Standing on Giant's Shoulders: Newcomer's Experience from the Amazon Robotics Challenge 2017

Gustavo Alfonso Garcia Ricardez, Lotfi El Hafi, and Felix von Drigalski

1 Introduction

Incentivizing the development of innovative solutions to real-world problems through competitions has become a popular means for many companies to gather new, diverse ideas [18]. The Amazon Robotics Challenge (ARC), formerly known as the Amazon Picking Challenge, focuses on warehouse automation and has become one of the most renowned robotics challenges. The challenge aims to combine the state-of-the-art robotic manipulation and computer vision technologies into practical solutions for warehouse operation.

With increasingly stricter rules, it models the problem of stocking and picking items, which may not be known in advance. For reference of scale, Amazon introduces 50,000 new items every day to warehouses and fulfills an estimated 35 orders per second. At the ARC 2016, the winning team achieved a success rate of nearly 84%, approaching the performance of human workers, who pick and stow about 400 items per hour at full speed with an almost 100% success rate [1].

In this chapter, we present our approach to the ARC 2017, with particular emphasis on the lessons learned from past participants, our design philosophy and our development strategy. Moreover, we describe two elements of our proposed system, the suction tool and the shelf or *storage system*, and their development process.

The remainder of this chapter is structured as follows: First, we summarize the ARC 2017 tasks and technical challenges in Sect. 2. In Sect. 3, we present our

G. A. Garcia Ricardez (✉) · L. El Hafi · F. von Drigalski
Team NAIST-Panasonic at the Amazon Robotics Challenge 2017, Nagoya, Japan

Graduate School of Information Science, Nara Institute of Science and Technology (NAIST), Ikoma, Nara, Japan
e-mail: garcia-g@is.naist.jp; lotfi.el_hafi.kx2@is.naist.jp; felix.von_drigalski.fp6@is.naist.jp

approach, the lessons learned from past competitors, our design philosophy, and our development strategy. In Sect. 4, we show an overview of our system and then present the details of the suction tool and storage system development. Finally, Sect. 5 concludes this chapter while highlighting the best lessons we learned.

2 Technical Challenges

The ARC consists of two tasks:

- *Stow*: Store 20 new items in the storage system, modeling the process of adding newly arrived items into the warehouse.
- *Pick*: Move 10 items from the storage system to three boxes, modeling the purchase process of Amazon.

The rules of the ARC 2017 have significant differences compared to previous years [2]:

(a) Half of the items are unknown until 30 min before the round starts.
(b) The storage system is designed by the teams.
(c) The volume of the storage system is 70% smaller than the previous years, for a total of 95 L.

(a) limits the applicability of conventional learning-based approaches in which a classifier is trained with large amounts of data (e.g., up to 150,000 images/item) to recognize items, which affects the approaches of eight of the top ten teams of the ARC 2016 [20, 21]. This requirement is realistic for warehouse applications, where new items are scanned and entered into the database and must be manipulated shortly after. For reference, Amazon enters about 50,000 new items to its inventory every day.

With (b), Amazon opened up a new design dimension in the challenge, allowing the teams to adapt the storage solution to their robot and to propose new ideas for the storage system. As the number of items remained the same as in previous years, the reduction of volume in (c) almost inevitably causes items to be stacked and occlusions to occur, which poses a significant challenge for object recognition, manipulation, and planning.

In summary, the main challenges are:

- **Object recognition:** Half of the items are unknown until 30 min before each round starts, which constrains following approaches where large amount of data can be used.
- **Robot manipulation:** The target items belong to very diverse categories (book, box, cylinder, deformable, wrapped, clamp-shell, and others), have
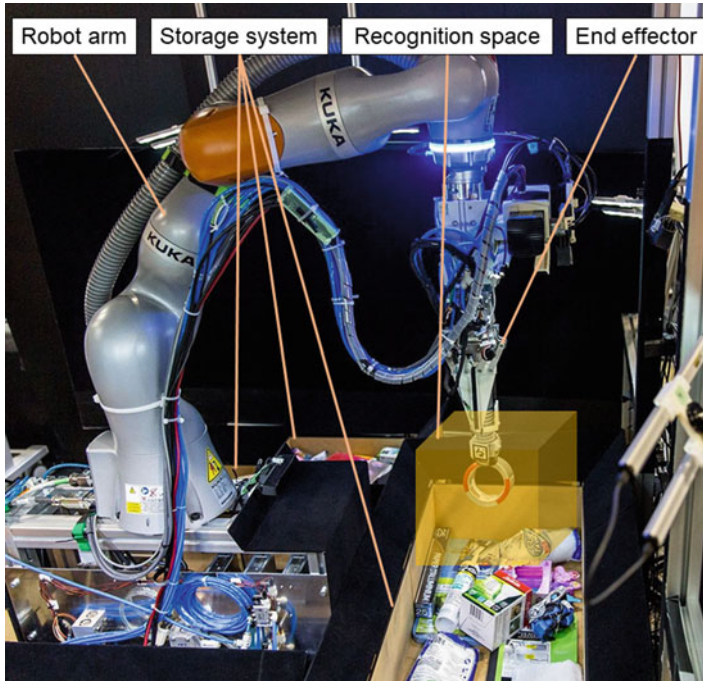
(continued)

Fig. 1 Proposed warehouse automation robotic solution deployed by Team NAIST-Panasonic at the ARC 2017. The system consists of a 7-DOF robot arm, a custom-made end effector, a weight-aware storage system, and a recognition space. Each picked item is examined in the recognition space equipped with four RGB-D cameras to quickly recognize items from multiple viewpoints using both the state-of-the-art learning-based and feature-based technologies. Photo courtesy of Amazon Robotics

> a maximum weight of 2 kg, and fit in a maximum volume of $0.42 \times 0.27 \times 0.14 \, \text{m}^3$.
>
> - **Storage system design:** Each team designs a storage system to fit a maximum of 32 items within 95 L of volume and 5000 cm^2 of area. It has to feature 2–10 bins, contain no actuators, a maximum of 50 USD in sensors (if any) and should be used for both tasks.

3 Approach

In general, our strategy consists of getting an initial estimation of the target item using the end effector camera, then pick it and move it to the *recognition space*, a

dedicated and controlled space as shown in Fig. 1, where multiple methods vote to determine its class. Our idea was to create a democratic sensor fusion system using RGB-D cameras [5, 16], as well as weight, contact, and force sensors to identify all items and reduce uncertainty to a minimum [11].

In the stow task, we move items from a container to the storage system relying on the deep learning classification and weight, and discard erroneous classifications with the item list provided before the round and the already moved items. In the pick task, we (1) search for known items which are recognizable by the deep learning classifier, and (2) explore the bins by grasping unknown items and moving them to the recognition space to be identified. Finally, we move the items to its corresponding box, if it is a target item, or to another bin or location inside the same bin otherwise.

In the rest of this section, we present the logic behind our development and design decisions during the ARC 2017. We also present a survey on the experiences of past competitors, describe our design philosophy, and present the development strategy that we followed.

3.1 Past Competitions

We have started our development by investigating other teams' efforts in the two previous editions of the competition, which are instructive both in terms of how some approaches succeeded and what went wrong for others. In the remainder of this section, we detail past work that helped us gain insights into past editions of the ARC, and how this knowledge shaped our own design approach.[1]

A number of reports and media coverage have summarized the state of the art of ARC, as well as the accumulated heuristics, such as [10] by Team RBO who took first place in ARC 2015.

Correll et al. [6] describe platforms, grippers, sensors, and perception and motion planning techniques used by the teams competing in the 2015 edition. They conclude that there is trade-off between customization and dependability of software developed by the teams and third parties.

This was complemented by an in-depth report from Nikkei [20, 21] about the solutions of the 2016 teams. Additional reports from the Robotics Society of Japan illuminated some more approaches and problems in [9, 22].

A number of previous competitors, such as Team C^2M [19], Team R U Pracsys [28], and Team MIT-Princeton [32], also provide implementations of their approaches, as well as the datasets generated during the competitions. These datasets have been useful as a starting point to train and test our object recognition algorithms, as many of the items are also found in the ARC 2017 practice kit.

---

[1] While this section summarizes our literature review preparing for the ARC 2017, we note that papers detailing the ARC 2017 results have been presented at the 2018 ICRA workshop on *Advances in Robotic Warehouse Automation* [3], and/or awarded *Amazon Robotics Best Paper* [14].

Table 1 Common failures and the potential impact on the performance

Failure	Potential impact
Collision with storage system	Round loss
Planning failure	Round loss
Items left on recognition space	Object recognition capability loss
Losing suction contact	Point loss due to dropped item
Two-item grasping	Point loss due to lost item
Object recognition errors	Point loss due to misplaced item
Grasping failures	Time loss
Slow path-planning	Time loss

Looking at the past competitions, it becomes clear that using suction cups and deep learning has tended to increase teams' success rates. Furthermore, the reports show that teams using a single robot manipulator perform better, and make a strong case for reusability by using the Robot Operating System (ROS) [24, 25].

We have identified the most common problems that have occurred during the competition and summarized their potential impact in Table 1.

With these failures in mind, we drew the following main conclusions to guide our development effort:

- Suction is an effective grasping tool, as 80% of the items are suctionable.
- A professional suction system is important for reliable operation.
- Learning-based object recognition can yield up to 90% success rate [20, 21].
- Using depth information does not improve object recognition significantly, and may even be counterproductive.
- Robust error recovery is fundamental for a competitive performance.
- A 7-DOF manipulator can save time by achieving the target pose quicker as demonstrated by the previous winners [10, 13].
- Task planning using state machines is effective [4].
- Modifying the code in the last minute must be avoided, as it leads to human error.
- Sensors can overheat and stability issues should be anticipated.
- Illumination in the venue significantly affects the object recognition performance.

3.2 Design Philosophy

Our design philosophy is centered around simplicity and reliability. As there is no miracle solution for vision and manipulation in unstructured environments, we focus on designing dependable systems that tolerate errors, recover from failures, and continue the tasks safely.

The concept of separation of functions allows the specialization of components and supports clean design [31]. In our solution, we separated the functions of the end effector into suction and gripper tools, as shown in Fig. 2. The suction and gripper tools are mounted on separate linear actuators, which are used to advance and retract each tool before the manipulation. This arrangement allows separating their functions and increases both versatility and redundancy, as the implementation of per-tool as well as per-item manipulation strategies become simpler.

From the mechanical design to the software implementation, we harden our system against errors and implement recovery strategies. We implement error avoidance and recovery strategies on both the hardware and software level to make the system more robust. For example, if the drawer gets stuck and cannot be closed, the linear actuator shakes the drawer to make the blocking items fall inside and avoid massive penalties in the score. Though less likely but critical errors are hard to recover from, protocols for quick restart should always be prepared.

We use learning-based methods such as YOLO v2 [26, 27] and feature-based methods such as color histogram, bounding box volume, weight, and histogram of
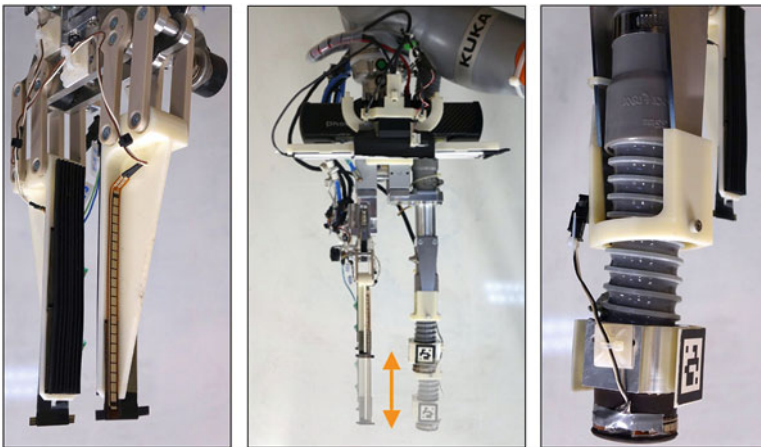

Fig. 2 The end effector (middle) consists of a suction tool (right) and gripper tool (left), each of which can move. The protruding hose contracts upon suction, which pulls the aluminum suction cup casing against the yellow ABS ring. Two markers on the casing allow the suction cup to be located easily during the recognition phase. A force-sensitive resistor (FSR) on the outside of the suction cup acts as a contact sensor. On the gripper, two protruding FSRs detect contact, and sensors underneath the rubber surface measure the contact force

oriented gradients [7] to detect and classify the items. The usage of multiple sensing technologies and methodologies helped to compensate for the weaknesses of each. For example, combining weight and volume by training a support vector machine yielded high accuracy in the recognition.

3.3 Development Strategy

In a first step, we aim to develop a minimal system that performs under limited and controlled conditions, so that we can test the basic functionality. We then iterate through short design and prototyping cycles, testing new ideas as early in the development process as possible.

In terms of hardware, we use the KUKA LBR iiwa [17] with a 14 kg payload, which has torque sensors in each of its seven joints. We use 3D printed parts and aluminum frames that are quickly accessible, versatile, and durable, and Arduino microcontrollers for both prototyping and deployment.

In our solution, the robot uses motion primitives to move between known configurations, as well as a planner [29] to pick up and deliver the items. This improves the overall performance of the robot, simplifies the high-level planning, and avoids getting stuck if planning solutions are time-consuming.

We use ROS to simplify the development, facilitate the integration, and increase the code reusability. To control the robot arm, we use the *iiwa_stack* [12] package to interface ROS and the KUKA API. We also use Git [30] version control and Docker [8] containers, which make it easy to share code and development environments across multiple devices and programmers. This helped us to quickly recover from an accidental deletion of source code and data that might have had a significant impact in our score, if no such tools have been employed.

The source code of our solution is publicly available online in the official ARC repository.[2]

4　Proposed Solution

The proposed system consists of a custom-made end effector mounted on a serial robotic manipulator, a controlled recognition space formed by an array of RGB-D cameras, and a storage system with weight sensors. The end effector features suction and gripper tools for manipulation, as well as an RGB-D sensor for object recognition and grasping point estimation. An overview of the components of the system is shown in Fig. 3.

---

[2]Warehouse Picking Automation Challenges, Team NAIST-Panasonic: https://github.com/warehouse-picking-automation-challenges/team_naist_panasonic.

In this section, we focus on describing the development of the storage system and the suction tool, which was the most used manipulation tool since 80% of the items are graspable by suction.

4.1 Suction Tool

The suction tool consists of a compliant, partially-constrained vacuum cleaner hose of 150 mm in length and 35 mm in diameter connected to an aluminum tube of 150 mm length and 35 mm in diameter. The tip of the suction tool is made of flexible rubber and has a force-sensitive resistor to detect contact with the items. The suction tool dimensions were determined by analyzing the suction force required to lift the maximum possible weight of an item set in the ARC rules as well as the flow over force ratio to ensure a reliable suction seal.

As shown in Fig. 2, several centimeters of the compliant hose extend beyond the fixation. This compliant connection allows the suction cup to move and incline, thus compensating for angular and position errors as the suction pulls the cup and object together. Furthermore, it enables the robot to move quickly into the containers, without danger of damaging articles during a collision. Lastly, when the suction seal is successful, the hose contracts, which pulls the suction cup casing and the suctioned item towards the fixation, making the formerly compliant connection more rigid and reducing the item movement (e.g., swinging) after suction.
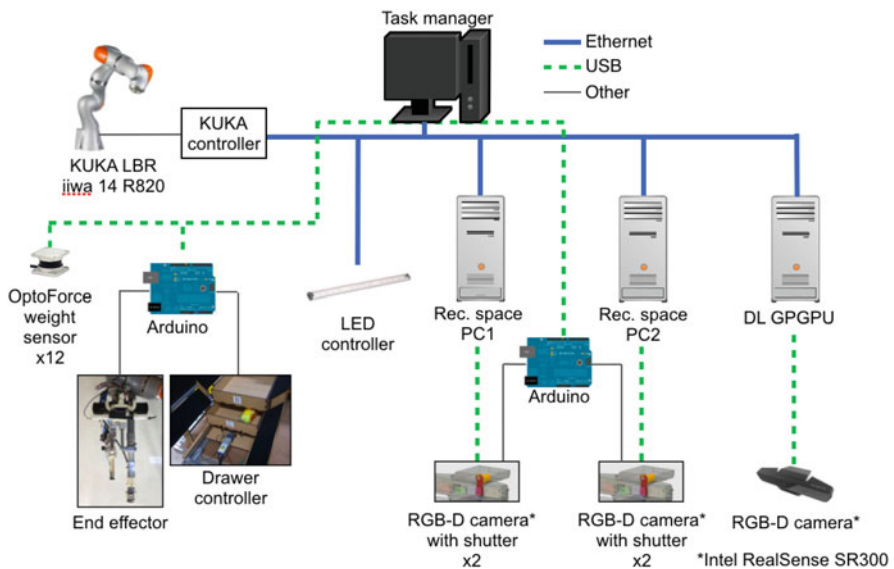

Fig. 3 Overview of the Team NAIST-Panasonic solution deployed at ARC 2017

In past competitions, teams have struggled with designing a suction mechanism with sufficient suction force. Commercial vacuum cleaners and similar solutions do not generate the necessary flow and pressure difference to secure all items. On the other hand, excessive suction force may damage the packaging of the item, e.g., clothes in PVC bags. To respond to this problem and to design a system that can suction all items safely, we have investigated the suction force systematically.

First, we modeled the suction tool as shown in Fig. 4 as a long tube with an opening at the end. Then, with the pressure difference and flow rate of the blower, the hose diameter, the suction cup size, and the relative opening at the end (assuming an imperfect seal), we calculate the resulting normal force. We performed preliminary experiments with suction cups of 30, 40, and 50 mm in diameter d_p, and hoses of 10, 20, 30, 40, and 50 mm in diameter d and 5 m in length L. Figure 5 shows the results.

The combination of $d_p = 40$ mm and $d = 30$ mm had the best performance when tried with all items from the ARC 2017 practice kit: 36 items can be suctioned (90%). The suction force has to be controlled because nine of the items (22.5%) can be damaged by excessive suction force. Thin and cylindrical items (e.g., wine glass, toilet brush, dumbbell) and the mesh cup were the most challenging to pick via suction. Nonetheless, it is notable that even porous, deformable, and irregularly shaped items such as the marbles and the body scrubber could be picked reliably.

The main conclusion is that with sufficient hose diameter and air flow, items can be held even if the suction seal is imperfect, such as when the item surface is uneven or rough. With smaller hose sizes, suction cups of smaller diameters break away significantly earlier, as they cannot transport enough air to sustain a leaking seal.

We decided to use an industrial grade blower [15] with a maximum vacuum of -40 kPa to power the suction tool. We use a pressure sensor to detect when an item has been suctioned and if the item has been dropped (i.e., when the suction seal has broken). To avoid damaging the delicate packaging of items, we added a waste gate with a PD controller to the hose that can regulate the static pressure at the suction cup between 0 and -40 kPa.
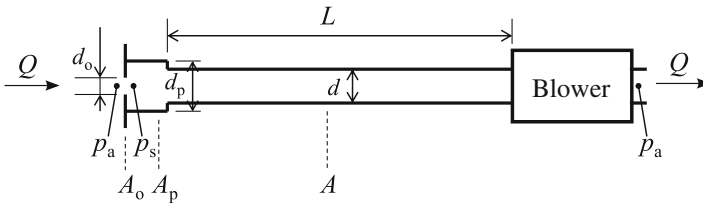

Fig. 4 Simplified model of the suction tool to systematically investigate the suction force. Q is the flow rate, p_a and p_s are the external and internal static pressures, respectively. A, A_p, and A_o are the cross sections of the tube, the suction cup chamber and the suction cup border, respectively, and their corresponding diameters are denoted with d. L is the length of the hose
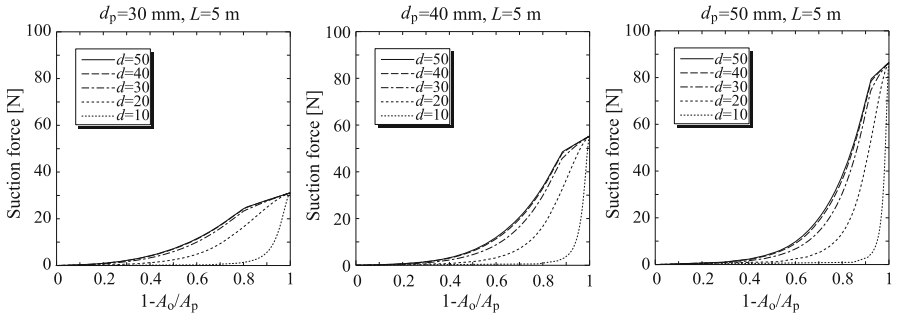
Fig. 5 Effective suction force for an imperfect suction seal, assuming the vacuum machine Induvac VC 355-720. $1 - A_o/A_p$ is 1 for a perfect suction seal, and 0 when no part of the object is in contact. It can be seen that larger hose diameters make the suction seal more robust, as they allow higher air flow
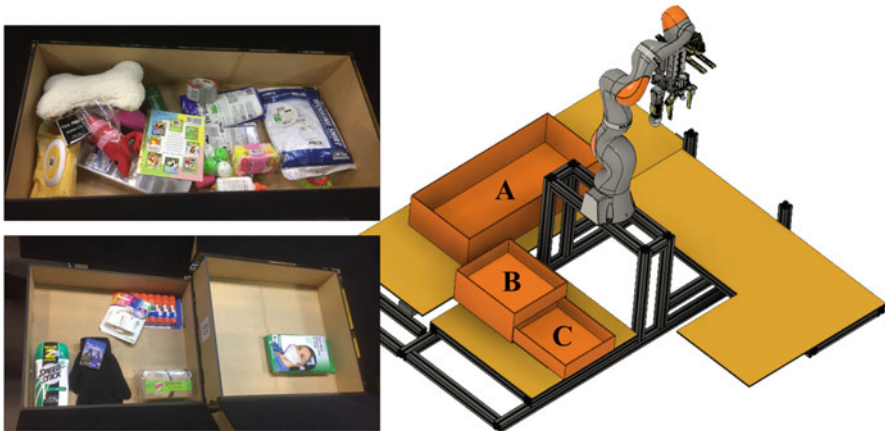

Fig. 6 Storage system featuring three bins. Bin A (top left) is designed to store large items. Bin B and C (bottom left) are shallower to store small items. Bin C is a drawer opened and closed by a linear actuator. The black skirts around the bins prevent items falling out of the storage area

4.2 Storage System

The final version of our storage system consists of three bins. It is made of wood and features one drawer, as shown in Fig. 6.

We started our development with a storage system composed of adjustable aluminum profiles. In retrospect, the ability to adjust division sizes was unnecessary, as we did not end up testing many different configurations. For the first prototype, we fixed the depth of the shelf to accommodate the second largest dimension of the item maximum size, i.e., 27 cm. The other two dimensions were set to

Table 2 Dimensions of the bins used in the horizontally oriented storage system

Name	Length (m)	Width (m)	Depth (m)
Bin A	0.90	0.40	0.19
Bin B	0.40	0.30	0.08
Bin C	0.38	0.29	0.08

an approximated square, making the shelf compliant with the rule of $95,000\,\text{cm}^3$ maximum volume.

During the preliminary tests with this initial design, we realized that it becomes cluttered very easily, and requires precise manipulation to avoid items falling out. We experimented with different configurations and drawers to decrease the clutter and finally opted for a horizontally oriented storage system with three bins, as described in Table 2. It is worth mentioning that most successful teams included a system that increased the available surface area of the storage system to spread out items and reduce clutter.

We aimed to maximize the surface available to place items considering both the maximum allowed volume and the kinematic reachability of the manipulator. To this end, we designed bin C as a drawer which is opened and closed by a linear actuator attached to the robot structure. This drawer holds small items which otherwise would be difficult to find in an unstructured pile of items.

We placed two sets of weight sensors: one under bin A and another under bins B and C. Each set of weight sensors consists of four 3D force sensors [23] attached to a rectangular base made of aluminum frames and can detect changes of up to 2 g. Taking into account the considerable effort and cost to implement a custom weight sensing solution, as well as the hysteresis effects in the force sensors that negatively affect overall precision, we would strongly recommend choosing an off-the-shelf alternative.

The data from the weight sensors is used to discriminate between events such as (un)successful grasping and accidental drops, as well as to classify the items by weight or a combination of weight and other features. Actually, we combined the weight of the item measured by the force sensors, and the bounding box volume calculated in the recognition space with a Support Vector Machine (SVM) for classification. Combining weight and volume not only allows the SVM to include a measure of density into its classification, which is especially helpful to differentiate the numerous light items, but also catches certain edge cases such as clamshell items (e.g., open books) effectively. On the other hand, the bounding box of highly deformable objects can be inconsistent, which makes them vulnerable to misclassification.

Finally, in order to avoid dropping items outside of the bins and to increase the chances of recovery, we installed ramps around the bins that help catch dropped items.

5 Conclusion

We presented our approach to the ARC 2017, with particular emphasis in the lessons learned from past participants, our design philosophy and our development strategy. We also described two particular features of our proposed system, namely the suction tool and the storage system, which we believe played an important role in our performance.

> The main lessons from our experience in the ARC 2017 can be summarized as follows:
>
> - Consider previous competitors' experience to avoid making similar mistakes.
> - Use code and datasets from past competitors to get started.
> - Keep it simple and only do what is necessary. Overengineering and unnecessary redundancy make the system more prone to fail.
> - Use tools such as Git, Docker, and ROS to facilitate development and recovery from unexpected errors.
> - Start with a minimal system that can be iterated upon quickly, to keep prototyping and development cycles short.
> - Develop a robust error recovery as perception and manipulation errors as well as uncertainty are unavoidable.
> - Avoid modifying the code in the last minute, as this leads to human error.
> - Take into account illumination as this significantly affects the object recognition performance in the venue.
> - Protect sensors against overheat and anticipate stability issues.
> - Work on logistics from the beginning as transporting robots is often time-consuming.

Appendix

Table 3 shows the final results obtained by Team NAIST-Panasonic during the ARC 2017.

Table 3 Final results of Team NAIST-Panasonic at ARC 2017

Tasks	Results	Rank	Participants
Stow task	110 points	Fourth place	16 teams
Pick task	80 points	Tenth place	16 teams
Combined final	90 points	Sixth place	8 teams
Overall challenge	Finalist prize	Sixth place	16 teams

References

1. Ackerman, E.: Team Delft wins Amazon Picking Challenge. IEEE Spectr. (2016). https://spectrum.ieee.org/automaton/robotics/industrial-robots/team-delft-wins-amazon-picking-challenge

2. Amazon: Amazon Robotics Challenge (2016). https://www.amazonrobotics.com/#/roboticschallenge/rules

3. Amazon: Amazon Robotics best paper announcement (2018). https://www.amazonrobotics.com/site/binaries/content/assets/amazonrobotics/pdfs/ar-best-paper-announcement.pdf

4. Bohren, J., Cousins, S.: The SMACH high-level executive. IEEE Robot. Autom. Mag. **17**(4), 18–20 (2010)

5. Carfagni, M., Furferi, R., Governi, L., Servi, M., Uccheddu, F., Volpe, Y.: On the performance of the Intel SR300 depth camera: metrological and critical characterization. IEEE Sensors J. **17**(14), 4508–4519 (2017)

6. Correll, N., Bekris, K.E., Berenson, D., Brock, O., Causo, A., Hauser, K., Okada, K., Rodriguez, A., Romano, J.M., Wurman, P.R.: Analysis and observations from the first Amazon Picking Challenge. IEEE Trans. Autom. Sci. Eng. (T-ASE) **15**(1), 172–188 (2018)

7. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), vol. 1, pp. 886–893 (2005)

8. Docker: Docker engine (2013). https://www.docker.com/

9. Domae, Y.: Technology trends and future about picking and manipulation by robots. J. Robot. Soc. Jpn. **35**(1), 13–16 (2017)

10. Eppner, C., Hofer, S., Jonschkowski, R., Martin-Martin, R., Sieverling, A., Wall, V., Brock, O.: Lessons from the Amazon Picking Challenge: four aspects of building robotic systems. In: 26th International Joint Conference on Artificial Intelligence (IJCAI 2017), Melbourne, pp. 4831–4835 (2017)

11. Garcia Ricardez, G.A., von Drigalski, F., El Hafi, L., Okada, S., Yang, P.C., Yamazaki, W., Hoerig, V.G., Delmotte, A., Yuguchi, A., Gall, M., Shiogama, C., Toyoshima, K., Uriguen Eljuri, P.M., Elizalde Zapata, R., Ding, M., Takamatsu, J., Ogasawara, T.: Warehouse picking automation system with learning- and feature-based object recognition and grasping point estimation. In: 2017 SICE System Integration Division Annual Conference (SI 2017), Sendai, pp. 2249–2253 (2017)

12. Hennersperger, C., Fuerst, B., Virga, S., Zettinig, O., Frisch, B., Neff, T., Navab, N.: Towards MRI-based autonomous robotic US acquisitions: a first feasibility study. IEEE Trans. Med. Imaging (T-MI) **36**(2), 538–548 (2017)

13. Hernandez, C., Bharatheesha, M., Ko, W., Gaiser, H., Tan, J., van Deurzen, K., de Vries, M., Van Mil, B., van Egmond, J., Burger, R., Morariu, M., Ju, J., Gerrmann, X., Ensing, R., Van Frankenhuyzen, J., Wisse, M.: Team delft's robot winner of the Amazon Picking Challenge 2016. arXiv Robotics (cs.RO), pp. 1–13 (2016)

14. IEEE: Advances in robotic warehouse automation (2018). http://juxi.net/workshop/warehouse-automation-icra-2018/

15. Induvac: Double stage side channel blowers 3 phase VC 355-720 (2010). http://www.induvac.com/en/downloads/data-sheets/side-channel-blowers/

16. Intel: RealSense Camera SR300 (2016). https://software.intel.com/en-us/realsense/sr300

17. KUKA: LBR iiwa 14 R820 (2013). https://www.kuka.com/en-de/products/robot-systems/industrial-robots/lbr-iiwa

18. MacCormack, A., Murray, F., Wagner, E.: Spurring innovation through competitions. MIT Sloan Manag. Rev. **55**(1), 25–32 (2013)

19. MPRG: Team C^2M: APC RGB-D+PointCloud Dataset 2015 (2015). http://mprg.jp/research/apc_dataset_2015_e

20. Nikkei: Amazon Picking Challenge 2016 in-depth report (Part 1). Nikkei Robotics **14**, 4–16 (2016)

21. Nikkei: Amazon Picking Challenge 2016 in-depth report (Part 2). Nikkei Robotics **15**, 16–22 (2016)

22. Okada, K.: Perspective from international robotics competitions. J. Robot. Soc. Jpn. **35**(1), 9–12 (2017)

23. OnRobot: OptoForce OMD-20-FG-100N (2016). https://onrobot.com/products/omd-force-sensor/

24. Open Source Robotics Foundation: Robot operating system (2009). http://www.ros.org/

25. Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., Ng, A.: ROS: an open-source robot operating system. In: 2009 IEEE Workshop on Open Source Software, Kobe (2009)

26. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. arXiv Computer Vision and Pattern Recognition (cs.CV), pp. 1–9 (2016)

27. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, pp. 779–788 (2016)

28. Rennie, C., Shome, R., Bekris, K.E., De Souza, A.F.: A dataset for improved RGBD-based object detection and pose estimation for warehouse pick-and-place. IEEE Robot. Autom. Lett. (RA-L) **1**(2), 1179–1185 (2016)

29. Sucan, I.A., Chitta, S.: MoveIt! (2011). https://moveit.ros.org/

30. Torvalds, L.: Git (2005). https://git-scm.com/

31. von Drigalski, F., El Hafi, L., Uriguen Eljuri, P.M., Garcia Ricardez, G.A., Takamatsu, J., Ogasawara, T.: Vibration-reducing end effector for automation of drilling tasks in aircraft manufacturing. IEEE Robot. Autom. Lett. (RA-L) **2**(4), 2316–2321 (2017)

32. Zeng, A., Yu, K.T., Song, S., Suo, D., Walker, E., Rodriguez, A., Xiao, J.: Multi-view self-supervised deep learning for 6D pose estimation in the Amazon Picking Challenge. In: 2017 IEEE International Conference on Robotics and Automation (ICRA 2017), Singapore, pp. 1386–1383 (2017)