Scene Collaging: Analysis and Synthesis of Natural Images with Semantic Layers – Supplementary Materials

Phillip Isola
MIT
phillipi@mit.edu

Ce Liu
Microsoft Research
celiu@microsoft.com

1. Maximum entropy prior

Our scenegraph grammar (Section 2.3, main text), serves as a binary prior: scenes are either valid or not. In order to compare against a method than can capture more subtle gradations in the prior probability, we also implemented a maximum entropy (maxent) prior over object co-occurrence statistics [3]. With this approach, we express the prior probability of a scene $\mathbf{X}$ as:

$$P(\mathbf{X}) = \frac{1}{Z(\Lambda)} \exp\left(-\sum_i \lambda_i f_i(\mathbf{X})\right), \tag{1}$$

where Z normalizes and the distribution is parameterized by parameters $\lambda_i \in \Lambda$. Many methods exist for learning Λ from training data. We employ gradient descent [2].

The functions f_i measure **object class count statistics** and **object class co-count statistics**.

We approximate the distribution of object class counts with a histogram with bins R. For each object class i,

$$f_{i,r}^{(count)}(\mathbf{X}) = \mathbb{1}(n_i(\mathbf{X}) \in r)), \tag{2}$$

where $n_i(\mathbf{X})$ is the number of objects of class i in the scene $\mathbf{X}$, that is:

$$n_i(\mathbf{X}) = \sum_{\ell \in \mathbf{L}} \mathbb{1}(\tilde{c}_\ell = i). \tag{3}$$

Recall from Section 2 of the main text that $\mathbf{L}$ is the set of dictionary indices of the objects used in a scene $\mathbf{X}$ and $\tilde{c}_\ell$ is the class of an object ℓ. $r \in R$ is a range of integer values. In our current implementation, $R = \{[0], [1], [2, 3], [4, 7], [8, \infty)\}$.

We approximate the distribution of object class co-counts with a 2D histogram $R \times R$. For each pair of object classes (i, j), and histogram bin $(r_1, r_2), r_1, r_2 \in R$, we have the feature:

$$f_{i,j,r_1,r_2}^{(co-count)}(\mathbf{X}) = \mathbb{1}(n_i(\mathbf{X}) \in r_1, n_j(\mathbf{X}) \in r_2). \tag{4}$$

In Section 5.2.1 of the main text, we compare our system with just the scenegraph grammar as a prior to our system with both the scenegraph grammar prior and the maxent prior described above. As shown in Table 4, the addition of the maxent prior decreases performance compared to our scenegraph grammar alone.

2. Segment transformation – details

Translation and scaling:

As summarized in Section 4.2 of the main text, each object from our dictionary is translated and scaled independently. We seek to place the object at a location and scale where it well explains the appearance of the query image, I. Using θ_t to represent the center position of the object's mask and θ_s to represent how much we scale the object's mask, we start with the following objective function:

$$E_1(\theta_t, \theta_s) = \sum_{q \in Q_\ell} \tilde{g}_\ell(f_q^{(I)}) + \lambda_t \sum_{q \in \tilde{Q}_\ell} p_t(\tilde{c}_\ell, q) + \lambda_s p_s(\theta_s), \tag{5}$$

$$Q_\ell = T_{ts}(\tilde{Q}_\ell, \{\theta_t, \theta_s\}), \tag{6}$$

where $\tilde{g}_\ell$ is the segment's appearance model, $p_t(\tilde{c}_\ell, q)$ is a prior over mask position for objects of class $\tilde{c}_\ell$, and $p_s(\theta_s)$ is a prior over scale factor. T_{ts} applies a similarity transformation to the dictionary object mask $\tilde{Q}_\ell$. In order to encourage objects to stay largely within the image frame, we assign a default penalty to all pixels in the transformed mask that fall outside the image frame.

We calculate $p_t(c, q)$ by averaging masks of class c across our dictionary:

$$p_t(c, q) \propto \sum_{\ell \in \mathcal{L} \text{ s.t. } \tilde{c}_\ell = c} \mathbb{1}(q \in \tilde{Q}_\ell). \tag{7}$$

We manually set $p_s(\theta_s) = \{0.75, 0.875, 1, 0.875, 0.75\}$ for the discrete set of scales $\theta_s \in \{0.5, 0.75, 1, 1.5, 2\}$. That is, objects slightly prefer to remain the scale at which they were found in our dictionary.

The objective E_1 prefers that objects scale down as much as possible until they achieve a precise fit onto the query image. In order to prevent all objects from choosing the smallest scale, we add a term that trades off between precision and coverage of the detections:

$$E_2(\theta_t, \theta_s) = \frac{E_1(\theta_t, \theta_s)}{|Q_\ell| + \eta(\tilde{Q}_\ell, \tilde{z}_\ell)}. \quad (8)$$

Here we have scaled E_1 by a softened version of the area the transformed object mask covers. The softening factor, $\eta(\tilde{Q}_\ell, \tilde{z}_\ell)$, controls preference for precision versus recall in the object detections. We set this term so as to prefer recall for large, background objects whereas to prefer precision for small, foreground objects. The intuition for this choice is that objects in the background are likely to be largely covered in the final scene explanation, and consequently, it is okay if only parts of them match the image well. This term is a function of the object's untransformed mask $\tilde{Q}_\ell$ and the layer of the object in its dictionary scene, $\tilde{z}_\ell$:

$$\eta(\tilde{Q}_\ell, \tilde{z}_\ell) \propto \frac{1}{|\tilde{Q}_\ell|\tilde{z}_\ell}, \quad (9)$$

We choose transformation parameters that maximize our objective:

$$\{\theta_t^*, \theta_s^*\} = \arg\max_{\theta_t, \theta_s} E_2(\theta_t, \theta_s). \quad (10)$$

To choose θ_s^* we try each of our discrete set of scales. To choose θ_t^*, we adopt a sliding window approach, searching over all possible placements of the object mask such that its center is within the image frame (however, for dictionary objects that touch an image border, we only consider placements such that the object still touches the image border). We use convolution between the object mask and image features to perform this search efficiently.

Trimming and growing:

We edit object mask silhouettes after each iteration of greedy optimization (Algorithm 1 in the main text; labeled as TRIMANDGROW). We formulate this part of the problem as 2D MRF-based segmentation, in which each object in the scene becomes a segment class label. Here, we denote object segment labels as a 2D array $\ell(q)$, and minimize the following energy function using BP-S [1]:

$$-\log p(\ell \mid I, \mathbf{X}) = \sum_q \tilde{g}_\ell(f_q^{(I)}) + \lambda_1 \sum_q \psi_\ell(q) + \quad (11)$$

$$\lambda_2 \sum_{\{p,q\} \in \epsilon} \phi(\ell(p), \ell(q); I) + \log Z, \quad (12)$$

where Z normalizes. The data term $\tilde{g}_\ell(\cdot)$ is our appearance model from Section 2.2 of the main text. Spatial priors $\psi_\ell(\cdot)$ are provided by our object visibility masks (Equation 4 in the main text):

$$\psi_\ell(q) = (G * V_\ell)(q), \quad (13)$$

where G is a Gaussian kernel with scale proportional to the object's mask area, $|Q_\ell|$.

For the spatial smoothness term $\phi(\cdot)$, we borrow the image-sensitive smoothness potential from [1]:

$$\phi(\ell(p), \ell(q); I) = \mathbb{1}(\ell(p) \neq \ell(q)) \left(\frac{\xi + e^{-\gamma \|I(p) - I(q)\|^2}}{\xi + 1} \right), \quad (14)$$

with $\gamma = (2 < \|I(p) - I(q)\|^2 >)^{-1}$.

This silhouette editing process leaves us with a 2D map of object labels $\ell(q)$. We use this map to update our object visibility masks: $\mathcal{V}_\ell = \cup_q \mathbb{1}(\ell = \ell(q))$. These visibility masks affect the synthesized scene's likelihood (Equation 6 in the main text), and are used to calculate pixel-wise and class-wise label accuracy (Section 5 of the main text). Upon each invocation of TRIMANDGROW, we discard the old silhouette edits and start afresh from the visibility masks given by Equation 4 in the main text.

3. Random scene synthesis – details

During random scene synthesis, we do not have a target image to match, so several changes need to be made to our collaging algorithm. First, the likelihood term is set to be proportional to the number of pixels the collage covers: $\log P(I|\mathbf{X}) \propto \sum_{\ell \in \mathbf{L}} |\mathcal{V}_\ell|$. This biases inference toward collages that fill the entire image frame. Second, segment transformation and trimming and growing are skipped – instead segments are placed exactly where they were found in the dictionary scene they came from. Third, we choose the layer on which to place each object segment ℓ by finding the layer in the collage that best matches the object's layer in the dictionary scene from which it was sampled. Let $A(\tilde{z}_\ell)$ be the histogram of object class counts above $\tilde{z}_\ell$ in the dictionary scene and B be the histogram below. Further, let $A'(z)$ and $B'(z)$ represent the histogram of object class counts above and below z in the scene collage being synthesized. We place the object on the layer z_ℓ^* that minimizes the following sum of histogram intersections:

$$z_\ell^* = \arg\max_z \sum_k \min(A(\tilde{z}_\ell)_k, A'(z)_k) + \min(B(\tilde{z}_\ell)_k, B'(z)_k). \quad (15)$$

4. Additional results

In Figures 1, 2, and 3 we display additional example parses on each dataset. Figures 4, 5, and 6 show the average per-class accuracy of our algorithm on the top 30 most frequent classes in each dataset. Figure 7 shows several characteristic failure cases for our algorithm.

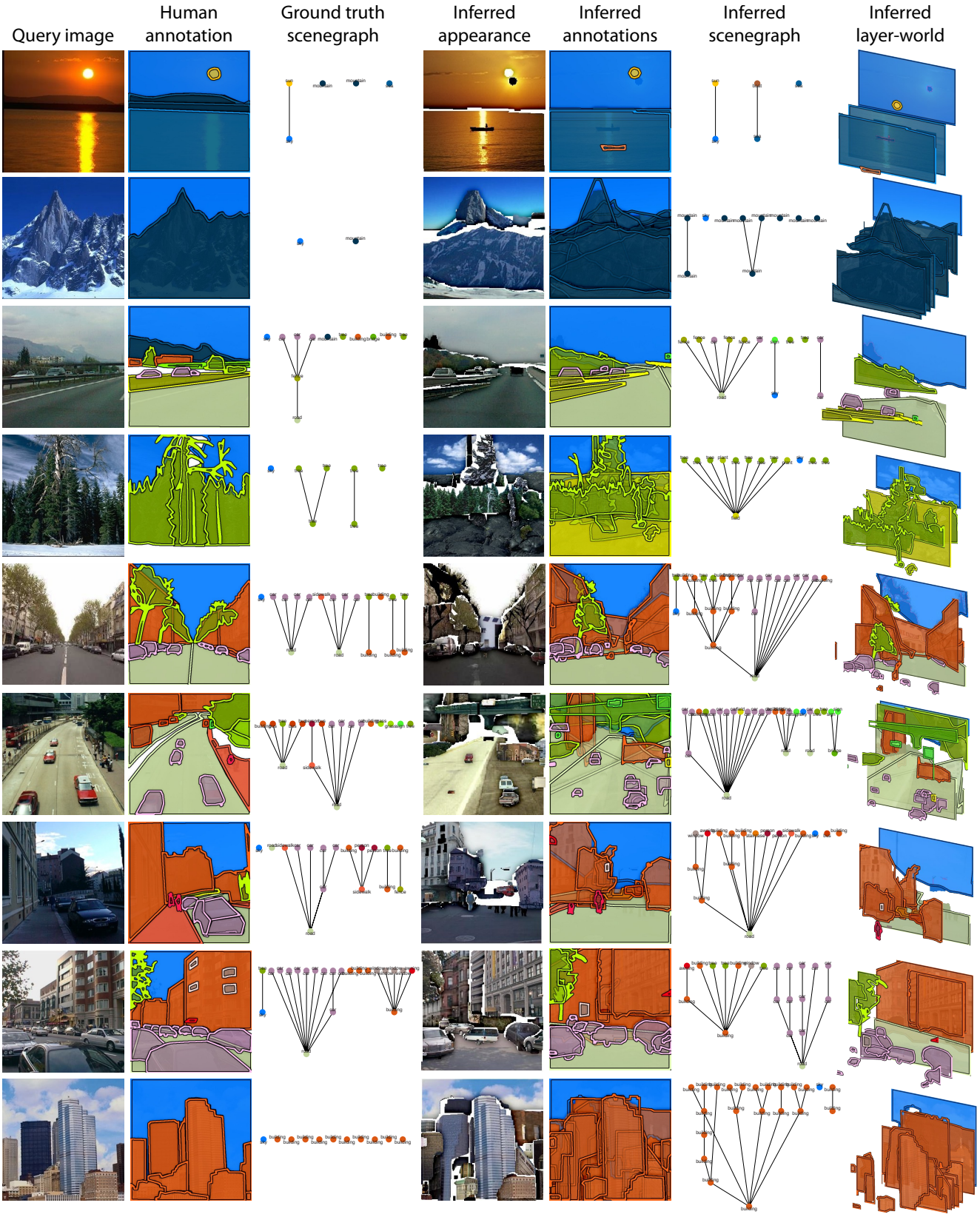

Figure 1: Additional example results parsing LMO images. Zoom in to see details, such as the object labels on the scene-graphs. The last row shows an example in which the dictionary contained the same building as in the query image (although a different photo of that building). This occurs fairly frequency in LMO and SUN since these datasets contain many cases of multiple photos of more or less the same place.

Figure 2: Additional example results parsing SUN images. Zoom in to see details, such as the object labels on the scene-graphs.

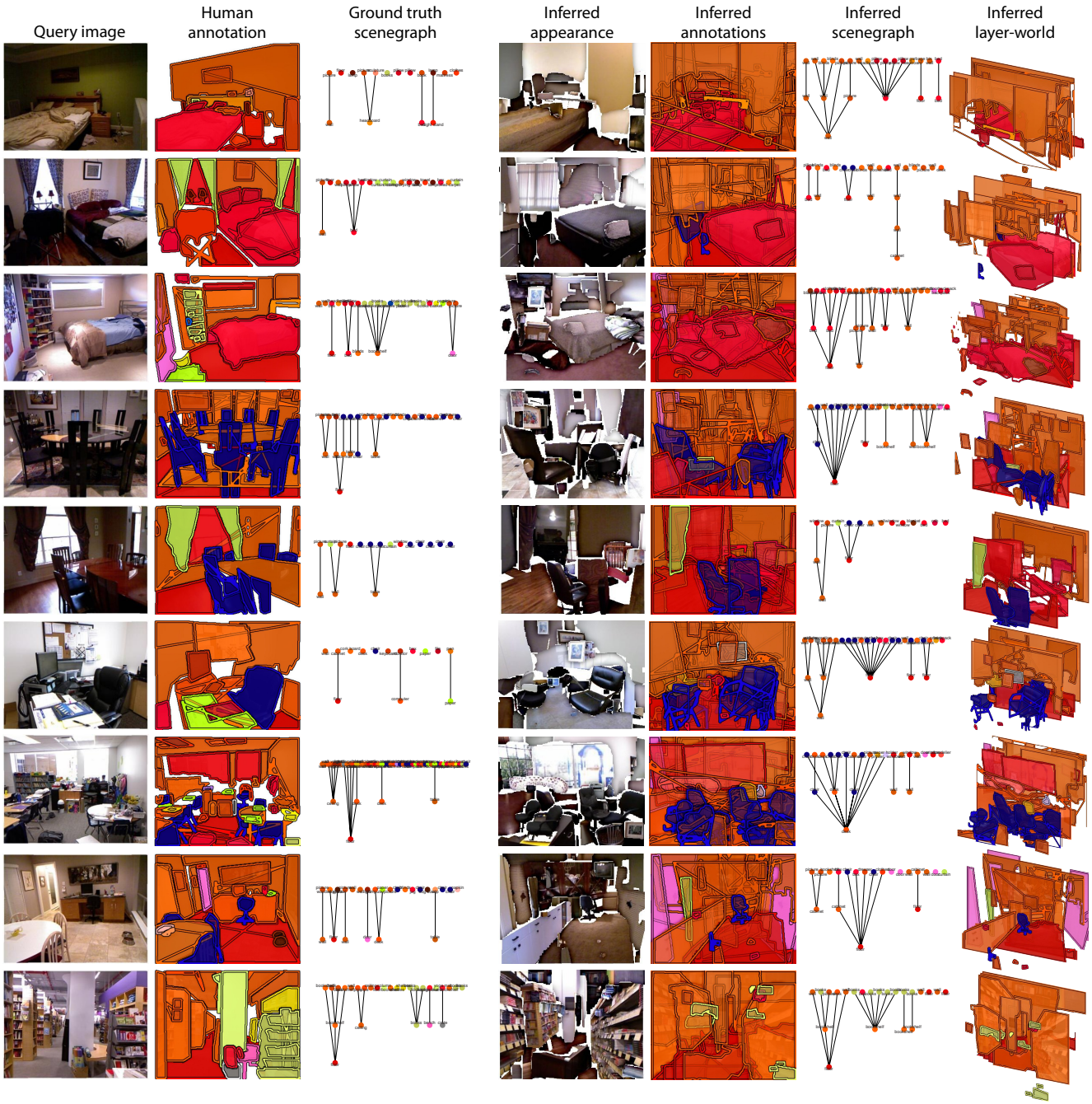

Figure 3: Additional example results parsing NYU RGBD images. Zoom in to see details, such as the object labels on the scenegraphs.

References

[1] C. Liu, J. Yuen, and A. Torralba. Nonparametric Scene Parsing via Label Transfer. *PAMI*, 2011. 2

[2] C. Liu, S. Zhu, and H. Shum. Learning inhomogeneous Gibbs model of faces by minimax entropy. *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, 1:281–287 vol. 1, 2001. 1

[3] J. Yuen, C. Zitnick, C. Liu, and A. Torralba. A Framework for Encoding Object-level Image Priors. *Microsoft Research Technical Report*, 2011. 1

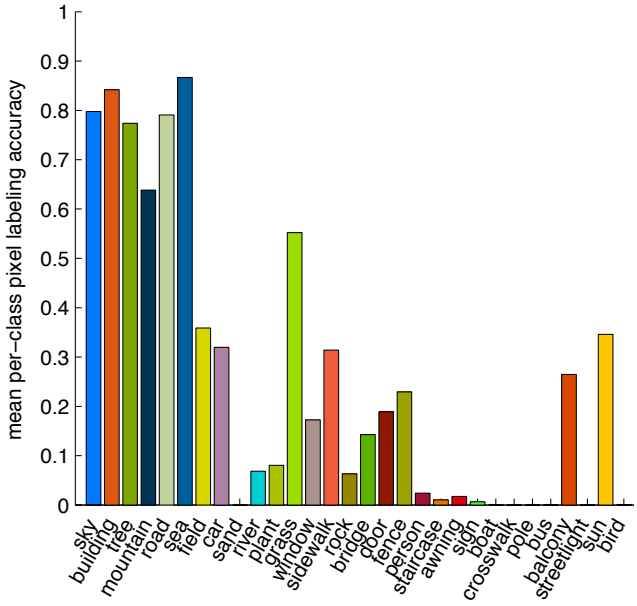

Figure 4: Mean per-class pixel labeling accuracy on the LMO dataset.

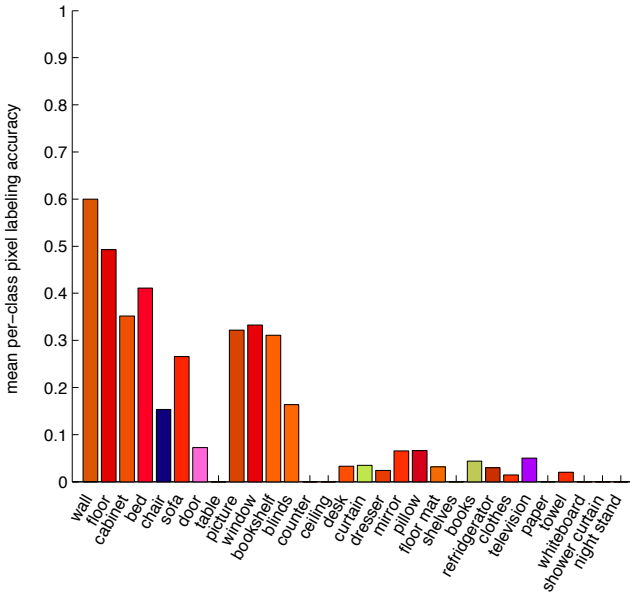

Figure 6: Mean per-class pixel labeling accuracy on the top 30 most common object classes in the NYU RGBD test set.

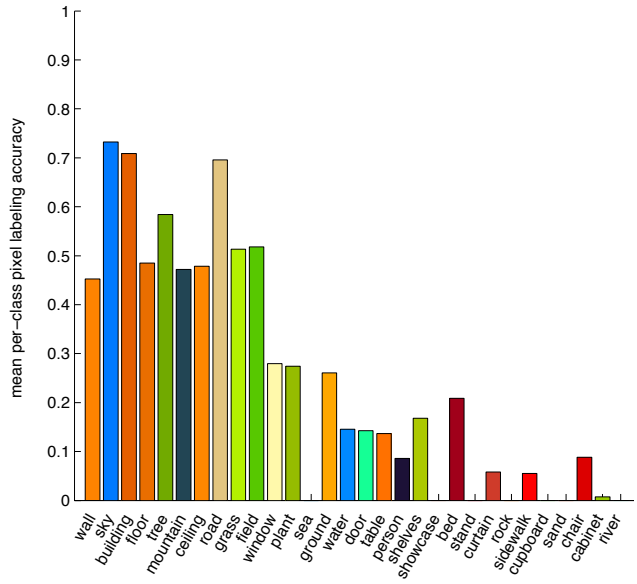

Figure 5: Mean per-class pixel labeling accuracy on the top 30 most common object classes in the SUN test set.

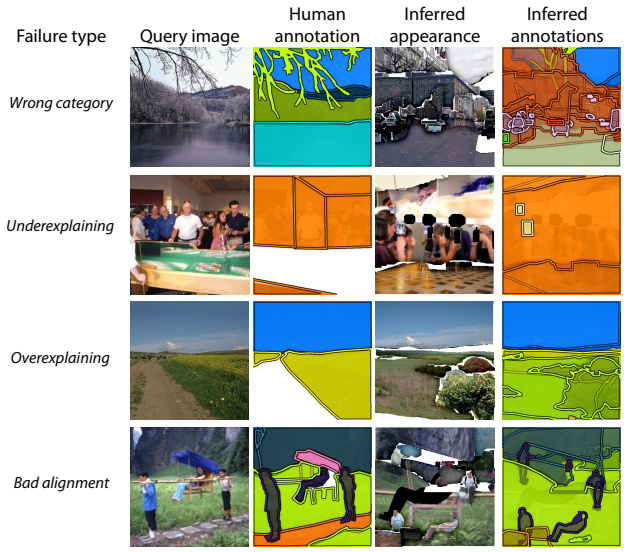

Figure 7: Characteristic failure modes of our algorithm. *Top row*: Sometimes the algorithm gets the entire scene category wrong. *Second row*: Our algorithm often runs into difficulty with cluttered scenes full of small objects. Groups of small objects are sometimes explained by one big object that resembles the ensemble appearance of the small objects. One reason for this is that often the human annotations do not mark all small objects in a scene. In this example, the human annotations did not mark the people in either the query image or in the scene whose segments were used in the scene collage. *Third row*: Conversely, sometimes the algorithm hallucinates small objects where none are necessary, as in the addition of the foreground plants in this simple image of a field. Together, the second and third rows demonstrate that the algorithm has difficulty choosing between explaining a region with several small objects versus one big object. *Bottom row*: Our current segment transformation algorithm produces fairly coarse alignments, and often makes false matches. The system often finds reasonable small objects to place into a scene collage (such as the people in this collage), but gets their position wrong, leading to low per-class accuracy; better small object detection and alignment is an important direction for future work.