DOI:10.1145/2436256.2436276

Discriminative Learning with Latent Variables for Cluttered Indoor Scene Understanding

By Huayan Wang, Stephen Gould, and Daphne Koller

Abstract

We address the problem of understanding an indoor scene from a single image in terms of recovering the room geometry (floor, ceiling, and walls) and furniture layout. A major challenge of this task arises from the fact that most indoor scenes are cluttered by furniture and decorations, whose appearances vary drastically across scenes, thus can hardly be modeled (or even hand-labeled) consistently. In this paper we tackle this problem by introducing latent variables to account for clutter, so that the observed image is jointly explained by the room and clutter layout. Model parameters are learned from a training set of images that are only labeled with the layout of the room geometry. Our approach enables taking into account and inferring indoor clutter *without* hand-labeling of the clutter in the training set, which is often inaccurate. Yet it outperforms the state-of-the-art method of Hedau et al.[7] that requires clutter labels. As a latent variable based method, our approach has an interesting feature that latent variables are used in direct correspondence with a concrete visual concept (clutter in the room) and thus interpretable.

1. INTRODUCTION

We address holistic understanding of indoor scenes from a single image. Our model takes an image of an indoor scene as input, and produces boundaries between the floor, the walls, and the ceiling (we call them the *box layout*), as well as segmentation of the clutter such as furniture and decorations (Figure 1). Learning the model parameters requires training images with hand-labeled box layout but not the clutter significantly reducing labeling effort.

Holistic scene understanding has attracted much attention in computer vision.[5, 9, 10, 11] One of the goals is to make use of *image context* to help improve traditional vision tasks, such as object detection. The image context can be represented by many aspects. For example, one could use a category label (e.g., street and kitchen), as it imposes a strong prior on the likely and unlikely objects to be detected (e.g., cars on the street). In this paper we focus on indoor scenes, for which we represent the image context by a box layout of the room and clutter. On one hand, such knowledge could be useful as a geometric constraint in a variety of traditional computer vision tasks such as object detection[14] and motion planning.[6] On the other hand it could also be used as input to infer more fine-grained contextual information such as the 3D layout.

Figure 1. Output of our method. First row: the inferred box layout illustrated by red lines representing face boundaries. Second row: the inferred clutter layout.

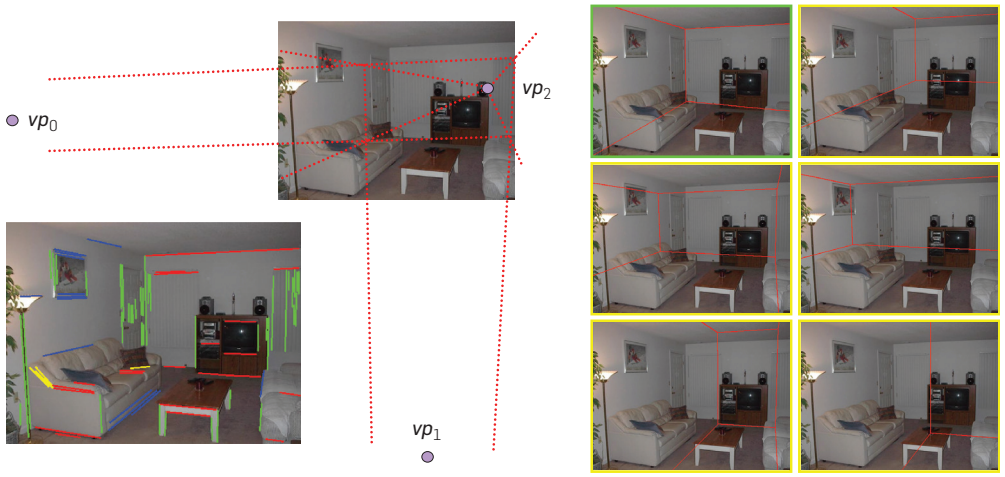
Given an image of an indoor scene, we perform two preprocessing steps. First, we detect the *vanishing points*, which play an essential role in characterizing the structure of an indoor scene. Due to the effect of perspective projection, a set of parallel lines in 3D are either parallel or intersect at a common point in the 2D image. The points where sets of 3D parallel lines meet in 2D are called vanishing points. Hedau et al.[7] observed that most indoor scenes are characterized by three vanishing points. Given these points, we can generate a family of box layouts. Specifically, as shown in Figure 2, we detect long lines in the image, and cluster them into three dominant groups corresponding to three vanishing points $\boldsymbol{vp}_0$, $\boldsymbol{vp}_1$, and $\boldsymbol{vp}_2$. Candidate box layouts can be generated by extending two rays from $\boldsymbol{vp}_0$, two rays from $\boldsymbol{vp}_1$, and connecting the four intersections with $\boldsymbol{vp}_2$. Infinitely many candidate box layouts can be generated in this manner, and the task becomes to identify the one that best fits the observed image.

Our method also infers clutter within the scene as shown in the second row of Figure 1. It is computationally expensive to perform reasoning on each pixel as to whether or not it belongs to clutter. We therefore perform an over-segmentation of the image using the mean-shift algorithm[2] and reason

The original version of this paper was published in *the 11th European Conference on Computer Vision, Part II, LNCS 6312 Proceedings*, Springer.

* This work was done when S. Gould was a Ph.D. candidate at Stanford University.

Figure 2. Lower-left: Three groups of lines (shown in R, G, B) corresponding to the three vanishing points. There are also "outlier" lines (shown in yellow). Upper-left: A candidate box layout is generated. Right: Different candidate box layouts (in yellow frames) are generated in the same way, and the hand-labeled true box layout (in green frame).



about the so-called superpixels, that is, we reason over small contiguous segments that have been predetermined according to their spatial and appearance properties. We typically have less than a hundred segments (superpixels) for each image. In reasoning about the clutter we treat each segment as a single entity.

Building on these preliminaries, we introduce our model, inference and learning approach in the next section. Experimental results are discussed in Section 3.

2. APPROACH

2.1. Model

We use $\boldsymbol{x}$ to denote the input, that is, the image together with its over-segmentation and vanishing points. To capture the appearance of pixels, we use a 21 dimensional vector to represent each pixel, including various color and texture cues. They are also included in $\boldsymbol{x}$. The output variable $\boldsymbol{y}$ defines the box layout, and the latent variable $\boldsymbol{h}$ specifies the clutter as discussed below.

As we have mentioned, the box layout is determined by four parameters (two rays sent from $\boldsymbol{vp}_0$ and two from $\boldsymbol{vp}_1$). We therefore parameterize $\boldsymbol{y}$ as $\boldsymbol{y} = \{y_i\}_{i=1}^4$ that specify where these four rays intersect with the image central line. We use vertical and horizontal central line for $\boldsymbol{vp}_0$ and $\boldsymbol{vp}_1$, respectively. The resulting box layout divides the image into at most five faces: *ceiling, left-wall, right-wall, front-wall, floor* as shown in Figure 2. Note that some of the five faces could be absent if one or multiple rays do not intersect within the extent of the image. Moreover, there exist ambiguous interpretations (*left/front* vs. *front/right*) when only two walls are visible. We also define a base distribution $p_0(\boldsymbol{y})$ (used in inference) over the four dimensional space of $\boldsymbol{y}$. This is estimated by fitting a multivariate Gaussian with diagonal covariance to the training data.

The latent variable $\boldsymbol{h}$ is a binary vector with one entry for each segment in the over-segmented image. The entries indicate whether or not each corresponding segment belongs to the clutter. The variables are called "latent" because they are never observed, that is, we do not require the clutter layout to be labeled in the training images. In fact, drawing the boundaries of the furniture and decorations by hand is not only time-consuming but also ambiguous in many cases. For example, should windows and floor rugs be labeled as clutter? In spite of these difficulties, accounting for clutter appropriately is essential to the success of modeling the scene geometry. This is because the clutter often obscures the geometric structure and occludes boundaries between faces. Moreover, appearance and layout of clutter can vary drastically across different scenes, so it is extremely difficult (if not impossible) to model it consistently. Our latent variable approach effectively addresses this issue.

Our energy-based model measures the consistency among $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{h}$, that is, scores how well the observed image agrees with a given box and clutter layout.

$$\mathbf{E}_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}) = \langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}) \rangle + \mathbf{E}^0(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}). \quad (1)$$

The joint feature mapping $\boldsymbol{\Psi} \in \mathbb{R}^n$ is a vector that contains many features that take into account image cues from various aspects including color, texture, perspective consistency, and overall layout. The parameter vector $\boldsymbol{w}$ specifies the weights of these features and is learned from the labeled training set. $\mathbf{E}^0$ is an energy term that captures our prior knowledge on the role of the latent variables. Given an energy function (with $\boldsymbol{\Psi}$, $\boldsymbol{w}$, and $\mathbf{E}^0$ specified), the problem of inferring the box and clutter layout can be formulated as:

$$(\bar{\boldsymbol{y}}, \bar{\boldsymbol{h}}) = \operatorname*{argmin}_{(\boldsymbol{y}, \boldsymbol{h})} \mathbf{E}_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}). \quad (2)$$

The prior energy term $\mathbf{E}^0$ consists of two parts,

$$\mathbf{E}^0(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}) = \alpha^a \mathbf{E}^a(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h}) + \alpha^c \mathbf{E}^c(\boldsymbol{y}, \boldsymbol{h}). \quad (3)$$

Here $\mathbf{E}^a$ summarizes the appearance variance of each major face excluding all clutter segments. This encodes the prior belief that the major faces should have a

relatively consistent appearance after the clutter is taken out. Specifically,

$$\mathbf{E}^a = \sum_{f\in\mathbb{F}}\sum_{a\in\mathbb{A}} \frac{\sum_{p\in f}\mathbf{1}(p_h=0)\cdot\left(p_a - p_a^f\right)^2}{\sum_{p\in f}\mathbf{1}(p_h=0)} \quad (4)$$

where $\mathbb{F}$ is the set of five major faces: *floor, ceiling, left wall, front wall, right wall*, $\mathbb{A}$ is the set of 21 appearance features, and $\mathbf{1}(\cdot)$ is the indicator function returning one if its argument is true and zero otherwise. Here p is a pixel; p_a is its appearance feature value indexed by a; p_a^f is the average of that appearance feature value within face f; and p_h is the value of the latent variable at that pixel. We define $p_h = 0$ to mean that pixel p is *not* a clutter. Note, however, that this term could be minimized by assigning almost everything as clutter and leaving only a tiny uniform piece with very consistent appearance. To avoid such degenerate solutions we introduce the second term:

$$\mathbf{E}^c = \sum_{f\in\mathbb{F}}\left[|f|\cdot\exp\left(\beta\cdot\frac{\sum_{p\in f}1(p_h=1)}{|f|}\right)\right] \quad (5)$$

which penalizes "clutterness" of each face. We adopt the exponential form because it exhibits superlinear penalty as the percentage of clutter increases. We will address how to determine the parameters β, α^a, α^c, as well as $\boldsymbol{w}$ in Section 2.3.

The features in $\boldsymbol{\Psi}$ are defined in a similar spirit: capturing the synergy of $\boldsymbol{y}$ and $\boldsymbol{h}$ in explaining the observation. For brevity we only give a high level description.

We have features to account for face boundaries. Ideally the boundaries between the five major faces should either be explained by a long line or occluded by some furniture. Therefore we introduce two features for each boundary: the percentage of its length in clutter regions (i.e., occlusion), and the percentage of its length that approximately overlaps with a detected line segment.

We also have features for perspective consistency, which we adopt from Hedau et al.[7] Note that the lines in the image fall into three groups corresponding to the three vanishing points (as in Figure 2). For each face, we are more likely to observe lines from two of the three groups. For example, on the front wall we are more likely to observe lines belonging to $\boldsymbol{vp}_0$ and $\boldsymbol{vp}_1$, but not $\boldsymbol{vp}_2$. We capture this with features that quantify the total length of line segments from each of the three groups in each of the five faces, and have a separate feature value for clutter and non-clutter regions.

In addition we have features that measure cross-face differences. For the 21 appearance values, we compute the difference between each pair of adjacent faces excluding clutter. Finally, we have features for some global properties of the box layout. For each of the five major faces, we use a binary feature indicating whether or not it is absent, and its percentage area in the image. For each of the four parameters $\{y_i\}_{i=1}^4$, we compute their likelihood under $p_0(\boldsymbol{y})$ as bias features.

2.2. Approximate inference

For now we assume that all model parameters are fixed and we want to solve the minimization problem (2). Because the joint feature mapping $\boldsymbol{\Psi}$ and prior energy $\mathbf{E}^0$ are defined in a rather complex way, it cannot be solved analytically. Our inference procedure is based on ICM (Iterated Conditional Modes[1] as shown in Algorithm 1. The basic idea is to iteratively perturb $\boldsymbol{h}$ and $\boldsymbol{y}$ and to accept the move if the energy decreases. To perturb $\boldsymbol{h}$ we flip one segment at a time (between *clutter* and *non-clutter*). To perturb y we sample one of its four components from a Gaussian centered at its original value.

Algorithm 1 Stochastic Hill-Climbing for Inference

```
Input: w, x
Output: ȳ, h̄,
for a number of random seeds do
    sample ȳ from p_0(y)
    h̄ ← argmin_h E_w(x, ȳ, h) by ICM
    repeat
        repeat
            perturb a parameter of y as long as it decreases
            the objective
        until convergence
        h̄ ← argmin_h E_w(x, ȳ, h) by ICM
    until convergence
end for
```

Complexity of the algorithm depends on a number of design choices. For example, a larger number of segments (dimensionality of $\boldsymbol{h}$) may be able to model the clutter at a finer scale but could potentially make inference slow to converge as introducing more latent variables generally increases the number of iterations required by the ICM algorithm. In our experimental setting, the inference running time is typically one or two minutes for each image.

In our experiments, we also compare to another baseline inference method that does not make use of the continuous parametrization of $\boldsymbol{y}$. Specifically, we independently generate a large number of candidate boxes from $p_0(\boldsymbol{y})$, infer the latent variable for each of these discrete choices, and pick the one with the lowest energy.

2.3. Parameter learning

First, consider the parameters in the prior term $\mathbf{E}^0$. When we introduce the latent variables we bear in mind that they should account for the *clutter* such as chairs, desks, sofas, *etc*. However, the algorithm has no access to any supervised information on the latent variables. Given the limited training data, it is hopeless to expect the learning process to figure out the concept of *clutter* by itself. To tackle this, we introduce the prior term $\mathbf{E}^0$ to capture the concept of clutter and constrain the learning process. Specifically, the parameters in $\mathbf{E}^0$, namely α^a, α^c, and β, are determined by cross-validation on the training set and fixed throughout the learning process.

Given the training set $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^m$, we learn the parameters $\boldsymbol{w}$ discriminatively using struct-SVM,[17, 20] which is a

large margin based learning formulation for structured prediction problems. Let $\underline{\mathbf{E}}_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}) = \min_{\boldsymbol{h}} \mathbf{E}_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{h})$. Our learning objective is:

$$\underset{\boldsymbol{w}, \xi}{\text{minimize}} \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{m}\sum_{i=1}^{m}\xi_i, \text{ s.t. } \forall i,\ \xi_i \geq 0 \text{ and} \quad (6)$$

$$\forall i, \boldsymbol{y} \neq \boldsymbol{y}_i,\ \underline{\mathbf{E}}_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{y}) - \underline{\mathbf{E}}_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{y}_i) \geq 1 - \frac{\xi_i}{\Delta(\boldsymbol{y}, \boldsymbol{y}_i)}, \quad (7)$$

where $\Delta(\boldsymbol{y}, \boldsymbol{y}_i)$ is the loss function that measures the difference between the candidate output $\boldsymbol{y}$ and the ground truth layout $\boldsymbol{y}_i$. Here we use the percentage of pixels that are labeled differently by the two box layouts as the loss function.

Roughly speaking, the learning objective states that the true box layout $\boldsymbol{y}_i$, when accompanied with proper clutter estimation $\boldsymbol{h}$, should better explain the observation $\boldsymbol{x}_i$ than any other combinations of $\boldsymbol{y}$ and $\boldsymbol{h}$. To account for model limitations, we introduce slack variables ξ_i scaled by the loss function. This scaling has the effect of requiring a larger margin (or confidence) from hypotheses that are far from the ground truth box layout $\boldsymbol{y}_i$ than from hypotheses that are similar to it.

Optimizing the learning objective is difficult because the number of constraints in equation (7) is infinite. Even if we discretize the parameter space of $\boldsymbol{y}$ in some way, the total number of constraints is still intractably large. And each constraint involves an embedded inference problem for the latent variables. Generally this is tackled by the *cutting plane method*, i.e., gradually adding violated constraints to the optimization problem,[12, 17] which involves an essential step of *loss augmented inference* that tries to find the output variable $\hat{\boldsymbol{y}}$ for which the constraint is most violated given the current parameters $\boldsymbol{w}$. In our problem, it corresponds to following inference problem:

$$(\hat{\boldsymbol{y}}, \hat{\boldsymbol{h}}) = \underset{\boldsymbol{y}, \boldsymbol{h}}{\text{argmax}} (1 + \mathbf{E}_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{h}_i) - \mathbf{E}_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{y}, \boldsymbol{h})) \cdot \Delta(\boldsymbol{y}, \boldsymbol{y}_i), \quad (8)$$

where the latent variables $\boldsymbol{h}_i$ should take the value that best explains the ground truth box layout under current model parameters:

$$\boldsymbol{h}_i = \underset{\boldsymbol{h}}{\text{argmin}}\ \mathbf{E}_{\boldsymbol{w}}(\boldsymbol{x}_i, \boldsymbol{y}_i, \boldsymbol{h}). \quad (9)$$

Equations (8) and (9) are solved by the same inference method as that we introduced in Section 2.2. However, we use a looser convergence criterion to speed up loss augmented inference as it has to be performed a large number of times in learning. The overall learning algorithm (following Tsochantaridis et al.[17]) is shown in Algorithm 2.

Algorithm 2 Overall Learning Procedure

```
Input: {(x_i, y_i)}_{i=1}^m, C, ε_final
Output: w
Cons ← ∅
ε ← ε_0
repeat
  for i = 1 to m do
    find (ŷ, ĥ) by solving (8) using Algorithm 2
    if the constraint in (7) corresponding to (ŷ, ĥ) is
    violated more than ε then
      add the constraint to Cons
    end if
  end for
  update w by solving the relaxed QP (6) given Cons
  for i = 1 to m do
    update h_i by solving (9)
  end for
  if number of new constraints in last iteration is less
  than threshold then
    ε ← ε/2
  end if
until ε < ε_final and number of new constraints in last
  iteration is less than threshold
```

3. EXPERIMENTAL RESULTS

We experimentally verified our method on the dataset introduced by Hedau et al.[7] The dataset consists of 314 images, each with hand-labeled box layout and clutter. It also specifies the training-test split (209 for training, 105 for test), which we use for reporting results. Performance is measured by pixel-error-rate: the percentage of misclassified pixels in the task of classifying them into one of the five classes. Our approach achieves an error-rate of 20.1% *without* clutter labels, compared to 26.5% in Hedau et al.[7] without clutter labels and 21.2% with clutter labels. Details are shown in Table 1.

Each row in Table 1 shows a different performance metric and each column represents a different algorithm. Briefly we have: *Row 1*: pixel error rate. *Row 2 and 3*: the number of test images (out of 105) with pixel error rate under 20% and 10%. *Column 1*: Hoiem et al.'s algorithm. *Column 2*: Hedau et al.'s method without clutter label. *Column 3*: Hedau et al.'s method with clutter label. The first three columns are directly copied from Hedau et al.[7] *Column 4*: Our method (without clutter label). The remaining columns will be explained below.

In order to validate the effects of prior knowledge in constraining the learning process, we take out the prior knowledge by adding the two terms $\mathbf{E}^a$ and $\mathbf{E}^c$ as ordinary features and try to learn their weights. The performance of recovering box layouts in this case is shown in Table 1, column 5 (labeled "without prior"). Although the difference between columns 4 and 5 is small, there are many cases where recovering more reasonable clutter does help in recovering the correct box layout.

One typical example is shown in Figure 3. In Figure 3(a), we can see that the boundary between the *floor* and the *front-wall* (the wall on the right) is correctly recovered even though it is largely occluded by the bed, which is correctly inferred as "clutter", and the boundary is probably found by the appearance difference between the floor and the wall. However, in the model learned without prior constraints, the bed is regarded as non-clutter whereas major parts of the floor and walls are

Table 1. Quantitative results.

	Hoiem et al.[10]	Hedau et al.[7] without	Hedau et al.[7] with	Ours without	without prior	$h = 0$	$h = GT$	cheat
Pixel	28.9%	26.5%	21.2%	*20.1 ± 0.5%*	21.5 ± 0.7%	22.2 ± 0.4%	24.9 ± 0.5%	19.2 ± 0.6%
≤20%	–	–	–	*62 ± 3*	58 ± 4	57 ± 3	46 ± 3	67 ± 3
≤10%	–	–	–	*30 ± 3*	24 ± 2	25 ± 3	20 ± 2	37 ± 4

See text for explanation.

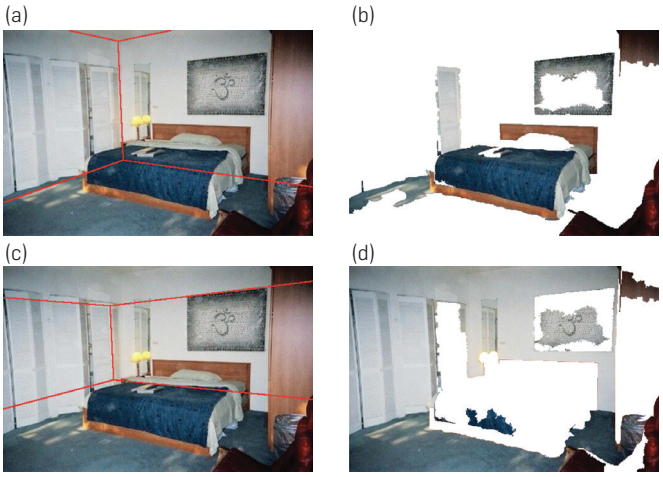
inferred as clutter (this is probably because the term $\mathbf{E}^c$ is not acting effectively with the learned weights), so it appears that the boundary between the *floor* and the *front-wall* is decided incorrectly due to the strong contrast between the white pillow and blue sheet.

More examples of inference result are shown in Figure 4.

In a second experiment, we fixed the latent variables $\boldsymbol{h}$ to be all zeros (i.e., assuming no clutter). The results are shown in column 6 of Table 1 (labeled $\boldsymbol{h} = 0$). The no clutter assumption gives rise to an accuracy of 22.2%, a 2% reduction over our model. However, it still considerably improves upon the 26.5% accuracy obtained by Hedau et al. in the case of no clutter labels. We note, however, that in this analysis, Hedau et al. only used "perspective consistency" features. When using the clutter labels during training, however, Hedau et al. incorporated other kinds of features and the supervised surface label classification method in Hoiem et al.[10] By fixing $\boldsymbol{h}$ to be all zeros we decompose our performance improvement upon Hedau et al.[7] into two parts: (i) using the richer set of features, and (ii) accounting for clutter with latent variables. Although the improvement brought by the richer set of features is larger, the effect of accounting for clutter is also significant.

We also tried to evaluate a supervised learning approach, in which we fix the latent variables h to be the hand-labeled clutter layout.

Figure 3. Example result of recovering box and clutter layout. The clutter layouts are shown by removing all non-clutter segments. (a) Inferred box layout using model learned *with* prior knowledge. (b) Inferred clutter layout using model learned *with* prior knowledge. (c) Inferred box layout using model learned *without* prior knowledge. (d) Inferred clutter layout using model learned *without* prior knowledge.



The results are shown in column 7 of Table 1 (labeled $\boldsymbol{h} = GT$). Somewhat surprisingly, the results are considerably worse than those obtained by our model, and even worse than assuming no clutter ($\boldsymbol{h} = 0$). To understand why, we quantitatively compare our recovered clutter to the hand-labeled clutter, and see that the average pixel difference is around 30% on both, the training and test set. A closer examination, shown in Figure 5, demonstrates the difference between the hand-labeled clutter and the clutter recovered by our method (on the test set). Generally, the hand-labels include much less clutter than our algorithm recovers. Because delineating objects by hand is very time consuming, usually only one or two pieces of major furniture are labeled as clutter. Some salient clutter is missing in the hand-labels such as the cabinet and the TV in the image of the 1st row, the smaller sofa in the image of the 5th row, and nothing is labeled in the image of the 3rd row. Therefore, it is not surprising that learning with hand-labeled clutter does not result in a better model. We also tried to fix the latent variable to be the hand-labeled clutter in *both* learning and inference. Note that the algorithm is actually "cheating" as it has access to the labeled clutter even in the testing phase, so it should be expected to perform well. Surprisingly, it only gives slightly better results (Table 1, column 8, labeled "cheat") than our method. It is also worth noting that there is around 6–7% (out of the 20.1%) of pixel error due to incorrect vanishing point detection results. The error rate of 6–7% is estimated by assuming a perfect model that always picks the best box generated from the vanishing point detection result, and performing stochastic hill-climbing to infer the box using the perfect model. The rest of the error attributes to the limitation of the model and inference algorithm. For example, the features we used do not perfectly characterize the structure of an indoor scene.

We compare our inference method (Algorithm 2) to the baseline method (of evaluating hypotheses independently) described in Section 3.2. Figure 6 shows the average pixel error rate over test set versus the number of calls (in logscale) to the joint feature mapping Ψ, which is an indication of running time.

The actual running time of the inference algorithm depends on the number of random starts and convergence criteria. In our current experiments we use 50 random re-starts and run a maximum of 100 inner loop iterations. It takes on average 16 seconds to run inference for one image, 25% of which are for performing ICM on latent variables.

In Figure 7 we show the performance of the learned model on test set versus the number of iterations in learning. Empirically the learning procedure attains a small

Figure 4. More results for comparing learning with and without prior constraints. The 1st and 2nd columns are the result of learning with prior constraints. The 3rd and 4th columns are the result of learning without prior constraints. In many cases, recovering more reasonable clutter does help in recovering the correct box layout.

error rate after a small number of iterations, and then fluctuates around this error rate due to the approximate loss-augmented inference step of learning.

4. RELATED WORK

Our method is closely related to a recent work of Hedau et al. [7]. We adopt their idea of generating box layouts from the vanishing points. However, they use supervised classification of surface labels[10] to identify clutter (furniture), and use the trained surface label classifier to iteratively refine the box layout estimation. Specifically, in each iteration they use the estimated box layout to add features to supervised surface label classification, and then use the classification result to lower the contribution of "clutter" image regions in estimating the box layout. Thus, their method requires the user to label clutter regions in the training set.

Latent variables have been exploited in the computer vision literature in various tasks such as object detection, recognition and segmentation. They can be used to represent visual concepts such as occlusion,[18] object parts,[3] and image-specific color models.[15] Introducing latent variables into structured prediction was shown to be effective in several applications.[20] An interesting aspect of our work is that latent variables are used in direct correspondence with a concrete visual concept (clutter in the room).

Since the publication of our initial work,[19] reasoning about the 3D geometry and semantics of scenes has been further addressed in many recent works such as Geiger et al.,[4] Gupta et al.,[6] Hedau et al.,[8] Lee et al.,[13] Pepik et al.,[14] and Tsai et al.[16] to name a few, which have demonstrated obtaining more fine-grained 3D context information or using it to help other vision tasks such as object detection.

5. DISCUSSION

In this paper we addressed the problem of recovering the geometric structure as well as clutter layout from a single

Figure 5. Sample results for comparing the recovered clutter by our method and the hand-labeled clutter in the dataset. The 1st and 2nd columns are recovered box and clutter layouts by our method. The 3rd column (right) is the hand-labeled clutter layouts. Our method usually recovers more objects as "clutter" than people would bother to delineate by hand. For example, the rug with a different appearance from the floor in the 2nd image, paintings on the wall in the 1st, 4th, 5th, and 6th images, and the tree in the 5th image. There are also major pieces of furniture that are missing in the hand-labels but recovered by our method, such as the cabinet and TV in the 1st image, everything in the 3rd image, and the small sofa in the 5th image.

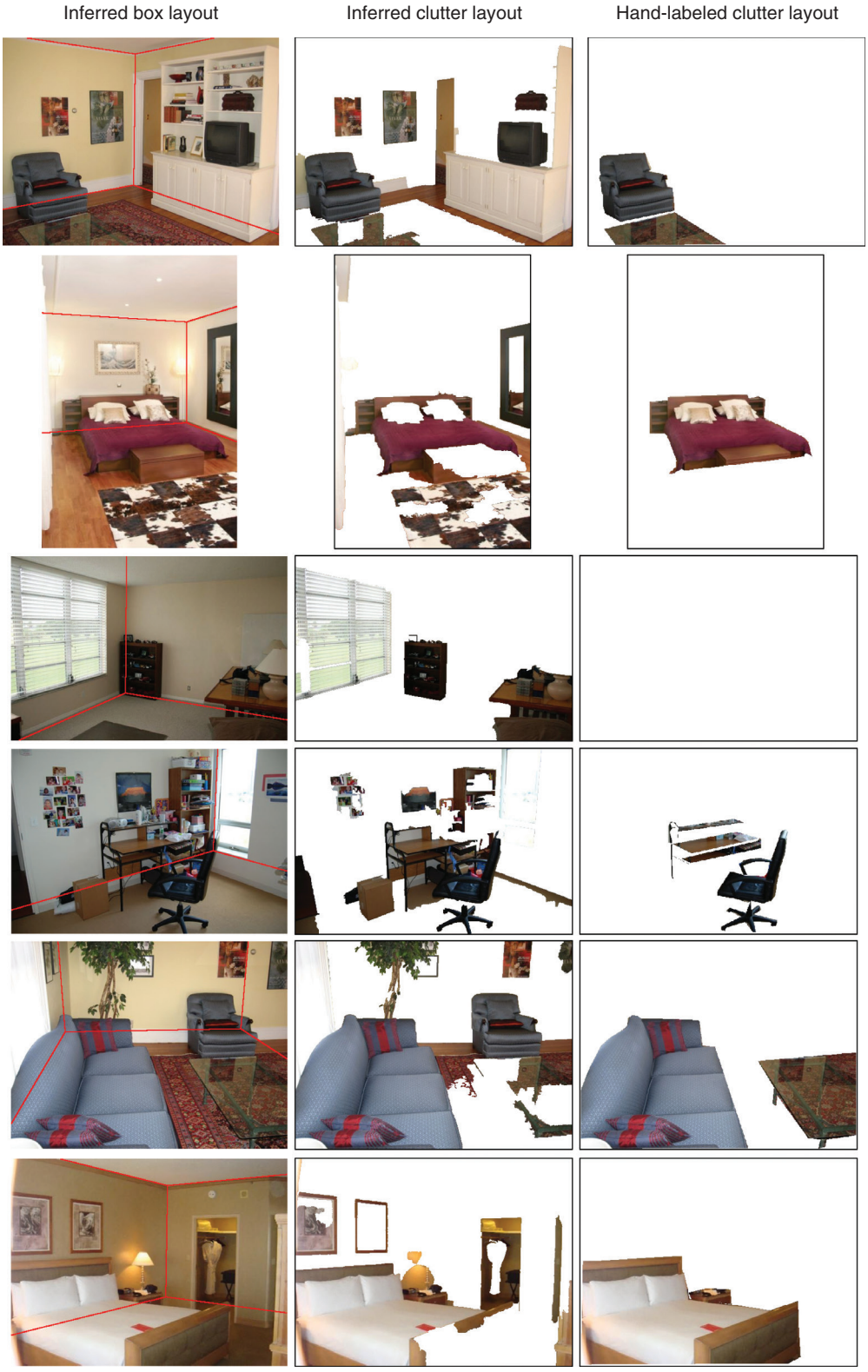


Figure 6. Comparison between the inference method described in Algorithm 2 and the baseline inference method that evaluates hypotheses independently.

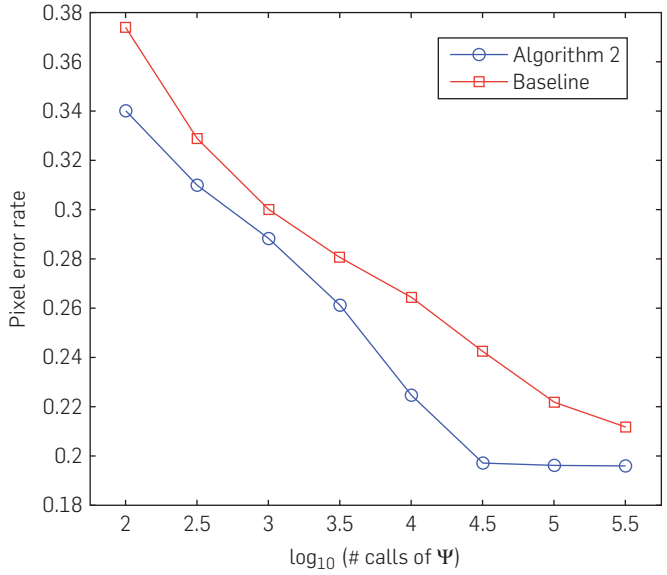


Figure 7. Empirical convergence of the learning procedure.

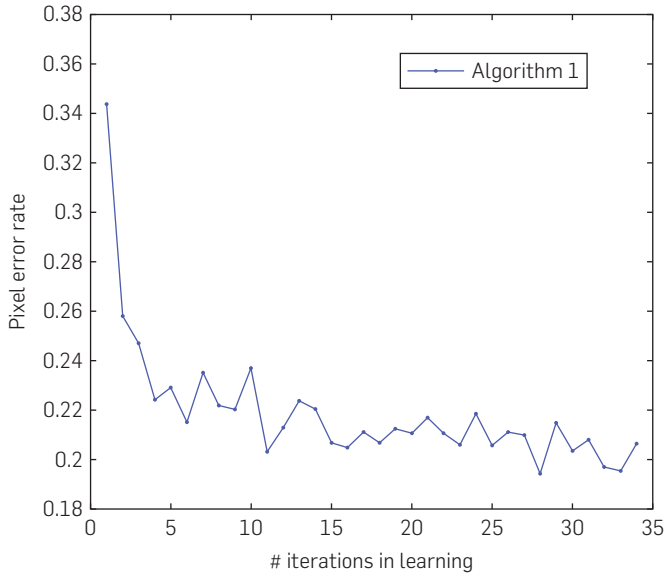


image. We used latent variables to account for indoor clutter, and introduced prior terms to define the role of latent variables and to constrain the learning process. Our approach, without using clutter labels in training, outperforms a baseline method that does use them.

This improvement can be attributed to three main technical contributions: (i) we introduce latent variables and the prior terms to account for the clutter in a principled manner; (ii) we design a rich set of joint features to capture the compatibility between image and the box-clutter layouts; and (iii) we perform more efficient and accurate inference by making use of the parametrization of the "box" space.

Learning latent variable based models is known to be susceptible to local optima of the learning objective. A lesson from this paper that we believe could be useful is that, imposing prior knowledge on how the model should function (using fixed prior energy terms in our case) can help guide the learning process to a desirable region of the parameter space, and thereby give rise to a better model.

Acknowledgments

This work was supported by the National Science Foundation under Grant No. RI-0917151, the Office of Naval Research under the MURI program (N000140710747) and the Boeing Corporation.

References

1. Besag, J. On the statistical analysis of dirty pictures. *J. Roy. Stat. Soc. B-48*, 3 (1986), 259–302.
2. Comaniciu, D., Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 5 (2002), 603–619.
3. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell. 32*, 9 (2010), 1627–1645.
4. Geiger, A., Wojek, C., Urtasun, R. Joint 3D estimation of objects and scene layout. In *NIPS* (2011), 1467–1475.
5. Gould, S., Fulton, R., Koller, D. Decomposing a scene into geometric and semantically consistent regions. In *Proceedings of the International Conference on Computer Vision (ICCV)* (2009).
6. Gupta, A., Satkin, S., Efros, A.A., Hebert, M. From 3D scene geometry to human workspace. In *Computer Vision and Pattern Recognition (CVPR)* (2011).
7. Hedau, V., Hoiem, D., Forsyth, D. Recovering the spatial layout of cluttered rooms. In *ICCV* (2009).
8. Hedau, V., Hoiem, D., Forsyth, D. Thinking inside the box: Using appearance models and context based on room geometry. *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, eds. Volume 6316 of *Lecture Notes in Computer Science*. (2010), Springer, 224–237.
9. Heitz, G., Koller, D. Learning spatial context: Using stuff to find things. In *ECCV* (2008), I: 30–43.
10. Hoiem, D., Efros, A.A., Hebert, M. Recovering surface layout from an image. *Int. J. Comput. Vis. 75*, 1 (Oct. 2007), 151–172.
11. Hoiem, D., Efros, A.A., Hebert, M. Closing the loop in scene interpretation. In *CVPR* (2008), 1–8.
12. Joachims, T., Finley, T., Yu, C.N.J. Cutting-plane training of structural SVMs. *Mach. Learn. 77*, 1 (2009), 27–59.
13. Lee, D., Gupta, A., Hebert, M., Kanade, T. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *Advances in Neural Information Processing Systems* (2010), J. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, eds. Volume 23, 1288–1296.
14. Pepik, B., Stark, M., Gehler, P., Schiele, B. Teaching 3d geometry to deformable part models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2012).
15. Shotton, J.D.J., Winn, J., Rother, C., Criminisi, A. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *Int. J. Comput. Vis. 81*, 1 (Jan. 2009), 2–23.
16. Tsai, G., Xu, C., Liu, J., Kuipers, B. Real-time indoor scene understanding using Bayesian filtering with motion cues. In *ICCV* (2011), IEEE, 121–128.
17. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y. Large margin methods for structured and interdependent output variables. *JMLR 6* (2005), 1453–1484.
18. Vedaldi, A., Zisserman, A. Structured output regression for detection with partial truncation. In *Advances in Neural Information Processing Systems* (2009).
19. Wang, H., Gould, S., Koller, D. Discriminative learning with latent variables for cluttered indoor scene understanding. In *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, eds. Volume 6314 of *Lecture Notes in Computer Science* (2010), Springer, 497–510.
20. Yu, C.N.J., Joachims, T. Learning structural svms with latent variables. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning* (2009), ACM, New York, NY, USA, 1169–1176.

Huayan Wang (huayanw@cs.stanford.edu), Computer Science Department, Stanford University.

Daphne Koller (koller@cs.stanford.edu), Computer Science Department, Stanford University.

Stephen Gould (stephen.gould@anu.edu.au), School of Computer Science, Australian National University.