Supplementary Material: Depth-aware Object Segmentation and Grasp Detection for Robotic Picking Tasks

Stefan Ainetter[1]
stefan.ainetter@icg.tugraz.at
Christoph Böhm[2]
christoph.boehm@ieee.org
Rohit Dhakate[2]
rohit.dhakate@ieee.org
Stephan Weiss[2]
stephan.weiss@ieee.org
Friedrich Fraundorfer[1]
fraundorfer@icg.tugraz.at

[1] Institute of Computer Graphics & Vision
Graz University of Technology
Graz, Austria

[2] Institute of Smart System Technologies
University of Klagenfurt
Klagenfurt, Austria

Additional Analysis of Depth-aware CoordConv

In addition to the results in Section 5.1, we report the performance using additional methods to calculate feature maps for depth-aware CoordConv. Hereafter, we describe other configurations for depth-aware CoordConv, resulting in different feature maps.

Depth similarity map $\mathbf{D}_{sim}$. It is based on the idea of [8] to calculate the similarity of the depth image related to a specific point (in our case the point proposal $p(x,y)$). The depth similarity $\mathbf{D}_{sim}$ is defined as

$$\begin{aligned} \mathbf{D}_{sim} &= \exp(\beta \cdot |\mathbf{D} - \mathbf{D}(p)|) - 1, \\ \mathbf{D}_{sim} &\in [0, \exp(\beta) - 1], \end{aligned} \tag{1}$$

with the normalized depth image $\mathbf{D} \in [0,1]$, the point proposal p and a scaling factor β.

HHA map $\mathbf{H}_{dist}$. HHA encoding [2] represents the depth image by three channels (horizontal disparity, height above ground, and angle with gravity). Fore each channel $c \in \{1,2,3\}$ of the HHA encoding $\mathbf{H}$ we calculate a distance map as

$$\begin{aligned} \mathbf{H}^c_{dist} &= \alpha \cdot (\mathbf{H}^c - \mathbf{H}^c(p)), \\ \mathbf{H}^c_{dist} &\in [-\alpha, \alpha], \end{aligned} \tag{2}$$

whereas $\mathbf{H}^c \in [0,1]$ is the c-th normalized channel of the HHA encoding.

Table 1 shows the results for the additional experiments, whereas the setup of the experiments is identical to the one in Section 5.1. As reported in the main paper, using the depth-distance map $\mathbf{D}_{dist}$ together with the 2.5D distance map $\mathbf{F}_{2.5D}$ achieves the highest

Method	Rel. Coord-Conv [7]	Depth-dist. Map	2.5D Dist. Map	Depth-sim. Map	HHA Map	Inst. Seg. IOU
Backbone + Seg. Branch	✗	✗	✗	✗	✗	83.01
Backbone + Seg. Branch	✓	✗	✗	✗	✗	85.63
Backbone + Seg. Branch	✓	✓	✓	✗	✗	**91.27**
Backbone + Seg. Branch	✓	✗	✓(only $\mathbf{X}_{rel}$,$\mathbf{Y}_{rel}$)	✓	✗	90.91
Backbone + Seg. Branch	✓	✗	✓(only $\mathbf{X}_{rel}$,$\mathbf{Y}_{rel}$)	✗	✓	89.68

Table 1: Comparison of different configurations of depth-aware CoordConv for instance segmentation on Siléane dataset. Setup and execution of experiments are identical to the ones in Section 5.1 in the main paper.

accuracy. Using $\mathbf{D}_{sim}$ instead of $\mathbf{D}_{dist}$ performs reasonable well, whereas the disadvantage of $\mathbf{D}_{sim}$ is that this similarity measure is not able to differentiate between positive and negative distances in z-direction, which results in an information loss. The HHA distance maps $\mathbf{H}_{dist}$ has the limitation that the HHA encoding already encode properties of geocentric pose, which leads to ambiguities when relating it to the point proposal p.

Detailed Information about Loss Function

For simultaneously learning the tasks of grasp detection, semantic segmentation and instance selection, we defined the composite loss function as

$$\mathcal{L} = \lambda_{grasp}\mathcal{L}_{grasp} + \lambda_{sem}\mathcal{L}_{sem} + \lambda_{inst}\mathcal{L}_{inst}, \tag{3}$$

with the grasp detection loss $\mathcal{L}_{grasp}$, the semantic segmentation loss $\mathcal{L}_{sem}$, and the instance selection loss $\mathcal{L}_{inst}$. All parts are weighted with a specific hyperparameter λ.

Grasp detection loss. The grasp detection loss $\mathcal{L}_{grasp}$ is defined as

$$\mathcal{L}_{grasp} = \mathcal{L}_{RPN} + \mathcal{L}_{box} + \mathcal{L}_{rot}, \tag{4}$$

where $\mathcal{L}_{RPN}$ defines the loss for training a Region Proposal Network (RPN), which is part of the grasp detection branch. $\mathcal{L}_{box}$ defines the regression loss for the box coordinates (x, y, w, h) and $\mathcal{L}_{rot}$ defines the classification loss for the grasp orientation θ. We refer to [3] for additional information about the RPN and corresponding loss $\mathcal{L}_{RPN}$. The grasp orientation loss $\mathcal{L}_{rot}$ is defined as

$$\mathcal{L}_{rot} = -\frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}_+}\log s_r^{c_\theta} - \frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}_-}\log s_r^{c_\emptyset}. \tag{5}$$

Note that $\mathcal{R} = \mathcal{R}_+ \cup \mathcal{R}_-$ is the set of valid and invalid region proposals, which are the output of the RPN. Each region proposal r consists of the parameters (x_r, y_r, w_r, h_r), which represent an initial axis-aligned bounding box. The score function $s_r^{c_\theta}$ defines the probability that the region proposal belongs to the ground truth orientation class c_θ, and $s_r^{\emptyset}$ defines the probability that the region proposal is invalid. Note that we discretized the grasp orientation θ into 18

intervals with equal length, where each interval is represented by its mean value, resulting in $c_\theta \in \{1,...,18\}$ for the orientation classes. The additional class $c_\emptyset$ is used to describe the region proposals which may be invalid.
For bounding box regression we use the loss $\mathcal{L}_{box}$ defined as

$$\mathcal{L}_{box} = \sum_{i \in \{x,y,w,h\}} smooth_{L_1}(t_i - t_i^*), \tag{6}$$

with the $smooth_{L_1}$ norm defined in [6]. The correction factors t_i are calculated by the grasp detection head, and t_i^* represent the offset between ground truth grasp candidates and a region proposal r. The correction factors t_i and the corresponding region proposal parameters (x_r, y_r, w_r, h_r), are then used to calculate the final box parameters (x, y, w, h).
Semantic segmentation loss. We denote $l = \{1,...,N\}$ as the set of semantic segmentation classes, with $N = 2$ for foreground/background segmentation. The semantic segmentation loss is a weighted per-pixel loss [5] defined as

$$\mathcal{L}_{sem} = -\sum_{j,k} w_{j,k} \log P_{j,k}(Y_{j,k}), \tag{7}$$

where (j,k) correspond to the pixel position in the image. Let $Y_{j,k} \in l$ be the semantic segmentation ground truth and $P_{j,k}$, the predicted probability of the semantic segmentation head for the same pixel, to be assigned to one of the semantic classes, respectively. The weights $w_{j,k}$ select the 25% of the lowest predicted probabilities $P_{j,k}$ for all (j,k) using $w_{j,k} = \frac{4}{WH}$, and $w_{j,k} = 0$ otherwise, whereas $(W \times H)$ is the spatial image resolution.
Instance segmentation loss. The instance segmentation is defined as binary segmentation problem, given that we want to calculate a specific object instance mask related to the point proposal $p(x,y)$. We use the Normalized Focal Loss as proposed in [7], which is defined as

$$\mathcal{L}_{inst} = -\frac{1}{Q(M)} \sum_{j,k} (1 - Q_{j,k})^\gamma \log Q_{j,k}, \tag{8}$$

whereas $Q(M) = \sum_{j,k}(1 - Q_{j,k})^\gamma$ defines the total weight of the values for all pixels in the image, with $Q_{j,k}$ the predicted probability of the instance selection network for the correct segmentation output at position (j,k), and the focusing parameter $\gamma \geq 0$. The Normalized Focal Loss concentrates on pixels that are misclassified by the network, in contrast to the binary cross entropy loss, which pays more attention to pixels that are correctly classified (as shown in [4]).

Additional Visualizations for OCID_grasp Experiment

Figure 1 shows comparison of qualitative results for [1] and (ours), corresponding to Section 5.2 in the paper. Please note the differences: 1) [1] performs semantic segmentation, where each color in the visualization corresponds to a certain object class, whereas multiple object instances of the same class share the same color. This makes it impossible to distinguish between multiple instances of the same class. (Ours) performs class-agnostic instance segmentation for multiple graspable objects, where each instance has a separate color, not related to classes. 2) Because [1] filters grasp candidates using the semantic segmentation, it is only possible to select one grasp candidate per class. Again, if multiple instance of one object class are in the scene, this algorithm fails to detect grasp candidates for more than

Figure 1: Comparison with previous state-of-the-art method for joint grasp detection and segmentation. As the method of [1] selects grasp candidates related to the semantic segmentation, they are missing out on grasp candidates if multiple instances of the same object class are present in the scene (e.g. for classes banana and orange in row one, column two). Furthermore, the semantic segmentation in [1] makes it impossible to distinguish between multiple object instances of the same class, whereas (ours) performs class-agnostic instance segmentation. Failure cases for (ours) are if no grasp candidate is detected for a graspable object, see dark bowl in second row, fourth column. Note that the colors for the seg. results in [1] and (ours) are independent and have no correlation.

one instance. (Ours) on the other hand, is free of restrictions regarding object classes, and is therefore able to predict grasp candidates for multiple instances of the same class. Failure cases for our method are if no grasp candidate is predicted for a graspable object (as one can see in Figure 1 second row, fourth column, no grasp candidate for bowl detected), which then results in no instance segmentation mask, as the center of the grasp candidate is used as point proposal for instance segmentation.

Technical Details for Real-World Robotic Picking

The system used for the real-world robotic picking experiments is a scaled hydraulic forest crane (see left-hand drawing in Figure 2). These cranes usually are human-operated to move tree logs from one place to another. To allow for automatic pick-and-place operation, we added electrically operated hydraulic valves to each joint of the crane in combination with a programmable logic controller (PLC). Linear or angular displacement sensors with electrically operated hydraulic valves make the automatic control of the crane possible.

The right-hand diagram of Figure 2 shows the overall structure of the system. The top left block includes the detection of feasible grasp candidates (see method in Section 4 in main paper). The best grasp candidate is projected into 3D and sent to the so-called motion sequencer client (Figure 2 top right) through a ROS message. This message contains the grasping 3D position, its orientation, the log's diameter and centroid. With this data (transformed from detection to the kinematics frame) and the periodically arriving sensor

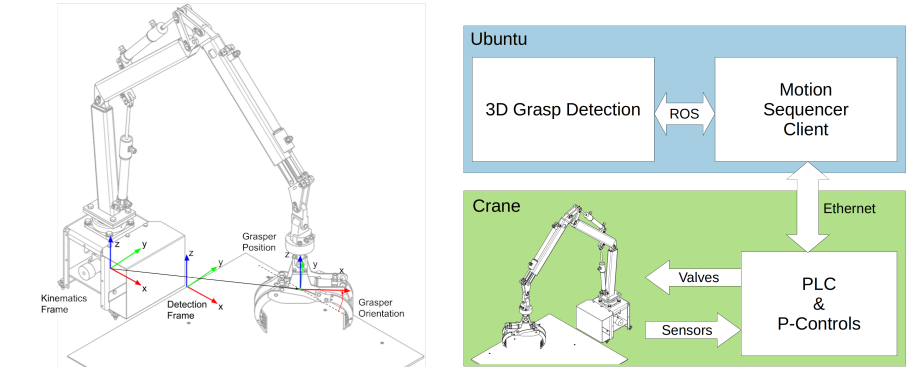

Figure 2: The drawing, left image, shows the crane used in real-world experiments and important reference frames; An overview of the system structure can be seen in the right image.

measurements from the PLC, the motion sequencer client can calculate the configuration of the joints needed to reach the requested point. A minimization method uses forward kinematics of the crane and the desired position and orientation to calculate the required joint configuration.

The PLC receives the calculated joint configuration via Ethernet. On the PLC, each joint has a proportional control loop with the received configuration as desired position/orientation, the measurements as current position/orientation, and the valve opening as output. The control loop interpolates linearly between the old and new desired setpoint, to avoid sudden motions and unwanted pressure spikes during setpoint changes. The motion sequencer client checks through the reported configuration if it matches the new setpoint. One needs to be aware that the hydraulics combined with a proportional control loop limits the precision in positioning, compared to just electrical joint-based systems.

Always heading directly for the detected grasping pose might result in erroneous behavior or worst-case crashes of the crane with its surroundings. Therefore, the motion sequencer client generates a sequence of so-called waypoints, based on the received grasp candidate, to ensure repeatability and safety. This sequence guarantees that side-ways motions are only done while high above the ground and only going down to pick up a log or put it into the container. The grasper itself limits grasping complexity as it is bulky because of its application, compared to other pick-and-place robotic manipulators. Thus, one needs to take care of the distance between logs. The grasper only opens as wide as the log's diameter plus safety margin to improve grasping performance in cluttered scenarios. A ROS service on the grasping detection, which triggers a prediction, allows the automation of the experiment. The motion sequencer client calls this service to start a new sequence. This procedure continues until no grasp candidates are detected.

References

[1] Stefan Ainetter and Friedrich Fraundorfer. End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from rgb. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 13452–13458, 2021.

[2] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *Proceedings of the European Conf. on Computer Vision (ECCV)*, pages 345–360. Springer, 2014.

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2961–2969, 2017.

[4] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE Int. Conf. on computer vision (ICCV)*, pages 2980–2988, 2017.

[5] Lorenzo Porzi, Samuel Rota Bulo, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 8277–8286, 2019.

[6] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.

[7] Konstantin Sofiiuk, Olga Barinova, and Anton Konushin. Adaptis: Adaptive instance selection network. In *Proceedings of the IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pages 7355–7363, 2019.

[8] Weiyue Wang and Ulrich Neumann. Depth-aware cnn for rgb-d segmentation. In *Proceedings of the European Conf. on Computer Vision (ECCV)*, pages 135–150, 2018.