HOUGH-BASED OBJECT DETECTION WITH GROUPED FEATURES

Abhilash Srikantha[1,2] *Juergen Gall*[1]

[1]University of Bonn [2]Max Planck Institute for Intelligent Systems, Tuebingen

ABSTRACT

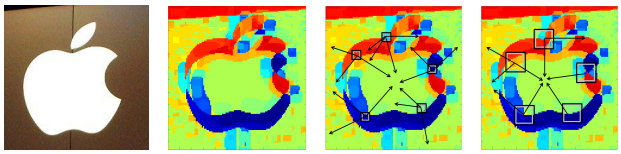
Hough-based voting approaches have been successfully applied to object detection. While these methods can be efficiently implemented by random forests, they estimate the probability for an object hypothesis independently for each feature. In this work, we address this problem by grouping features in a local neighborhood to obtain a better estimate of the probability. To this end, we propose oblique classification-regression forests that combine features of different trees. We further investigate the benefit of combining independent and grouped features and evaluate the approach on RGB and RGB-D datasets.

Index Terms— feature grouping, random forest

1. INTRODUCTION

Hough-based voting approaches or implicit shape models [1, 2, 3] model an object by a codebook of features and their spatial offsets to the center or root of the object. For object detection, features of the test image are matched to the codebook where each codebook entry models a distribution over the space of object hypotheses. This might not only encode the class and bounding box of the object, but could also include additional information like depth [4].

There is a significant body of work on voting approaches that rely on contour features. The approach in [5] groups contour features and dynamically finds their optimal transformations to associate with training data during testing. In [6], a codebook of recurring contour-pairs is learned from positive examples and an active appearance model for the object boundary is used for detection. During testing, Hough voting is performed by contour-pairs to identify hypotheses. However, the lack of a strong global model requires an additional verification stage [7]. Methods based on individual contours are also proposed in [8, 9]. While the former ranks them against a predefined model to evaluate its voting confidence, the latter jointly optimizes their location. Similar approaches are proposed in [10, 11] where discriminative models that optimize joint contour locations are learnt. Further, [12] models objects jointly on contour and appearance information of superpixels. While these methods are shown to work well on datasets with relatively few examples and where contour information is relatively easy to extract, they do not generalize to datasets where object contours are less reliable.



(a) (b) (c) (d)

Fig. 1. Illustration of Hough-based voting with independent features. Each pixel of an image (a) is assigned to a codeword. Each color in (b) corresponds to a codeword. For independent features, each pixel votes according to its assigned codebook entry (c). For grouped features, the voting depends also on the assigned codebook entries in its neighbourhood (d).

Consequently, there has been a shift towards employing more general image features in the recent past. Works in this context include [13, 14] that introduce a max-margin framework, to learn voting discriminatively. In addition, Codebook learning has been addressed in [15, 16] where random forests [17, 18] that simultaneously solve classification and regression problems have been proposed. Other improvements include self-similarity as a feature [19], global loss functions for training [20] and ordinal statistics for splitting criteria [21]. Further, several models per object class are learned in [22] to gain performance. These approaches, however, accumulate votes independently during testing in order to preserve computational efficiency.

In this work, we address the assumption of feature independence. We propose to model the object hypothesis on features grouped in a local neighborhood as illustrated in Fig. 1. Our approach is inspired by [23] that employs semantic texton forests to solve classification problems like segmentation and image-categorization tasks very efficiently. In our experiments, we show that the split functions used in segmentation forests [23] are too weak for the task of object detection. We therefore propose oblique classification-regression forests that use a linear combination of feature histograms instead of a binary feature selection as splitting functions. We further investigate the benefit of combining independent and grouped features and evaluate the approach on four RGB and RGB-D datasets.

2. HOUGH-BASED OBJECT DETECTION

Hough-based voting approaches represent an object by features like image patches or contour fragments that appear at certain locations with respect to its center. In other words, the probability for an object hypothesis $\mathbf{h}$, which encodes the label, position, scale and aspect ratio of an object, is written as:

$$p(\mathbf{h}|I) \approx \sum_{\mathbf{y}\in\Omega} p(\mathbf{h}|I(\mathcal{N}(\mathbf{y}))) \tag{1}$$

where $I(\mathcal{N}(\mathbf{y}))$ represents a collection of features from image I extracted within a neighborhood of location $\mathbf{y}$. In this regard, we present previous work in Section 2.1 where features are treated independently by setting $\mathcal{N}(\mathbf{y}) = \mathbf{y}$. Treatment of grouped features is then introduced in Section 2.2.

2.1. Independent Features

In this work, we use a Hough forest [15] that combines a classification and regression objective for training [17].

Training data. Each tree T of a forest is trained on a set of uniformly sized patches $\{P_i = (I_i, c_i, \mathbf{d}_i)\}$ randomly sampled from positive and negative examples. A patch is represented by low level features such as histogram of gradients and color, denoted by I_i. If extracted from a positive example, $\mathbf{d}_i$ denotes the offset of the patch to its center. Otherwise, the vector is left unused.

Splitting function. Splitting functions $f_\phi(I_i) \rightarrow \{0, 1\}$ at each non-leaf node split incoming training data and forward it to the left and right child:

$$f_\phi = \begin{cases} 0 & \text{if} \quad I^l(\mathbf{p}) - I^l(\mathbf{q}) < \tau \,, \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

where $I^l(\mathbf{p})$ and $I^l(\mathbf{q})$ are the values of low-level feature l at pixel locations $\mathbf{p}$ and $\mathbf{q}$ of patch P_i. The family of splitting functions is therefore defined by $\phi = (l, \mathbf{p}, \mathbf{q}, \tau)$.

Training. Starting from the root node, each node is optmized by selecting the best of a randomly generated pool of splitting functions. Given a set of training patches $\mathcal{P}$ arriving at a node, the goodness of a split $\mathcal{P}_0$ and $\mathcal{P}_1$ is measured by

$$\Delta G_o = H_o(\mathcal{P}) - \sum_{l\in\{0,1\}} \frac{|\mathcal{P}_l|}{|\mathcal{P}|} H_o(\mathcal{P}_l) \tag{3}$$

where classification or regression objective is chosen randomly. Training continues until the tree reaches a maximal depth or the size of a resulting split is below a threshold. Each leaf L_T of tree T then holds the class probability $p(c|L_T)$ and the distribution of the offset vectors $p(\mathbf{d}|c, L_T)$ for the positive class.

Testing. For object detection, the peaks of $p(\mathbf{h}|I)$ are detected for various scales s and aspect ratios a. The probability of a hypothesis for class c and location $\mathbf{x}$ is then given by

$$p(\mathbf{h}(c, \mathbf{x}, s, a)|I(\mathbf{y})) = \frac{1}{|\mathcal{T}|} \sum_{T\in\mathcal{T}} p(\mathbf{h}(c, \mathbf{x}, s, a)|L_T(\mathbf{y})),$$

$$p(\mathbf{h}|L_T(\mathbf{y})) = p(\mathbf{d}(\mathbf{y}, \mathbf{x}, s, a)|c, L_T(\mathbf{y})) \cdot p(c|L_T(\mathbf{y})) \tag{4}$$

where $L_T(\mathbf{y})$ is the leaf corresponding to the patch at location $\mathbf{y}$ and $\mathbf{d}(\mathbf{y}, \mathbf{x}, s, a)$ is the scale and aspect ratio normalized offset vector between $\mathbf{y}$ and $\mathbf{x}$.

2.2. Grouped Features

In order to model $p(\mathbf{h}|I(\mathcal{N}(\mathbf{y})))$, we learn the probability of a hypothesis based on all leaf assignments within a neighborhood of $\mathbf{y}$. To this end, we employ a second forest that is trained on the patch-to-leaf assignments obtained from the original forest, defined as follows.

Training data. Instead of training a forest on patches of low-level image features, the forests are now trained on histograms of leaves, HOL. While the class label c_i and the offset vector $\mathbf{d}_i$ of a group of features $G_i = (\text{HOL}_i, c_i, \mathbf{d}_i)$ are the same as before, histograms of leaves consist of a histogram for each tree HOL_T where the entries are given by $L_T(\mathcal{N}(\mathbf{y}))$, *i.e* by the leaf assignments within a rectangular region around image location $\mathbf{y}$.

Splitting function. The splitting function is given as

$$f_\phi = \begin{cases} 0 & \text{if} \quad \sum_{T\in\mathcal{T}} w_T \cdot \text{HOL}_T(L_T) < \tau \,, \\ 1 & \text{otherwise} \end{cases} \tag{5}$$

with $\phi = (\{w_T\}_{T\in\mathcal{T}}, \{L_T\}_{T\in\mathcal{T}}, \tau)$. While T selects the histogram HOL_T, L_T is the index of one bin in the histogram. We investigate two families of splitting functions: Axis aligned forests utilizing information from a single tree [23] by forcing $w_T \in \{0, 1\}$ and $\sum_T w_T = 1$. We will show that this family of splitting functions is not powerful enough for our task. Instead, we propose to set $w_T \in \mathbb{R}$ where features from different trees are combined in oblique forests [18].

Training and Testing. The training is performed as in Section 2.1. For testing, the first forest is applied to all scales and aspect ratios in order to assign each image patch to leaves $\{L_T\}_{T\in\mathcal{T}}$. The probability of an object hypothesis $\mathbf{h}$ is then given by the two forests:

$$p(\mathbf{h}(c, \mathbf{x}, s, a)|I(\mathcal{N}(\mathbf{y}))) = p(\mathbf{h}(c, \mathbf{x}, s, a)|\{L_T(\mathcal{N}(\mathbf{y}))\}_{T\in\mathcal{T}})$$
$$= \frac{1}{|\mathcal{T}_{gr}|} \sum_{T_{gr}\in\mathcal{T}_{gr}} p(\mathbf{h}|L_{T_{gr}}(\{L_T(\mathcal{N}(\mathbf{y}))\}_{T\in\mathcal{T}})) \tag{6}$$

where $\mathcal{T}$ is the first forest with independent features and $\mathcal{T}_{gr}$ is the second forest with grouped features. The approach for a single tree is illustrated in Fig. 1. Lastly, we investigate a version that combines independent (4) and grouped (6) features by defining the joint hypothesis as

$$p(\mathbf{h}|I(\mathcal{N}(\mathbf{y})), \lambda) \propto p(\mathbf{h}|I(\mathcal{N}(\mathbf{y})))^\lambda \cdot p(\mathbf{h}|I(\mathbf{y}))^{1-\lambda} \tag{7}$$

where $\lambda \in [0, 1]$ steers the impact of the two probabilities.

Table 1. ETHZ dataset: Recall at 0.3/0.4 fppi (%) for spatial support $\mathcal{N}$ and depth of the first forest to generate HOL

support:	7×7			13×13		
depth:	5	10	16	5	10	16
Applelogos	55.0/60.0	**90.0/90.0**	75.0/75.0	15.0/15.0	75.0/75.0	10.0/10.0
Bottles	**92.8/92.8**	89.2/89.2	75.0/75.0	57.1/57.1	71.4/71.4	32.2/32.2
Giraffes	72.3/74.5	78.7/80.8	83.0/83.0	61.7/61.7	**83.0/85.1**	74.5/74.5
Mugs	61.3/61.3	**67.7/74.2**	61.3/61.3	45.2/45.2	61.3/61.3	51.6/51.6
Swans	70.6/76.5	70.6/70.6	58.8/58.8	58.8/58.8	**82.3/88.2**	41.2/58.8

3. EXPERIMENTS

We evaluate our method on four different datasets of increasing difficulty and compare its performance with previously published results. In each case, we adopt evaluation protocols of previous works to facilitate comparison.

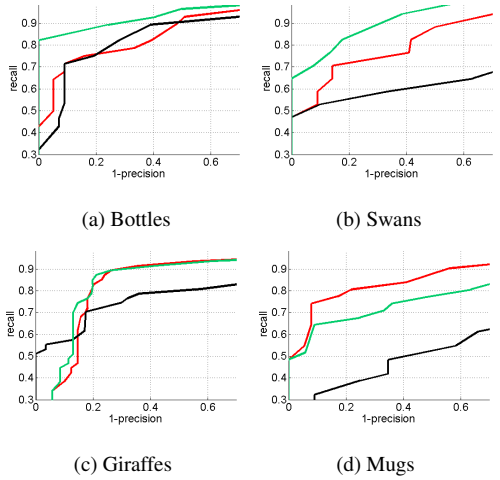
ETHZ Dataset. The dataset consists of 255 images classified into five categories: *Applelogos, Bottles, Giraffes, Mugs* and *Swans*. Class *Giraffes* is the largest, consisting of 87 examples and *Swans* the smallest, with 32 examples. We use the protocol as in [13] where the training set consists of half the images of a class and an equal number of negative images sampled uniformly from all other classes. The baseline with independent features is trained using 5 trees with maximal depth of 25 and a minimum leaf size of 20 samples. The features consist of color and histogrammed gradients. The patches have a fixed size of 16×16 pixels and each tree is trained on 20k positive and negative patches each. The negative patch pool is built using equal contributions from negative examples and the backgound of positive examples. Testing is performed by spanning a space of three aspect ratios and five scales. On Average, training a tree took 312s (Intel i7, 3.4GHz) and 1GB RAM and testing took 3.5s per image.

Object detection with grouped features depends on the spatial support of neighborhood $\mathcal{N}$ and on the depth of the first forest generating HOL. These parameters are fixed by grid search on tree depth $\in \{5, 10, 16\}$ and spatial support $\in \{7 \times 7, 13 \times 13\}$. The resulting performance for each case is shown in Table 1. Training a tree took 187s and 480MB RAM on average and testing took 13.8s took per image.

It can be noted that the tree depth is an important parameter. Intuitively, deep trees result in highly specific features thereby overfitting training data. On the other hand, shallow trees result in generic features that are less effective in describing objects. Since the configuration pair $\{10, 7 \times 7\}$ performs reasonably well for all classes, we use it for all experiments henceforth.

Table 2. EHTZ Dataset: Performances of independent features (4), grouped features (6) and their combination (7)

	Average Precision			optimal λ	Recall at 0.3/0.4 fppi		
	(4)	(6)	(7)		(4)	(6)	(7)
Applelogos	77.8	77.4	**85.4**	0.9	80.0/80.0	90.0/90.0	**90.0/90.0**
Bottles	85.9	84.3	**93.8**	0.7	92.9/96.4	89.2/89.2	**96.4/96.4**
Giraffes	82.6	76.9	**83.4**	0.1	**91.5/93.6**	78.7/80.8	91.5/91.5
Mugs	**84.9**	62.6	84.1	0.1	**90.3/90.3**	67.7/74.2	87.1/90.1
Swans	83.2	63.3	**90.2**	0.6	100/100	70.6/70.6	**100/100**



(a) Bottles (b) Swans

(c) Giraffes (d) Mugs

Fig. 2. ETHZ Dataset: Precision-recall curves for independent features (red), group features (black) and the best combination (green).

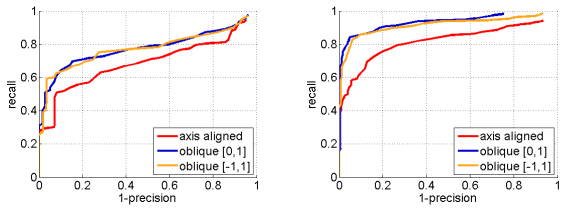
To investigate the effect of combination according to (7), we vary the parameter λ in $[0, 1]$. We record the best performance in Table 2 and visualize the same in Fig. 2. Two important observations can be made drawn: Firstly, there is a higher performance gain for rigid object classes against those with high intraclass variations, namely articulations in *Giraffes* and handle variations in *Mugs*. And secondly, the choice of optimal λ is class dependent and should therefore be set using a validation dataset. Both conclusions point towards the need for more data.

Further, comparing with state-of-the-art at (0.3/0.4) fppi averaged over all five classes, [12, 11] perform at (94.0/95.8) and (95.2/95.6) respectively against the best possible performance of (93.6/93.6) from our setup. With average precision, [10, 9] perform at 88.2% and 87.7% respectively against our best possible performance of 87.5%, thereby indicating that the performance is comparable to the state-of-the-art.

Note that in contrast to most methods, our approach neither performs an additional verification step nor relies on specific contour features typical for this shape dataset.

INRIA Horse Dataset. The dataset consists of 170 positive and 170 negative examples; of which the first 50 in each case are used for training. The forest of independent features is made up of 5 trees each trained with 40k positive and negative patches each. Further, hard-negative training is performed by mining the 50 hardest negative examples in the training data. This contributes to an additional gain of 8% recall at 0.3 fppi.

Here, three types of splitting functions are investigated on grouped features as discussed in Section 2.2. Firstly, axis aligned forests as used in [23]. Secondly, oblique forests that employ splitting functions with weights $w_T \in [0, 1]$ and lastly, oblique forests with weights $w_T \in [-1, 1]$. Perfor-



(a) INRIA Horse Dataset (b) Weizmann Dataset

Fig. 3. Precision-recall plots showcasing the performance of three types of splitting functions for grouped features. The oblique forests outperform axis aligned forests.

mance of each of the three variants is shown in Fig. 3(a). While the oblique forests outperform axis aligned forests, also allowing negative weights does not change the performance.

Table 3 compares state-of-the-art methods with the grouped features using an oblique forest combined with independent features with $\lambda = 0.1$, obtained by splitting the training data in half to generate a validation set.

Weizmann Horse Dataset. The dataset comprises 200 images for training and validation and 456 images for testing. For the independent features, we train a forest of 5 trees with 40k positive and negative patches each followed by hard-negative training. The retraining, however did not yield a significant gain in performance. As for the INRIA horse dataset, we evaluate three types of splitting functions shown in Figure 3(b). The oblique forests again outperform the axis-aligned forests. Table 3 compares state-of-the-art methods with the grouped features using an oblique forest and combined with independent features at $\lambda = 0.1$, fixed as in INRIA Horse Dataset.

Berkeley 3-D Object Dataset. The dataset is a collection of real-world images captured with a Kinect sensor consisting of RGB-D image pairs, gathered for over 50 classes. The dataset comes bundled with post-processed data that interpolates missing depth information. The dataset also comes with a six-fold split for 8 of these classes, resulting in 48 splits, over which baseline performances using a part-based model [26] for various RGB, D and RGBD features are presented in [25]. Interestingly, the authors report a significant drop in performance upon including depth information.

The forests of the independent features are ensembles of 10 trees, each trained with 100,000 positive patches and an equal number of negative patches. Hard negative training was not incorporated.

We first investigate the utility of depth by comparing forests built on independent RGB-only features ignoring depth information, and independent RGBD features where the depth of a pixel is used as an additional feature. The results are tabulated in Table 4. Unlike [25], we see a significant improvement of RGBD over RGB features.

For the grouped features, we use an oblique forest with $w_T \in [0, 1]$. Each forest consists of 10 trees and is trained with the same protocol as the forest for independent features. The performance of grouped features is at most equal to that of independent features with an exception for *bottles* where a gain of 0.5% in average precision is seen.

We also combined both forests as in (7) by fixing the parameter λ for each split using a validation dataset that is obtained by splitting the training data in half. Table 4 presents performances of the best possible combination, value of λ set using validation and the resulting performance. It is to be noted that combining both forests mostly results in an improved performance indicating that though the grouped features alone do not outperform their individual counterparts, they contain complementary information.

The proposed approach with λ estimated on the validation set achieves an average precision of 0.314 and is comparable or better than the state-of-the-art methods [25] and [27], which achieve 0.280 AP and 0.312 AP, respectively. The approach [28] reports 0.592 AP, but it follows a different evaluation procedure by using custom annotation of the dataset.

4. CONCLUSION

We have presented an approach for grouping features for Hough-based object detection in RGB and RGB-D images. For evaluation, we have used four datasets of varying difficulty. While the approach performs comparable to the state-of-the-art, the experiments give some interesting insights. We have found that it is important that the features used for grouping are not too specific and that the proposed oblique forests outperform axis aligned forests for this task. The experiments also showed that a combination of independent and grouped features improves the performance, indicating that both feature sets encode complementary information.

Acknowledgements: Authors acknowledge financial support from the DFG Emmy Noether program (GA 1927/1-1).

Table 3. INRIA, Weizmann Dataset: Recall at 1.0 fppi

	measure	proposed	[10]	[12]	[5]	[13]	[15]	[24]
INRIA	recall	88.0	93.7	92.4	87.3	85.3	×	×
Weizmann	AP	97.2	×	×	×	×	98.0	96.0
Weizmann	recall	94.3	×	×	×	×	95.1	91.5

Table 4. VOCB3DO Dataset: Average precision

Class	RGB	RGBD	Group	bestComb.	λ	Combin.	[25]
bowl	0.231	0.402	0.394	0.423	0.5	0.420	**0.430**
cup	0.123	0.346	0.339	0.358	0.5	**0.357**	0.260
monitor	0.282	0.540	0.530	0.547	0.4	0.547	**0.750**
mouse	0.208	0.282	0.275	0.302	0.4	**0.301**	0.190
phone	0.076	0.163	0.129	0.172	0.3	0.163	**0.180**
keyboard	0.085	0.314	0.283	0.321	0.4	**0.321**	0.170
chair	0.028	**0.208**	0.161	0.211	0.4	0.206	0.140
bottle	0.022	0.178	0.183	0.201	0.2	**0.195**	0.120

5. REFERENCES

[1] B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 259–289, 2008.

[2] A. Opelt, A. Pinz, and A. Zisserman, "Learning an alphabet of shape and appearance for multi-class object detection," *International Journal of Computer Vision*, vol. 80, no. 1, pp. 16–44, 2008.

[3] J. Shotton, A. Blake, and R. Cipolla, "Multiscale categorical object recognition using contour fragments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 7, pp. 1270–1281, 2008.

[4] M. Sun, G. Bradski, B.-X. Xu, and S. Savarese, "Depth-encoded hough voting for joint object detection and shape recovery," in *European Conference on Computer Vision*, 2010, pp. 658–671.

[5] P. Yarlagadda, A. Monroy, and B. Ommer, "Voting by grouping dependent parts," in *European Conference on Computer Vision*, 2010, pp. 197–210.

[6] V. Ferrari, F. Jurie, and C. Schmid, "From images to shape models for object detection.," *International Journal of Computer Vision*, vol. 32, pp. 284–303, 2010.

[7] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration.," *Computer Vision and Image Understanding*, vol. 89, no. 2–3, pp. 114–141, 2003.

[8] H. Riemenschneider, M. Donoser, and H. Bischof, "Using partial edge contour matches for efficient object category localization.," in *European Conference on Computer Vision*, 2010, pp. 29–42.

[9] T. Ma and L. Latecki, "From partial shape matching through local deformation to robust global shape similarity for object detection.," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 1441–1448.

[10] P. Yarlagadda and B. Ommer, "From meaningful contours to discriminative object shape.," in *European Conference on Computer Vision*, 2012, pp. 776–779.

[11] P. Srinivasan, Q. Zhu, and J. Shi, "Many-to-one contour matching for describing and discriminating object shape.," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1673–1680.

[12] A. Toshev, B. Taskar, and K. Daniilidis, "Object detection via boundary structure segmentation.," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 950–957.

[13] S. Maji and J. Malik, "Object detection using a max-margin hough transform.," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1038–1045.

[14] Y. Zhang and T. Chen, "Implicit shape kernel for discriminative learning of the hough transform detector," in *British Machine Vision Conference*, 2010, pp. 105.1–105.11.

[15] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky, "Hough forests for object detection, tracking, and action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2188–2202, 2011.

[16] R. Okada, "Discriminative generalized hough transform for object dectection," in *International Conference on Computer Vision*, 2009, pp. 2000–2005.

[17] A. Criminisi and J. Shotton (Editors), *Decision Forests for Computer Vision and Medical Image Analysis*, Springer, 2013.

[18] Leo Breiman, "Random forests," in *Machine Learning*, 2001, pp. 5–32.

[19] N. Razavi, N.S. Alvar, J. Gall, and L. Van Gool, "Sparsity potentials for detecting objects with the hough transform," in *British Machine Vision Conference*, 2012, pp. 11.1–11.10.

[20] S. Schulter, P. Wohlhart, C. Leistner, A. Saffari, P.M. Roth, and H. Bischof, "Alternating decision forests," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[21] S. Schulter, P.M. Roth, and H. Bischof, "Ordinal random forests for object detection," in *German Conference for Pattern Recognition*, 2013.

[22] N. Razavi, J. Gall, P. Kohli, and L. Van Gool, "Latent hough transform for object detection," in *European Conference on Computer Vision*, 2012, pp. 312–325.

[23] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation.," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[24] J. Shotton, A. Blake, and R. Cipolla, "Efficiently combining contour and texture cues for object recognition.," in *British Machine Vision Conference*, 2008.

[25] A. Janoch, S. Karayev, Y. Jia, J.. Barron, M. Fritz, K. Saenko, and T. Darrell, "A category-level 3d object dataset: Putting the kinect to work," in *Consumer Depth Cameras for Computer Vision*, pp. 141–165. Springer, 2013.

[26] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1627–1645, 2010.

[27] B. Kim, S. Xu, and S. Savarese, "Accurate localization of 3d objects from rgb-d data using segmentation hypotheses.," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3182–3189.

[28] T. Wang, X. He, and N. Barnes, "Learning structured hough voting for joint object detection and occlusion reasoning.," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1790–1797.