Surface Based Object Detection in RGBD Images

Siddhartha Chandra*[1]
siddhartha.chandra@inria.fr
Grigorios G. Chrysos*[2]
grigoris.chrysos@gmail.com
Iasonas Kokkinos[1]
iasonas.kokkinos@inria.fr

[1] INRIA GALEN
Centrale Supélec
Paris, France
[2] Imperial College
London, U.K.

*These authors contributed equally to this work. G. Chrysos was at Centrale Supélec while conducting this research.

Introduction Intra-class variation in object detection is due to both intrinsic (texture and shape), and extrinsic (viewpoint and illumination) factors. RGBD data simplifies the problem of detection by introducing depth maps, which are invariant to texture and illumination. However, viewpoint variation still remains the Achilles heel for most object-detection frameworks. In this work we describe strategies to improve a HOG based object detection pipeline by introducing viewpoint based mixture components. We learn accurate mixtures of object detectors for RGBD data using the latent SVM framework. Our contributions are threefold.

- To better exploit depth information, we develop a novel depth-based dense feature extraction method that provides a robust statistical description of scene geometry.
- We use publicly available 3D mesh models to learn strongly supervised viewpoint classifiers. These are used to guide the first stages of model learning, and help avoid inaccurate local minima of latent SVM training.
- We develop a geometric dataset augmentation scheme that uses scene geometry to 'take another look' at the training data, simulating the effect of camera viewpoint changes.

We evaluate our learned detectors on the NYU dataset, and demonstrate that each of our advances results in systematic performance improvements over the traditional HOG-based detection pipeline.

Viewpoint Mixture Model Our object detection framework consists of a mixture model where each mixture component corresponds to a viewpoint. Formally, we denote the parameters of our mixture model by $\mathbf{w} = \{\mathbf{w}_1, \cdots, \mathbf{w}_C\}$, where C is the total number of mixture components. Given a bounding box $\mathbf{x}$ of an RGBD image, as well as a component ID $c \in \{1, \cdots, C\}$, the model provides a score, which is equal to the dot product of its parameters $\mathbf{w}_c$ and the feature vector of the bounding box $\Psi(\mathbf{x})$. The feature vector $\Psi(\mathbf{x})$ consists of two types of features: (i) the HOG (histogram of gradient) features, denoted by $\phi_g(\mathbf{x})$; (ii) the depth features denoted by $\phi_d(\mathbf{x})$, which are a combination of HOD (histograms of depth) features and our novel displacement features. In other words, the score $s_{\mathbf{w}}(\mathbf{x}, c)$ for a bounding box $\mathbf{x}$, and component ID c is given by

$$s_{\mathbf{w}}(\mathbf{x}, c) = \mathbf{w}_c^\top \Psi(\mathbf{x}),$$
$$\Psi(\mathbf{x}) = \left[\phi_g(\mathbf{x}); \phi_d(\mathbf{x})\right] \quad (1)$$

Displacement Features Displacement features are local, depth-based descriptors that are computed over dense grid. Given a region/bounding box $\mathbf{x}$ in a depth image, and a cellsize s, we first subsample $\mathbf{x}$ using average pooling. Each $s \times s$ non-overlapping cell in $\mathbf{x}$ is replaced by the average depth of the pixels in the cell. In the subsampled image, we compute the displacement of each pixel p_i from the center pixel p_0. This displacement is given by, $\delta_i = depth^{p_0} - depth^{p_i}$. We quantize δ_i into a set of N displacement bins, using a hard quantization function: $q(\delta) : \mathbb{R} \rightarrow \mathbb{R}^N$. Thus, each cell in $\mathbf{x}$ is represented by a sparse indicator vector of size N. We concatenate these sparse vectors to get the displacement feature of $\mathbf{x}$. The displacement features capture the depth variations in a region with respect to the center, and give substantial gains in performance when used alongside HOG and HOD features.

Learning A Viewpoint Mixture Model Given a training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i), i = 1, \cdots, N\}$, we wish to estimate the parameters of our mixture model such that it provides accurate detections for previously unseen test images. Here $\mathbf{x}_i$ is a training sample (specifically, a bounding box of an RGBD image) and $y_i \in \{+1, -1\}$ indicates whether the sample $\mathbf{x}_i$ belongs to the object class or the background class. We denote the indices of all object samples by $\mathcal{O}$ and the indices of all background samples as $\mathcal{B}$. In the absence of explicit ground truth viewpoint annotations, we treat these viewpoints as latent variables. We employ the following latent support vector machine (latent SVM) formulation:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}||\mathbf{w}||^2 + C_O \sum_{i \in \mathcal{O}} \xi_i + C_B \sum_{j \in \mathcal{B}} \xi_j, \\ \text{s.t.} \quad & \max_c \mathbf{w}_c^\top \Psi(\mathbf{x}_i) \geq 1 - \xi_i, \xi_i \geq 0, \forall i \in \mathcal{O}, \\ & \max_c \mathbf{w}_c^\top \Psi(\mathbf{x}_j) \leq -1 + \xi_j, \xi_j \geq 0, \forall j \in \mathcal{B}. \end{aligned} \quad (2)$$

The above optimization problem is a difference-of-convex program whose local minimum can be obtained using the CCCP algorithm. However the CCCP algorithm is sensitive to its initialization and the initial values of the component IDs can greatly affect its performance in practice. To overcome this difficulty, we exploit publicly available 3D models to better initialize the viewpoints.

Component Initialization with 3D Models The aspect ratio of an object is a reasonable- yet not an invincible- cue to what its viewpoint is. It can help distinguish the front and side views of a car, however, it cannot tell the front and rear views apart. In this work, we use a combination of aspect ratio cues, and viewpoint classifiers learnt from publicly available 3D models to estimate viewpoints of objects. We develop a two step strategy to initialize the component IDs for the object samples $\mathcal{O}$. First, we use 3D models of object categories from the Google Warehouse to render synthetic images with known viewpoints. The synthetic samples are used to learn a viewpoint classifier. Next the cues obtained by the viewpoint classifier are combined with the aspect ratio information, in an energy minimization framework, to obtain the component IDs of the object samples.

Dataset Augmentation with Geometric Jittering Camera viewpoint variation is a major challenge in object recognition; camera rotations affect object appearance radically and mixtures of viewpoint-tuned classifiers are imperative for multi-view detection. Here we exploit DIBR (Depth-Image-Based-Rendering) methods to simulate the effects of small camera rotations around the object, and obtain new samples that see the object from novel viewpoints. We use these synthesized images from novel views to augment our training set and enhance its variability.

Experimental Results

Method	Bed	Chair	M.+TV	Sofa	Table	Avg.
rgbd-hog	0.4660	0.2773	0.2480	0.2295	0.1430	0.2628
rgbd-hog + disp	0.5178	0.2771	0.2591	0.3440	0.1683	0.3133
rgbd-hog + disp + aug	0.5406	0.2919	0.2583	0.3470	0.1653	0.3206
rgbd-hog + disp + aug + vp	**0.5675**	**0.3023**	**0.3093**	**0.3719**	**0.1957**	**0.3493**
improvement over rgbd-hog	0.1015	0.0250	0.0613	0.1424	0.0527	0.0766

The table reports Average Precision values for the object detection task on 5 categories in the NYU dataset. The baseline method uses rgbd-hog (HOG+ HOD) features. We get systematic improvements in detection performance as we introduce displacement features (+disp), augmented data (+aug), and viewpoint initialization (+vp) to the baseline detector. Compared to the *rgbd-hog* baseline, we show an average improvement of 7.7%. While the displacement features allow us to better capture the depth variations of the surface, the augmentation and viewpoint initialization allow us to better model viewpoint effects.

Future Work The number of viewpoints dictates the number of components in our mixture model, and thus controls the model complexity. In the future we would like to develop models where we can express the viewpoint/orientation of an object in continuous space, while at the same time keeping a check on the model complexity. We would also like to extend our objective to solve the tasks of object detection and pose estimation simultaneously.