3rd Workshop on Semantic Perception, Mapping and Exploration (SPME), Karlsruhe, in conjunction with ICRA 2013.

# Depth-Enhanced Hough Forests for Object-Class Detection and Continuous Pose Estimation

Ishrat Badami Jörg Stückler Sven Behnke

*Abstract*— **Much work on the detection and pose estimation of objects in the robotics context focused on object instances. We propose a novel approach that detects object classes and finds the pose of the detected objects in RGB-D images. Our method is based on Hough forests, a variant of random decision and regression trees that categorize pixels and vote for 3D object position and orientation. It makes efficient use of dense depth for scale-invariant detection and pose estimation. We propose an effective way to train our method for arbitrary scenes that are rendered from training data in a turn-table setup. We evaluate our approach on publicly available RGB-D object recognition benchmark datasets and demonstrate state-of-the-art performance in varying background and view poses, clutter, and occlusions.**

## I. INTRODUCTION

Unstructured environments pose severe challenges to the perception capabilities of autonomous robots. One major theme in mobile manipulation is the detection and localization of objects that shall be handled by the robot. Much previous work within this context has considered the detection and pose estimation of specific object instances. If a robot, however, needs to perceive objects within classes that contain a large amount of instances, shallow object instance detection pipelines do not scale well. Moreover, unknown instances cannot be detected. Efficient detection pipelines at the instance as well as category level can be obtained by grouping the instances in a class taxonomy [1], [2]. Detection can then be successively refined from coarse object categories up to the individual instance. Perceiving object classes, however, imposes challenges over object instances, since the detection must handle intra-class variation but still needs to distinguish view poses onto the objects.

In this paper, we propose a novel approach to object-class detection and pose estimation in RGB-D images. We utilize Hough forests, a learning architecture that combines discriminatively trained ensembles of random regression trees with the Hough transform. The random trees not only model the probability distribution over class labels an image pixel belongs to, but also cast votes for the pose of the object. On the category level, we detect a canonical pose for the instances of an object class, which can be further used for decision-making or to initialize tracking.

We extend Hough forests to efficiently operate on RGB-D images. Exploiting dense depth, we normalize image features in the decision cascade of the trees for scale changes. The use of depth also allows for incorporating the right scale directly into the position and orientation votes of the pixels, making scale as an additional voting parameter obsolete. Complementarily, dense estimates of the local surface orientation at each pixel allow for view-point invariant voting for the object pose. Finally, we utilize 3D information in training to render arbitrary amounts of novel training scenes that capture background variability, clutter, and occlusions in real imagery.

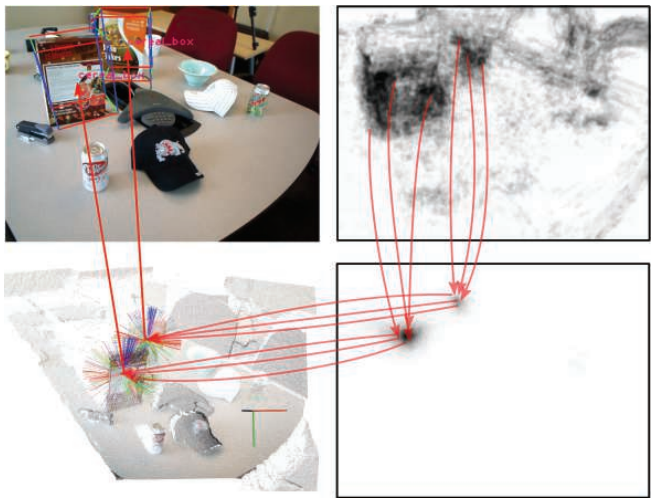


Fig. 1. Object-class detection and 6-DoF pose estimation in RGB-D images. We discriminatively train random decision forests to classify pixels into object classes (upper right) and to vote for the 3D position of objects (lower right). The pose of the detected object is estimated in a second stage of Hough voting for the position clusters (lower left).

We evaluate our approach on publicly available RGB-D objects and scenes datasets. We demonstrate that our approach outperforms Hough forests that operate on RGB images only and would vote for the 2D pixel location of the object. Furthermore, our method recovers the pose of the objects with good accuracy.

## II. BACKGROUND

### A. Instance-Level Object Detection and Pose Estimation

The development of local scale-invariant features has been a leap forward in the detection and pose estimation of object instances in intensity images. Frameworks such as MOPED [3] that are based on SIFT [4] or SURF [5] features work well for textured objects, but face difficulties if the objects are less textured. For 3D data, various local point features have been proposed that describe shape through the local constellation of points and surface normals. Prominent

All authors are with Autonomous Intelligent Systems, Computer Science Institute VI, University of Bonn, 53113 Bonn, Germany badami at cs.uni-bonn.de, stueckler at ais.uni-bonn.de, behnke at cs.uni-bonn.de

Source code: http://ibadami.github.io/RGBD-Hough-Forest

examples are spin images [6] and signatures of histograms of local surface orientation (SHOT) [7]. Variants of point feature histograms (PFH) [8], [9], [10] have been used to describe and detect whole objects or parts within segments. Recently, point-pair features (PPF) [11], [12] have been demonstrated as a robust approach to detecting objects in 3D measurements. PPF methods find locally consistent arrangements of surfel pairs between model and scene through either Hough voting [11] or RANSAC [12]. Our approach discriminatively trains a codebook of point-pair-feature votes to estimate the pose of object classes within a Hough forest framework.

### *B. Category-Level Object Detection and Pose Estimation*

In recent years the computer vision community developed powerful object-class detection methods.

The Implicit Shape Model (ISM) of Leibe et al. [13] combines the ideas of visual appearance codebooks and Hough transform. Each visual word is augmented with the spatial distribution of the displacements between the object center and the respective visual word location. At detection time, descriptors are matched with visual words that cast votes for the object center. Deformable Parts Models (DPM) [14] make the arrangement of parts explicit in a star-model. The appearance of parts is encoded using Histograms of Oriented Gradients (HOG) [15]. The authors propose a latent SVM formulation to discriminatively train the model. Hough forests learn a dense pixel-wise codebook in a discriminative way within a random forest framework. A forest consists of multiple random decision trees that decide on the local appearance of a pixel in binary decision cascades. At the leaves of the trees, it stores the spatial distribution of relative object locations towards the training pixels, which are used to cast location votes during recall. The learning objective separates classes and produces Hough votes that focus well.

Detecting objects and concurrently estimating their pose at the category level recently received much attention in the computer vision community. The approaches can be classified into methods along two dimensions that estimate discrete or continuous view poses and methods that utilize only RGB images or that exploit dense depth.

**Discrete vs. continuous pose estimation.** Many approaches learn detectors for the object class in discrete view poses [16], [2], [17]. Interpolation techniques need then to be applied to obtain continuous pose estimates [18], [19]. Sun et al. [20] apply depth registration to find an accurate pose in a post-processing step. Some approaches extract local image features either from 3D models obtained through Structure-from-Motion [21] or from views synthesized from CAD models [22], [23], and model the 3D constellation of the features. We propose a method that is discriminatively trained to cast continuous votes for the 3D location and orientation of objects. While our method is trained from discrete views, we exploit local shape properties to transform the orientation vote of a pixel into a reproducible view-pose-invariant local coordinate frame to cast continuous pose votes.

**Exploiting dense depth.** Only few approaches in the computer vision literature make additional use of dense depth. Sun et al. [20] propose Depth-Encoded Hough Voting (DEHV). They formulate a probabilistic model for joint object detection and shape recovery that utilizes dense depth, but is also applicable if no depth is available during recall. To estimate the pose of the object, a 3D model is registered to the recovered shape. Our approach directly votes for the pose of the object, utilizing local shape properties, and consistency of the votes is integrated as a discriminative training objective of our random forest. We also propose to use depth as a feature cue but scale-normalize the features using depth. In this way, the random forest is not required to capture multiple scales within its codebook.

Wang et al. [24] use depth to improve Hough forests during the training stage. In addition to 2D offset uncertainty, they also incorporate 3D offset dispersion as a split measure into the Hough forest framework. They incorporate votes from the spatial context of objects and use depth to store the relative scale of the votes with respect to the object size. Since depth is not used during recall, the votes have to be cast across multiple scales. This approach only votes for the object location and a bounding box, while ours retrieves the full 3D position and orientation of the object.

Wohlkinger et al. [25] propose 3DNet, a large-scale object-class recognition method based on CAD models of instances. They train classifiers on 3D descriptors extracted from synthetic views that are generated using the CAD models. Our approach does not require CAD models of the objects and makes use of combined texture and shape features that are discriminatively trained for each object class.

## III. OBJECT-DETECTION IN RGB-D IMAGES USING HOUGH FOREST

Hough forests are ensembles of random decision trees. Each tree maps training pixels in an image to one of its leaves through a cascade of binary decisions over local appearance. These leaves can be seen as a discriminative appearance codebook of visual words. Each leaf stores the distribution of class labels that reached it. Additionally, the leaves carry spatial information about the object, e.g., of the relative location of the object center. During recall, this information is used to classify test pixels into object classes and to cast votes in a Hough space parametrized in object location, scale or orientation.

### *A. Training*

**Training data**. In a Hough forest $\mathcal{F}$, each of the decision trees $\mathcal{T}$ is built using a set of sampled image pixels $S_0 = \{(\mathcal{I}(\mathbf{y}), c(\mathbf{y}), \mathbf{d}(\mathbf{y}), \mathbf{n_y})\}$, where $\mathcal{I} = \{I^1, I^2, \ldots, I^N\}$ is the appearance of the training image, $I^j$ is the $j^{th}$ appearance channel, $c(\mathbf{y}) \in \mathcal{C} : \{0, 1\}$ is a class label (0 for negative sample and 1 for positive sample), if $\mathbf{p}(\cdot)$ defines 3D point corresponding to 2D pixel then, $\mathbf{d}(\mathbf{y})$ is the relative 3D location of an object center to the sampled training point $\mathbf{p}(\mathbf{y})$ and $\mathbf{n_y}$ is a normal vector at $\mathbf{p}(\mathbf{y})$. Note that $\mathbf{d}(\mathbf{y})$

is undefined in case of image pixels not belonging to an object class.

**Tree construction.** During training, each node $n$ is ascribed a pixel-pair-based binary test $t : \mathbf{y} \rightarrow \{0, 1\}$ over an appearance channel of the image to separate the training samples $\mathbf{y} \in \mathcal{S}_n$ that reach the node. For appearance channel $I^j$ and offset vectors $\mathbf{u}_1$, $\mathbf{u}_2$, the test $t_{j,\mathbf{u}_1,\mathbf{u}_2,\tau}(\mathbf{y})$ is then defined as:

$$t_{j,\mathbf{u}_1,\mathbf{u}_2,\tau}(\mathbf{y}) = \begin{cases} 0, & \text{if } I^j(\mathbf{y}+\mathbf{u}_1) - I^j(\mathbf{y}+\mathbf{u}_2) < \tau \\ 1, & \text{otherwise.} \end{cases} \tag{1}$$

It compares the value of a pixel pair in one of the appearance channels with some threshold $\tau$.

At each node $n$, for all the positive class samples in the node $\mathcal{P}_n = \{\mathbf{y} \in \mathcal{S}_n | c(\mathbf{y}) = 1\}$, we also compute a transformation matrix from the local frame of the query pixel to an object frame. For each training sample $\mathbf{y}$ and point at the two offsets $\mathbf{y} + \mathbf{u}_k, k \in \{1, 2\}$, the local frame of the query pixel in the camera frame $C$, ${}^C\mathbf{T}^k_\mathbf{y} = [{}^C\mathbf{R}^k_\mathbf{y}, {}^C\mathbf{t}^k_\mathbf{y}] = [\mathbf{l}_k, \mathbf{m}_k, \mathbf{n}_k, {}^C\mathbf{t}^k_\mathbf{y}]$ is calculated as

$$\begin{aligned} \mathbf{l}_k &= \frac{\mathbf{n}_\mathbf{y} \times \mathbf{p}(\mathbf{u}_k)}{||\mathbf{n}_\mathbf{y} \times \mathbf{p}(\mathbf{u}_k)||_2}, \\ \mathbf{m}_k &= \mathbf{l}_k \times \mathbf{n}_\mathbf{y}, \\ \mathbf{n}_k &= \mathbf{n}_y, \\ {}^C\mathbf{t}^k_\mathbf{y} &= \mathbf{d}(\mathbf{y}). \end{aligned} \tag{2}$$

If we denote ${}^C\mathbf{T}_O$ as frame of an object in camera frame then the relative transformation from object frame to query pixel frame ${}^\mathbf{y}\mathbf{T}^k_O$ is.

$${}^\mathbf{y}\mathbf{T}^k_O = ({}^C\mathbf{T}^k_\mathbf{y})^{-1} \times {}^C\mathbf{T}_O, \tag{3}$$

Hough forests are trained in a supervised way. At each node, a pool of binary tests $\{t\}$ is generated by randomly sampling $I^j$, $\mathbf{u}_1$, $\mathbf{u}_2$, and $\tau$. The idea is to pick the test in a way such that uncertainty in the class label and object pose votes decreases. In order to minimize the class label uncertainty, a well known entropy measure is used, which is defined over a set of image pixels as:

$$M_1(n) = -\sum_{l=0}^{1} log\left(\frac{|\{\mathbf{y} \in \mathcal{S}_n | c(\mathbf{y}) = l\}|}{|\mathcal{S}_n|}\right). \tag{4}$$

During recall, displacement vector $\mathbf{d}(\mathbf{y})$ and rotation matrix ${}^\mathbf{y}\mathbf{R}^j_O$, saved at the reached leaf node, vote for object location and orientation, respectively. Hence it is important to minimize the uncertainty of votes coming from a single leaf node. For the displacement vector, we use the same dispersion (uncertainty) measure as in [26].

$$M_2(n) = \sum_{\mathbf{y} \in \mathcal{P}_n} \left\| \mathbf{d}(\mathbf{y}) - \frac{1}{|\mathcal{P}_n|} \sum_{\mathbf{y}' \in \mathcal{P}_n} \mathbf{d}(\mathbf{y}') \right\|_2. \tag{5}$$

While training, one of the above mentioned measures is chosen randomly at each node.

**Leaf information.** During training each leaf-node $l$ saves information about the training samples reached to that node.

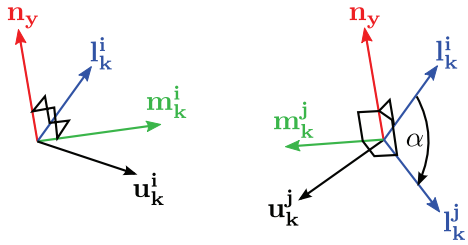



Fig. 2. We construct local reference frames from the local surface normal $\mathbf{n_y}$ at pixel $\mathbf{y}$ and the directions towards the offset pixels $\mathbf{u_k}$. All local frames differ only by an angle around the normal. This property allows for efficiently saving relative orientations towards the object rotation by angles for training examples.

In our case, relative class frequencies $\mathcal{C}_l$, displacement vectors $\mathcal{D}_l$ and orientations $\mathcal{R}_l$ are saved.

The orientation votes are not only cast for the two point-pairs $(\mathbf{y}, \mathbf{y} + \mathbf{u}_{1/2})$ of the split test at the leaf node, but from the point-pairs of all the tests along the path of the training samples from the root to the leaf. Since the local frames at the point pairs differ only by an angle around the normal $\mathbf{n_y}$ at $\mathbf{y}$ (Fig. 2), we can store these $2 \times d$ rotations memory-efficiently by one reference rotation ${}^{\mathbf{y},0}\mathbf{R}^k_O$ for the orientation at the root node and angular differences $\alpha^k_{\mathbf{y},n}$ around $\mathbf{n_y}$ for each other point-pair in the decision cascade, i.e.,

$$\begin{aligned} {}^{\mathbf{y},n}\mathbf{R}^k_O &= {}^{\mathbf{y},n}\mathbf{R}_{\mathbf{y},0} \times {}^{\mathbf{y},0}\mathbf{R}^k_O, \text{ where} \\ {}^{\mathbf{y},n}\mathbf{R}_{\mathbf{y},0} &= \mathbf{R}\left(\mathbf{n_y}, \alpha^k_{\mathbf{y},n}\right). \end{aligned} \tag{6}$$

### *B. 6-DoF Object Detection*

During recall, each image pixel $\hat{\mathbf{y}}$ traverses through all the trees $t \in \{1, \ldots, N\}$ and the class probability of the image pixel is computed by averaging the frequency of class labels at the reached leaf nodes $l_t(\hat{\mathbf{y}})$ that have been recorded during training, i.e.

$$p(c \mid \mathcal{F}, \hat{\mathbf{y}}) = \frac{1}{N} \sum_{t=1}^{N} p(c \mid l_t(\hat{\mathbf{y}})), \tag{7}$$

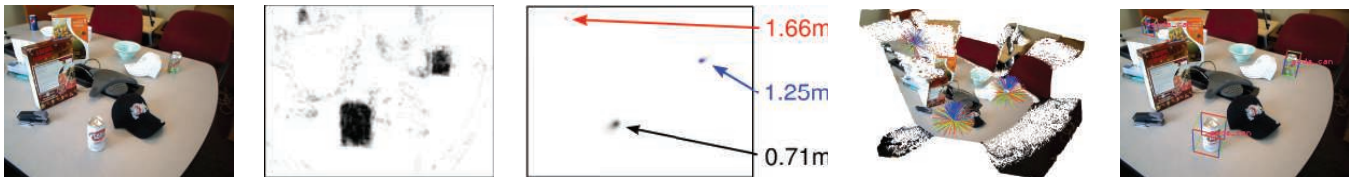
where $p(c \mid l_t(\hat{\mathbf{y}})) = C_{l_t}(c)$ is set to the class frequency in leaf $l_t$ of tree $t$.

Detection is done in two phases. In a first phase, each reached leaf node casts probabilistic votes for the object position in a 3D Hough space and maxima are sought. Training samples which contributed to position maxima then again vote for the orientation of the object in a 4D Hough space, parameterized in quaternions.

We discretize the 3D Hough space into image locations and scales. The latter is represented in inverse depth to model higher position accuracy at closer distances. For each relative object position stored in the leaf that has been reached by pixel $\hat{\mathbf{y}}$, we add the weight

$$w = C_l(c) \cdot \text{dist}(\hat{\mathbf{y}}) \tag{8}$$

to the corresponding bin in Hough space. The weighting is proportional to the relative class frequency and scaled by the




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

Fig. 3. During detection, for each test pixel in test image (a), class probability (b) is computed, pixels labeled as object class then vote for object location in 3D Hough space (c) and maxima are sought. Votes those contributed to found maxima then again vote for object orientation in the 4D Hough space corresponding to each maxima (d). Similarly maxima in orientation Hough space is searched. Once complete 6-DOF object pose is found, pre-computed bounding boxes are projected (e).

distance towards the pixel to account for the projected size of the object in the image.

The orientation votes are computed for all nodes that pixel $\hat{\mathbf{y}}$ has passed on its path from root to leaf in each tree. At the root node, the orientation votes ${}^{C}\mathbf{R}_{O}^{k} = {}^{C}\mathbf{R}_{\hat{\mathbf{y}},0}^{k} \times {}^{\mathbf{y},0}\mathbf{R}_{O}^{k}$ are computed from the local frames for the offsets $k \in \{1, 2\}$. All other nodes vote for the orientations

$$ {}^{C}\mathbf{R}_{O} \quad = \quad {}^{C}\mathbf{R}_{\hat{\mathbf{y}},n}^{k} \times \mathbf{R}\left(\mathbf{n}_{\mathbf{y}}, \alpha_{\mathbf{y},n}^{k}\right) \times {}^{\mathbf{y},0}\mathbf{R}_{O}^{k} \tag{9} $$

which are recovered from the relative rotations $\mathbf{R}\left(\mathbf{n}_{\mathbf{y}}, \alpha_{\mathbf{y},n}^{k}\right)$ towards the reference orientation in the root.

Once a full 6-DoF pose is detected, a pre-measured 3D bounding box is projected into the image (see Fig. 3).

### C. Features

We train our binary decision functions at each node on different appearance channels such as *color* in Lab space, $1^{st}$- and $2^{nd}$-order *gradients* in $x$ and $y$ dimensions on the intensity channel, *depth*, *surfel-pair features*, and *HoG*. HoG channels are produced as a soft bin count of gradient orientation in a depth-normalized window around each pixel. To boost invariance against noise and disturbance, we further perform *min* and *max* filtration with depth-normalized kernel-size in a local neighborhood.

**Depth**. It has been observed that depth cues can improve object detection tremendously [27], [28]. It enriches the information about object in terms of geometry, shape, contour etc. We thus use depth as an additional appearance channel.

Unlike [26], we use depth-normalized offset vectors in binary node tests. This way, the size of the offset vectors is automatically adjusted according to the scale of objects in the image, which obviates the need of presenting object class training images at multiple scales and handles variable scales efficiently during recall. Using the depth information at each pixel $I_d(\mathbf{y})$, the binary test function Eq. (1) is changed as below:

$$ t_{j,\mathbf{u}_1,\mathbf{u}_2,\tau}(\mathbf{y}) = \begin{cases} 0, & \text{if } I^j\left(\mathbf{y} + \frac{\mathbf{u}_1}{I_d(\mathbf{y})}\right) - I^j\left(\mathbf{y} - \frac{\mathbf{u}_2}{I_d(\mathbf{y})}\right) < \tau \\ 1, & \text{otherwise.} \end{cases} \tag{10} $$

**Surfel-pair features**. The availability of dense depth images allows incorporating geometry features into the decision cascade. For example, a *soda can* has a cylindrical shape whereas *cereal boxes* are cuboid. In order to capture such characteristic shape, we include tests on 4 dimensional surfel-pair features [8] as additional node splitting criteria. These features characterize the relative position and local surface orientation between two points in the scene. For any two points $\mathbf{p_1}$ and $\mathbf{p_2}$ and their corresponding normals $\mathbf{n}_1$, $\mathbf{n}_2$, the surfel-pair feature is computed as:

$$ S(\mathbf{p}_1, \mathbf{p}_2) = \left(\|\mathbf{d}\|_2, \angle(\mathbf{n}_1, \mathbf{d}), \angle(\mathbf{n}_2, \mathbf{d}), \angle(\mathbf{n}_1, \mathbf{n}_2)\right), \tag{11} $$

where $d := \mathbf{p}_2 - \mathbf{p}_1$. If one channel of the surfel-pair features is chosen for the test function, the function thresholds on the value of the feature directly through

$$ t_{j,\mathbf{u}_1,\mathbf{u}_2,\tau}(\mathbf{y}) = \begin{cases} 0, & \text{if } S^j\left(p\left(\mathbf{y} + \frac{\mathbf{u}_1}{I_d(\mathbf{y})}\right), \left(\mathbf{y} - \frac{\mathbf{u}_2}{I_d(\mathbf{y})}\right)\right) < \tau \\ 1, & \text{otherwise.} \end{cases} \tag{12} $$

## IV. GENERATION OF RICH TRAINING DATA

Visual object detection in unstructured environments is challenging due to background variability, clutter, occlusions, changes in illumination, different viewpoints and variable scales. It is important to capture these variations during training to achieve robustness of the system. However, manually annotating a rich training dataset with large variety in scenes with ground truth object detections and poses is barely feasible. Instead we propose to make use of a simple controlled training setup that provides ground truth conveniently, and to artificially render a variety of scenes from this data.

We obtain many view points onto the objects in a turn-table setup with varying angles in pitch and yaw. Object segmentation and ground truth pose are easily obtained through segmenting the object above the turn table plane and utilizing knowledge about the turn-table rotation and the relative location and orientation of the camera towards the turn-table. The object's 3D center and bounding box is found through overlaying object segments from 360° viewing directions into a single point cloud and measuring extents of the points.

To generate new training scenes, we extract RGB-D segments of the objects from the turn-table views. We render table planes with varying texture, color, and lighting conditions, and distribute the object views on the plane. For training examples for the object classes, the object to be trained is placed at its original location and orientation in the

Fig. 4. We render new RGB-D training images by increasing context information and introducing intensity variations similar to real world scenes

turn-table scene. To simulate clutter and occlusions, we place object views of other classes around the object. Example images for the background class are simply populated with views of all other object classes. Although our approach does not provide a photo-realistic rendering, our goal is to achieve a similar statistical distribution in intensity, color and depth as in natural scenes. Fig. 4 shows training images rendered with our approach using the RGB-D Objects Dataset [29].

## V. EXPERIMENTS

TABLE I

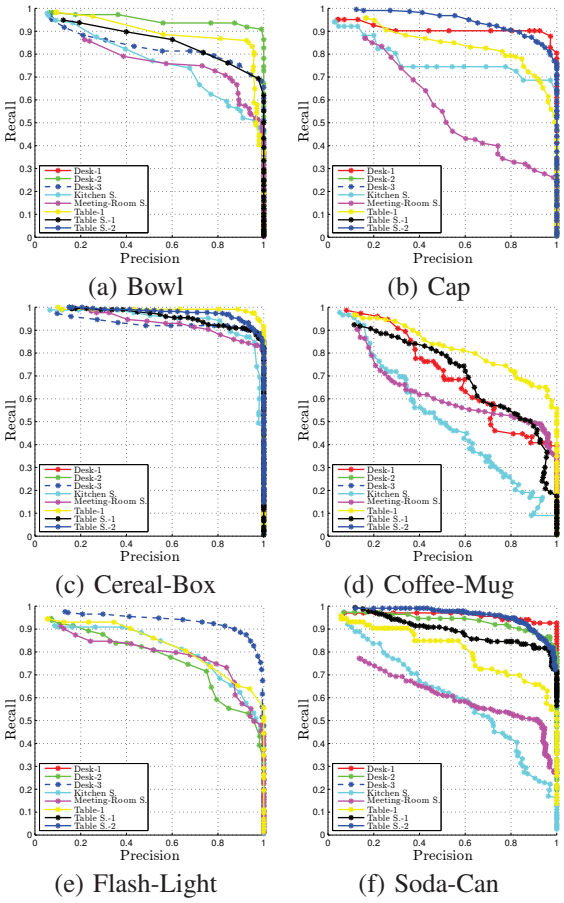
AVERAGE ACCURACY AT EER FOR DIFFERENT CHANNEL COMBINATIONS (C: COLOR, D: DEPTH, S: SURFEL-PAIR, H: HOG) AND ANGULAR DEVIATION ON THE SCENES SEQUENCES. SEE TEXT FOR DETAILS.

| category | appearance channel | | | | c+d angle deviation(°) |
|---|---|---|---|---|---|
| | c+d (%) | c+d+s (%) | c+d+s+h (%) | c+h [26] (%) | ($\mu \pm \sigma$) |
| bowl | 66 | **68** | 49 | 21 | 15.98 ± 18.55 |
| cap | **64** | 62 | 64 | 8 | 17.86 ± 17.15 |
| cereal box | **83** | 77 | 80 | 20 | 13.91 ± 10.34 |
| coffee mug | 44 | **45** | 43 | 17 | 13.22 ± 13.79 |
| flashlight | 64 | **66** | 60 | 16 | 12.35 ± 9.86 |
| soda can | 65 | 65 | **66** | 27 | 12.0 ± 11.21 |

We evaluate our approach on the publicly available RGB-D Objects and Scenes Datasets [29]. The Scenes Datasets contain RGB-D images of annotated objects in the 6 classes bowl, coffee mug, cap, cereal box, flashlight, and soda can[1]. In the Objects Dataset, the objects are placed on a turn-table and viewed in 3 pitch angles (30°, 45°, 60°) and approx. 10° steps in yaw. The same object instances have been placed in scene imagery for the Scenes Dataset. It comprises video sequences of common indoor environments, including office workspace, kitchens, and meeting rooms.

Our training settings are as follows: Every class-specific Hough forest consists of 5 decision trees. For each tree we

[1]We only evaluate on these 6 classes for this reason.




(a) Bowl (b) Cap (c) Cereal-Box (d) Coffee-Mug (e) Flash-Light (f) Soda-Can

Fig. 5. Precision recall curves for the objects: (a) Bowl, (b) Cap, (c) Cereal-Box, (d) Coffee-Mug, (e) Flash-Light and (f) Soda-Can, computed for all the eight *RGB-D Scene Datasets* [29]. All plots are computed for color + depth channel combination

randomly choose 250 images of the object class and 250 images of background. The background images are mixed from rendered scenes (Sec. IV) and real scene imagery from the datasets that do not contain the objects to be classified[2]. We choose 1000 and 1000 random pixels from each image, respectively. At every node, 2000 tests are generated, while the trees are trained up to a maximum depth of 20.

During detection we discretize scale into 10 bins for the 3D position Hough space. We count detections ( with score higher than a set threshold ) as true positives, if its bounding box overlaps by at least 50% with the ground truth. Each ground truth bounding box may only be associated once with a detection. For our 3D approach, we determine an enclosing 2D bounding box on the projected corner points of the found 3D bounding box. To generate a fair comparison with the method in [26], we augmented their approach to suppress local submaxima within the bounding boxes of strongest maxima detections.

We evaluate precision and recall for varying detection threshold on the RGB-D Scenes dataset [29] (Fig. 5). In Table I, the average accuracy at equal precision/recall error

[2]Note that our results are not comparable to [2] since we do not evaluate on the turn-table scenes.

rate (EER) over all sequences is tabulated for different channel combinations and compared to the method in [26]. The last column shows mean and standard deviation in the upward orientation estimate. We observe that our method outperforms pure RGB based object recognition as in [26] with significant margin. The depth channel introduces essential information about object shape. It allows spatial information to be represented in 3D, which reduces the smearing of votes in Hough space and increases the overall recognition rate. Among all the objects we used for testing, performance of *coffee-mug* is lowest mainly due to its high shape resemblance with other object categories such as soda-can and bowl. HoG channels are highly informative about object shape near object boundaries. We expect the performance of surfel-pair features or HoG to make a difference with larger numbers of sample pixels and sample node tests during training. Such large sample densities are currently prohibitive by the time-expensive training with our CPU implementation. For low number of samples the addition of HoG channels seems to reduce the probability of selection of other useful channels during the node test optimization and hence reduces the performance. A faster GPU implementation could make HoG channels applicable. The size of the objects also influences detection rate as for large objects, detection is achievable even at further distances, e.g. for the *cereal-box*. Our method provides good estimates of object orientation with an average mean error of ca. $14°$. It could therefore be useful for initializing a pose optimization method such as ICP or pose trackers. Naturally, objects with spherical shapes, such as the caps or bowls, yield higher angular deviation.

## VI. CONCLUSIONS

In this paper, we proposed a novel approach to object-class detection and continuous pose estimation in RGB-D images. We discriminatively train random decision forests to classify pixels and to vote for 3D object location and orientation. We exploit depth at various stages of the processing pipeline. For training, we extract object views and render new training scenes with varying background, clutter, lighting changes, and occlusions. The features used in the random decision forest are made scale-invariant through depth-normalization. We furthermore use depth cues to make use of the geometry information contained in the RGB-D images for detection. Finally, 3D orientation votes are cast from local reference frames that are created from local surface normals and 3D point configurations.

Our experiments demonstrate that our approach yields good accuracy in detecting objects and recovering their pose. It well compares with a state-of-the-art approach to object-class detection that only utilizes RGB information during recall.

In future work, we will evaluate our approach for scalable multi-class detection that detects classes in a taxonomy. For scalable training on large datasets or on-line interactive learning of the trees, we will pursue a GPU implementation of the method.

## REFERENCES


[1] N. Razavi, J. Gall, and L. J. V. Gool, "Scalable multi-class object detection," in *Proc. of CVPR*, 2011.

[2] K. Lai, L. Bo, X. Ren, and D. Fox, "A scalable tree-based approach for joint object and pose recognition," in *Proc. of AAAI Conf.*, 2011.

[3] A. Collet, M. Martinez, and S. S. Srinivasa, "The MOPED framework: Object recognition and pose estimation for manipulation," *IJRR*, 2011.

[4] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[5] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," in *Proc. of ECCV*, 2006.

[6] A. Johnson, *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, CMU, Pittsburgh, PA, August 1997.

[7] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *Proc. of ECCV*, 2010.

[8] E. Wahl, U. Hillenbrand, and G. Hirzinger, "Surflet-pair-relation histograms: a statistical 3d-shape representation for rapid classification," in *Proc. of 3DIM*, 2003.

[9] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. of ICRA*, 2009.

[10] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the viewpoint feature histogram," in *Proc. of IROS*, 2010.

[11] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3D object recognition," in *Proc. of CVPR*, 2010.

[12] C. Papazov, S. Haddadin, S. Parusel, K. Krieger, and D. Burschka, "Rigid 3D geometry matching for grasping of known objects in cluttered scenes," *IJRR*, vol. 31, no. 4, pp. 538–553, 2012.

[13] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *In ECCV workshop on statistical learning in computer vision*, pp. 17–32, 2004.

[14] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Trans. on PAMI*, vol. 32, no. 9, pp. 1627–1645, 2010.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of CVPR*, 2005.

[16] H. Su, M. Sun, L. Fei-Fei, and S. Savarese, "Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories.," in *Proc. of ICCV*, 2009.

[17] R. J. Lopez-Sastre, T. Tuytelaars, and S. Savarese, "Deformable part models revisited: A performance evaluation for object category pose estimation.," in *Proceedings of the ICCV Workshops*, 2011.

[18] C. Gu and X. Ren, "Discriminative mixture-of-templates for viewpoint classification," in *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, pp. 408–421, 2010.

[19] B. Pepik, P. Gehler, M. Stark, and B. Schiele, "3D2PM - 3D deformable part models," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012.

[20] M. Sun, G. Bradski, B.-X. Xu, and S. Savarese, "Depth-encoded hough voting for joint object detection and shape recovery," in *Proc. of ECCV*, 2010.

[21] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich, "Viewpoint-aware object detection and continuous pose estimation," *Image and Vision Computing*, 2012.

[22] M. Zia, M. Stark, B. Schiele, and K. Schindler, "Revisiting 3D geometric models for accurate object shape and pose," in *Proc. of ICCV Workshops*, 2011.

[23] J. Liebelt, C. Schmid, and K. Schertler, "Viewpoint-independent object class detection using 3d feature maps," in *Proc. of CVPR*, 2008.

[24] T. Wang, X. He, and N. Barnes, "Learning hough forest with depth-encoded context for object detection," in *Proc. of DICTA*, 2012.

[25] W. Wohlkinger, A. Aldoma, R. Rusu, and M. Vincze, "3DNet: Large-scale object class recognition from CAD models," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[26] J. Gall and V. Lempitsky, "Class-Specific Hough Forests for Object Detection," in *Proc. of CVPR*, 2009.

[27] J. Stückler, N. Biresev, and S. Behnke, "Semantic mapping using object-class segmentation of RGB-D images," in *Proc. of IROS*, 2012.

[28] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon, "Efficient regression of general-activity human poses from depth images," in *Proc. of ICCV*, 2011.

[29] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *Proc. of ICRA*, 2011.