13th IEEE International Conference on Automation Science and Engineering (CASE), Xi'an, China, August 2017.

Robolink Feeder: Reconfigurable Bin-Picking and Feeding with a Lightweight Cable-Driven Manipulator

Seongyong Koo[1], Grzegorz Ficht[1], Germán Martín García[1],
Dmytro Pavlichenko[1], Martin Raak[2], and Sven Behnke[1]

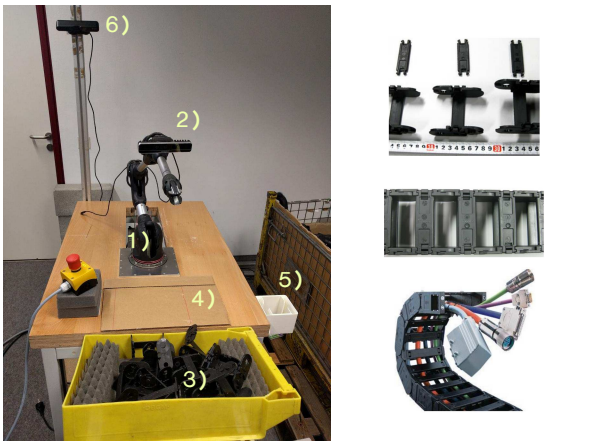
***Abstract*— The feeding of parts from bulk supply to production lines is a common but still challenging task in industrial automation. In this paper, we present a part feeding system that has two objectives: first, it needs to be easy reconfigurable to handle new part variants, and second, it should be safe for collaboration with human workers. We propose a reconfigurable vision system that finds generic graspable parts and robustly estimates 6D pose given CAD data with a learning method based on synthetic data. To guarantee that the system is human safe, we use a cable-driven lightweight igus-robolink® WR manipulator that can move at high velocities without risk of harming human workers. We improve the safety by detecting persons entering the workspace of the arm and quickly stopping its movement. We demonstrate in our experiments a high success rate in feeding industrial parts with safe human collaboration.**

I. Introduction

Picking parts from bulk supply and feeding them to production lines are common tasks in industrial automation [1], [2]. Existing automation solutions to part feeding, such as vibratory bowl feeders and robotic pickers, require specialized engineering for each product variant [3].

We present a solution for the automatic bin-picking and feeding of textureless parts such as the ones shown in Fig. 1b. These, so called, chain links are assembled to energy chains for guiding cables as shown in the bottom of Fig. 1b. The parts can have different sizes and variations of the shape in a total of 63 possible configurations. This variability has prevented until now the automation of the feeding of parts for their assembly into chain links.

Our solution[3] addressees two requirements: 1) it is easily reconfigurable for new part variants, and 2) it is safe for the collaboration with human workers. An overview of the system is given in Fig. 1a: a cable-driven 6 DOF robolink® WR manipulator[4] picks parts from a bin (from 3 to 4 in Fig. 1a), estimates their pose and feeds them in the slot with a specific orientation (from 4 to 5 in Fig. 1a). The parts are sensed by means of an Intel RealSense SR300 RGB-D camera mounted on the wrist of the arm (2) in Fig. 1a). Furthermore, a workspace camera, ASUS Xtion Pro, (6) in Fig. 1a) allows us to detect humans entering the workspace in order to halt the motion of the arm. This improves the safety of human workers together with the intrinsic collision safety feature of the lightweight compliant cable-driven robolink® manipulator.



(a) (b)

Fig. 1: (a): 1) The robolink manipulator. 2) The SR300 RGB-D camera mounted on the arm. 3) A pile of chain link parts. 4) The surface where the parts are placed to estimate their pose. 5) The part feeder. 6) Asus Xtion RGB-D workspace camera.
(b) From top to bottom: different variants of the chain link parts; the assembled energy chain; and the energy chain in its use for conducting cables.

A key component of the easy reconfiguration is an adaptive vision system that can detect new part variants and estimate their 6D poses. Precisely estimating the 6D pose of the part is crucial in order to correctly place it in the feeding slot. Several approaches have been proposed in the literature for estimating the pose of objects. Given the CAD model of a new part, one can apply iterative 3D registration methods [4] when an initial guess of the pose is available. Feature-based initial alignment methods [5] can be used for this purpose; however, they are not robust against noisy and partial measurements and require specific hand-crafted features. Another approach is generating a synthetic dataset by rendering the CAD model from multiple view poses, and estimating the pose of the object given 2D camera measurements. For example, Liu *et al.* [6] generated rendered depth edge maps for each pose, based on which a template matching algorithm was used to find the best-matching pose. However, this

[1]Authors are with the Autonomous Intelligent Systems (AIS) Group, Computer Science Institute VI, University of Bonn, Germany, {koosy, ficht, martin, pavlichenko, behnke}@ais.uni-bonn.de

[2]Martin Raak is with igus® GmbH, Spicher Str. 1a, 51147, Köln, Germany, mraak@igus.de

[3]This system was developed as one of the solutions for the European Robotics Challenge (EuRoC), Challenge 1. Reconfigurable Interactive Manufacturing Cell. http://www.euroc-project.eu

[4]http://www.igus.de/wpck/6076/robolink

approach requires a specially designed multi-flash camera to extract good edge features and exhaustive template search due to the different depth-edge silhouettes in different poses.

Instead, we propose a deep neural network (DNN) to estimate the 6D pose of the part. Since CAD models of the part variants are available, we produce a synthetically rendered dataset in order to train the network. This end-to-end approach does not need manually designed features nor distance functions which are critical to target objects in the aforementioned methods. Therefore, the proposed method reduces reconfiguration time with minimal engineering effort.

II. Related Work

Since the related literature spans two different fields, we first present the related work on applications of cable-driven manipulators, and second, related approaches on bin-picking as well as learning-based object detection and pose estimation methods.

A. Cable-driven Manipulators

The WAM arm pioneered the research on cable-driven robotic manipulators, and was commercially launched by Barret Technology [7]. An inherent characteristic of cable-driven robotic arms is that they can be backdriven, and thus, a common application of this technology is in medical rehabilitation [8].

Quigley *et al.* [9] present a 6 DOF arm with a human-safe design. The shoulder of the arm is driven by three stepper motors, which have cables attached to perfom the movement of the joints. This ensures safety for a human. The distal three joints are actuated by compact Dynamixel servos. So, the arm is controlled by motors of mixed type. To demonstrate the capabilities of the arm, authors used the arm to play chess via teleoperation and to bake pancakes using a tool.

Whitney and Hodgins [10] describe a passively safe and gravity-counterbalanced anthropomorphic robot arm with joints controlled via cables by stepper motors. The shoulder design is done in a way that the motors are completely grounded, and hence, the counterbalancing weights are placed away from the shoulder as well.

B. Bin Picking Methods

On the topic of bin picking, one can distinguish methods that aim at finding graspable parts of the object and those that try to estimate the pose of the object in the pile. An example of the first category is the work presented by Domae *et al.* [11]. The authors propose a method for finding graspable surfaces of textureless objects in unordered piles. For a given gripper, they define two convolutional filters to find collision-free graspable points in depth maps. Liu *et al.* [6] use fast directional chamfer matching [3] to find the pose of objects in random piles on depth edge images obtained from a multi-flash camera.

In the second category, Buchholz *et al.* [12] use point-pair features and a random sampling algorithm for locating objects in bins using depth data. They use CAD models of the objects. Rodrigues *et al.* [13] propose a method for estimating the pose of texture-less shiny objects in bin-picking scenarios. They construct a multi-light system that produces images where colour changes encode changes in surface orientation. In their work, a voting scheme is designed where small image patches vote for the pose of the object. Holz *et al.* [1] estimate the refined part pose based on the initially detected object candidate. A workspace camera was used to detect multiple object candidates, and a wrist camera then recognizes and localizes the part to grasp.

C. Deep Learning for Grasping Objects

Levine *et al.* [14] propose a method that uses a deep network to learn motion commands to a robotic arm directly from image data. In their setup, the bin contains coloured objects of different categories, and the network controls the movement of the arm in order to grasp an object with multiple sequential images. In our case we have no distinctive colour features since all the parts are black and use a single image to detect grasping pose. Redmon and Angelova [15] propose a method for detecting grasping poses in RGB-D images. The authors assume that a single object is present in the input image. This is not applicable in our case where we have to deal with a random pile of untextured objects. Lenz *et al.* [16] propose a two-stage network for detecting grasps in images containing multiple objects. In their approach, the first stage of the network—swallow and fast—is in charge of ranking candidate grasps in an exhaustive-search. The second stage taking these filtered candidates is able to find the optimal grasp for each object by means of a deeper architecture. Pinto and Gupta [17] propose a CNN to predict whether a grasp is suitable for an image patch at 18 possible orientations. They collect a dataset with 50 thousand trials and errors of object grasps which they use to train their network. Schwarz *et al.* [18] develop a deep-learning approach that combines object detection and semantic segmentation in the manipulation scenes captured with RGB-D cameras. The combined components complement each other and yield reliable perception for the bin-picking in clutter.

III. System Description

A. Hardware

For our feeding demonstration we have used the robolink® arm produced by igus GmbH, which has two main components: customisable manipulator and motor unit. The motor unit houses six stepper motors (M1 – M6 in Fig. 3) with 1:16 reduction gearing to which pulley disks are mounted. The manipulator is constructed out of thin aluminum tubes connected with high-grade plastic mechanisms which act as joints. The order and type of joints as well as length of the tubes can be exchanged, which yields a versatile solution. The arm is actuated at its base, where the stepper motors are located. The cables are routed through the tubing and connect motor pulleys with their respective joints. Motion is transferred through cables to the joint by rotating one of the pulleys, which in turn produces a rotation in the corresponding joint. The result of this arrangement is a lightweight, reconfigurable arm which is compliant and safe

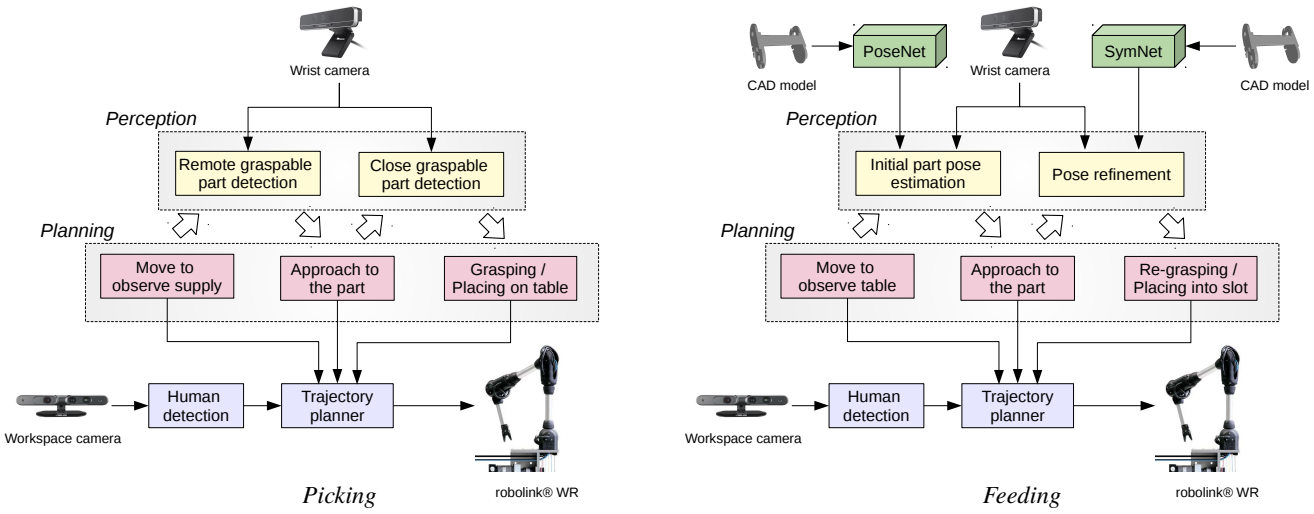


Fig. 2: Software system overview for the two tasks of the proposed robolink® feeder.

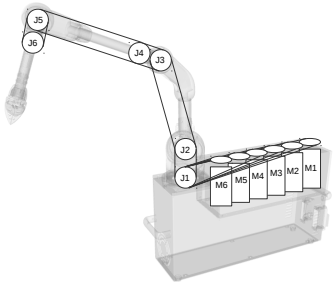


Fig. 3: Schematic of the igus-robolink® WR manipulator: joints J1 to J6 and their corresponding actuators M1 to M6.

to interact with. In our setup, we have used a configuration with six degrees of freedom, assigning yaw and pitch axes with three of them each.

B. Active Perception System

The robolink® feeder is capable of singularizing parts from a bin (*picking*) and placing them into a slot (*feeding*). Fig. 2 shows the workflow of the software modules for the two tasks. The main sensor of the system is a short-range RGB-D camera (Intel RealSense SR300) mounted on the robot wrist, which is able to actively observe the scene at different distances and perspectives. We implemented a coarse-to-fine search strategy that moves the camera close to objects for both tasks.

In the *picking* task, the manipulator first moves to a predefined pose to observe all parts in the bin. With the depth measurements from the wrist camera, multiple graspable parts are detected by a local depth minima detection method, and the highest one is selected as the most graspable part. In order to grasp the part, the robot approaches the selected graspable part and detects the part with the wrist camera again, by which the pose of the graspable part can be estimated more accurately in higher resolution. Then, the robot is able to move to the part, grasp and pick it up, and place it on the table.

The *feeding* task requires reorienting the part so that its pose fits to the slot of the chain link assembly machine. The main perception task is estimating accurate 6D pose of the part on the table, with which the manipulator plans where to grasp, how to rotate, and where to place the part on the slot. In the same manner as the active perception in the *picking* task, accurate part pose estimation is achieved using the wrist camera and robot movements as shown in Fig. 2b. First, the robot moves to the predefined observation pose and estimates initial part pose using *PoseNet* which is pretrained offline given the CAD model of the part. Then the robot moves the camera above the part and refines the pose with higher resolution depth measurements of the part. Here, another pre-trained neural network, *SymNet*, is used to identify ambiguous symmetric poses and correct the estimated pose.

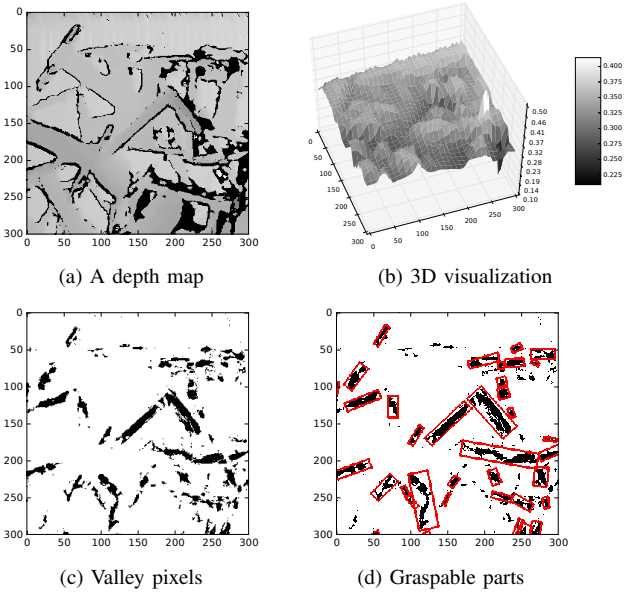
Another RGB-D camera (ASUS Xtion Pro) covers the workspace of the arm. When a human worker intervenes in the robolink® feeder's tasks, the human movement is detected by the workspace camera, and the robot stops moving until the person exits the workspace. The stopping and resuming robot movements is controlled by a trajectory planner, which generates joint trajectories given a task and sends the reference commands to the robot controller.

IV. Perception System for Picking

The perception system for the *picking* task aims at finding multiple graspable parts by the gripper, which is fast, applicable to texture-less objects in arbitrary shapes, and invariant to the position and orientation of the camera.

A. Graspable Part Detection

A two-finger gripper can grasp a sticking out area which is closer to the wrist camera than the neighborhood of that area. As shown in Fig. 4b, the sticking out parts form valleys of 3D surfaces in the depth map. The graspable area of the valley is the part whose width is less than the opening of the gripper.



(a) A depth map (b) 3D visualization

(c) Valley pixels (d) Graspable parts

Fig. 4: Finding graspable parts from a depth map.

Similar to the skeletonization algorithm in [19], we propose an operator to find pixels that belong to the valley whose width is smaller than w. Given the maximum width (w), the operator traverses each depth pixel $d(u,v)$ and observes eight boundary pixels $d(u\pm s,v), d(u,v\pm s), d(u\pm s,v\pm s)$ with the depth-dependent pixel size s corresponding to w,

$$s = \frac{w}{\|p_1 - p_2\|}, \tag{1}$$

$$p_1 = deproj(u,v,d(u,v)), \tag{2}$$

$$p_2 = deproj(u+1,v,d(u,v)), \tag{3}$$

where $deproj(u,v,d)$ yields a de-projected 3D point of a pixel (u,v) in the depth image with the depth value d.

For all valid pixels (which are non-black pixels in Fig. 4a), the operator computes the number $c(u,v)$ of neighboring pixels having an equal or smaller depth-level. Fig. 4c shows pixels which have $c(u,v) < 0$. This means that there are no pixels that have smaller depth values than the central pixel in 8-directions within the area of $\{(i,j) | \|(i,j)-(u,v)\| < s\}$, which corresponds to the part of a valley with the width of w. Multiple graspable parts are then computed by region growth clustering of the connected local depth minima, and described by rotated bounding boxes $\mathcal{B} = \{b_k\}$ of clusters $\mathcal{C} = \{c_k\}$ (depicted as red rectangles in Fig. 4d):

$$c_k = \{(\mathbf{u},\mathbf{v},\mathbf{d}) | max(u_i - u_{i+1}, v_i - v_{i+1}) < 1\}, \tag{4}$$

$$b_k = minRect(c_k) \tag{5}$$

$$= \{x_k, y_k, w_k, h_k, \theta_k\}. \tag{6}$$

B. Grasping pose estimation

To obtain a reliable grasping pose of the most graspable part, first the robot observes all parts at the pre-defined observation pose, and selects the closest graspable part,

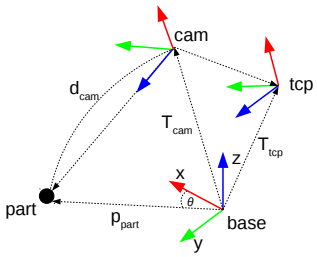


Fig. 5: Observation pose to locate the part at the center of the wrist camera view.

which has the minimum averaged depth of the cluster:

$$c_{min} = \arg\min_{c_k} \bar{\mathbf{d}}, \quad \mathbf{d} \in c_k. \tag{7}$$

Next, the robot moves towards the selected part so that the wrist camera can observe the part in detail. Based on the 3D position of the part $p_{part} = \{p_x, p_y, p_z\}$, the camera tilt angle ϕ, and distance from the part d_{cam}, the target gripper pose T_{tcp} is calculated as follows:

$$\theta = \arctan\frac{p_y}{p_x}, \tag{8}$$

$$T_{cam} = T_{rpy}(p_x, p_y, p_z, \phi, 0, \theta - \pi/2), \tag{9}$$

$$\cdot\, T_{rpy}(0,0,-d_{cam},0,0,0), \tag{10}$$

$$T_{tcp} = T_{cam} \cdot T_{tcp}^{cam}, \tag{11}$$

where T_{rpy} is a homogeneous transformation matrix given 3D positions and roll, pitch, yaw angles.

After moving to T_{tcp}, the robot detects multiple graspable parts again, and selects the centered graspable part (c_{grap}, b_{grasp}). Then, a 3D point cloud of the graspable part $\mathcal{P}$ can be obtained by de-projecting the pixels of the selected cluster:

$$\mathcal{P} = deproj(\mathbf{u},\mathbf{v},\mathbf{d}), \quad (\mathbf{u},\mathbf{v},\mathbf{d}) \in c_{grasp}. \tag{12}$$

The final grasping pose T_{grasp} is calculated using three eigenvectors $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ of the covariance matrix of $\mathcal{P}$:

$$T_{grasp} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 & p_{part} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{13}$$

V. Perception System for Feeding

Feeding a part into the slot requires an accurate estimation of its pose. We propose a learning-based active pose estimation method, which first estimates an initial pose by *PoseNet*, then moves the wrist camera close to the object, and refines the initial pose by checking symmetry of the shape using *SymNet*. The two deep neural networks (DNN) are easy reconfigurable due to the use of synthetic training data given a CAD model, which does not require manual annotations for labeling ground truth poses.

A. Network Architecture

Fig. 6 shows our network structures. *PoseNet* and *SymNet* have the same structure for training deep features with six convolutional and four pooling layers. This structure is

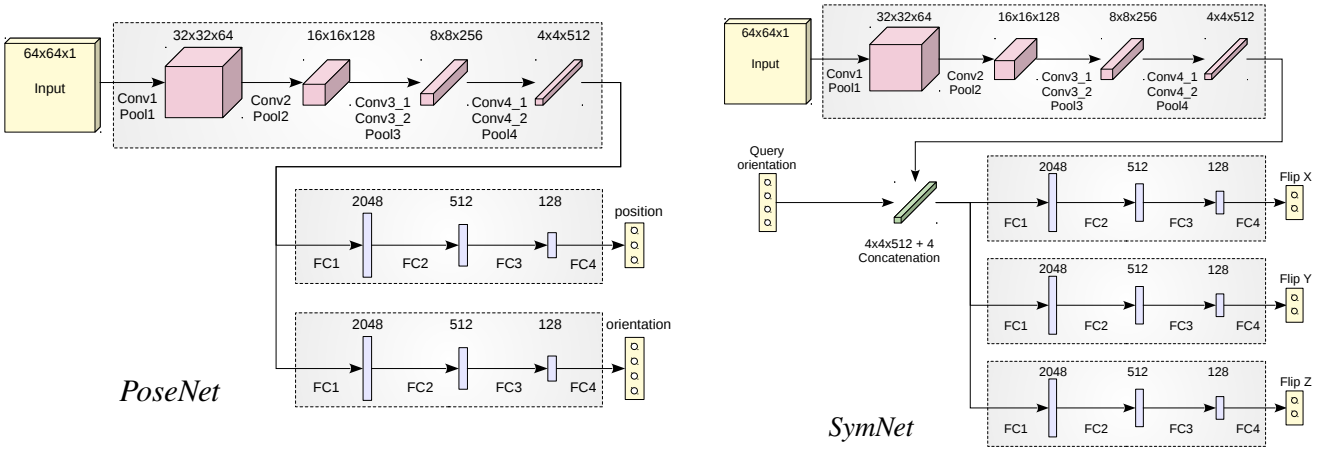


Fig. 6: Deep Neural Networks for estimating 6D part pose (*PoseNet*) and for defining symmetries (*SymNet*).

adopted from the first six weight layers of the VGG-11 model suggested in [20]. Each convolutional layer has 3×3 receptive field size and batch-normalization [21] is applied with ReLU activation function.

Following convolutional layers, *PoseNet* consists of two regression networks to produce 3D position and orientation of a part in the input depth map. Here, we represent 3D orientation as a quaternion with four variables. Each regression network is composed by three fully connected layers with batch-normalization and ReLU activation functions, and the last fully connected layer with linear activation.

If the given object shape is nearly symmetric, *PoseNet* could estimate an orientation that is flipped along the symmetry axis. One way to correct this is by flipping the orientation once the wrong estimation has been detected. *SymNet* aims to learn distinctive features that discriminate between depth images of symmetric shapes given the estimated orientation.

First, *SymNet* extracts the symmetry-sensitive features of the input image through the convolutional and pooling layers, and then concatenates the features with given query orientation. The following fully connected network for each symmetric axis estimates if the given orientation is opposite or not along the axis. Here, the fully connected layers all have batch-normalization and ReLU activation functions.

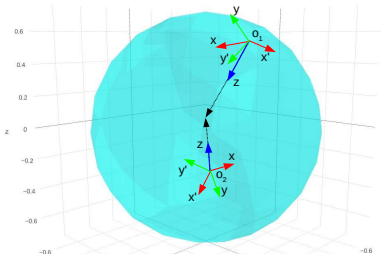


Fig. 7: Examples of rendering camera poses on the sphere.

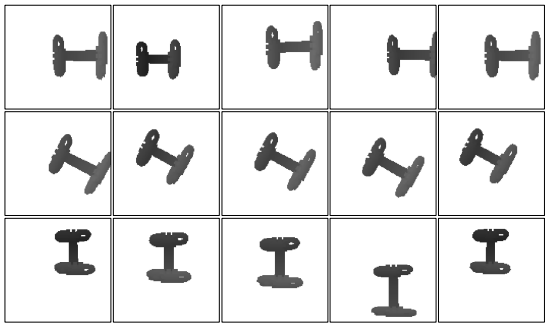


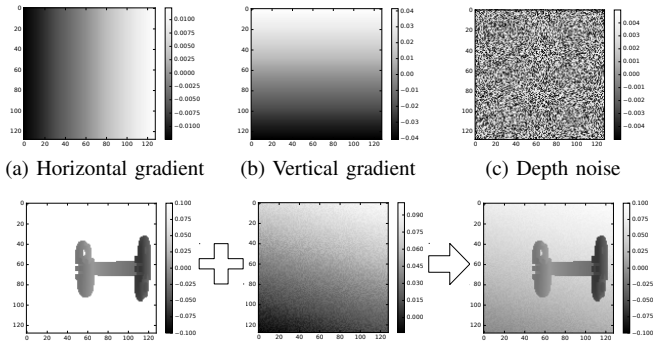
Fig. 8: Rendered depth map samples for five translations and three camera poses. Images at each row are rendered by the same camera pose with different translations.

B. Data Generation

Given a CAD model of the object, training depth images are generated by rendering the model with a number of different poses. Orientation variants are generated by allocating various camera poses on a sphere with 2-level subdivision of icosahedron, which has 162 vertexes as shown in Fig. 7. On each vertex, 16 camera poses are defined by rotating $\pi/8$ radian per step along the z axis toward the origin. By doing so, a total 2,592 orientations are regularly sampled.

For each camera pose, an initial depth map of the model is rendered, where the object is positioned at the image center. To generate translation variants, 16 pixels on the initial depth map are randomly sampled. We use the offset from the centre of the image to each sampled pixel to translate the initial depth map of the model. With the randomly sampled 16 translations for each camera pose, 41,472 poses were sampled in total. Among them, randomly sampled 33,178 poses are used for training, and the remaining 8,294 poses are used for validation. Fig. 8 shows rendered depth maps for five translations and three camera poses.

Each rendered depth map is combined with a synthesized background as shown in Fig. 9. The background is a combination of four sources: horizontal depth gradient (Fig. 9a), vertical depth gradient (Fig. 9b), uniform depth noise (Fig. 9c), and a constant depth offset. The max depth values of the horizontal and vertical gradients are randomly sampled within the given limit depth value. The fully random



(a) Horizontal gradient (b) Vertical gradient (c) Depth noise

(d) Synthesized depth by combining normalized foreground and the background composed by (a), (b), (c), and a randomly sampled constant depth.

Fig. 9: Online data augmentation process.

variables for the background composition are sampled for each image at each epoch of training. The on-line data augmentation prevents the network from over-fitting to a particular background data distribution.

For each epoch, the training set for *PoseNet* consists of 33,178 pairs of synthesized depth images $\mathbf{I}_{syn} \in \mathbb{R}^{64\times64}$ and ground truth poses (position $\mathbf{p} \in \mathbb{R}^3$ and quaternion $\mathbf{q} \in \mathbb{R}^4$) of the model with respect to the camera. Due to the duality of quaternion, we require the w value of all quaternions to be positive:

$$\mathcal{D}_{posenet} = \{P_1, ..., P_i, ..., P_{33178}\}, \tag{14}$$

$$P_i = \{\mathbf{I}_{syn}, \mathbf{p}, \mathbf{q}\}. \tag{15}$$

The total loss in the batch of training data is computed by the sum of L2 losses between *PoseNet* output and ground truth pose.

Training data for *SymNet* requires labeling for a proper query orientation $\mathbf{q}$. The output of the network $\mathbf{f}$ indicates whether the query orientation should be flipped for each axis:

$$\mathcal{D}_{symnet} = \{P_1, ..., P_i, ..., P_{33178\times4}\}, \tag{16}$$

$$P_i = \{\mathbf{I}_{syn}, \mathbf{q}, \mathbf{f}\}. \tag{17}$$

For a ground truth orientation q, we can create three false data pairs by flipping about each axis:

$$\mathbf{q} = \begin{cases} \mathbf{q} & \text{if } i = k \times 4 \\ flip_x(\mathbf{q}) & \text{if } i = k \times 4 + 1 \\ flip_y(\mathbf{q}) & \text{if } i = k \times 4 + 2 \\ flip_z(\mathbf{q}) & \text{if } i = k \times 4 + 3. \end{cases} \tag{18}$$

The output of *SymNet* is encoded as a one-hot vector for three axes:

$$\mathbf{f} = \begin{cases} (1,0),(1,0),(1,0) & \text{if } i = k \times 4 \\ (0,1),(1,0),(1,0) & \text{if } i = k \times 4 + 1 \\ (1,0),(0,1),(1,0) & \text{if } i = k \times 4 + 2 \\ (1,0),(1,0),(0,1) & \text{if } i = k \times 4 + 3. \end{cases} \tag{19}$$

Given the label data, the loss of the network is computed by the averaged sigmoid cross entropy in the batch of training data.

C. Part Pose Estimation

When a part is placed on the table by the *picking* task, the robot moves to a pre-defined table observation pose. First, the object is detected by using table top segmentation and Euclidean clustering using the PCL library implementation [22]. Based on the detected position, a depth map around the object is cropped and fed into *PoseNet*. The output pose of the network is relative to the image center, so the initial part pose can be computed by de-projecting the center pixel of the cropped image.

The robot approaches the object to observe the part closely as Eq. (8-11). The wrist camera observes the part and performs 3D registration using Iterative Closest Point (ICP) with the estimated initial pose. This is a fast and efficient method as long as the initial pose is correctly estimated. Then, the depth map of the part and the refined orientation are fed to *SymNet* to check if the estimated orientation is flipped or not due to the symmetry of the part. If the network outputs false for an axis, the orientation is flipped about the axis. Then, the refinement step using ICP is performed again.

VI. Safe Motion Control

In order to detect person intrusion into the workspace of the robot, we utilize ASUS Xtion Pro depth sensor, which observes the workspace from the top. We design a method to filter out any known objects in the workspace (e.g., the bin), and the robot itself, by representing them as a set of cuboids. Points measured by the workspace camera that correspond to these elements are filtered out of the point cloud. Points that fall out of the workspace of the arm are also not considered. On the remaining points, we set a threshold on the number of points to detect the presence of unknown objects entering the workspace. We set the value of this threshold empirically to 800 points.

To control the motion of the arm, we use the estimated pose of the object as a source for creating the desired observation, approach and grasp pose for the tip of the gripper in Cartesian space. Movement to pre-recorded arm configurations is realised in joint space. We utilise the Reflexxes library [23] for interpolating between the current and desired position in Cartesian or joint space. Interpolated Cartesian poses are converted to joint positions through inverse kinematics with the selectively damped least squares (SDLS) method [24].

VII. Experiments

The proposed system was evaluated in three aspects: i) accuracy and computation time of pose estimation, ii) success rate of picking and feeding tasks, and iii) human safety. The demonstration took place during the official showcase evaluation of the European Robotics Challenge (EuRoC) at Fraunhofer IPA. We recorded data and evaluated following performance measures in the presence of an examiner from the challenge host[6].

[6]Video link: `https://www.ais.uni-bonn.de/videos/robolink_feeder/`

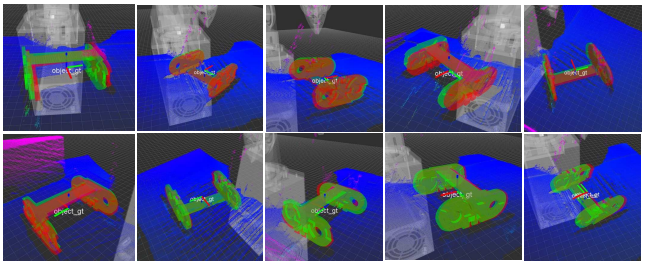


Fig. 10: Results of the estimated pose (green) and manually found ground truth pose (red).

A. Part Pose Estimation

The examiner randomly placed a part on the table, and the system detected the part and estimated its pose. After saving the estimated pose, the examiner manually annotated ground truth pose by overlaying the part CAD model on the scene using a 3D visualizer and GUI interface. The evaluation was repeated ten times. Fig. 10 shows the difference between estimated pose (green) and annotated ground truth pose (red). Table I shows the results of the pose accuracy and the computation time.

Even the maximum position and orientation errors (see fourth image at top row in Fig. 10) are precise enough to be grasped by the gripper of robolink® feeder, which has 30 mm stroke. Errors from the other cases are within the manual labeling.

B. Picking and Feeding

The second evaluation is measuring success rate and cycle time for the *picking* and *feeding* tasks. In the first *picking* task, the robot attempted 20 times bin-picking, where the graspable part detection is critical to estimate reliable grasping pose. In the second *feeding* task, the examiner randomly placed a part on a table, then the robot estimated part pose and planned trajectories to grasp and feed the part into the slot. The *feeding* task was repeated 10 times. Table II shows the evaluation results.

Due to the typical noise and incorrect depth measurements, false positive graspable parts are possibly detected. This results in failed picking at the first attempt, as the images at the first row in Fig. 11. Once the robot cannot find any part on the table, it returns back to the observation pose and selects the second highest graspable part, which is the second attempt. As shown in the second row in Fig. 11, the robot successfully picked all parts at the second attempt in our experiments.

TABLE I: Evaluation result of part pose estimation

	Position	Orientation
Avg. error	3.9704 mm	2.7369 degree
Median error	3.6124 mm	2.7826 degree
Max. error	6.7180 mm	4.9921 degree
Avg. computation time	1.5564 sec	

TABLE II: Evaluation result of part picking and feeding

	Picking (1st attempt)	Picking (2nd attempt)	Feeding
Success rate	65 %	100 %	90 %
Avg. cycle time	32.53 sec	79.82 sec	86.42 sec

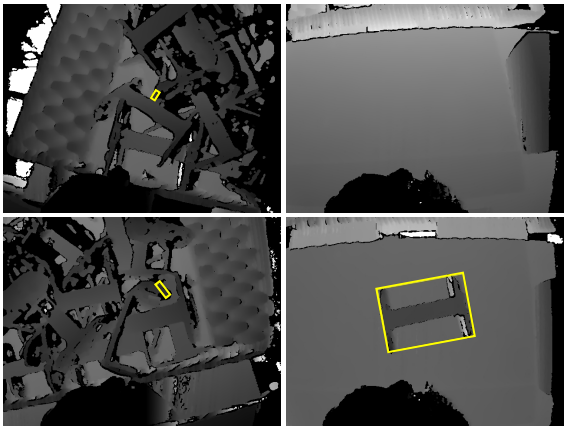


Fig. 11: The first (top) and second (bottom) attempts of picking in the case of false positive graspable part (yellow rectangle in the top left image). Left images show the selected graspable parts, and right images show the placed objects on the table.

C. Complete Pipeline Test with Collaboration with a Human

We performed complete test of *picking* and *feedinig* with human interventions. During picking and feeding parts ten times in sequence, the examiner intentionally reached into the workspace several times. The response time is measured by a duration between the time the workspace camera detects and sends stop request to the robot and the time all robot joint velocities are zero. The robolink® successfully fed nine parts out of ten. One missing feeding was caused by wrong pose estimation when the part was placed outside of the observation field. There were 46 intervention attempts in total, and the robot successfully stopped its movement fast enough to avoid colliding to the human.

VIII. CONCLUSION

We have presented a demonstrator for the feeding of parts from bulky supply. Our goal was to develop a system that is easy to reconfigure for new part variants, and that is safe for the presence of human workers. To these ends, we proposed a vision system that finds generic graspable points in RGB-D data, and a learning-based method that robustly estimates the pose of the grasped part. The chosen hardware,

TABLE III: Evaluation result of safe motion control

# of attempts	46
Success rate	100 %
Avg. response time	0.2057 sec

a lightweight cable-driven igus-robolink® WR manipulator allows for performing fast motions without posing safety issues for human workers in the vicinity. We have additionally proposed a fast method for detecting the presence of humans in the workspace of the arm in order to stop its motion. We have shown in the experiments the reliability of our system in performing 1) the detection of graspable parts in random piles of objects, 2) estimating the 6D pose of the part that was grasped, and 3) feeding the part with the right orientation.

ACKNOWLEDGMENTS

We acknowledge the contributions of igus® GmbH and Michael Schreiber of University of Bonn to the development of the system, and thank Bernd Winkler and Ramez Awad of Fraunhofer IPA for their support on system evaluation.

REFERENCES

[1] D. Holz, A. Topalidou-Kyniazopoulou, J. Stückler, and S. Behnke, "Real-time object detection, localization and verification for fast robotic depalletizing," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, IEEE, 2015, pp. 1459–1466.

[2] M. Schwarz, A. Milan, C. Lenz, A. Munoz, A. S. Periyasamy, M. Schreiber, S. Schüller, and S. Behnke, "Nimbro picking: Versatile part handling for warehouse automation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[3] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa, "Fast directional chamfer matching," in *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2010, pp. 1696–1703.

[4] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke, "Registration with the point cloud library: A modular framework for aligning in 3-d," *IEEE Robotics & Automation Magazine*, vol. 22, no. 4, pp. 110–124, 2015.

[5] Q.-Y. Zhou, J. Park, and V. Koltun, "Fast global registration," in *European Conference on Computer Vision*, Springer, 2016, pp. 766–782.

[6] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T. K. Marks, and R. Chellappa, "Fast object localization and pose estimation in heavy clutter for robotic bin picking," *The International Journal of Robotics Research*, vol. 31, no. 8, pp. 951–973, 2012.

[7] F. Smith and B. Rooks, "The harmonious robot," *Industrial Robot: An International Journal*, vol. 33, no. 2, pp. 125–130, 2006.

[8] W. S. Harwin, J. L. Patton, and V. R. Edgerton, "Challenges and opportunities for robot-mediated neurorehabilitation," *Proceedings of the IEEE*, vol. 94, no. 9, pp. 1717–1726, 2006.

[9] M. Quigley, A. T. Asbeck, and A. Y. Ng, "A low-cost compliant 7-dof robotic manipulator.," in *ICRA*, IEEE, 2011, pp. 6051–6058.

[10] J. P. Whitney and J. K. Hodgins, "A passively safe and gravity-counterbalanced anthropomorphic robot arm," in *2014 IEEE International Conference on Robotics and Automation*, 2014, pp. 6168–6173.

[11] Y. Domae, H. Okuda, Y. Taguchi, K. Sumi, and T. Hirai, "Fast graspability evaluation on single depth maps for bin picking with general grippers," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 1997–2004.

[12] D. Buchholz, M. Futterlieb, S. Winkelbach, and F. M. Wahl, "Efficient bin-picking and grasp planning based on depth data," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2013, pp. 3245–3250.

[13] J. J. Rodrigues, J.-S. Kim, M. Furukawa, J. Xavier, P. Aguiar, and T. Kanade, "6d pose estimation of textureless shiny objects using random ferns for bin-picking," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2012, pp. 3334–3341.

[14] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *ArXiv preprint arXiv:1603.02199*, 2016.

[15] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 1316–1322.

[16] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[17] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *2016 IEEE International Conference on Robotics and Automation, ICRA*, 2016.

[18] M. Schwarz, A. Milan, A. Selvam, and S. Behnke, "RGB-D object detection and semantic segmentation for autonomous manipulation in clutter," *The International Journal of Robotics Research*, to appear 2017.

[19] S. Behnke, M. Pfister, and R. Rojas, "Recognition of handwritten digits using structural information," in *Neural Networks, 1997., International Conference on*, IEEE, vol. 3, 1997, pp. 1391–1396.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ArXiv preprint arXiv:1409.1556*, 2014.

[21] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *ArXiv preprint arXiv:1502.03167*, 2015.

[22] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2011, pp. 1–4.

[23] T. Kröger, "Opening the door to new sensor-based robot applicationsthe reflexxes motion libraries," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2011, pp. 1–4.

[24] S. R. Buss and J.-S. Kim, "Selectively damped least squares for inverse kinematics," *Journal of graphics, gpu, and game tools*, vol. 10, no. 3, pp. 37–49, 2005.