Occlusion-Aware Object Localization, Segmentation and Pose Estimation

Samarth Brahmbhatt
http://www.cc.gatech.edu/~sbrahmbh
Heni Ben Amor
http://henibenamor.weebly.com
Henrik Christensen
http://www.cc.gatech.edu/~hic

School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA USA

Figure 1: Object localization and segmentation example. Left: Image, Right: Refined mask from SD-HOP

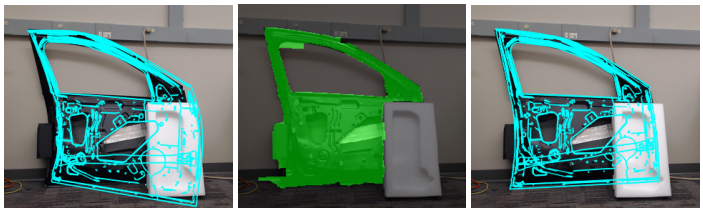

Figure 2: 3D pose estimation. Left to right: Pose estimation with IRLS, SD-HOP refined segmentation, Pose estimation with OR-IRLS.

We present a learning approach for localization and segmentation of objects in an image in a manner that is robust to partial occlusion. Our algorithm, *Segmentation and Detection using Higher-Order Potentials* (SD-HOP) produces a bounding box around the full extent of the object and labels pixels in its interior that belong to the object. This is different from semantic segmentation, which does not provide information about the spatial position of labelled pixels inside the object.

A common theme in the literature is to model occlusion geometrically or appearance-wise, thereby allowing it to contribute to the detection process. The former often make simplifying assumptions about occluder and scene geometry. Our appearance-based approach avoids these assumptions and performs better than existing appearance-based approaches due to the use of higher-order potentials for modelling neighbour influence and a loss function that targets both localization and segmentation.

SD-HOP discriminatively learns HOG templates for objects and occlusion. Whereas the object templates model the objects of interest, the occlusion templates provide discriminative support and do not model a specific occluder. Segmentation is done by considering the response of patches to these templates, and influence of neighbouring patches through a CRF with higher-order connections. The training phase requires a set of images with different occlusions of the object(s) of interest. Each training sample is (1) over-segmented and (2) annotated with a bounding box around the full extent of the object and a binary segmentation of the area inside the box into object vs. non-object pixels. Given these, we train a structured Support Vector Machine (SVM) that learns the HOG templates and CRF weights. Object segmentation is done by assigning binary labels to HOG cells within the bounding box, 1 for visible and 0 for occluded. Neighbour influence for segmentation can take two forms: (1) pairwise terms that impose a cost for 4-connected neighbours to have different labels and (2) higher-order potentials that impose a cost for cells to have a different label than the dominant label in their segment of the image. These segments are produced separately by an unsupervised segmentation algorithm.

The label for an object in an image $\mathbf{x}$ is represented as $\mathbf{y} = (\mathbf{p}, \mathbf{v}, a)$, where $\mathbf{p}$ is the bounding box, $\mathbf{v}$ is a vector of binary variables indicating the visibility of HOG cells within $\mathbf{p}$ and $a \in [1, A]$ indexes the discrete viewpoint. $\mathbf{p} = (p_x, p_y, p_\sigma)$ indicates the position of the top left corner and the level in a scale-space pyramid. The width and height of the box are fixed per viewpoint as w_a and h_a HOG cells respectively. Hence $\mathbf{v}$ has $w_a \cdot h_a$ elements. Given a labelled image, a sparse joint feature vector $\Psi(\mathbf{x}, \mathbf{y})$ is formed by stacking A vectors, each corresponding to a different discretized viewpoint. These vectors consist of vectorized HOG features and visibility labels of cells, count of cells in $\mathbf{p}$ that lie outside the image boundary, statistics of visibility agreement between 4-connected neighbouring cells and cells in the same unsupervised segment, and a constant bias. All vectors except for the one corresponding to viewpoint a are zeroed out.

Learning involves determining linear weights $\mathbf{w}$ such that the score $\mathbf{w}^T\Psi(\mathbf{x}_i, \mathbf{y}_i)$ of any ground truth labelled image $\mathbf{x}_i$ must be smaller than the score $\mathbf{w}^T\Psi(\mathbf{x}_i, \hat{\mathbf{y}}_i)$ of any other labelling $\hat{\mathbf{y}}_i$ by the distance between the two labellings $\Delta(\mathbf{y}_i, \hat{\mathbf{y}}_i)$ minus the slack variable ξ_i, where $\|\mathbf{w}\|_2$ and ξ_i are minimized. Hence we learn $\mathbf{w}$ by solving the following constrained Quadratic Program

$$\min_{\mathbf{w},\xi} \frac{1}{2}\|\mathbf{w}\|_2 + C\sum_{i=1}^{N} \xi_i \tag{1}$$

$$\begin{aligned} \text{s.t.} \quad & \mathbf{w}^T(\Psi(\mathbf{x}_i, \hat{\mathbf{y}}_i) - \Psi(\mathbf{x}_i, \mathbf{y}_i)) + \xi_i \geq \Delta(\mathbf{y}_i, \hat{\mathbf{y}}_i) \ \forall i, \hat{\mathbf{y}} \in Y_i \\ & \xi_i \geq 0 \ \forall i \\ & \mathbf{D}^2\mathbf{w} \geq \mathbf{0} \end{aligned}$$

$\mathbf{D}^2$ is a second order curvature constraint on the K + 1 weights for the higher-order potentials, which forces them to make a concave lower envelope. Training is performed by using the cutting plane training algorithm of [3], with adaptation for training higher-order potentials as described in [2]. The loss function between two labels $\mathbf{y}$ and $\hat{\mathbf{y}}$ depends on the amount of overlap between the two bounding boxes and the Hamming distance between the visibility labellings

$$\Delta(\mathbf{y}, \hat{\mathbf{y}}) = \left(1 - \frac{\text{area}(\mathbf{p} \bigcap \hat{\mathbf{p}})}{\text{area}(\mathbf{p} \bigcup \hat{\mathbf{p}})}\right) + \frac{\text{area}(\mathbf{p} \bigcap \hat{\mathbf{p}})}{\text{area}(\mathbf{p} \bigcup \hat{\mathbf{p}})} \cdot H(\mathbf{v}, \hat{\mathbf{v}}) \tag{2}$$

Inference is performed by finding the labelling that minimizes the dot-product energy: $\mathbf{y}^* = \text{argmin}_{\mathbf{y}}\, \mathbf{w}^T\Psi(\mathbf{x}, \mathbf{y})$. Due to the linear parametrization of energy and decomposability of the loss function over the unary terms, inference is efficient. At every bounding box location in a pyramid, it is performed by a single $s-t$ mincut on a graph constructed as described in [1] and [2].

We implemented SD-HOP in Matlab, with MVC search and inference implemented in CUDA since they are massively parallel problems. Inference on a 640x480 image with 11 scales takes 3s for a single object with a single viewpoint on our 3.4 GHz CPU and NVIDIA GT-730 GPU. SD-HOP achieves 13.52% segmentation error and 0.81 area under the false-positive per image vs. recall curve on average over the challenging CMU Kitchen Occlusion Dataset. This is a 42.44% decrease in segmentation error and a 16.13% increase in localization performance compared to the state-of-the-art. Figure 1 shows a sample output on this dataset.

We demonstrate that the segmentation output of SD-HOP can be used to ignore edges produced by occlusion, thereby making model-based 3D pose estimation robust to partial occlusion as shown in Figure 2

[1] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 2004.

[2] Stephen Gould. Max-margin learning for lower linear envelope potentials in binary markov random fields. In *ICML*, 2011.

[3] Thorsten Joachims, Thomas Finley, and Chun-Nam John Yu. Cutting-plane training of structural SVMs. *Machine Learning*, 2009.