches, we could train a detector to find them, and if we had a detector for a geometrically consistent visual concept, we could find instances. We propose an iterative solution for the optimization. Given $\mathbf{y}$, we can compute $\mathbf{N}$ and learn $\mathbf{w}$. Once we have $\mathbf{w}$, we can refine our membership $\mathbf{y}$ and repeat.

3.1. Iterative Optimization

Our approach alternates between optimizing membership $\mathbf{y}$ and detector weights $\mathbf{w}$ while ensuring both visual discriminativity and geometric informativity. Given an initialization in terms of membership $\mathbf{y}$ (Section 3.2), we train a detector $\mathbf{w}$ to separate the elements of the cluster from geometrically dissimilar patches from negative examples $\mathcal{V}$ (found via the canonical form $\mathbf{N}$). In this work, we also supplement our negative data with 6000 randomly chosen outdoor images $\mathcal{W}$. Once a detector has been trained, we scan our training set $\mathcal{I}$ for the top most-similar patches to $\mathbf{w}$, and use these patches to update membership $\mathbf{y}$.

From y to w: given the primitive instances, we want to train a detector that can distinguish the primitive instances from the rest of the world. To help find dissimilar patches, we first compute the canonical form $\mathbf{N}$ of the current instances as the per-pixel average surface normal over the instances (i.e., after aligning and rescaling patches). We then train a linear SVM $\mathbf{w}$ to separate the positive instances from geometrically dissimilar patches in the negative set $\mathcal{V}$ and all patches in $\mathcal{W}$. We define geometrically dissimilar patches as patches with mean per-pixel cosine distance from $\mathbf{N}$ greater than $70°$. We use a high-threshold to include definite mistakes only and promote generalization.

From w to y: Given $\mathbf{w}$ is found, we pick $\mathbf{y}$ by selecting a

| (a) Input | (b) Sparse Detections | (c) Primitive-aligned Label Transfer | (d) Final Result |
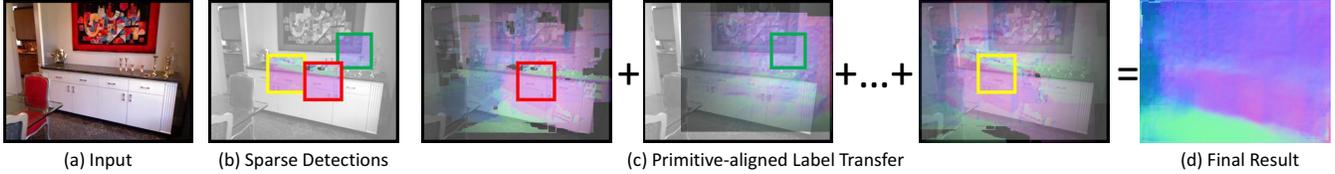
Figure 3. An illustration of the our simple inference approach. We align the training images via detections and predict the test image as a weighted linear sum of the training images.

geometrically consistent set among the top detections of $\mathbf{w}$ in $\mathcal{I}$; in our experiments, we use the s-member subset that approximately minimizes the intra-set cosine distance.

If done directly, this sort of iterative technique (akin to discriminative clustering [34]) has been demonstrated to overfit and produce sub-optimal results. We therefore adopt the cross-validation technique of [8, 29] in which the datasets are divided into partitions. We use two partitions; we initialize identities $\mathbf{y}$ and train the detector $\mathbf{w}$ on partition 1; then we update the identities $\mathbf{y}$ and train $\mathbf{w}$ on partition 2; following this, we return to partition 1, etc.

3.2. Implementation Details

Initialization: We initialize our algorithm with a greedy approach to find independently visually and geometrically compact groups in randomly sampled patches. First, we sample random square patches throughout the training set $\mathcal{I}$ at multiple scales. For every patch, we find $s-1$ nearest neighbors in appearance and geometry space by intersecting the nearest neighbor lists (computed with Euclidean distance on HOG and mean per-pixel cosine distance respectively). We group the query patch with its neighbors to initialize the primitive instances. For a training set of 800 images, we produce $3,000$ primitive candidates.

Calibrating the Detectors: Our discovery procedure will produce a collection of geometric primitives with detectors trained in isolation. We therefore calibrate our detectors on held-out data. To produce calibrated scores, we use the Extreme-Value-Theory-based calibration system of Scheirer et al. [25], which requires only negative data. In our case, this is ideal, since there are no ground-truth positive/negative annotations as in object detection, and we only set a threshold for what counts as an inaccurate detection (in our experiments, for calibration this is $30°$). This calibration step also suppresses any ineffective primitives.

Parameters: We represent the visual data of the patch $\mathbf{x}_i^A$ with HOG features [7], calculated at a canonical size (8×8 cell with a stride of 8 pixels per cell) and the geometric representation $\mathbf{x}_i^G$ as the surface normal patch scaled to a canonical size (10×10). For our detectors $\mathbf{w}$, we use linear SVMs trained with $C = 0.1$ fixed throughout.

4. Interpretation via Data-driven Primitives

Our discovery algorithm extracts geometrically consistent and visually discriminative primitives from RGBD data. These primitives can then be detected with high precision and accuracy in new RGB images to develop a 3D in-

terpretation of the image. However, not all surface normals are equally easy to infer. Consider the image shown in Fig. 3(a). While the primitives are detected in discriminative regions such as the cupboard and painting, other regions such as a patch on a textureless wall are hopelessly difficult to classify in isolation. While a sparse interpretation from discriminative primitives (Fig. 3(a),(b)) might be sufficient for many vision applications, others might require a more dense interpretation. A dense interpretation would require propagating information from the confident regions to the uncertain regions and a variety of methods have been proposed to do this (e.g., [13, 14]) To demonstrate the effectiveness of our primitives, we propose a simple label-transfer method to propagate labels, which we show outperforms the state of the art in Section 5.

Technical Approach: Given a test image, we run the bank of the 3D primitives' learned detectors $\mathbf{w}$. This yields a collection of T detections ($\mathbf{d}_1 \ldots \mathbf{d}_T$). We warp the surface normals so the primitive detections and s training instances per detection are aligned, producing a collection of sT aligned surface normal images ($\mathbf{M}_{1,1} \ldots \mathbf{M}_{1,T} \ldots, \mathbf{M}_{s,T}$). We infer the pixels of a test image as a linear combination of the surface normals of these aligned training images with weights determined by detections:

$$\hat{M}(\mathbf{p}) = \frac{1}{Z} \sum_{i,j=1,1}^{s,T} \text{score}(\mathbf{d}) \exp(-||\mathbf{p} - \mathbf{d}_j||^2 / \sigma_{\text{spatial}}^2) \mathbf{M}_{i,j}(\mathbf{p}),$$

where Z is a normalization term. The first term gives high weight to confident detections and to detections that fire consistently at the same absolute location in training and test image (e.g., floor primitives should not be at the top of an image). The second term is the spatial term and gives high weight for transfer to pixels near the detection and the weight decreases as a function of the distance from the location of the primitive detection. Each direction is processed separately and the resulting vector renormalized, corresponding to a maximum-likelihood fit of a von Mises-Fisher distribution. This procedure is illustrated in Figs. 3,4.

5. Experimental Evaluation

Dataset: We evaluate our approach on the NYU Depth v2 [28] dataset. This dataset contains $1,449$ registered RGB and depth images from a wide variety of real-world and cluttered scenes. We ignore values for which we cannot obtain an accurate estimate of the surface normals due to missing depth data. We compute the "ground truth" surface normals with respect to the camera axes from the depth data
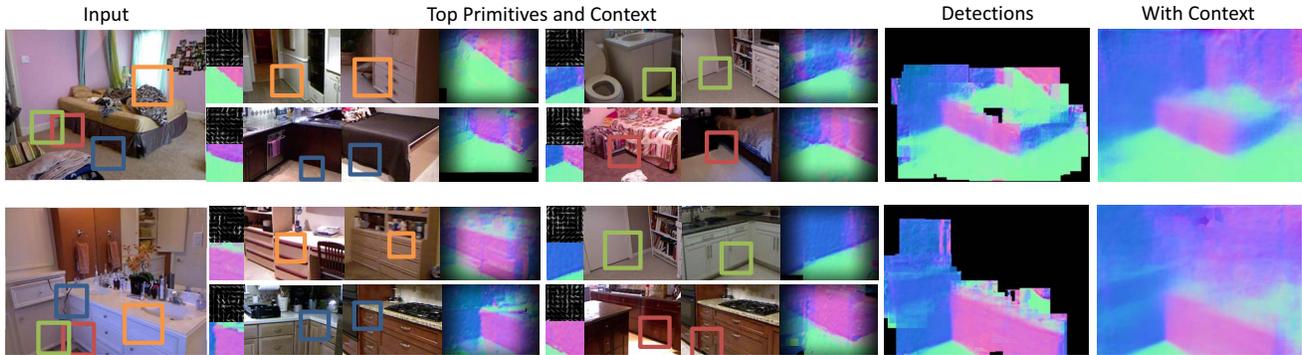
Figure 4. Qualitative Results: The above figure shows two examples of how our primitives help us in accurate but sparse 3D interpretation of images. This sparse understanding can then be used to produce an accurate dense 3D interpretation even using a simple transfer approach.

using a least-squares fit followed by bilateral smoothing to reduce noise. For all our experiments, we use four-fold cross validation while ensuring no room appears in both the train and test parts of a split. Fig. 2 shows some examples of the top primitives of one fold.

Baselines: We qualitatively and quantitatively compare against state-of-the-art methods for depth and surface normal prediction. Specifically, we compare against eight baselines in sparse and dense prediction. The first five are the state-of-the-art; the sixth tests the contribution of geometric supervision; the last two test against direct regression of surface normals.

(1) Lee et al. [19]: The plane-sweeping orientation map finds sparse vanishing-point aligned planar surfaces from lines. For dense predictions, we use NN-based filling.

(2) Hoiem et al. [14]: The geometric context approach predicts quantized surface normals in five directions using multiple-segmentation based classifiers. We retrain the classifiers using the NYU data.

(3) Hedau et al. [13]: This baseline builds on geometric context classifiers (including one for clutter), which we retrain on NYU and uses structured prediction to predict a vanishing-point-aligned room cuboid.

(4) Karsch et al. [17]: One can also produce surface normals by predicting depth and computing normals on the results; we do this with the depth prediction method of Karsch et al. [17], which out-performs other methods for depth prediction by a considerable margin in all evaluation criteria.

(5) Saxena et al. [23]: We also compare with surface normals computed from depth predicted by Make 3D using the pre-trained model.

(6) Singh et al. [29]: We compare against this appearance based primitive discovery approach. We replace our geometric primitives with mid-level patches discovered by [29], and use the same inference pipeline.

(7) RF + SIFT: We train a random forest (RF) regressor to predict surface normals using a histogram of dense-SIFT [20] features (codebook size 1K) over SLIC [1] superpixels ($S = 20$, $M = 100$) as well as location features.

(8) SVR + SIFT: We also train a ϵ-Support Vector Regres-

sor (SVR) using a Hellinger kernel to predict surface normal orientations using the same input features as above.

Evaluation Criteria: Characterizing surface normal predictor performance is difficult because different metrics encode different objectives, not all of which are desirable. For instance, the root-mean-squared error (RMSE) so severely penalizes large errors that in practice it characterizes the tail of the error distribution. This was noted in the stereo benchmark [24], which favors alternate metrics that count pixels as correct or not according to a threshold. Therefore, in addition to reporting the mean, median, and RMSE on a per-pixel-basis, we report three pixel-accuracy metrics, or percent-good-pixels (i.e., the fraction of pixels with cosine distance to ground-truth less than t) with $t = 11.25°, 22.5°, 30°$. To characterize the noise in our data, we annotated a randomly chosen subset of planar surfaces (ideally with the same surface normal) in 100 images and evaluated the angular error between pairs of pixels; the median error was $5.2°$.

Evaluating 3D Primitives: Fig. 4 shows qualitative examples of the top few primitive detections in two images. The detections are accurate and convey 3D information about the scene despite their sparsity. Fig. 5 shows additional sparse results compared to [19].

We also quantitatively evaluate our primitives by producing a precision-vs-coverage curve that trades off between precision (the fraction of pixels correctly predicted) and coverage (the fraction of pixels predicted). Fig. 7 shows the precision-coverage curve using a threshold of $22.5°$ to determine a predicted pixel's correctness. We compare with Geometric Context [14] (sweeping over classifier confidence), and the appearance-only primitives of Singh et al. [29]. Additional precision-coverage curves appear in the supplementary materials. Finally, we report results using only a single round of the iterative procedure to test whether the primitives improve with iterations. Our approach works considerably better than all baselines and the initialization at every coverage level. The gains are especially strong in the low-recall, high-precision regime. The appearance only baseline [29] does not produce good results since its vo-

| Input	Ground-Truth	Lee et al.	3D Prim. (Sparse)	Karsch et al.	RF+SIFT	3D Prim. (Dense)
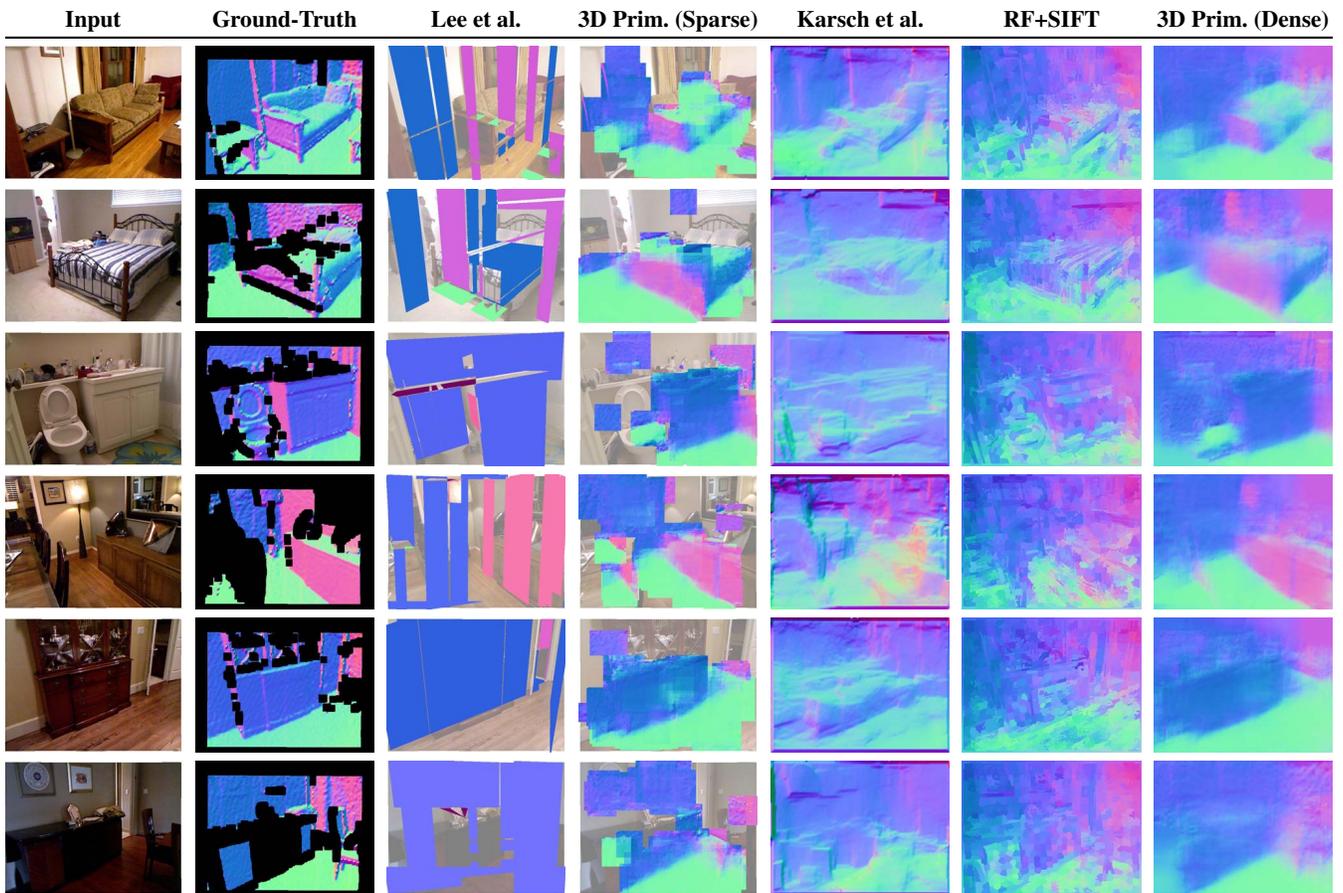
Figure 5. Qualitative results of our technique in comparison to some of the baselines.
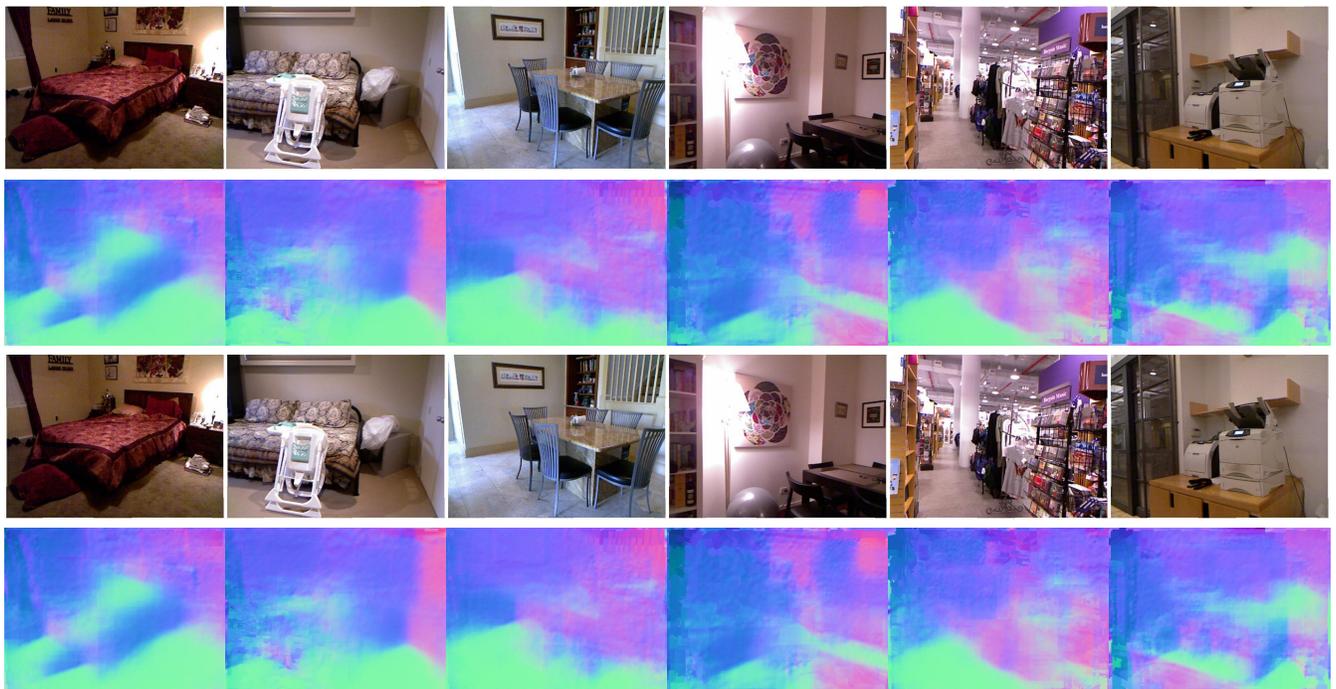
Figure 6. Example normal maps inferred via our data-driven 3D primitives. **Top row:** selected results; **bottom row:** results automatically regularly sampled showing the (left-to-right/good-to-bad) performance range of the proposed method from 1/7th to 6/7th percentile. Additional results appear in the supplementary material.
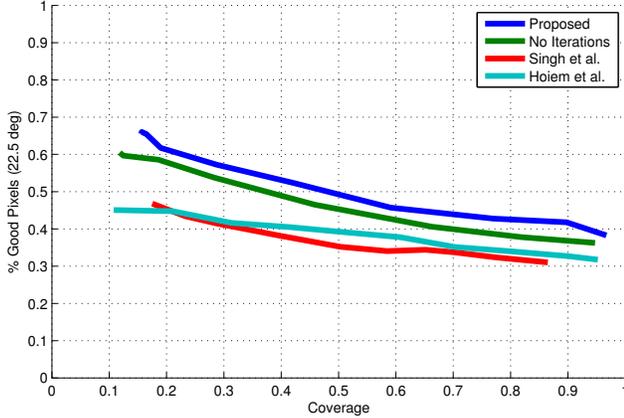
Figure 7. Fraction of pixels with error below $22.5°$ vs. coverage for the proposed approach and sparse baselines.

cabulary does not contain crucial 3D primitives which are difficult to cluster using appearance alone (e.g., corners of rooms in Fig. 2, row 7, col. 1). Note that our primitive method does not reach 100% coverage in Fig. 7. The remaining unpredicted pixels correspond to textureless surfaces and cannot be predicted accurately from local evidence, thus requiring our context-transfer technique.

As seen in Table 2, the most confident primitives perform much better, showing that our technique can identify which predictions are accurate; in addition to enabling high performance for applications using sparse normals, this confidence is a crucial cue for any subsequent reasoning.

Evaluating Dense Prediction: Fig. 4 shows how a few detections can be used to align training normals and test images. The primitives create an accurate dense interpretation from sparse detections. We qualitatively compare the results of our technique with several baselines in Fig. 5 and show the entire spectrum of our results in Fig. 6. Our results accurately convey the 3D structure of scenes, including fine-grained objects such as filing cabinets even when these details deviate from a typical scene (Fig. 6, upper-left scene). Additionally, although we do not use a segmentation, our results accurately capture inter-object boundaries.

Quantitatively, we compare our approach with all the baselines in Table 1. Predictions are qualitatively and quantitatively different if one assumes there are three orthogonal surface normal directions (the Manhattan-world assumption). We therefore evaluate approaches making this assumption separately. We add this assumption to our approach by adjusting each predicted pixel to the nearest vanishing point calculated by [13]. This results in gains in some metrics and losses in others by making most predictions almost perfect or completely wrong; for error thresholds above $t = 40°$, the assumption degrades results. Our approach outperforms all methods by a substantial margin in all metrics. This is important since each metric captures a different aspect of performance and no one metric is sufficient: Hedau et al.'s method, for instance, does well on me-

Table 1. Results on the NYU Depth v2 dataset. Our technique outperforms the baselines by a substantial margin in all metrics.

	Summary Stats. (°) (Lower Better)			% Good Pixels (Higher Better)		
	Mean	Median	RMSE	$11.25°$	$22.5°$	$30°$
With Manhattan World Constraints						
Lee et al.	44.9	34.6	54.8	24.8	40.5	46.7
Hedau et al.	41.2	25.5	55.1	33.2	47.7	53.0
3D Primitives	**33.5**	**18.0**	**46.6**	**37.4**	**55.0**	**61.2**
Without Manhattan World Constraints						
Karsch et al.	40.8	37.8	46.9	7.9	25.8	38.2
Hoiem et al.	41.2	34.8	49.3	9.0	31.7	43.9
Singh et al.	35.0	32.4	40.6	11.2	32.1	45.8
Saxena et al.	47.1	42.3	56.3	11.2	28.0	37.4
RF + SIFT	36.0	33.4	41.7	11.4	31.1	44.2
SVR + SIFT	36.5	33.5	42.4	10.7	30.8	44.1
3D Primitives	**33.0**	**28.3**	**40.0**	**18.8**	**40.7**	**52.4**

Table 2. Evaluation of the approach at several coverage levels.

	Mean	Median	RMSE	$11.25°$	$22.5°$	$30°$
25% coverage	27.2	18.4	36.8	33.0	57.3	67.7
50% coverage	31.9	23.5	41.4	26.9	48.5	58.4
75% coverage	33.7	27.8	42.1	21.8	42.0	53.0
Full Coverage	33.0	28.3	40.0	18.8	40.7	52.4

dian error but produces many wildly inaccurate results and thus does poorly in mean error. Analysis of the significance of results may be found in the supplementary material.

Cross-Dataset Prediction: We also want to demonstrate that our primitives generalize well and do not overfit to the NYU data. Therefore, using identical parameters, we use models learned on one split of the NYU dataset to predict dense surface normals on the Berkeley 3D Object Dataset (B3DO) [16] and the subset of SUNS dataset [32] used in [22]. Fig. 8 shows some qualitative results from these datasets. Some scenes are atypical of layout prediction datasets (e.g., close-up views) but our method still produces accurate results. We also quantitatively characterize the generalization performance on the B3DO dataset in Table 3, since it has depth data available. Our approach again outperforms the baselines in all metrics.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, F. P., and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 34(11):2274–2281, 2012. 5

[2] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94:115–147, 1987. 2
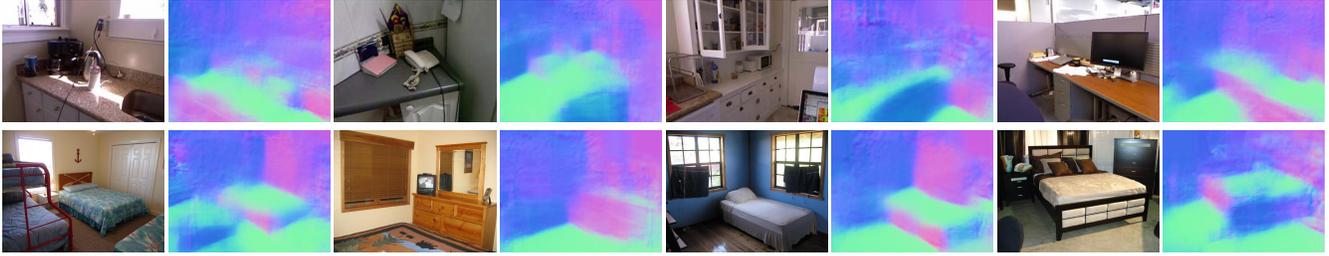
Figure 8. Cross-dataset results on B3DO (top), SUNS (bottom). More results appear in the supplementary material.

Table 3. Cross-dataset results. We train the method on one fold of the NYU dataset [28] and test it on the B3DO dataset [16].

	Mean	Median	RMSE	11.25°	22.5°	30°
With Manhattan World Constraints						
Lee et al.	41.9	28.4	56.6	32.7	45.7	50.8
Hedau et al.	43.5	30.0	58.1	32.8	45.0	50.0
3D Primitives	**38.0**	**24.5**	**51.2**	**33.6**	**48.5**	**54.5**
Without Manhattan World Constraints						
Hoiem et al.	42.1	37.4	49.7	8.2	25.5	38.1
Singh et al.	36.7	34.2	42.3	9.9	29.4	42.9
Saxena et al.	45.6	41.2	53.5	8.4	25.5	36.6
RF + SIFT	36.8	34.3	42.6	10.2	29.5	42.8
SVR + SIFT	37.0	34.0	42.6	9.5	29.1	42.9
3D Primitives	**34.7**	**30.7**	**41.1**	**14.3**	**35.9**	**49.0**

[3] T. Binford. Visual perception by computer. In *IEEE Conference on Systems and Controls*, 1971. 2

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

[5] R. Brooks, R. Creiner, and T. Binford. The acronym model-based vision system. In *IJCAI*, 1979. 2

[6] M. Clowes. On seeing things. *Artificial Intelligence*, 2:79–116, 1971. 1

[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 3, 4

[8] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros. What makes Paris look like Paris? *ACM Transactions on Graphics (SIGGRAPH)*, 31(4), 2012. 2, 4

[9] B. Epshtein and S. Ullman. Semantic hierarchies for recognizing objects and parts. In *CVPR*, 2007. 2

[10] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 32(9), 2010. 2

[11] A. Gupta, A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*, 2010. 2

[12] T. Hassner and R. Basri. Example based 3D reconstruction from single 2D images. In *CVPR Workshop: Beyond Patches*, 2006. 2

[13] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, 2009. 2, 4, 5, 7

[14] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. In *IJCV*, 2007. 2, 4, 5

[15] D. Huffman. Impossible objects as nonsense sentences. *Machine Intelligence*, 8:475–492, 1971. 1

[16] A. Janoch, S. Karayev, Y. Jia, J. Barron, M. Fritz, K. Saenko, and T. Darrell. A category-level 3-D object dataset: Putting the kinect to work. In *Workshop on Consumer Depth Cameras in Computer Vision (with ICCV)*, 2011. 7, 8

[17] K. Karsch, C. Liu, and S. B. Kang. Depth extraction from video using non-parametric sampling. In *ECCV*, 2012. 2, 5

[18] D. C. Lee, A. Gupta, M. Hebert, and T. Kanade. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *NIPS*, 2010. 2

[19] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, 2009. 2, 5

[20] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2):91–110, 2004. 5

[21] X. Ren, L. Bo, and D. Fox. RGB-(D) scene labeling: Features and algorithms. In *CVPR*, 2012. 2

[22] S. Satkin, J. Lin, and M. Hebert. Data-driven scene understanding from 3D models. In *BMVC*, 2012. 2, 7

[23] A. Saxena, M. Sun, and A. Y. Ng. Make3D: Learning 3D scene structure from a single still image. *TPAMI*, 2008. 2, 5

[24] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1):7–42, 2002. 5

[25] W. J. Scheirer, N. Kumar, P. N. Belhumeur, and T. E. Boult. Multi-attribute spaces: Calibration for attribute fusion and similarity search. In *CVPR*, 2012. 4

[26] A. G. Schwing and R. Urtasun. Efficient exact inference for 3D indoor scene understanding. In *ECCV*, 2012. 2

[27] A. Shrivastava and A. Gupta. Building parts-based object detectors via 3D geometry. In *ICCV*, 2013. 2

[28] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, 2012. 4, 8

[29] S. Singh, A. Gupta, and A. A. Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*, 2012. 2, 4, 5

[30] K. Sugihara. *Machine Interpretation of Line Drawings*. MIT Press, 1986. 1

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

[32] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 7

[33] J. Xiao, B. Russell, and A. Torralba. Localizing 3D cuboids in single-view images. In *NIPS*, 2012. 2

[34] J. Ye, Z. Zhao, and M. Wu. Discriminative k-means for clustering. In *NIPS*, 2007. 4

[35] S. X. Yu, H. Zhang, and J. Malik. Inferring spatial layout from a single image via depth-ordered grouping. In *Workshop on Perceptual Organization*, 2008. 2