NimbRo Picking: Versatile Part Handling for Warehouse Automation

Max Schwarz*, Anton Milan, Christian Lenz, Aura Muñoz, Arul Selvam Periyasamy,
Michael Schreiber, Sebastian Schüller, and Sven Behnke

***Abstract*— Part handling in warehouse automation is challenging if a large variety of items must be accommodated and items are stored in unordered piles. To foster research in this domain, Amazon holds picking challenges. We present our system which achieved second and third place in the Amazon Picking Challenge 2016 tasks. The challenge required participants to pick a list of items from a shelf or to stow items into the shelf. Using two deep-learning approaches for object detection and semantic segmentation and one item model registration method, our system localizes the requested item. Manipulation occurs using suction on points determined heuristically or from 6D item model registration. Parametrized motion primitives are chained to generate motions. We present a full-system evaluation during the APC 2016 and component-level evaluations of the perception system on an annotated dataset.**

I. Introduction

Bin-picking problems arise in a wide range of applications, from industrial automation to personal service robots. In the case of warehouse automation, the problem setting has unique properties: While the surrounding environment is usually very structured—boxes, pallets and shelves—the sheer number and diversity of objects that need to be recognized and manipulated pose daring challenges to overcome.

In July 2016, Amazon held the second Amazon Picking Challenge (APC)[1], which provided a platform for comparing state-of-the-art solutions and new developments in bin picking and stowing applications. The challenge consisted of two separate tasks, where contestants were required to pick twelve specified items out of chaotically arranged shelf boxes shelf—and to stow twelve items from an unordered pile in a tote into the shelf. Amazon provided a set of objects from 39 categories, representing a large variety of challenging properties, including transparency (e.g. water bottle), shiny surfaces (e.g. metal or shrink wrap), deformable materials (e.g. textiles), black surfaces (difficult to measure depth), white textureless surfaces, heavy objects, and non-solid objects with many holes (not easy to grasp with a suction cup). Also the shiny metal floors of the shelf boxes posed a considerable challenge to the perception systems, as all objects are also visible though their mirrored image. Before the run, the system was supplied with a task file, which specified the desired objects and the object location (in terms of shelf boxes or the tote). After the run, the system was expected to output the new locations of the items.

Our team developed a robotic system for the APC with some unique properties, which will be presented in this work.

*University of Bonn, max.schwarz@uni-bonn.de

[1] http://amazonpickingchallenge.org/

Fig. 1. Picking objects from the APC shelf.

Our main contributions include:

1) Development of two deep-learning based object perception methods that employ transfer learning to learn from few annotated examples (Section V),
2) integration of said deep-learning techniques into a robotic system,
3) and a parametrized-primitive-based motion generator which renders motion planning unnecessary (Section VI).

II. Related Work

Bin picking is one of the classical problems in robotics and has been investigated by many research groups in the last three decades, e.g. [1]–[9]. In these works, often simplifying conditions are exploited, e.g. known parts of one type being in the bin, parts with holes that are easy to grasp by sticking fingers inside, flat parts, parts composed of geometric primitives, well textured parts, or ferrous parts that can be grasped with a magnetic gripper.

During the APC 2015, various approaches to a more general shelf-picking problem have been proposed and evaluated. Correll *et al.* [10] aggregate lessons learned during the APC 2015 and show a general overview and statistics of the approaches. For example, 36 % of all teams (seven of the top ten teams) used suction for manipulating the objects.

Eppner *et al.* [11] describe their winning system for APC 2015. Mechanically, the robot consists of a mobile base and a 7-DOF arm to reach all shelf bins comfortably. In contrast, our system uses a larger arm and can thus operate without

a mobile base (see Section III). The endeffector of Eppner *et al.* [11] is designed as a fixed suction gripper, which can execute top and side picks; front picks are, however, not possible. For object perception, a single RGB-D camera captures the scene. Six hand-crafted features are extracted for each pixel, including color and geometry-based features. The features are then used in a histogram backprojection scheme to estimate the posterior probability for a particular object class. The target segment is found by searching for the pixel with the maximum probability. After fitting a 3D bounding box, top or side grasps are selected heuristically. Similar to our system, motion generation is based on parametrized motion primitives and feedback is used to react safely to collisions with the environment, rather than performing complex motion planning beforehand. The system could not manipulate the pencil cup object, which our system can pick with a specialized motion primitive. The team performed very well at APC 2015 and reached 148 out of 190 points.

Yu *et al.* [12] reached second place with Team MIT in the APC 2015. Their system uses a stationary industrial arm and a hybrid suction/gripping endeffector. The industrial arm provides high accuracy and also high speed. Similar to our approach, an Intel RealSense sensor mounted on the wrist is used for capturing views of the bin scenes (together with two base-mounted Kinect2 sensors). A depth-only GPU-based instance registration approach is used to determine object poses. Again, motion primitives were chosen in favor of motion planning. Specialized motion primitives can be triggered to change the configuration inside the bin when no picking action can be performed (such as tipping an object over). Team MIT achieved 88 points in the competition.

In contrast to the APC 2015, the 2016 challenge introduced more difficult objects (e.g. the heavy 3 lb dumbbell), increased the difficulty in the arrangements, and introduced the new stowing task.

III. Mechatronic Design

Our robot consists of a 6-DOF arm, a 2-DOF endeffector, a camera module, and a suction system.

To limit system complexity, we chose to use a stationary manipulator. This means the manipulation workspace has to cover the entire shelf, which places constraints on the possible robotic arm solutions. In our case, we chose the UR10 arm from Universal Robotics, because it covers the workspace nicely, is cost-effective, lightweight, and offers safety features such as an automatic (and reversible) stop upon contact with the environment.

Attached to the arm is a custom-built endeffector (see Fig. 2). For reaching into the deep and narrow APC shelf bins, we use a linear actuator capable of 37 cm extension. On the tip of the linear extension, we mounted a rotary joint to be able to carry out both front and top grasps. The rotary joint is actuated by a pulley mechanism, with the servo motor residing on the other end of the extension (and thus outside of the shelf during picking). This means that the cross section that needs to be considered during motion generation is only 3 cm×3 cm.

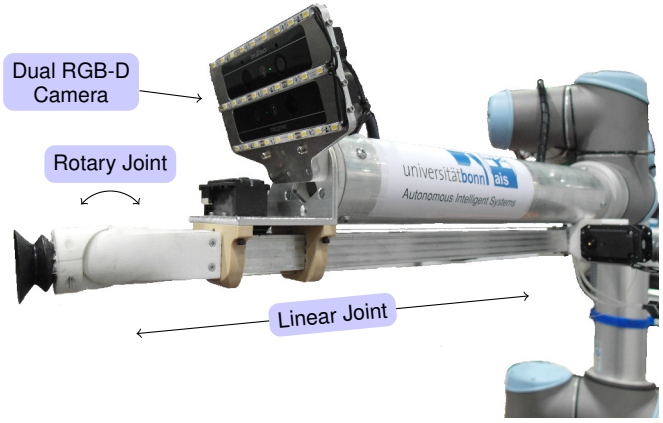


Fig. 2. Endeffector with suction finger and dual camera setup.

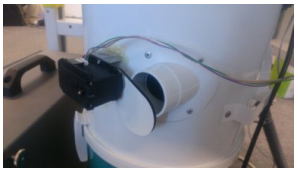


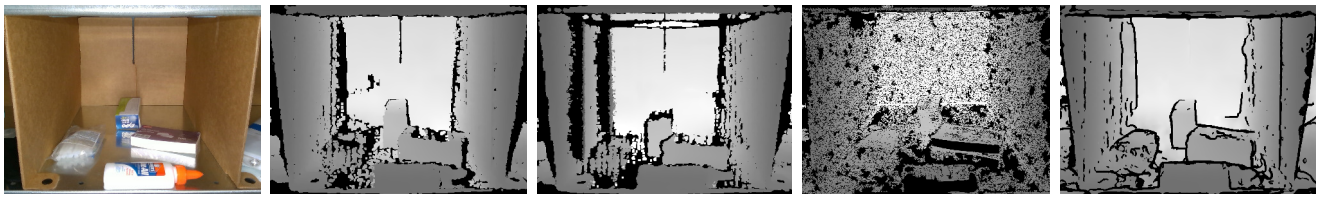
Fig. 3. Bleed actuator for suction regulation. Left: CAD model. Right: Final installation.

For grasping the items, we decided to employ a suction mechanism. This choice was motivated by the large success of suction methods during the last APC [10], and also due to the presented set of objects for the APC 2016, most of which could be manipulated easily using suction. Our suction system is designed to generate both high vacuum *and* high air flow. The former is needed to lift heavy objects, the latter for objects on which the suction cup cannot make a perfect vacuum seal.

Air flow is guided from a suction cup on the tip of the endeffector through the hollow linear extension, and then through a flexible hose into the robot base. The vacuum itself is generated by a 3100 W vacuum cleaner meant for central installation. For binary on/off control, it offers a 12 V control input. Since it overheats quite easily if the air flow is completely blocked, we added a "bleed" actuator (see Fig. 3), which can regulate the amount of air sucked into an additional intake. By closing the intake, we achieve maximum suction strength, while complete opening reduces suction to zero. Air flow is measured by a pitot tube inside the linear extension. This is used to detect whether an object was successfully grasped or lost during arm motion.

In summary, our kinematic design allows us to apply suction on all points of the object hemisphere facing the robot, control suction power quickly and precisely, and monitor air flow to recognize success or failure.

For control and computations, two computers are connected to the system. The first one, tasked with high- and low-level control of the robot, is equipped with an Intel Core i7-4790K CPU (4 GHz). The second one is used for vision processing, and contains two Intel Xeon E5-2670 v2



(a) RGB frame (b) Upper depth (c) Lower depth (d) Stereo depth (e) Fused result

Fig. 4. RGB-D fusion from two sensors. Note the corruption in the left wall in the lower depth frame, which is corrected in the fused result.

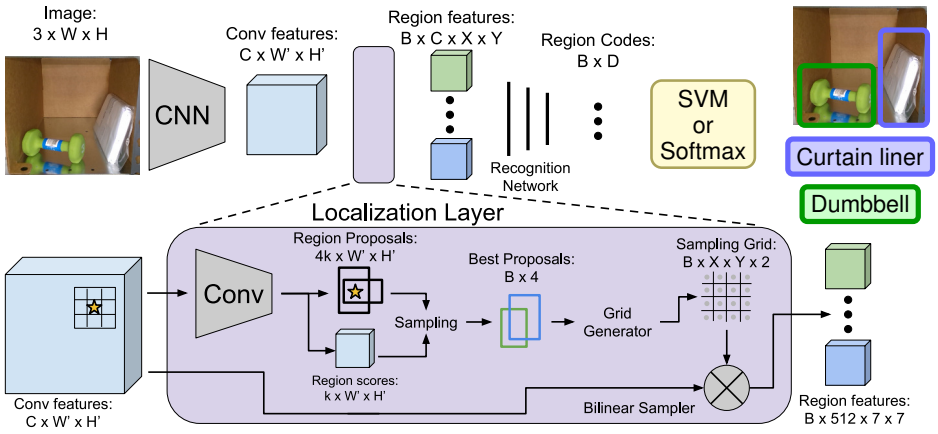


Fig. 5. Architecture of the object detection pipeline. Adapted from Johnson *et al.* [13].

(2.5 GHz) and four NVIDIA Titan X GPUs. For training, all four GPUs can be used to accelerate training time. At test time, two GPUs are used in parallel for the two deep learning approaches (see Section V).

IV. RGB-D Preprocessing

After testing multiple sensors in the APC setting, we settled on the Intel RealSense SR300 RGB-D sensor due to its lightweightness, high resolution, and short-range capabilities. However, we noticed that the depth sensor produced systematic artifacts on the walls of the shelf. The artifacts seem to depend on the viewing angle, i.e. they were present only on the right side of the image. To rectify this situation, we designed a dual sensor setup, with one of the sensors rotated 180° (see Fig. 2).

Using two separate sensors also makes a second RGB stream available. To exploit this, we also calculate dense stereo disparity between the two RGB cameras using LIBELAS [14]. The three depth sources are then projected into a common frame and fused using a majority voting scheme. Figure 4 shows an exemplary scene with the fused depth map. The final map is then filled and regularized using a guided TGV regularizer [15] implemented in CUDA.

V. Perception

For perceiving objects in the shelf or tote, we developed two independent methods. The first one solves the object detection problem, i.e. outputs bounding boxes and object classes for each detection. The second one performs semantic segmentation, which provides a pixel-wise object classification.

Since training data and time is limited, it is crucial not to train from scratch. Instead, both methods leverage convolutional neural networks (CNNs) pre-trained on large image classification datasets and merely adapt the network to work in the domain of the APC.

A. Object Detection

We extend an object detection approach based on the DenseCap network [13]. DenseCap approaches the problem of dense captioning, i.e. providing detailed textual descriptions of interesting regions (bounding boxes) in the input image. Figure 5 shows the general architecture of the DenseCap network. A large number of proposals from an integrated region proposal network are sampled to a fixed number (1000 in our case) using an objectness score network. Intermediate CNN feature maps are interpolated to fixed size for each proposal. The proposals are then classified using a recognition CNN. The underlying CNN was pretrained on the ImageNet [16] dataset. Afterwards, the entire pipeline was trained end-to-end on the Visual Genome dataset [17]. In order to make use of this pretraining, we use a trained model distributed by the DenseCap authors and either train a custom classifier or finetune the entire pipeline (see Sections V-A.1 and V-A.2).

Since the region proposals do not make use of depth, we augment the network-generated proposals with proposals from a connected components algorithm running on the RGB and depth frames (see Fig. 6). Two pixels are deemed connected if they do not differ more than a threshold in terms of 3D position, normal angle, saturation and color. Final

Fig. 6. RGB-D based additional region proposals. Left: RGB frame. Center: Regions labeled using the connected components algorithm. Right: Extracted bounding box proposals.

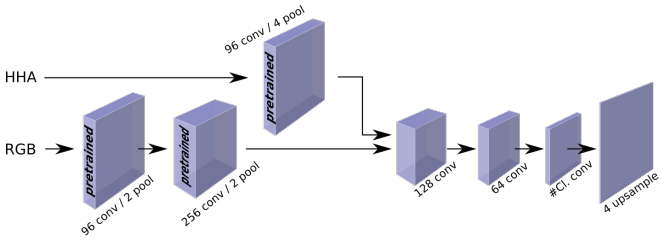


Fig. 7. Our network architecture for semantic object segmentation.

bounding boxes are extracted from regions which exceed an area threshold.

While the textual descriptions are not interesting for bin-picking scenarios, the descriptions are generated from an intermediate feature vector representation, which is highly descriptive. To exploit the power of this feature representation, we use the network without the language model for feature extraction, and do classification using a linear SVM. As an alternative, we investigate a soft-max classifier layer, which allows us to fine-tune the network during training.

1) Linear SVM: In the first case, we remove the language generation model and replace it with a linear SVM for classification. We also introduce two primitive features based on depth: The predicted bounding box is projected into 3D using the center depth value. The metric area and size are then concatenated to the CNN features. Since linear SVMs can be trained very efficiently, the training can happen just-in-time before actual perception, exploiting the fact that the set of possible objects in the bin is known. Restricting the set of classes also has the side-effect that training time and memory usage are constant with respect to the set of all objects present in the warehouse.

The SVM is used to classify each predicted bounding box. To identify a single output, the bounding box with the maximum SVM response is selected. This ignores duplicate objects, but since the goal is to retrieve only one object, this reduction is permissible.

2) Finetuning: For finetuning the network, we use a soft-max classification layer instead of the SVM. All layers except the initial CNN layers (see Fig. 5) are optimized. In contrast to SVM training, the training is performed offline on all object classes. At test time, all predicted boxes are classified and the bounding box with the correct class and highest objectness score is produced as the final output.

B. Semantic Segmentation

Manipulation of real-world objects requires a more precise localization that goes beyond a bounding box prediction. Therefore, we also investigated pixel-level segmentation approaches. To that end, we adapt our previous work [18] to the scenario at hand. The method employs a 6-layer fully convolutional neural network (CNN) similar to OverFeat [19]. The full network architecture is illustrated in Fig. 7.

As a first step, low-level features are extracted from the captured RGB-D images using a set of filters that was pretrained on ImageNet [20]. We then finetune the network to the APC domain by training the last three layers of the network.

C. Combination

During APC, we used a combination of the SVM object detection approach and the semantic segmentation. The bounding boxes predicted by the object detection were rendered with a logistic estimate of their probability and averaged. This process produced a "probability map" that behaved like a pixel-wise posterior. In the end, we simply multiplied this probability map with the class probabilities determined in semantic segmentation. A pixel-wise max-probability decision then resulted in the final segmentation mask used in the rest of the pipeline.

After APC, we replaced the hard bounding box rendering with a soft gaussian, which yielded better results.

D. Item Pose Estimation

For certain objects, manipulation in the constrained space of the shelf is only possible if the 6D object pose is known. For example, large objects such as the pack of socks can only be grasped near the center of mass. Other grasps will result in tilting the object, making it impossible to remove it in a controlled manner and without collisions.

To that end, we modeled a dense representation of such objects using the method and implementation by Prankl *et al.* [21], where we capture a 360° turntable sequence of point clouds of the object with the robot's sensors. We select a subset of frames that fully captures the object from various angles. Subsequently, the extrinsic camera parameters are estimated given the correspondences between frames, which are finally refined using bundle adjustment. Based on a manually positioned bounding box, the scene is filtered to exclude background and false measurements. From the camera poses and masks of the selected frames, a 3D model of the object is reconstructed by optimizing correspondences between frames. Once the object geometry is captured,

Fig. 8. Pose registration. Left column: Turntable capture, resulting model. Center column: Scene, scene with initializations. Right column: Final registered model.

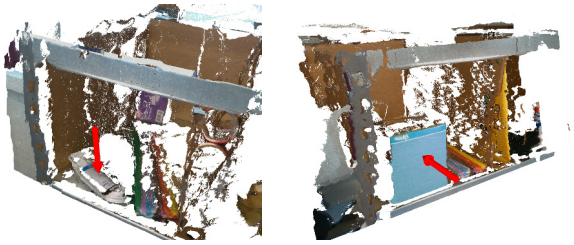


Fig. 9. Heuristic grasp selection. Left: Top grasp on an extension cord. Right: Front grasp on the kleenex tissue box.

we manually attach the desired grasping poses for each object in turn in order to guarantee a stable grasp. We experimented with multiple 6D pose estimation methods, and finally adopted an ICP-based approach, which gave fastest and most accurate results in our setting.

From the segmentation mask for the particular object, we can extract object points from the scene point cloud. As we know that particular models are likely to be positioned in few orientations (e.g. standing or lying on the ground), we can define a set of predefined orientations to initialize the registration. At the moment of performing registration, we position the model at the center of mass of the filtered point cloud and perform Generalized ICP [22] on the predefined set of orientations and choose the 6D pose with the shortest Euclidean registration distance between scene and model. Figure 8 shows a tote scene with the tube socks object.

Note that 6D pose registration was only required for three objects: the duct tape (little supports for suction), the pack of tube socks (large), and the paper towel roll (large). All other APC objects can be grasped using generic grasp positions directly computed from the segmentation mask, which is described in Section VI-A.

VI. Motion Generation

At first glance, the kinematic constraints imposed on motions in the shelf appear quite severe: The available space is very narrow, and objects can be partially occluded, meaning that the robot has to reach around other objects.

A. Heuristic Grasp Selection

For objects which are not registered (see Section V-D), we select grasps heuristically. Our system supports two basic grasps: Top grasp and center grasp.

The top grasp is determined using the 3D bounding box of the object. We select the point belonging to the segmentation mask, whose projection onto the ground plane is closest to the projection of the 3D bounding box center. The grasp height is chosen as the maximum height of object points in a cylinder around the chosen position. The grasp position is then refined to the next object point.

Center grasps are defined as grasps close to the 2D image-space bounding box center. Again, the closest object point to the center is chosen, this time in image space. The surface normal is estimated using the local neighborhood and used as grasp direction.

Figure 9 shows center and top grasps on exemplary scenes.

B. Inverse Kinematics

In order to simplify the problem, we focused on an intelligent inverse kinematics solver first. The solver is driven by two ideas: First, the suction pose itself is invariant to rotations around the suction axis, and second, the solver should resolve the inherent redundancy in the kinematic chain so as to minimize the chance of collisions with the environment.

As a basis, we use a selectively damped least squares (SDLS) solver [23]. We augment it with a null-space optimization step, which projects the gradient of a secondary objective f to the null space of the SDLS Jacobian J.

We first define a joint-level null space objective g:

$$\begin{aligned} g_i(q) = w_l \max\{0, q - (q_i^+ - q_\delta)\}^2 \\ + w_l \min\{0, q - (q_i^- + q_\delta)\}^2 \\ + w_c (q - q_i^{(c)})^2, \end{aligned} \tag{1}$$

where i is the joint index, q is the joint position, q_i^+ and q_i^- are the upper and lower joint limits, q_δ is a joint limit threshold, $q_i^{(c)}$ is the "convenient" configuration for this joint, and w is used to form a linear combination of the costs. As can be seen, this objective prefers a convenient configuration and avoids joint limits.

More interestingly in this application, we also specify Cartesian-space costs using a plane-violation model:

$$h_{\vec{n},d}(\vec{x}) = (\max\{0, (-\vec{n}\vec{x}^T + d)\})^2, \tag{2}$$

where $\vec{n}$ and d specify an oriented plane $\vec{n}\vec{x}^T - d = 0$, and $\vec{x}$ is some Cartesian point. This model is used to avoid specified half-spaces with parts of the robot.

Finally, we obtain the combined costs f:

$$\begin{aligned} f(\vec{q}, \vec{x_l}, \vec{x_w}) = \sum_{i \in Q} g_i(\vec{q_i}) + h_{\vec{n}_s, d_s}(\vec{x_l}) \\ + h_{\vec{n}_t, d_t}(\vec{x_l}) + h_{\vec{n}_b, d_b}(\vec{x_w}), \end{aligned} \tag{3}$$

where $\vec{q}$ is the vector of joint positions, $\vec{x_l}$ and $\vec{x_w}$ are Cartesian positions of the linear extension and the camera module, and $\vec{n_i}, d_i$ describe three half spaces which are avoided (see Fig. 10). This half space penalization ensures that we do not enter the shelf with the cameras, that the linear

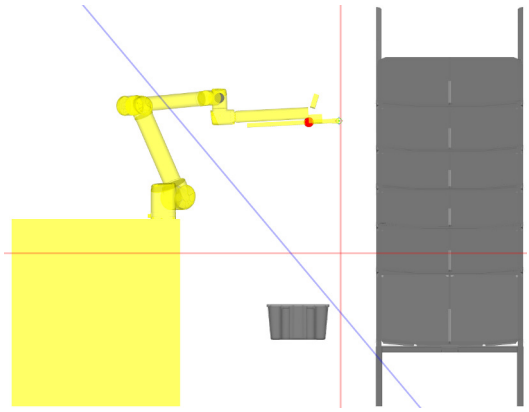


Fig. 10. Penalizing planes in IK solver. The red/blue planes penalize violation by the red/blue spheres, respectively.

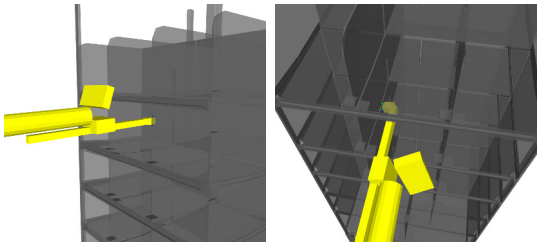


Fig. 11. Nullspace-optimizing IK. Left: Front grasp. Right: Side grasp.

extension is horizontal during manipulation in the shelf[2], and that collisions with the robot base are avoided.

One iteration of the solver calculates the update δ_q as follows:

$$\bar{J}, \bar{J}^+ = \text{SDLS}(RPR^T J) \tag{4}$$

$$N = I - \bar{J}^+ \bar{J} \tag{5}$$

$$\delta_q = \bar{J}^+ \Delta x - \alpha N \nabla f(\vec{q}, \vec{x_l}, \vec{x_w}), \tag{6}$$

where R is the target orientation of the endeffector, P is a projector zeroing the roll component (allowing rotation around the suction axis), J is the $6 \times n$ kinematic Jacobian matrix, N is the null space projector of $\bar{J}$, Δx is the remaining 6D pose difference, and α is the step size for null space optimization.

Using this custom IK solver, it is possible to reach difficult target poses in the shelf and tote without collisions (see Fig. 11). Note that we used a null-space optimizing solver before [24], but limited the cost function to joint-space posture costs. In contrast, the null-space costs are now used to avoid collisions in task space.

C. Retract planner

For approaching an object, we can follow the camera ray to the object to get a collision-free trajectory. Retracting with the object, however, can be more difficult, especially in the shelf, since other objects might be in front of our target object.

[2]This also uses the penalization of linear extension in Eq. (1).

Fig. 12. Retract planning. Left: RGB image of the scene. Center: Front-projected collision mask for retrieval of the black pencil cup. The sippy cup is not fully masked because of missing depth values on the transparent surface. Right: Distance transform.

As gravity keeps the objects on the floor of the shelf bin, we can always lift the object as high as possible to increase the chance of collision-free retraction. For simplicity, we decided to restrict further retract planning to find an optimal Y coordinate (with the Y axis pointing sideways).

To do this, we first calculate a 2D "skyline" view of the potential colliding objects (see Fig. 12). After performing a distance transform, we can easily identify an ideal Y coordinate with the maximum distance to colliders.

D. Parametrized Motion Primitives

For actual motion generation we use our keyframe-based interpolation system [24]. Each keyframe specifies either joint- or Cartesian space configurations for parts of the robot. It also specifies joint and/or Cartesian velocity and acceleration constraints which limit the motion to this keyframe. Keyframes can be edited in a dedicated 3D GUI for predefined motions such as dropping items into the tote, or adapted live to perception results, such as grasp motions. Finally, motions are smoothly interpolated in joint space and executed on the robot.

Since our motion generation—while very robust—makes several strong assumptions, it is still possible that unwanted collisions with the shelf or other objects are generated. In particular, there is no mechanism that detects whether it is actually possible to retrieve the target object. For example, it may be necessary to move other occluding objects before attempting actual retrieval. In our experience, however, the combination of the inverse kinematics solver and the retract planner are sufficient to solve most situations. As a final preemptive measure, we configure the UR10 to stop and notify the control software whenever the exerted force exceeds a threshold. The software then releases the stop, executes a retract primitive, and continues with the next object. Failed objects are retried at the end of the picking sequence.

VII. Results

A. Amazon Picking Challenge 2016

The system proposed in this work attempted both the picking and stowing task successfully during the APC 2016. For stowing, our system stowed eleven out of twelve items into the shelf.[3] However, one of the successfully stowed items was misrecognized, which meant that the system could not recognize the final item (a toothbrush). Even though a

[3]Video at `https://youtu.be/B6ny9ONfdx4`

TABLE I

PICKING RUN AT APC 2016

Bin	Item	Pick	Drop	Report
A	duct tape	×	×	×
B	bunny book	✓	✓	×[2]
C	squeaky eggs	✓	×	✓
D	crayons[1]	✓	×	✓
E	coffee	✓	✓	×[2]
F	hooks	✓	×	✓
G	scissors	×	×	×
H	plush bear	✓	×	✓
I	curtain	✓	×	✓
J	tissue box	✓	×	✓
K	sippy cup	✓	×	✓
L	pencil cup	✓	✓	×[2]
	Sum	10	3	7

[1] Misrecognized, corrected on second attempt.
[2] Incorrect report, resulting in penalty.

fallback mechanism was built in, which would attempt to recognize all known objects, this method failed due to an object size threshold. The misrecognition of the item led to the attainment of the second place in the stow task.

In the picking task, our system picked ten out of twelve items.[4] Despite the high success rate (the winning team DELFT achieved a success pick-up rate of only nine items), only a third place was achieved as a consequence of dropping three items during picking. While this was recognized using the air velocity sensor, the system incorrectly deduced that the items were still in the shelf, while they actually dropped over the ledge and into the tote. Since the system was required to deliver a report on the final object locations, the resulting penalties dropped our score from 152 points to 97 points—just behind the first and second place with both 105 points.

On the final day of the competition, the teams had the chance to showcase their system in an open demonstration. We chose to retry the picking task in a slightly different configuration, which allowed us to show our ability to handle the most difficult objects: The pencil cup, which can only be suctioned on the bottom side, and the dumbbell, which is quite heavy (3 lb) for suction-based systems. For the former, we first push it over on the side. The latter is possible using our powerful vacuum system.

B. Object Detection

Apart from the system-level evaluation at the APC, we evaluated our perception approaches on our own annotated dataset, which was also used for training during APC. The dataset contains 190 shelf frames, and 117 tote frames. The frames vary in the number of objects and location in the shelf. As far as we are aware, this number of frames is quite low in comparison to other teams, which highlights the effectiveness of our transfer learning approach. Figure 13 shows an exemplary scene from the dataset with object detection and segmentation results.

[4]Video at `https://youtu.be/q9YiD80vwDc`

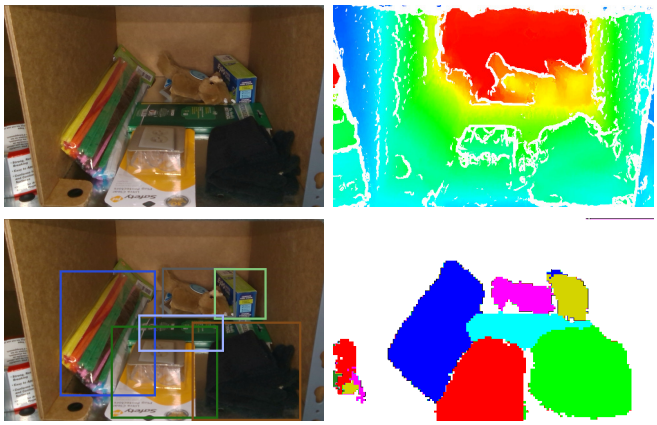


Fig. 13. Object perception example. Upper row: Input RGB and depth frames. Lower row: Object detection and semantic segmentation results (colors are not correlated).

TABLE II

F1 SCORES FOR OBJECT PERCEPTION

	Shelf		Tote	
Method	Uninformed	Informed	Uninformed	Informed
SVM (plain)	-	0.654	-	0.623
SVM (tailor)	-	0.661	-	0.617
Finetuned CNN	0.361	0.783	0.469	0.775
Segmentation	0.757	0.787	0.789	0.816
Combination	0.787	0.805	0.813	0.829

SVM (plain): trained on all object classes.
SVM (tailor): trained just-in-time for the objects present in the image.
Combination: Finetuned CNN + Segmentation.

For evaluation, we define a five-fold cross validation split on the shelf dataset. To see the effect of each design choice, we evaluate each approach in an informed case (the set of objects in the bin is known) and in an uninformed case. For object detection, we calculate area-based precision and recall from the bounding boxes. For segmentation, pixel-level precision and recall are calculated. Resulting F1 scores are shown in Table II. As expected, knowledge of the set of possible objects improves the performance. Finetuning the network yields a large gain compared to the SVM approach. As far as the box-level and pixel-level scores can be compared, the finetuning object detection approach and the semantic segmentation approach yield similar results. Finally, the combination of the finetuned object detector

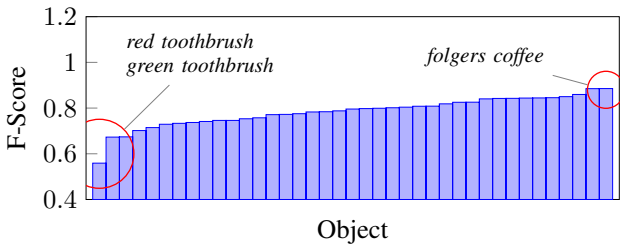


Fig. 14. F-Score distribution over the objects for object detection. Results are averaged over the cross validation splits using the finetuned model.

TABLE III
PERCEPTION RUNTIMES

Phase	Object detection			Segmentation
	RGB-D proposal	SVM	Finetuned	
Train	-	-	45 min	~5 h
Test	1006 ms	3342 ms[1]	340 ms	~900 ms

[1] Includes just-in-time SVM training

and the semantic segmentation yields a small but consistent increase in performance.

Figure 14 gives an impression of the distribution of difficulty across the objects. We also measured the runtime of the different modules on our setup (see Table III). Note that the two perception approaches usually run in parallel.

VIII. CONCLUSION

In this work, we described our system for the APC 2016, explained design choices, and evaluated the system in the competition and on our own dataset. As always, the results of single-trial competitions are very noisy. Teams may fail due to technical problems, misunderstandings, and pure chance. During training, we had better runs than in the competition. Still, our result proves that the components work in isolation and together under competition conditions.

As this was the first time in our group that deep-learning techniques were actually used in a live robotic system, this was a valuable learning opportunity for us. Indeed, only the tight integration of perception and action made success possible—as already noted by Eppner *et al.* [11]. Maybe for this reason, there are few ready-to-use deep-learning implementations for robotics contexts. We hope to reduce this problem with our source-code release which is planned together with the publication of this paper. Finally, we will also release our APC dataset annotated with object polygons and class labels.

REFERENCES

[1] D. Buchholz, D. Kubus, I. Weidauer, A. Scholz, and F. M. Wahl, "Combining visual and inertial features for efficient grasping and bin-picking," 2014, pp. 875–882.

[2] M. Nieuwenhuisen, D. Droeschel, D. Holz, J. Stückler, A. Berner, J. Li, R. Klein, and S. Behnke, "Mobile bin picking with an anthropomorphic service robot," in *Robotics and Automation (ICRA), IEEE International Conference on*, 2013, pp. 2327–2334.

[3] A. Pretto, S. Tonello, and E. Menegatti, "Flexible 3D localization of planar objects for industrial bin-picking with monocamera vision system," in *IEEE International Conference on Automation Science and Engineering (CASE)*, 2013, pp. 168–175.

[4] Y. Domae, H. Okuda, Y. Taguchi, K. Sumi, and T. Hirai, "Fast graspability evaluation on single depth maps for bin picking with general grippers," 2014, pp. 1997–2004.

[5] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: efficient and robust 3d object recognition," 2010, pp. 998–1005.

[6] A. Berner, J. Li, D. Holz, J. Stückler, S. Behnke, and R. Klein, "Combining contour and shape primitives for object detection and pose estimation of prefabricated parts," in *Proceedings of IEEE International Conference on Image Processing (ICIP)*, 2013.

[7] C. Martinez, R. Boca, B. Zhang, H. Chen, and S. Nidamarthi, "Automated bin picking system for randomly located industrial parts," in *2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, 2015, pp. 1–6.

[8] K. N. Kaipa, A. S. Kankanhalli-Nagendra, N. B. Kumbla, S. Shriyam, S. S. Thevendria-Karthic, J. A. Marvel, and S. K. Gupta, "Addressing perception uncertainty induced failure modes in robotic bin-picking," *Robotics and Computer-Integrated Manufacturing*, vol. 42, pp. 17–38, 2016.

[9] K. Harada, W. Wan, T. Tsuji, K. Kikuchi, K. Nagata, and H. Onda, "Iterative visual recognition for learning based randomized bin-picking," *arXiv preprint arXiv:1608.00334*, 2016.

[10] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, "Lessons from the amazon picking challenge," *arXiv preprint arXiv:1601.05484*, 2016.

[11] C. Eppner, S. Höfer, R. Jonschkowski, R. Martín-Martín, A. Sieverling, V. Wall, and O. Brock, "Lessons from the amazon picking challenge: four aspects of building robotic systems," in *Proceedings of Robotics: Science and Systems*, AnnArbor, Michigan, Jun. 2016. [Online]. Available: `http://www.redaktion.tu-berlin.de/fileadmin/fg170/Publikationen_pdf/apc_rbo_rss2016_final.pdf`.

[12] K.-T. Yu, N. Fazeli, N. Chavan-Dafle, O. Taylor, E. Donlon, G. D. Lankenau, and A. Rodriguez, "A summary of team MIT's approach to the amazon picking challenge 2015," *arXiv preprint arXiv:1604.03639*, 2016.

[13] J. Johnson, A. Karpathy, and L. Fei-Fei, "DenseCap: Fully convolutional localization networks for dense captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[14] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Asian Conference on Computer Vision (ACCV)*, 2010.

[15] D. Ferstl, C. Reinbacher, R. Ranftl, M. Rüther, and H. Bischof, "Image guided depth upsampling using anisotropic total generalized variation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 993–1000.

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[17] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, *et al.*, "Visual genome: connecting language and vision using crowdsourced dense image annotations," *arXiv preprint arXiv:1602.07332*, 2016.

[18] F. Husain, H. Schulz, B. Dellen, C. Torras, and S. Behnke, "Combining semantic and geometric features for object class segmentation of indoor scenes," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 49–55, May 2016, ISSN: 2377-3766.

[19] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," *CoRR*, vol. abs/1312.6229, 2013. [Online]. Available: `http://arxiv.org/abs/1312.6229`.

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, *ImageNet Large Scale Visual Recognition Challenge*. 2014.

[21] J. Prankl, A. Aldoma, A. Svejda, and M. Vincze, "RGB-D object modelling for object recognition and tracking," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, IEEE, 2015, pp. 96–103.

[22] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP.," in *Robotics: Science and Systems*, vol. 2, 2009.

[23] S. R. Buss and J.-S. Kim, "Selectively damped least squares for inverse kinematics," *Graphics, GPU, and Game Tools*, vol. 10, no. 3, pp. 37–49, 2005.

[24] M. Schwarz, T. Rodehutskors, D. Droeschel, M. Beul, M. Schreiber, N. Araslanov, I. Ivanov, C. Lenz, J. Razlaw, S. Schüller, D. Schwarz, A. Topalidou-Kyniazopoulou, and S. Behnke, "NimbRo Rescue: solving disaster-response tasks through mobile manipulation robot Momaro," *Journal of Field Robotics (JFR)*, vol. 34, no. 2, pp. 400–425, 2017.