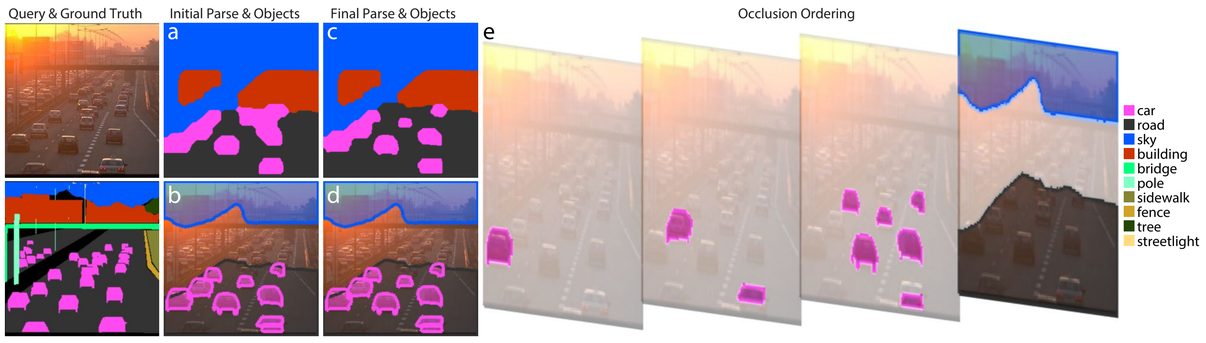
Scene Parsing with Object Instances and Occlusion Ordering*

Joseph Tighe, UNC Marc Niethammer, UNC Svetlana Lazebnik, UIUC

*To appear in CVPR 2014

Overview

We propose a method to interpret a scene by assigning a semantic label at every pixel and inferring the spatial extent of individual object instances together with their occlusion relationships. Starting with an initial pixel labeling and a set of candidate object masks for a given test image, we select a subset of objects that explain the image well and have valid overlap relationships and occlusion ordering. Then we alternate between using the object predictions to refine the pixel labels and vice versa.



Pixel Inference

Given a test image, first we infer a semantic label at each pixel.

- Our "Finding Things" system (Tighe and Lazebnik 2013) infers a field of pixel labels $\mathbf{c}$ by minimizing the following CRF objective function

$$E(\mathbf{c}) = \sum_i \underbrace{\psi_u(p_i, c_i)}_{\text{unary (eq.2)}} + \sum_i \underbrace{\psi_o(p_i, c_i, \mathbf{x})}_{\text{object potential}} + \sum_{i<j} \underbrace{\psi_{sm}(c_i, c_j)}_{\text{smoothing}} . \quad (1)$$

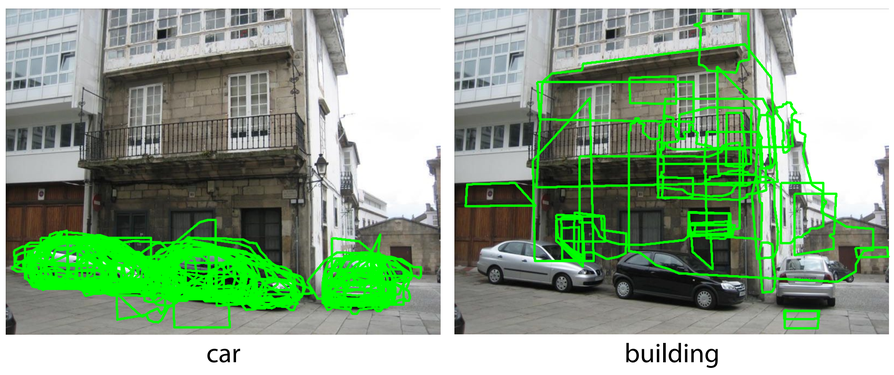
- The unary is obtained by combining region- and detector-based dataterms via 1-vs-all SVMs

$$\psi_u(p_i, c_i) = -\log \sigma(E_{\text{SVM}}(p_i, c_i)) , \quad (2)$$

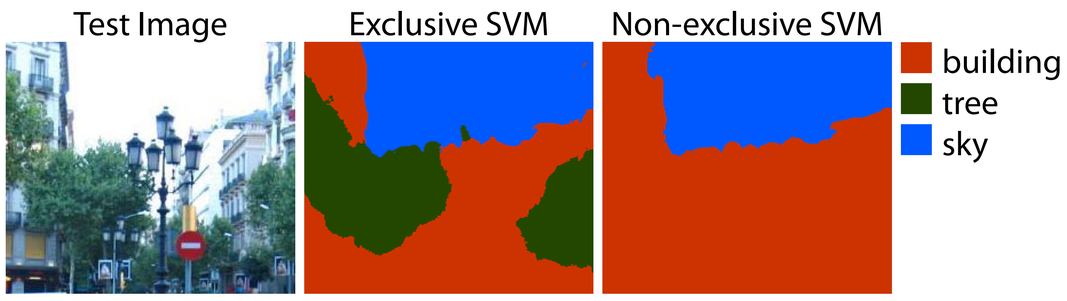
Instance Inference

Guided by these pixel labels we infer a set of object instance masks to cover the image (possibly incompletely), as well as the occlusion ordering of these masks.

- Obtain "Thing" candidate object instances from the positive per-exemplar detections



- Obtain "Stuff" candidate object instances from foreground and backgrounds stuff parses



- Score each candidate using the predicted pixel labels c_i and non-exclusive SVM output

$$s_m = w_m \hat{s}_m , \quad \text{where } \hat{s}_m = \sum_{p_i \in O_m} (1 + E_{\text{NXSVM}}(p_i, c_m) \text{ and } w_m = \frac{1}{|O_m|} \sum_{p_i \in O_m} V_{\text{p}}(c_m, c_i) . \quad (3)$$

V_{p} is 1 if the instance label c_m can appear *behind* the pixel label c_i and 0 otherwise

- Determine valid pairwise overlap relationships ($V_{\text{o}}(m, n)$) for all candidate instances (m, n) by thresholding counts on the training set

  For example, cars must overlap other cars by 50% or less, while headlights must overlap cars by 90% or more.

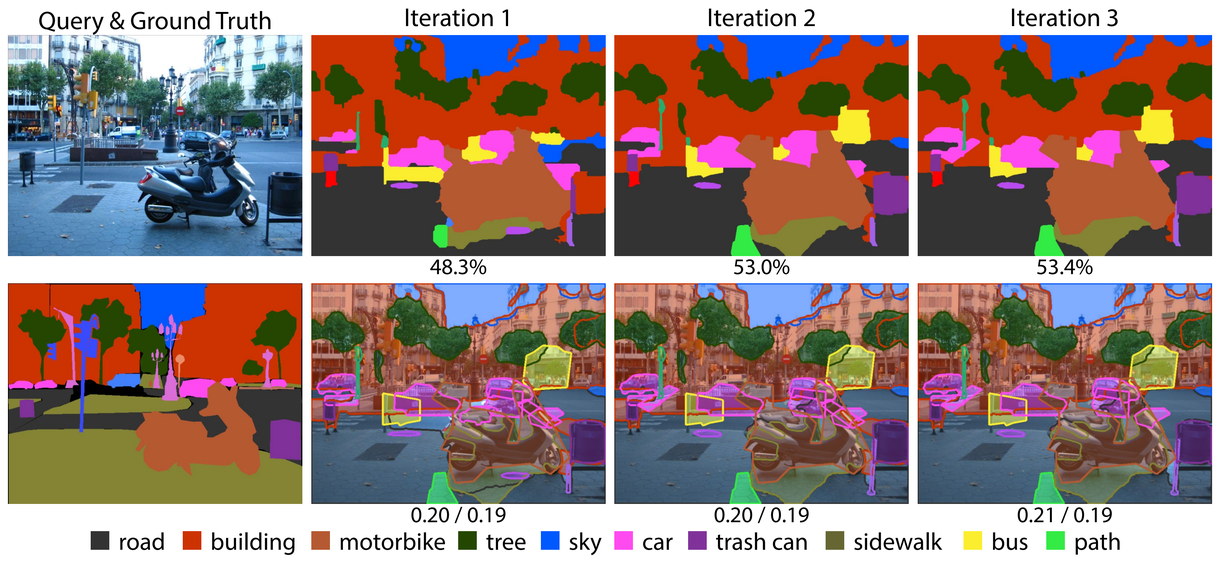
- Given the score for each candidate and the pairwise overlap constraints, infer the configuration (x) that gives the highest score via quadratic integer programing

$$\begin{aligned} &\text{minimize}_{\text{x}} \sum_m s_m x_m - \sum_{n \neq m} s_{mn} x_m x_n 1_{\{c_m = c_n\}} \\ &\text{s.t. } \forall (x_m = 1, x_n = 1, n \neq m) \;\; V_{\text{o}}(m, n) = 1 , \end{aligned} \quad (4)$$

- Infer the occlusion order graph by finding the the most likely order for each pair of occluding objects while insuring there are no loops

Alternating Pixel/Instance Inference

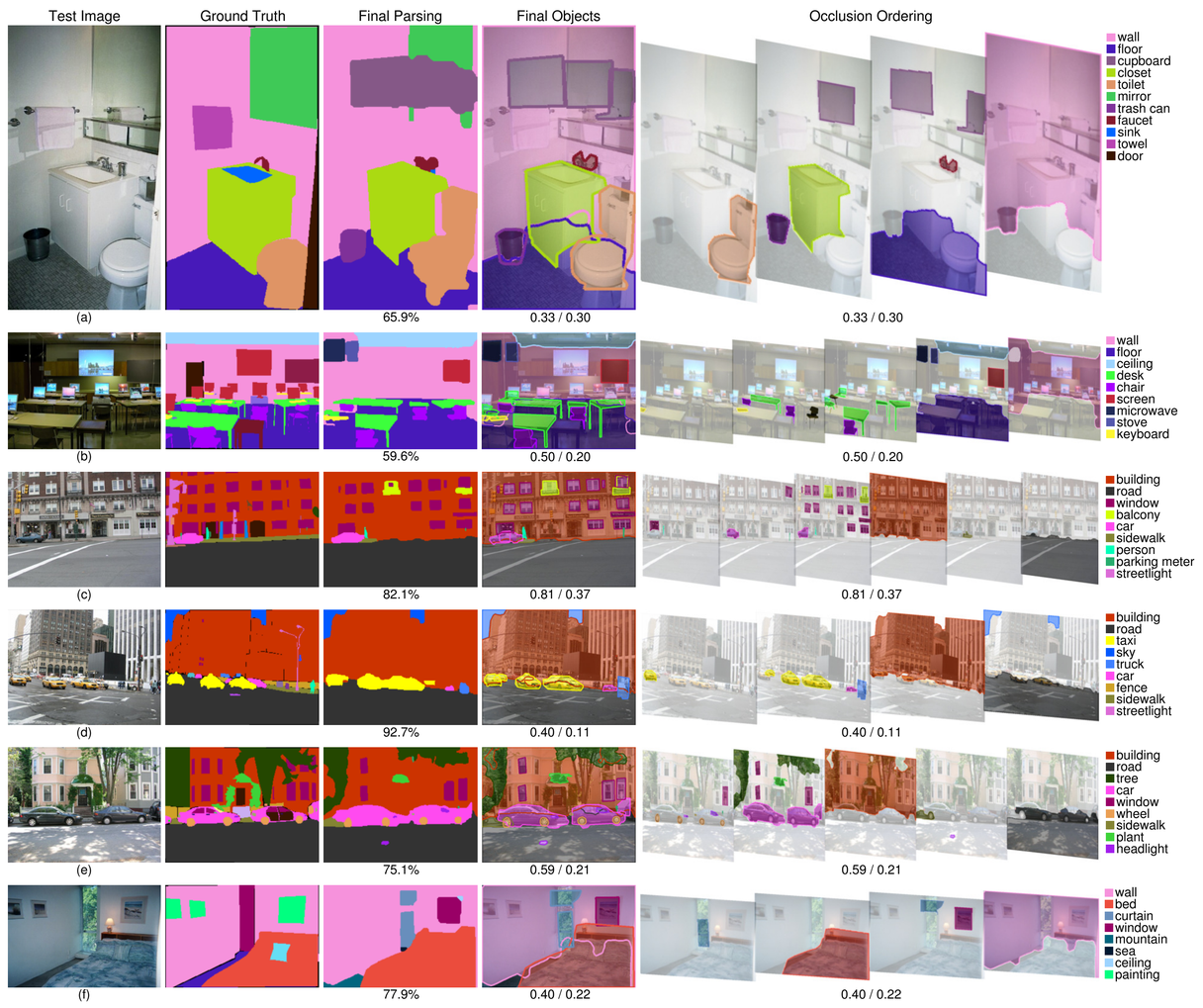
- Use the object instance predictions (x) with their occlusion ordering as an additional potential (ψ_o) for equation 1 to refine the pixel labels (c)
- In turn, use the updated pixel labels (c) from equation 1 in the instance inference to refine the inferred objects



LM+SUN Results

LM+SUN dataset: 45,176 training images, 500 test images, 232 class labels.

	Instances	Object P/R	Pixel P/R	Pixel Parse
Initial Pixel Parse				61.8 (15.5)
NMS Detector	146101	0.8 / 12.2	27.8 / 25.1	60.9 (15.1)
NMS SVM	12839	9.3 / 12.9	53.3 / 52.8	61.8 (15.9)
Greedy	4909	**24.5** / 13.1	**60.9** / 59.8	**62.1 (16.2)**
CPlex QP	5435	22.3 / **13.3**	**60.9 / 59.9**	**62.1 (16.2)**



LMO Results

LMO dataset: 2,488 training images, 200 test images, 33 class labels.

	Instances	Object P/R	Pixel P/R	Pixel Parse
Initial Pixel Parse				**78.6 (39.3)**
NMS Detector	13734	3.1 / 21.4	58.0 / 50.5	77.8 (39.1)
NMS SVM	3236	11.8 / 18.4	75.7 / 62.0	78.1 (38.8)
Greedy	918	**44.3** / 20.0	**75.4 / 71.8**	78.4 (38.5)
CPlex QP	993	42.8 / **21.0**	**75.4 / 71.8**	78.4 (38.6)