Object Pose Estimation via Pruned Hough Forest With Combined Split Schemes for Robotic Grasp

Huixu Dong, *Member, IEEE*, Dilip K. Prasad, *Senior Member, IEEE*, and I-Ming Chen, *Fellow, IEEE*

***Abstract*— Robotic grasp in complex open-world scenarios requires an effective and generalizable perception. Estimating object's pose is needed in a variety of practical grasping scenarios. Here we present a novel approach of pose estimation of textureless and textured objects. The algorithm utilizes a single RGB-D image to exploit depth invariant, oriented point pair feature as well as local contextual sensitivity in cluttered environments. To enhance the performance of the voting process and improve learning efficiency, we employ a global pruning algorithm that reduces the risk of overfitting and simplifies the structure of decision trees after compensating for the complementary information among multiple trees by optimizing a designed global objective function. Finally, we also refine the pose obtained from the above stage. The proposed approach of estimating 6-D (degree of freedom) poses of textured and textureless objects is evaluated on publicly available data sets against the recent works under various conditions. It illustrates that our framework is superior to these recent works. Further, we perform extensive qualitative experiments of robotic grasp to illustrate the proposed approach can be applied to practical scenarios.**

***Note to Practitioners*—This article is motivated by the problem of the pose estimation of textured and textureless objects in clutter environments. It is difficult for conventional works to address the issue of estimating textured or textureless objects' poses in such scenarios. We considered that a novel system should be able to obtain the 6-D poses of objects. Therefore, we investigate the combined use of multiple split functions with different characteristics. Learning the model based on Hough forests always cost much computational resource; therefore, we construct a novel pruned Hough forest for solving this issue. Through the comparison and robotic grasp verifications, the behavior of our system can be used in practical applications. In future, we will deploy the proposed system in robotic assembling tasks.**

***Index Terms*— Hough voting, local context, pose estimation, robotic grasp, split function.**

I. Introduction

VISION for robotics is often approached differently from general computer vision as it involves interacting directly with the environment [4], [5]. Pose estimation of objects is frequently needed in robotic manipulations and scene understanding. A few important challenges in estimating poses of objects in cluttered and occluded scenarios still remain.

There exists a vast amount of research activities regarding object detection and pose estimation, including template-, deep learning-, and feature-based methods [7]–[9]. The common approach of detecting objects and predicting coarse poses is template matching [3]. Due to template matching based on 3-D descriptors, template-based techniques can work accurately in practical applications, but also suffer in cases of occlusions. Currently, the convolutional neural network (CNN) have been applied to learning RGB-D or RGB features. Learning RGB-D representations is used for detecting objects and estimating 6-D poses [7], [8]. Training on real images may require a significant energy of the data collection, which limits the applications of learning-based methods. Dense features with small variance are applied to estimating object pose, which achieved a high prediction accuracy [2], [12], [13]. There are a lot of feature-based strategies by pixel voting to improve the performance of pose estimation [6], [12], [14]. It has been verified that the depth comparison features can be employed to improve pose estimation tremendously since the depth enriches the object's information such as geometry, shape, contour, and so on [15]. Moreover, the feature on a pair of two oriented points was used for estimating object's pose [6], [14]. Also, the contextual feature is employed to improve the accuracy of pose estimation [12]. In terms of dense features, it means each pixel in image generates some predictions corresponding to the desired outputs.

Given the availability of RGB-D image, we design a system to obtain 6-D poses (3-D rotation and 3-D translation) of everyday rigid objects accurately even in the presence of clutter. Our feature-based work of object's pose estimation is motivated by the concept of Hough forest employed in [6] and [14]. Since the splitting strategy in a binary tree determining pairwise pixel relationships has proven to be well-suited on object detection and pose estimation [16], we utilize three split schemes with stable characterizes, such as depth invariant, oriented point pair feature as well as local contextual sensitivity, to enhance the estimation performance of our method.

In terms of given scenarios, due to a great deal of clutter, the feature on surface point pair is not well suited for estimating object's pose. To be more robust to scale changes, depth invariance is considered as a desirable feature to integrate into decision trees. Aside from their simple operations, depth-based

Manuscript received May 23, 2020; revised July 13, 2020; accepted August 20, 2020. Date of publication September 17, 2020; date of current version October 6, 2021. This article was recommended for publication by Associate Editor C. Yang and Editor K. Saitou upon evaluation of the reviewers' comments. *(Corresponding author: Dilip K. Prasad.)*

Huixu Dong is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA.

Dilip K. Prasad is with the Department of Computer Science, UiT The Arctic University of Norway, 9037 Tromso, Norway (e-mail: dilipprasad@gmail.com).

I-Ming Chen is with the Robotics Research Centre, Nanyang Technological University, Singapore 639798.

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2020.3021119

features are evaluated rapidly. For the feature on oriented point pair, it is considered as a low-dimensional description of object surfaces that have generally rich variations at each split node. For instance, a coke can have a cylindrical shape whereas milk boxes are cuboid. The surfaces of these two objects present different geometric attributes such as surface normal and so on. However, it is not very efficient as a large number of different point pairs on planar or self-symmetric objects fall into the same hash slot. We adopt the binary tests for surflet-pair features as a splitting criterion at nodes, which is invariant to translation and rotation. The surflet-pair features characterize the intrinsic geometric relation in the scene. In addition, we argue that integrating contextual information into random forests obviously enhances the learning performance. "Context" indicates the interrelationship of the nearby pixels, also called the neighborhood. It is generally accepted that the surroundings of one pixel have a profound inherent relation with this pixel. The spirit of a context-sensitive decision tree is to utilize split functions integrated by contextual information of image pixels at each binary test by coupling the outputs of pixels in a small neighborhood, before determining where to route the node. In summary, the good performance of our approach in estimating 6-D pose of objects attributes to the reason that it makes use of the macroscopic physical invariance such as depth information and also utilizes the microscopic pixel information including oriented point surflet-pair and neighborhood contextual information against occluded and cluttered scenarios. By adding these features, the casted votes in Hough space tend to get real pose candidates. Moreover, in terms of a texture-less object, the surface has the similar appearance and significantly different texture information. Thus, it is difficult for the methods relying on abundant appearance texture to predict the object class and estimate the pose. The proposed approach extracts geometric information such as depth, the space vector relations of pixels rather than just the surface appearance to realize an excellent performance on estimating the pose of texture-less objects. Indeed, Hough forest with a large size also causes high computational and storage cost, which is a serious issue, especially for real applications. To alleviate the computational burden and maintain the high performance of Hough forest, inspired by [17] but differing from them, we implement to a new application-pose estimation by pruning the leaves of decision trees under the global optimization in training and also, enforce dependencies between pixels rather than make predictions for each training pixel independently. From obtaining the pose hypothesis, we further refine the final pose by optimizing the cost function structured. These features are derived from the following contributions.

1) We combine local contextual information based on the depth invariant, geometric surflets as well as contextual sensitivity into binary test pool at each node of decision trees using a new jointed split function defined as a predictor, which presents a superior performance when compared with several pose estimators.
2) We incorporate complementary information among the decision tree for enhancing the learning performance of Hough forest and prune insignificant leaves to improve the learning effectiveness. It is proven that such strategy is effective for pose estimation.
3) We achieve considerable successful rates on robotic grasping experiments in cluttered and occluded scenarios. It is verified that the proposed algorithm can be used in practical applications in RGB-D images with certain occlusions.

We present the proposed algorithm in Section II. We provide quantitative and qualitative analysis of the experiments in Section III, following by the conclusion in Section IV.

II. Methodology

A. Construction of Hough Forest

For Hough forest, during the training period, a large amount of training patches reach each node of a decision tree. To optimize the information gain, a defined split function is used for guiding the sample patches to different directions at each intermediate node and a threshold is assigned to a split function. When the patch is at the maximum depth and the remaining number of patches is less than a threshold, the split node is regarded as a leaf.

In a random forest $\mathcal{F}$, each binary tree is built depending on a set of sampled image pixels. The local patch appearance $\mathcal{P}_i$ that is a 3-D patch (e.g., $V \times V \times 4$) extracted from an RGB-D image is made up of a few parts: $\{\mathcal{P}_i = (\mathcal{I}_i, c_i, \theta_i, s_i, d_i, n_i)\}$. i is the center of the patch $\mathcal{P}_i$. $\mathcal{I}_i = (I_i^0, I_i^1, I_i^2, \ldots, I_i^g, \ldots)$ represents obtained features at the patch $\mathcal{P}_i$, and I_i^g is the gth feature channel at the patch $\mathcal{P}_i$ [4]. Here the feature channels, such as depth, surflet-pair features, color, the first- and second-order gradients in x- and y-dimensions for the intensity space, LBP, and HoG, are applied. The vector θ_i includes the pose parameters $\theta^x, \theta^y, \theta^z, \theta^{\text{yaw}}, \theta^{\text{pitch}}, \theta^{\text{roll}}$ associated with each patch. θ^x, θ^y, and θ^z represent offsets from the point in the camera falling on the center of the training patch to the object position in 3-D, while $\theta^{\text{yaw}}, \theta^{\text{pitch}}$, and θ^{roll} are the object rotation angles denoting the object orientation. s_i indicates a binary class label (0 for background and 1 for object patch) and d_i is the offset from the centroid of the object to the center of the sample patch. It is noted that d_i is undefined in case of image pixels not belonging to an object class and n_i is the normal of the center i. c_i denotes the object class.

B. Split Functions Based on Features

The critical step of designing the pose estimators based on random forests is to create an effective split function. While training, every nonleaf node B in the decision tree is appointed a binary test for the sample patch.

1) Designing a Split Function Based on Depth Information: We design a split function based on the pixelwise depth information [15]. We adopt depth-normalized offset vectors in binary tests. For the pixel i in an image G, the feature response $F_i(G)$ is defined

$$F_i(G) = \frac{1}{R_1} \sum_{o_1 \in R_1} I_i^g \left(i + \frac{o_1}{D_G(i)} \right) - \frac{1}{R_2} \sum_{o_2 \in R_2} I_i^g \left(i + \frac{o_2}{D_G(i)} \right).$$

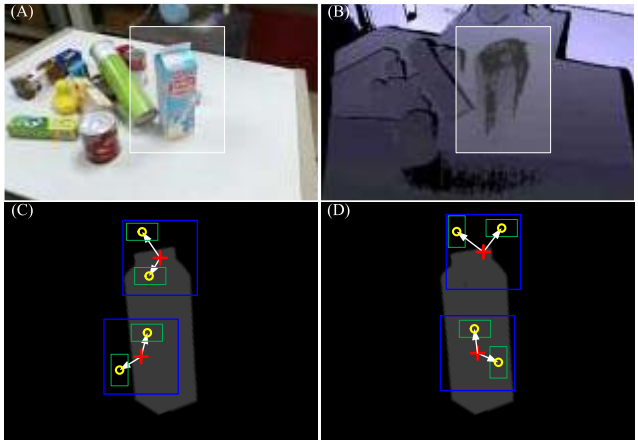

Fig. 1. (a) RGB image. (b) Depth image. (c) Large response of depth difference for two example features. (d) Small response of depth difference with the same two example features above. A sample patch is marked by a bounding box in blue and the two randomly generated regions R_1 and R_2 as parts of a binary test are enclosed by the green frames; the white arrow represents the 3-D offset vector (in red) to the offset pixel locations (yellow circles).

Thus, the binary test $V_{B,R_1,R_2,g,\tau}(\mathcal{I})$ is defined as

$$V_{B,R_1,R_2,g,\tau}(\mathcal{I}) = \begin{cases} 0(\text{left}), & \text{if } F_i(G) < \tau \\ 1(\text{right}), & \text{otherwise} \end{cases} \tag{1}$$

where I_i^g represents the feature channel g, R_1 and R_2 denote two rectangle regions within the patch boundaries, and τ represents a corresponding threshold. In an image G, $D_G(i)$ is the depth at pixel i, and o_1, o_2 represent two offsets from the pixel i. Indeed, these features can implicitly encode contextual information. The two rectangular area average values instead of two pixels are used in the binary test, which is less sensitive to the noise, as illustrated in Fig. 1. R_1, R_2, g are generated randomly while training.

The normalization of the offsets by $(1/D_G(i))$ makes the features largely depth invariant. In this case, the size of the offset vectors is adaptive with the scale of objects in the image, which eliminates the need of transforming object class training images at multiple scales and solves the issue of variable scales efficiently during recall. Moreover, offset pixels with undefined depth or which lies on the background or outside of the image is set a positive value.

2) Binary Geometric Surflet Split Function: Here a pair of oriented surface points are referred as surflets; each oriented point with its position p and its local surface normal n. They reflect a generalization of curvatures that measure geometric relations between neighboring surflets. Inspired but differ from [16], we introduce more shape information and use the encoding strategy to calculate the split value. Position and surface normal are estimated from multiple neighboring points in the 3-D space based on objects' models. For a pair of surflets (p_1, n_1) and (p_2, n_2), we set p_1 as the origin. As shown in Fig. 2, for the surflets (p_1, n_1) and (p_2, n_2), the surflet-pair is described by the parameters

$$\angle(n_1, n_2), \quad \angle(n_2, \delta), \quad \angle(n_1, \delta), \quad \delta = \|p_2 - p_1\|_2$$

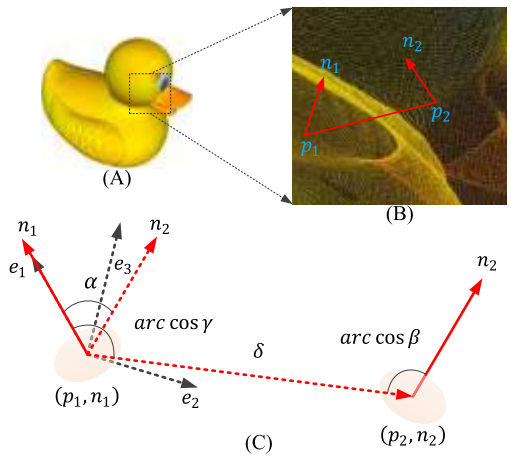

Fig. 2. (a) 3-D model. (b) Patch: two surface points p_1, p_2 and the orientations n_1, n_2. (c) Four parameters representing the surflet feature.

where $\angle(n_1, n_2)$ and $\angle(n_2, \delta)$ denote n_2 as a direction angle and the cosine of a polar angle, respectively; $\angle(n_1, \delta)$ and δ represent the direction and the translation from p_1 to p_2, respectively. We define the surflet feature $S = (\angle(n_1, n_2), \angle(n_2, \delta), \angle(n_1, \delta))$. Thus, a unique set of parameters can be exactly mapped onto the geometric configuration of a surflet pair. A total of $(m(m-1)/2)$ features is obtained from a surface with m surflets. Due to such huge amount of features, we have to propose an efficient processing scheme. Similar to [18], we encode every element using 8-bit binary number in the surflet feature S by the following way: first judge whether the normal of a pair of orientated points are parallel or not. The most significant bits of all the elements are set as 1 when they are parallel, otherwise 0; and then, the rest elements are quantized into the remaining 7 bits. Indeed, the encoding can improve the computational speed since normalization step is skipped. Second, all the elements in the feature vector can be integrated together as the most dominant factor by the means of mathematical bitwise operation. We define a novel split function drastically, which improves the accuracy of random forests with the binary tests on the pose estimation of object, as discussed in Section III as follows:

$$B(S) = \sum_{\substack{p_1 \in R_1 \\ p_2 \in R_2}} \sigma\left(E_{n_1 n_2} \otimes E_{n_1 \delta} \otimes E_{n_2 \delta}\right)$$

where $\sigma(\cdot)$ is a binary function that returns 1 if "$\cdot$" is true, 0 otherwise; $E_{n_1 n_2}$, $E_{n_1 \delta}$, and $E_{n_2 \delta}$ represent the binary encoding numbers of $\angle(n_1, n_2)$, $\angle(n_2, \delta)$, and $\angle(n_1, \delta)$, respectively; $\otimes$ is the bitwise AND operation, which is highly efficient since only binary bitwise operations and the addition are involved; p_1 and p_2 are from the rectangles R_1 and R_2, respectively (see Fig. 3). Thus, at the nonleaf node B, we can define the split function $F_{B,R_1,R_2,\tau}(S)$ as

$$F_{B,R_1,R_2,\tau}(S) = \begin{cases} 0, & B(S) < \tau \\ 1, & B(S) \geq \tau \end{cases} \tag{2}$$

where τ is a threshold stored at the test pool.

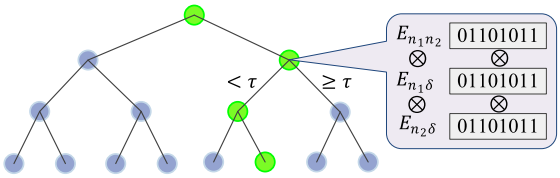

Fig. 3. Bitwise operation.

3) Split Function by Integrating Contextual Sensitivity: Inspired by [12], we use the idea of prioritized node training to increase the learning effectiveness in a Hough forest. But, we do not use loss function proposed in [12] to determine the order of a priority queue. The new split function is parametrized by a context-sensitive decision tree producing contextual information based on displacement vectors o_1, o_2 relative to a location i, a threshold τ that are used to perform a binary test for coupling the context sensitivity. The split function that takes a patch $\mathcal{P}_i$ as input is provided as follows:

$$\mathcal{W}(\mathcal{P}_i|o_1, o_2, \tau) = \begin{cases} 0(\text{left}), & \text{if } E_{(s,d,s',d')\sim Q_1,Q_2}(\mathcal{Z}) < \tau \\ 1(\text{right}), & \text{otherwise.} \end{cases} \tag{3}$$

$$\mathcal{Z} = \begin{cases} K_\sigma(d - d'), & (s, s') = (1, 1) \\ 0, & (s, s') \neq (1, 1) \end{cases} \tag{4}$$

where $E_{X\sim Q}(f(\mathcal{X}))$ represents the expectation of $f(\mathcal{X})$ regarding $\mathcal{X}$ based on a probability distribution Q that denotes the posterior probability of votes for sampling patches arriving at leaf nodes; $c \in \{0, 1\}$ indicates the binary class label illustrating the presence of the object; we use d as the displacement of the center of object. In addition, we predict (s', d') sampled depending on Q instead of (s, d). From a binary tree T, we can obtain the associated posterior probability of every vote element for a patch $\mathcal{P}_i$ by calculating Q. $K_\sigma(\cdot) = \exp(-(\|\cdot\|^2/\sigma^2))$ with σ-algebra is just a Gaussian (or radial basis function kernel). The bandwidth σ is a free parameter controlling the "window" of the kernel. $Q_k = Q(\cdot|(i + o_k), T.), k \in \{1, 2\}$, are the posterior probabilities by T at positions $(i+o_1, i+o_2)$ of the patch $\mathcal{P}_i$, respectively. For Hough forest, the novel split function $\mathcal{W}(\mathcal{P}_i|o_1, .o_2, \tau)$ tries to separate the training set into elements that have similar voting length and direction, which indeed reduces an uncertainty uniformly from root to leaf nodes and leads to more stable contextual information.

C. Training

1) Traversing Decision Trees: Predicting the object class is considered as a classification problem. The binary test, which allows the stored entropy to be minimal, is provided as

$$U_1(A) = -|A| \cdot \sum M_c \ln(M_c) \tag{5}$$

where $|A|$ denotes the patch number in a set A and M_c represents the proportion of patches with the object class label c in a set A.

Estimating 6-D pose θ at each node is regarded as a regression problem with a multivariate Gaussian distribution, i.e., $p(\theta) = \mathcal{N}(\theta; \tilde{\theta}, \Gamma)$. $\tilde{\theta}$ indicates the mean of 6-D pose θ and Γ represents the full covariance matrix

$$U_2(A) = \ln(|\Gamma(P)|) - \sum_{r\in\{\mathcal{L},\mathcal{R}\}} \frac{|P_i|}{|P|} \ln(|\Gamma_n(P_i)|) \tag{6}$$

where $\mathcal{L}$ and $\mathcal{R}$ represent the left and the right, respectively; P_i is the set of patches reaching node r and P is the set of patches at the parent node of r. The determinant of the covariance matrix Γ tends to be minimized by maximizing (6). The covariance matrix $\Gamma = \text{diag}(\Gamma^q, \Gamma^a)$ is block-diagonal, Γ^q and Γ^a denote the covariance matrix among the position vectors and among the rotation angle vectors, respectively [4]. Thus, we can obtain the following equation:

$$U_2(A) = \ln(|\Gamma^q| + |\Gamma^a|) - \sum_{r\in\{\mathcal{L},\mathcal{R}\}} \frac{|P_i|}{|P|} \ln(|\Gamma^q_r| + |\Gamma^a_r|). \tag{7}$$

2) Pruning Random Forests by Complementary Information: Here we make good use of complementary information among multiple trees for improving the performance of object's pose estimation and alleviate the computational burden. We define the prediction of a decision tree as

$$Y = w\Phi(x) \tag{8}$$

where the vector Y denotes the output including the object class and the object's pose. The indicator $\Phi(x)$ represents a mapping function from the input data x to the structure of a tree. w depicts the leaf matrix mapped by all the leaf vectors. By means of minimizing the loss function we define the learning process for the leaf vectors in the random forest

$$\min_w \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} L(Y_i^t, \hat{Y}_i)$$
$$\text{s.t. } Y_i^t = w_t \Phi_t(x_i) \quad \forall_i \in [1, N] \quad \forall t \in [1, T] \tag{9}$$

where N denotes the training patch sample number and T is the number of decision trees; Y_i^t denotes the prediction such as the object class and object's pose of the tth tree; W_t and $\Phi_t(X)$ are the leaf matrix and the indicator vector of the tth tree; $L(Y_i^t, \hat{Y}_i)$ indicates the loss function between the prediction Y_i^t and the ground truth $\hat{Y}$. As for regression regarding the

Algorithm 1 Training Stage

Initialize the root nodes with all $\mathcal{P}_i$
For *all* $\mathcal{P}_i$ *in the data set* ***do***
  For *trees in forest* ***do***
    For *depth from 1 to maximum depth* ***do***
      Check stopping criteria
      Choose a split function from ***Eq.(1,2,3)***
      For *binary test in binary test pool* ***do***
        Calculating the best test
      End
      Determine $\mathcal{P}_i$ *to the left or the right*
    End
    Store the information of object's class and pose
  End
  Minimize the loss ***Eq.(9)*** *with all* $\mathcal{P}_i$ *and trees*
End

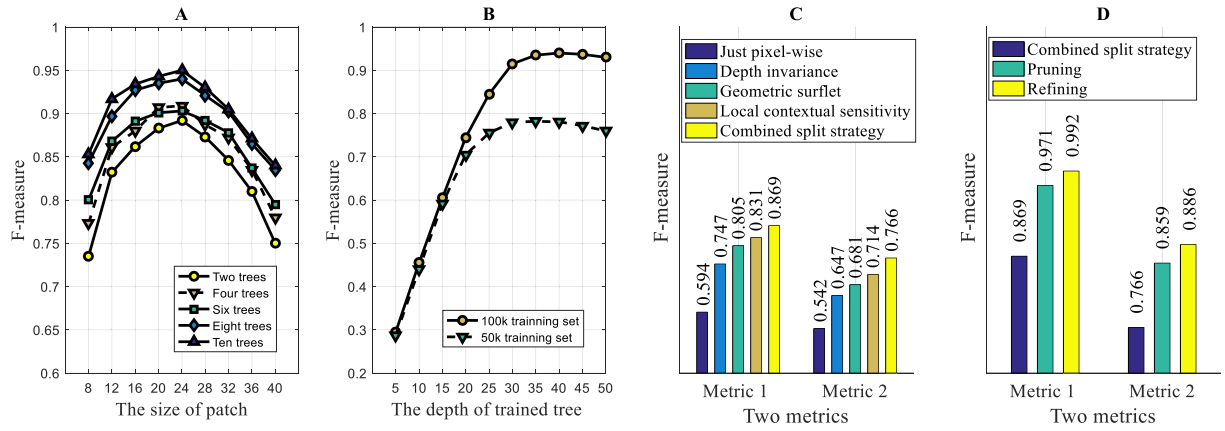

Fig. 4. Parameter determination [(A) for the patch size and (B) for the tree depth] and self-comparison [(C) for verifying the effect of each integrated strategy and (D) for the effect of two post-processes on the performance].

object's pose, the mean square error is applied to calculating this loss.

Indeed, the loss function $(1/T)\sum_{t=1}^{T} L(Y_i^t, \hat{Y}_i)$ summarized by Hough forest-based model is different from the ideal loss function $L(Y_i, \hat{Y}_i), Y_i = (1/T)\sum_{t=1}^{T} Y_i^t$ related to the final prediction of Hough forests. We integrate the ideal loss function into (9) to retrieve the complementary information among decision trees. Thus, we define the global reformulation of random forest as follows:

$$\min_{w} \frac{1}{2}\|W\|^2 + \frac{C}{N}\sum_{i=1}^{N} L(Y_i, \hat{Y}_i)$$
$$\text{s.t. } Y_i = W\Phi(x_i), \quad \forall i \in [1, N] \tag{10}$$

along with

$$W = [w_1, \ldots, w_t, \ldots, w_T]$$
$$\Phi(x) = [\Phi_1(x); \ldots; \Phi_t(x); \ldots; \Phi_T(x)]$$

where the L2 regularization term $(1/2)\|W\|^2$ can alleviate the over-fitting issue and C represents the control parameter. The proposed global objective reformulation is efficiently solved by the convex optimization based on the linear support vector.

D. Testing

While testing, each sample patch $\hat{\mathcal{P}}_i$ from a novel RGB-D image passes through all the trees in Hough forests. The probability $p(c|\mathcal{F}., \hat{\mathcal{P}}_i)$ of object class c is obtained via averaging the probabilities of object class labels at the reached leaf nodes. The pose is estimated based on nonmaximum suppression (NMS).

E. Refining the Pose

Here as an optional step for estimating object's pose, we use a similar solution proposed in [19] to refine the pose hypothesis obtained from the stage above to further improve the estimation accuracy and address the pose ambiguity problem.

III. Discussions and Experiments

We first determine the parameters of the algorithm (A) and use self-comparison to show the effect of the use of designed split functions and pruned Hough forest (B). Subsequently, the performance of the proposed method is assessed on public data sets using the same metrics [1], [20] (C). F-measure is used as the criterion as well. The robotic grasping experiments are conducted (D). It is noted that rigid objects with textureless and texture are invovled in all the compared experiments and robotic grasping experiments for varifying the proposed method can address pose estimation of textured and textureless objects.

A. Framework Parameters

Tuning parameters of the proposed algorithm is implemented on training images generated via the virtual camera in OpenGL in Section II. We train the model by varying a parameter while fixing all other parameters.

The patch size V generates an important influence on the performance since a patch with a big size always obtains a holistic matching, which results in being sensitive to clutter and occlusions. When the patch size becomes small relatively, it tends to be considered as noise. Moreover, a forest including more than eight binary trees costs much computational resource. The size V of the patch is set to 24 and the number of trees is 8, as shown in Fig. 4(a).

Regardless of using either 50k or 100k training images, the depth of trees is of importance in detection performance [see Fig. 4(b)]. Fig. 4(b) illustrates that increasing the tree depth results in the improvement of F-measure. The deeper tree requires needs more memory. The depth of trees is 35.

B. Performance Evaluations by Self-Comparison

The proposed method based on pruned Hough forest with combined split functions delivers a relative accurate performance on the metrics used in [1] and [20] for the data set [1] where the object class is known in advance. On the whole, the integration of split strategies on depth invariant, geometric orientation, and local contextual sensitivity into Hough forest is helpful to improving pose estimation's performance. For the metric [1], the proposed method is almost reaching to 90% of pose correctly, as illustrated in Fig. 4(c). Specifically, in terms of estimation performance, the estimator with just pixelwise split scheme is less than one with just depth invariance

Fig. 5. Some cases of pose estimation of object 3-D models constructed by ourselves. The results of pose estimation are visualized by the bounding boxes.

Fig. 6. Pose estimation examples on the data set of [1].

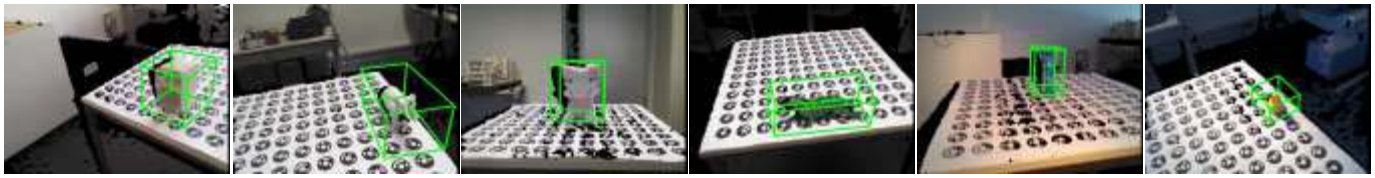

Fig. 7. Pose estimation examples on the data set of [2]. The estimated results are visualized via the bounding boxes.

(15.3%), one with just geometric surflet (21.1%) and one with just contextual sensitivity (23.7%). Contrastively, as the metric [20] is more rigorous than the metric [1] on the rotation evaluation, our approach still reaches 76.6% accuracy, which is considerably more than one with just pixelwise split function (−22.4%), one with just depth invariance (−12.9%), one with just geometric surflet (−9.5%), and one with just contextual sensitivity (−5.2%).

The use of the pruning strategy integrated into Hough forest leads to a considerable improvement of F-measure (10.2% and 9.3% for the metrics of [1] and [20], respectively) than one with the combined split scheme, as shown in Fig. 4(d). After pruning Hough forest, the size of the learning model changes to 167M from 682M. The training time reduces from 486 to 133 min. Indeed, the refinement stage is optional, and without it, the estimator already has got a good performance. For the data set of [1], the use of the refinement improves the performance of the final 6-D pose (−2.1% and −2.7% without the refinement for two metrics in [1] and [20], respectively), as shown in Fig. 4(d).

We demonstrate some results of pose estimation of objects in office scenarios, as shown in Fig. 5. Fig. 5 illustrates that the proposed pose estimator can deal with some occluded scenarios in terms of estimating object's pose.

C. Performance Evaluations on the Public Data Sets

1) Results on the Data Set Including Single Instance: Our estimator is assessed on the public data set of [1] consisting of 13 (out of 15) texture-less and textured objects in cluttered environments against the recent works. Table I summarizes the performance comparisons of pose estimation. The average accuracy over the 13 objects is computed and reports 99.5%. Nonetheless, the shape of Ape is irregular so that the patches sampled sometimes miss, which results in a lower accuracy than other items. From the average performance, our comparison with other methods provides further illustration of the advantage of our method to deal with one object pose estimation in a cluttered scenario, however, without occlusions among items. The proposed estimator illustrated that this model could generalize well in practical scenarios, without Gaussian noise [2]. We demonstrate some examples of pose estimation in Fig. 6 as well.

2) Results on the Data Set Under Different Light Conditions: Our approach is compared against several estimators on the data set of [2], which includes textured and texture-less objects with a variety of illuminations, such as the brightness, darkness, and directional spot light (spot), as shown in Table II. The spot image set was tested for showing the performance of pose estimator under a new lighting scenario, realizing an average accurate rate of 92.4%. The proposed estimator has a robust generalized capability for different appearances in multiple lighting conditions, even without the training set with Gaussian noise. Since the features extracted by us are geometric rather than just the appearance, the change of appearances does not have obvious effect on the pose estimation. Our approach realizes a better performance on estimating the pose of object with small size such as Duck. In terms of objects with less features such as Driller, the proposed estimator is superior to several recent methods because the integration of context sensitivity enhances the capability to learn a variety of features. Several estimation examples are illustrated in Fig. 7.

D. Robotic Grasping Experiments in Clutter Environments

To verify that our pose estimator can predict objects' pose efficiently and accurately in practical applications, we conduct

TABLE I

RESULTS ON THE DATA SET OF [1] FOR THE METRIC PRESENTED IN [1]. ALL THE DATA CAN BE FOUND IN THE CORRESPONDING WORKS

Methods	Object													
	Ape	Bench Vise	Driller	Cam	Can	Iron	Lamp	Phone	Cat	Hole Punch	Duck	Box	Glue	Average
Hinterstoisser[1](%)	95.8	98.7	93.6	97.5	95.4	97.5	97.7	93.3	99.3	95.9	95.9	99.8	91.8	96.3
Cabrera[5, 10] (%)	95.0	98.9	94.3	98.2	96.3	98.4	97.9	95.3	99.1	97.5	94.2	99.8	96.3	97.1
Kouskouridas [11] (%)	95.7	**99.7**	99.2	**99.6**	96.1	98.5	**99.6**	96.7	**99.8**	**99.5**	96.1	98.1	98.2	98.2
Ours(%)	**99.0**	99.5	**99.9**	99.4	**99.5**	**99.7**	99.4	**99.5**	99.6	99.3	**99.1**	**99.9**	**99.7**	**99.5**

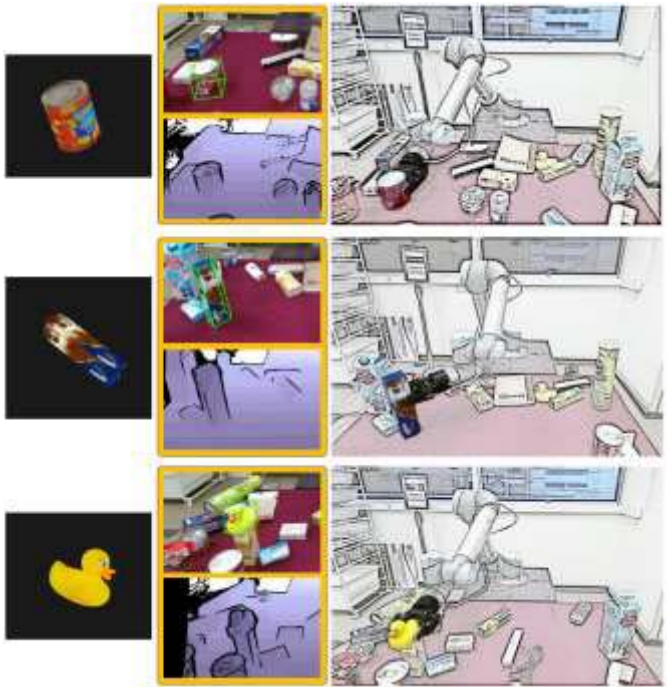

Fig. 8. Examples for robotic grasps in cluttered environments.

TABLE II

RESULTS FOR THE METRIC [1] ON THE DATA SET OF [2]

Object	[3]	[2]	[6]	Ours
Audio Box	-	**75.4**%	71.5%	73.6%
Carry Case	-	95.9%	90.7%	**96.7**%
Dish Soap	-	**100**%	**100**%	**100**%
Helmet	-	77.6%	74.5%	**80.1**%
Hole Puncher	-	98.1%	94.3%	**99.2**%
Pump	-	69.3%	67.4%	**71.8**%
Teapot	-	91.9%	89.8%	**94.7**%
Toolbox	-	99.5%	**100**%	**100**%
Toy (Battle Cat)	70.2%	91.8%	**92.4**%	90.5%
Toy (Panthor)	-	96.9%	94.2%	**97.3**%
Toy (Stridor)	-	94.0%	**94.3**%	93.6%
Stuffed Cat	-	98.3%	94%	**99.1**%
Duck	-	81.6%	87.7%	**92.7**%
Dwarf	-	67.6%	65.6%	**83.4**%
Mouse	-	89.1%	90.1%	**94.8**%
Owl	-	60.5%	90.27%	**91.4**%
Elephant	-	94.7%	96.13%	**98.3**%
Samurai	-	98.5%	**99.6**%	98.6%
Sculpture 1	-	82.7%	89.5%	**93.4**%
Sculpture 2	-	**100**%	**100**%	**100**%
Average	-	88.2%	89.1%	**92.4**%

robotic grasping experiments based on the presented pose estimator in cluttered environments. The experimental platform is made up of a robotic arm-UR5, a two-finger Robotiq 85 gripper with under-actuated grasping property [21], [22], and a Kinect v2 camera, as shown in Fig. 8.

We define that the robotic grasp is considered as a success grasp based on the proposed perception strategy if an object is successfully picked up from an initial position. It is well known that many factors, such as suitable motion planning, robotic control, gripper, perception strategy, generate important effects on a success of robotic grasp since the robotic grasping is a systemic work. That is, any factor may lead to a failure grasp. The robotic grasping actions rely on the feedback from perception prediction. If objects to be grasped are put in occluded scenarios, the robot has to use a complex motion planning to avoid obstacles, which results in that motion planning becomes the main factor affecting a grasping success. To focus on that the proposed pose estimator can be used in robotic grasps, we just implement the robotic grasping experiments in cluttered scenarios. Based on the inverse kinematics, the robot can grasp the object by the proposed pose estimator predicting the object's pose. Here we introduce the grasp planning briefly. The robot arrives at the defined pregrasp position with around 100 mm before the final grasp. From here, the robot approaches straightforward until the final grasp pose is reached. The gripper is clamped and the robotic-arm lifts the object. We perform the grasping task with each object for 10 robot trials for a total of 60 trials, as illustrated in Fig. 8. Our pose estimator performs equally well for grasping texture-less and textured objects with different shapes. For the object set, we achieved a grasp success rate of 98.33% (59/60). One failure is usually caused by occasional deviations during the execution of grasping due to slipping.

To illustrate the importance of the pose estimation accuracy for robotic grasping, we set up the experiments of robot grasping the canned tomato can and the soya milk carton box on purpose. Since the widths of these two objects are almost the same as the maximum opening value of the gripper. If the difference between the practical result and predicted pose is not small enough, the grasp tends to be failed due to some collisions. From the experimental result, our method can obtain an accurate pose to the robot such as the robot grasps objects with the size almost equal to the maximum opening value of the gripper. A key potential application is that the proposed method can be used in robotic assembling tasks based on an accurate pose estimation. Moreover, a perception circle approximately costs average 1 s and its far from real time. However, such perception speed can cope with most of robotic grasping cases.

IV. Conclusion

We developed a learning-based method of estimating 6-D pose of everyday objects in crowded scenes for task-oriented grasps by designing a novel Hough forest structure. Extensive evaluation experiments on challenging public data sets depicting realistic scenarios are performed by comparing with several recent approaches. The experiments on data sets illustrate that the proposed estimator has a good performance in estimating object's pose in cluttered and occluded environments. Moreover, the robotic grasping experiments demonstrate that our pose estimator can be used in practical environments. The limitation in this work is that this pose estimator cannot be applied to real-time applications. In future, for real-time use we will try to improve the pose estimation speed.

References

[1] S. Hinterstoisser *et al.*, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *Proc. Asian Conf. Comput. Vis.*, 2012, pp. 548–562.

[2] E. Brachmann, A. Krull, F. Michel, S. Gumhold, J. Shotton, and C. Rother, "Learning 6D object pose estimation using 3D object coordinates," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 536–551.

[3] S. Hinterstoisser *et al.*, "Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 858–865.

[4] H. Dong, D. K. Prasad, Q. Yuan, J. Zhou, E. Asadi, and I.-M. Chen, "Efficient pose estimation from single RGB-D image via Hough forest with auto-context," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 7201–7206.

[5] H. Dong, E. Asadi, G. Sun, D. K. Prasad, and I.-M. Chen, "Real-time robotic manipulation of cylindrical objects in dynamic scenarios through elliptic shape primitives," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 95–113, Feb. 2019.

[6] A. Doumanoglou, R. Kouskouridas, S. Malassiotis, and T.-K. Kim, "Recovering 6D object pose and predicting next-best-view in the crowd," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3583–3592.

[7] Y. Hu, J. Hugonot, P. Fua, and M. Salzmann, "Segmentation-driven 6D object pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3385–3394.

[8] F. Manhardt *et al.*, "Explaining the ambiguity of object detection and 6D pose from visual data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6841–6850.

[9] H. Dong, G. Sun, W.-C. Pang, E. Asadi, D. K. Prasad, and I.-M. Chen, "Fast ellipse detection via gradient information for robotic manipulation of cylindrical objects," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 2754–2761, Oct. 2018.

[10] R. Rios-Cabrera and T. Tuytelaars, "Discriminatively trained templates for 3D object detection: A real time scalable approach," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2048–2055.

[11] A. Tejani, R. Kouskouridas, A. Doumanoglou, D. Tang, and T.-K. Kim, "Latent-class Hough forests for 6 DoF object pose estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 119–132, Jan. 2018.

[12] P. Kontschieder, S. R. Bulò, A. Criminisi, P. Kohli, M. Pelillo, and H. Bischof, "Context-sensitive decision forests for object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 431–439.

[13] E. Brachmann, F. Michel, A. Krull, M. Y. Yang, S. Gumhold, and C. Rother, "Uncertainty-driven 6D pose estimation of objects and scenes from a single RGB image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3364–3372.

[14] A. Tejani, R. Kouskouridas, A. Doumanoglou, D. Tang, and T.-K. Kim, "Latent-class Hough forests for 6 DoF object pose estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 119–132, Jan. 2018.

[15] G. Fanelli, J. Gall, and L. Van Gool, "Real time head pose estimation with random regression forests," in *Proc. CVPR*, Jun. 2011, pp. 617–624.

[16] C. Choi and H. I. Christensen, "3D pose estimation of daily objects using an RGB-D camera," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 3342–3349.

[17] S. Ren, X. Cao, Y. Wei, and J. Sun, "Global refinement of random forest," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 723–730.

[18] D. Tang, Y. Liu, and T.-K. Kim, "Fast pedestrian detection by cascaded random forest with dominant orientation templates," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 1–11.

[19] H. Zhang and Q. Cao, "Texture-less object detection and 6D pose estimation in RGB-D images," *Robot. Auton. Syst.*, vol. 95, pp. 64–79, Sep. 2017.

[20] J. Shotton, B. Glocker, C. Zach, S. Izadi, A. Criminisi, and A. Fitzgibbon, "Scene coordinate regression forests for camera relocalization in RGB-D images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2930–2937.

[21] H. Dong, E. Asadi, C. Qiu, J. Dai, and I.-M. Chen, "Geometric design optimization of an under-actuated tendon-driven robotic gripper," *Robot. Comput.-Integr. Manuf.*, vol. 50, pp. 80–89, Apr. 2018.

[22] H. Dong, E. Asadi, C. Qiu, J. Dai, and I.-M. Chen, "Grasp analysis and optimal design of robotic fingertip for two tendon-driven fingers," *Mechanism Mach. Theory*, vol. 130, pp. 447–462, Dec. 2018.

Huixu Dong (Member, IEEE) received the B.Sc. degree in mechatronics engineering from the Harbin Institute of Technology, Harbin, China, in 2013, and the Ph.D. degree from the Robotics Research Centre, Nanyang Technological University, Singapore, in 2018.

He is currently a Post-Doctoral Fellow with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. His research interests include robotic perception and grasp in unstructured environments, robot-oriented image processing, computer vision and robot-oriented artificial intelligence, the navigation of mobile robot, and optimal design of robotic gripper.

Dilip K. Prasad (Senior Member, IEEE) received the B.Tech. degree in computer science and engineering from IIT (ISM) Dhanbad, Dhanbad, India, in 2003, and the Ph.D. degree in computer science and engineering from Nanyang Technological University, Singapore, in 2013.

He is currently an Associate Professor with The Arctic University of Norway, Tromsø, Norway. He has authored over 60 internationally peer-reviewed research articles. His research interests include image processing, pattern recognition, computer vision, and machine learning.

I-Ming Chen (Fellow, IEEE) received the B.S. degree from National Taiwan University, Taipei, Taiwan, in 1986, and the M.S. and Ph.D. degrees from the California Institute of Technology, Pasadena, CA, USA, in 1989 and 1994, respectively.

He is currently a Full Professor with the School of Mechanical and Aerospace Engineering, directions of the Robotics Research Centre and the Intelligent System Centre, Nanyang Technological University, Singapore.

Dr. Chen is a fellow of the ASME and the General Chairman of the 2017 IEEE International Conference on Robotics and Automation (ICRA2017). He is a Senior Editor of the IEEE TRANSACTIONS ON ROBOTICS and also the Editor-in-Chief of the IEEE TRANSACTIONS ON MECHATRONICS. He also acts as the Deputy Program Manager of the A*STAR SERC Industrial Robotics Program to coordinate project and activities under this multi-institutional program involving NTU, NUS, SIMTech, A*STAR I2R, and SUTD. He works on many different topics in robotics, such as mechanism, actuators, human–robot interaction, perception and grasp, and industrial automation.