A Bin-Picking Benchmark for Systematic Evaluation of Robotic Pick-and-Place Systems

Hussein Mnyusiwalla[1], Pavlos Triantafyllou[1], Panagiotis Sotiropoulos[1], Máximo A. Roa[2], Werner Friedl[2], Ashok M. Sundaram[2], Duncan Russell[1], and Graham Deacon[1]

***Abstract*—Pick-and-place operations constitute the majority of today's industrial robotic applications. However, comparability and reproducibility of results has remained an issue that delays further advances in this field. Evaluation of manipulation systems can be carried out at different levels, but for the final application the performance of the overall system is the critical one. This paper proposes a benchmarking framework for pick-and-place tasks, inspired by a typical task in the logistic domain: picking up fruits and vegetables from a container and placing them in an order bin. The framework uses an easy-to-reproduce environment, a publicly available object set, and guidelines for creating scenarios of different complexity. The proposed benchmark is applied to evaluate the performance of four variants of a robotic system with different end-effectors.**

***Index Terms*—Performance Evaluation and Benchmarking; Factory Automation; Grasping**

I. Introduction

ROBOTIC manipulation has been a very popular field of research in the last decades, with numerous groups working on the development of highly dexterous and capable robots. However, comparing advances in research with previous work has proven a challenging task, and reproducibility and repeatability of studies is still a pending topic for enabling effective comparison of developments across multiple groups [1].

There have been several initiatives to address this issue with a number of benchmarks and competitions for manipulation and grasping in particular. Typical examples of system-level evaluations are competitions such as the Amazon Picking Challenge (APC) [2], or the IROS Robotic Grasping and Manipulation Competition [3], where tasks with predefined rules must be accomplished. However, the frequency of these competitions and the limited number of participants remain a drawback. These competitions consider the robotic system as a whole, i.e. the full pipeline of perception, planning and control is evaluated when performing a predefined set of tasks.

Manuscript received: August, 15, 2019; Revised November, 18, 2019; Accepted December, 14, 2019.

This paper was recommended for publication by Editor Ding Han upon evaluation of the Associate Editor and Reviewers' comments.

This work has received funding from the European Union's H2020 programme under grant agreement No. 645599, project SOMA.

[1]Hussein Mnyusiwalla, Pavlos Triantafyllou, Panagiotis Sotiropoulos, Duncan Russell and Graham Deacon are with Robotics Research Team, Ocado Technology, Hatfield, UK. h.mnyusiwalla@ocado.com

[2]Máximo A. Roa, Werner Friedl and Ashok M. Sundaram are with Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany. maximo.roa@dlr.de

Digital Object Identifier (DOI): see top of this page.

Some other system-level benchmarks related to the pick-and-place problem have been proposed for a tabletop scenario [4], and for a pick-from-the-shelf scenario [5], building upon experience from participating in the APC.

The scenario presented by Amazon in the APC is only one instance of warehouse automation tasks in the logistic domain, but the general problem of picking items from a container pops up in multiple companies, e.g. Ali Baba, DHL, Costco, and in general any e-commerce warehouse that tries to find an efficient solution for managing huge amounts of orders in the shortest possible time. With such an abundance of logistic challenges and solutions existing in the industry, attempting to derive a benchmark that covers all of them would be unrealistic. Nevertheless, by creating an easily reproducible protocol and systematic benchmark, a wider audience in the robotic pick-and-place community could be engaged and able to evaluate and/or improve previously published approaches.

This work introduces a new benchmarking framework for pick-and-place actions inspired by the grocery use case of Ocado, the world's largest online-only supermarket. The proposed benchmark builds upon the common scenario of picking items from a container and placing them on an output bin. An initial version of this protocol was previously published in [6]. Compared to that work, the benchmark scenario has been simplified and systematized in a way that easily allows the generation of scenarios with different degrees of difficulty for the picking system, and the metrics that evaluate the tests have been revised to distill a set of highly meaningful measures qualifying the system performance. The benchmark remains still general enough to be applied to test and compare any bin-to-bin pick-and-place integrated system.

The benchmark and protocol templates proposed in the YCB dataset [4] are followed to create the evaluation procedure, and typical objects are identified (Sec. II). The protocol is used to evaluate picking systems using different end-effectors [7], [8], [9] developed within the SOMA project[1], an EU project that aimed to test novel soft manipulation solutions by encouraging interactions with the environment (in the form of environmental constraints) rather than avoiding them, especially in relatively cluttered and geometrically constrained set-ups. The evaluated systems and benchmark results are presented in Sec. III and IV, respectively, and Sec. V concludes the paper.

[1]SOMA project website: http://soma-project.eu/

II. Protocol and Benchmark Description

This section focuses on the description of the evaluated manipulation task (Sec. II-A), describes the setup used in the experiments (Sec. II-B and II-C), provides instructions for the experimental procedure (Sec. II-D), details the protocol requirements in terms of hardware and software (Sec. II-E), and describes how to evaluate the experimental results (Sec. II-F). The proposed tests follow the guidelines of the YCB benchmarks[2], including: i) an assessment protocol that details the procedure and constraints for setting up and performing the experiments, and ii) a set of evaluation metrics that constitute the benchmark. The protocol and evaluation procedure are included as multimedia attachments to this paper.

A. Purpose and Task Description

The first goal of the protocol is to assess the robustness, reliability and operation speed of robotic pick-and-place systems when manipulating objects of the fruit-and-veg class in an industrial grocery setting. To this end, we define a simple *task*: the system must autonomously pick one-by-one all objects placed in a non-mixed storage container, transport and place them in a delivery container in the minimum possible time.

Most robotic systems consist of a large number of software and hardware components (e.g. planning, visual perception, controllers, end-effector, etc.) that work together towards achieving the task at hand. Understanding which components have the most significant role for system performance is crucial for efficient system redesign, as this allows for better work prioritization. Unfortunately, this is usually very difficult, especially when a benchmark only evaluates a system on the completion of the task. To put it simply, just knowing that the pick-and-place task as a whole failed gives limited information on i) why it failed, and ii) what is the most important component to improve in order to avoid failure.

In an effort to facilitate the design process, our protocol defines a gated task decomposition. More specifically, we break down the task in the following phases to allow for better isolation of failures in the pick-and-place pipeline: *pre-grasping*, *grasping*, *transport*, and *placement*, as illustrated in Fig. 1. Pre-grasping is the phase in which the system moves the end-effector from its initial position to the vicinity of the object to be grasped. This phase includes any stages of pre-grasp manipulation necessary to render the object accessible for grasping. The grasping phase consists of the system's attempt to grasp the object, and ends when the object loses contact with the storage crate, i.e. when the end-effector supports all of the object's weight. The transport phase comprises the motion of the system from the moment when the hand fully supports the object, up to when the end-effector is over the placement area. Finally, the placement phase is the stage where the object is placed inside the delivery crate in a controlled manner, i.e. the system *intentionally* releases the object inside the delivery crate from a maximum allowed height ($20cm$) over the crate.

This analysis helps the system developer to focus on smaller parts of the pick-and-place task. This means that, for example, one could first deal with grasping failures before working on potential transport failures. Of course, as we move deeper in the task pipeline, system performance is not only affected by the components active at the current phase, but also implicitly by all phases before that. For instance, a transport failure might be caused only by the robot controller used in the transport phase or by an unstable grasp performed during the grasping phase that the transport controller cannot compensate for. Such component-level correlations are, in general, difficult to foresee a priori, but they could be uncovered and tracked using the per-phase analysis. More specifically, repeating the benchmark after redesigning a single system component should offer the user information on the importance of this component both to the whole task performance and its separate phases (Sec. IV-B).

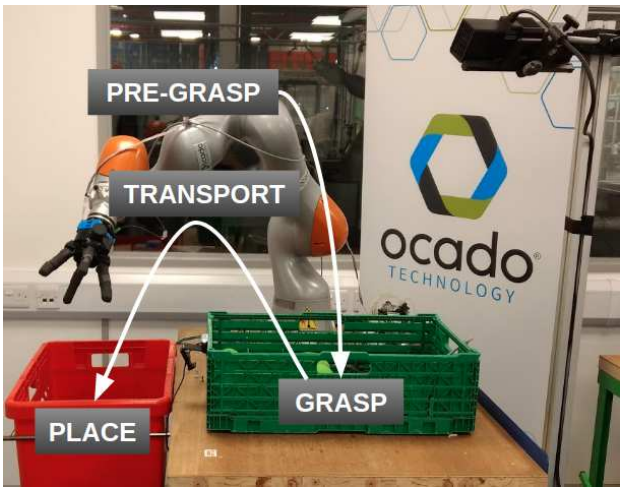

Fig. 1. Phases considered for benchmarking pick-and-place tasks.

B. Setup Description

The experimental setup consists of a storage and a delivery container, and the objects to be manipulated. The containers must have an opening of exactly $60cm \times 40cm$ $(L \times W)$ and a minimum height of $15cm$. For example, in our experiments we used a $60cm \times 40cm \times 18.4cm$ Green Plus 6416 IFCO container for storage, and a $60cm \times 40cm \times 36cm$ custom-made container for delivery. The positioning of the robot and containers in the setup is free, and can vary between systems, as it highly depends on the reachability and workspace of the robotic arm. The pose selected for the storage and delivery containers must be fixed with respect to a static coordinate frame, and must be included in the assessment report.

In our previous work [6], we modified the storage container environment (adding ramps on the walls and an inlay on its surface) in an attempt to investigate the effect of environmental constraints on grasp success. However, in this paper we decided to not allow any such modifications in order to standardize the storage container environment.

We have identified five classes of objects as representative examples of most grocers' fruit and vegetables product range in terms of packaging, shape and weight. Limiting the benchmark to these five objects is required for the tractability

[2]YCB-Benchmarks website: http://www.ycbbenchmarks.com/protocols-and-benchmarks/

Fig. 2. Real and surrogate objects for the proposed protocol.

of the study. The object set for this framework comprises: netbag of limes, mango, loose leaf salad bag, cucumber and punnet (small plastic box) of blueberries (Fig.2), which pose various challenges for both perception and manipulation. The mango, cucumber and punnet were chosen as representative of general classes of objects that resemble basic geometrical shapes (sphere, cylinder and box); solutions for picking these items are very likely general enough to be applied to other objects with the same geometrical characteristics. The netbag of limes behaves as an articulated body, shifting its centre of mass when manipulated, while its accurate segmentation in a cluttered scene is quite difficult even for humans. Also, salads and blueberries have transparent, low-friction and highly deformable packaging. Note also that these objects are difficult for suction cups (which is a solution commonly used in the logistics industry), due to factors such as their geometry or the use of nets or perforated bags that prevent a proper suction strategy.

Even though instances of fruit and vegetables exist in the Food Items YCB Object Set, their weight does not reflect the properties of the actual objects. The proposed object set includes mock-up objects with size and weight close to the real ones, as illustrated in Fig. 2. Of course, other physical characteristics, e.g. texture, are more difficult to emulate. The mock-up mangos, cucumbers, limes and blueberries are 3D printed, while the loose leaves for the salad are made of shredded paper. The fact that our objects are either 3D printed or built from widely available materials guarantees their worldwide availability in the future. To facilitate adoption of the benchmark by the wider community, we commit to maintain a *travelling* object set that will be lent to interested research groups. For those who prefer to reproduce the object set, CAD files and instructions are publicly available[3] and provided as additional material. However, to prevent overfitting to the problem, the benchmarked solutions must not make use of the CAD models of the objects.

C. Object Placement

In order to maximize the reproducibility of the benchmark and the comparability of the results, we introduce 15 predefined scenarios (Fig. 3) that specify the objects' initial poses within the storage container. The scenarios span different levels of clutter and test various conditions of inter-object and object-environment positioning. Our hypothesis is that the complexity of the manipulation task increases as the free space around an object decreases. This can be either the result of clutter (object surrounded by other objects) or proximity to an environmental constraint, e.g. a wall of the container.

The high-clutter scenarios are designed to mimic the initial placement of objects corresponding to optimal packing, as commonly encountered in warehouses (Fig. 3). In a real world scenario, transport of the storage containers through a warehouse will inevitably introduce some disorder to the objects they contain. In the context of this paper, we consider the disorder that occurs naturally once a robotic system starts manipulating objects inside a storage container to be a good enough approximation, as grasping one object will perturb the position of the remaining ones.

Accurate and repeatable positioning of the objects for the initial setup can be performed using the images of Fig. 3 as a guideline. In most of the scenarios, a good level of placement accuracy can be achieved by exploiting the storage container's geometric features (e.g. aligning objects with walls, etc.). In contrast to [5] and [6], we do not use stencils to position the objects, as this slows down the experiments. Moreover, a robust system should be able to cope with variations in the initial pose of the objects. Finally, it should be noted that the scenarios must be considered as unknown to the benchmarked systems (i.e. the system must not assume any knowledge of the initial object poses or ordering inside the container).

D. Experimental procedure

In this section we describe a single execution of a scenario. We first place the objects inside the storage container using the procedure described in Sec. II-C. Then we command the robotic system to autonomously pick and place the objects one-by-one (in any order) until there is no object left to pick, or until the maximum execution time defined for the scenario (see the attached protocol) has been exceeded.

For each pick-and-place cycle we report the success or the type of task failure. Grasping and placing multiple items in a single cycle should be considered as a failure. No external intervention is allowed during the execution, therefore the objects dropped outside of the storage container should not be introduced again. The experiment execution should be stopped in case of a system failure.

E. Software-Hardware Requirements

The proposed protocol is defined in a generic way in order to accommodate as wide a range of systems as possible while ensuring comparability of results. To this end, the poses of the crates and any other obstacle in the environment and the type of object to be grasped can be considered as known. There are no strict perception capabilities (visual or other) required. For example, a system could use only tactile (instead of visual) information to perform the task. However, the system should be able to understand whether the storage container is empty or not in order to automatically stop picking.

[3]Benchmark repository: https://github.com/SoMa-Project/pick_and_place_benchmarking_framework

Fig. 3. First three rows: Protocol scenarios sorted by object type and amount of clutter. Last row: Real placement scenes.

F. Metrics for System Performance Evaluation and Introspection

The benchmark evaluates the system's performance over multiple executions of the different scenarios presented above. The success rate R used to evaluate the system's performance in each scenario is defined by:

$$R = \frac{n_p}{n_0}$$

where n_p is the number of objects successfully picked, transported and placed, and n_0 is the initial number of objects in the storage crate.

To allow for a more complete view of the system's performance, the following data should also be reported for each experiment:

- Mean picks per hour (MPPH). This is one of the most common metrics used in the logistics business for measuring both human and machine efficiency and throughput [10].
- Successful task executions over total attempts (SETA). This is an estimate of the probability that a single task execution attempt is going to be successful.
- Average duration (AVGCT) and standard deviation of duration (STDCT) of a successful pick-and-place cycle.

In a compromise between experimental time and statistical significance, the aforementioned data should be reported over a minimum of three consecutive runs for each experiment. More specifically, the average of the aforementioned metrics over the total number of runs should be reported.

For system introspection purposes, the percentage of the total failures that happen in each of the phases defined in Sec. II-A must be reported, along with non phase-related system failures. Moreover, a list of the system components that are actively used during each phase must also be reported.

III. Description of the Benchmarked Pick-and-Place Systems

In this section we describe four system configurations that have been tested using the proposed evaluation framework. The robot arm, the vision system and parts of the planning pipeline are shared among the system configurations, while we vary the end-effector. The different end-effector capabilities dictates the need for separate grasping controllers, while the end-effectors' physical characteristics require different approaches for grasp planning. Therefore, these four system configurations could be considered as different systems, even if seemingly it is only the end-effector that varies.

A. Benchmarked systems - Hardware components

The four end-effectors used in this work (Fig. 4) were developed within the SOMA project to explore different dimensions of the design space for compliant end-effectors. In what follows we describe the unique characteristics that these end-effectors exhibit and how they affect the planning and control modules of the particular system configuration.

System A - Modified version of Pisa/IIT SoftHand [7]: a humanoid hand with a single degree of actuation, featuring

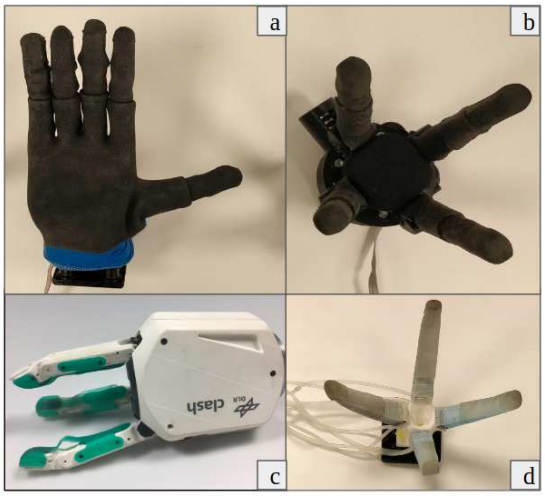

Fig. 4. End-effectors of the benchmarked systems: (a) Pisa/IIT SoftHand, (b) Pisa Softgripper, (c) DLR CLASH Hand, and (d) RBO Gripper.

rigid links connected with soft ligaments, making the hand compliant and robust. The single degree of actuation simplifies the control problem but poses challenges for the planner, since it limits the hand's configurations that one can control.

System B - Pisa Softgripper: a four finger gripper based on the same design and actuation principles as the Pisa/IIT SoftHand. A gripper design is better suited for grasping from within a container scenario such as the one that we consider here. The smaller footprint could simplify the planning problem on a cluttered scene.

System C - DLR CLASH Hand: a three-finger (thumb and two fingers) gripper developed by the German Aerospace Centre (DLR) [8]. The gripper uses variable-impedance actuation inspired by the Awiwi hand in the DLR humanoid DAVID [11]. The thumb has 3 DOFs with a N+1 tendon coupling driven by 4 motors, while the other two fingers have also 3 DOFs actuated by 4 motors for both fingers. The hand has a proximity sensor integrated in its palm. The additional degrees of actuation and the extra sensing allow different grasping behaviours, depending on the type of objects and the complexity of the scene.

System D - RBO Gripper: a soft gripper based on modular PneuFlex actuators developed for the RBO Hand 2 [9]. It has four pneumatically actuated two-chamber fingers built with soft materials, and is actuated using six control valves. The inherent compliance of the fingers allows safe interactions with the environment while conforming with the shape of the objects being grasped.

All robotic systems are based on a KUKA LBR iiwa 14 R820 robot arm equipped with a 6-axes Optoforce Force-Torque sensor and a 3D-printed attachment to mount the different end-effectors. An external Kinect2 depth sensor is used to perceive the contents of the storage container.

B. Vision and planning pipeline

Using a single RGB-D image of the current scene, the systems' visual perception pipeline estimates the pose of the storage container as well as the bounding boxes and the poses of the objects that are placed inside it.

The output of the perception component is the input to the task planning component, which uses additional knowledge of the type of object to be grasped and the end-effector characteristics to devise a kinematically feasible plan. The planning pipeline is based on [12] and can plan grasps exploiting environmental constraints (ECs). In this case, the ECs are the walls of the storage container. In our case, the generated plan is performed in an open-loop manner, using the wrist mounted Force-Torque sensor to sense contacts with the object to be grasped and the environment (neighbour objects and storage container).

IV. Experiments and Results

This section presents and analyses the benchmark results for the systems described in Sec. III.

A. System Evaluation Results

The benchmarking results for the evaluated systems are presented in Table I, with the highest success rate R per scenario across the four systems highlighted in green. Also, for an overall comparison of the different systems, we averaged R per object across scenarios. As illustrated in Fig. 5, the Pisa Softgripper performs best for the mango, the netbag of limes and the salad bag, closely followed by the RBO Gripper. The DLR CLASH Hand performs best for the cucumber; its additional degrees of actuation allow system C to pre-shape the hand more precisely to fit the shape of the cucumber, which leads to better performance in cluttered scenarios.

From the above, it is apparent that the overall performance of the grippers is better than that of the Pisa/IIT SoftHand for this task. That could be partially explained by the constrained storage container environment that favours top grasp approaches. In these cases, the large footprint of the modified version of the Pisa/IIT SoftHand used in System A leads to unwanted interactions of the hand with the neighbouring objects and, eventually, to less successful grasps. On the other hand, Systems B and D were not tested at all on scenarios P1 - P3 because the Pisa Softgripper and the RBO Gripper were not able to grasp the punnet of berries due to their small aperture. As a result, in these scenarios the Pisa/IIT SoftHand achieved the highest success rate.

As far as the rest of the benchmarking metrics are concerned, we should note that the evaluated systems are still in development and are not optimized for speed of operation. For example, the robotic arm operates at a low velocity to prevent hardware damage in case of system failure. As a consequence, AVGCT is quite high and MPPH is quite low, even in the cases where a system manages to complete the task.

For medium-clutter scenarios, even though the objects are close to each other initially, there is still enough free space in the storage container for the systems to perform their manipulation strategies mostly unobstructed. For this reason the results in Table I are similar between low-clutter and medium-clutter scenarios. However, that is not the case for high-clutter scenarios: the lack of free space for the end-effectors' fingers

TABLE I
SYSTEM EVALUATION RESULTS

Object	Mango			Netbag of limes			Cucumber			Punnet of berries			Salad bag		
Scenarios	M1	M2	M3	L1	L2	L3	C1	C2	C3	P1	P2	P3	S1	S2	S3
System A (Pisa/IIT SoftHand)															
Avg. R	2.67/3	2.67/4	0/18	2/3	1.33/4	6/15	1.67/3	2/4	0/9	2.33/3	2/4	0/9	2.67/3	2.67/4	0.67/6
MPPH	27.25	17.34	0	19.11	8.76	12.20	14.13	13.37	0	22.78	17.38	0	38.32	29.79	16.89
Avg. SETA	0.62	0.38	n/a	0.46	0.21	0.29	0.29	0.27	n/a	0.39	0.20	n/a	0.89	0.62	0.40
AVGCT (s)	88.47	92.58	n/a	89.28	84.17	89.49	93.14	84.95	n/a	83.01	84.07	n/a	84.67	83.71	86.25
STDCT (s)	7.85	7.50	n/a	2.77	0.51	8.68	0.44	3.53	n/a	3.10	1.54	n/a	6.33	2.03	1.99
System B (Pisa Softgripper)															
Avg. R	3/3	3.33/4	18/18	2.33/3	3/4	6/15	1.33/3	2/4	0/9	n/a	n/a	n/a	3/3	4/4	4/6
MPPH	40.42	24.77	23.76	28.45	23.38	24.77	11.39	12.52	0	n/a	n/a	n/a	41.03	44.19	31.41
Avg SETA	0.90	0.50	0.51	0.64	0.50	0.55	0.22	0.25	0.00	n/a	n/a	n/a	0.90	1.00	0.67
AVGCT (s)	80.97	83.48	82.65	81.26	79.81	80.50	75.31	77.49	n/a	n/a	n/a	n/a	80.26	81.46	78.78
STDCT (s)	4.34	2.10	3.28	5.36	4.31	2.30	1.11	1.30	n/a	n/a	n/a	n/a	5.16	4.51	1.52
System C (DLR CLASH Hand)															
Avg. R	2.33/3	3/4	7/18	1/3	1.67/4	9/15	2.67/3	2.33/4	1.93/9	2/3	1/4	0/9	3/3	4/4	3/6
MPPH	24.59	20.50	13.17	9.47	14.05	14.35	35.02	15.15	5.22	16.43	6.29	0	36.27	32.14	23.94
Avg. SETA	0.64	0.53	0.32	0.25	0.36	0.39	0.89	0.35	0.12	0.40	0.14	n/a	0.90	0.86	0.43
AVGCT (s)	99.04	95.40	98.44	94.99	106.27	101.64	93.68	96.02	93.07	89.11	89.30	n/a	93.45	95.71	94.55
STDCT (s)	4.81	3.83	3.09	0	0.91	3.63	4.57	7.02	1.35	2.28	0	n/a	2.97	6.03	4.36
System D (RBO Gripper)															
Avg. R	2.67/3	4/4	15/18	2/3	2.67/4	8/15	2/3	1.33/4	0/9	n/a	n/a	n/a	1.67/3	3.33/4	6/6
MPPH	28.60	29.66	22.58	25.13	26.26	17.95	21.54	10.76	0	n/a	n/a	n/a	28.61	39.36	42.64
Avg. SETA	0.47	0.5	0.48	0.5	0.38	0.38	0.21	0.17	n/a	n/a	n/a	n/a	0.56	0.83	1
AVGCT (s)	80.72	79.27	82.11	76.82	80.51	81.55	70.53	80.67	n/a	n/a	n/a	n/a	82.01	80.78	84.42
STDCT (s)	4.31	6.7	7.16	1.21	11.9	3.25	4.39	8.07	n/a	n/a	n/a	n/a	0.78	1.16	5.84

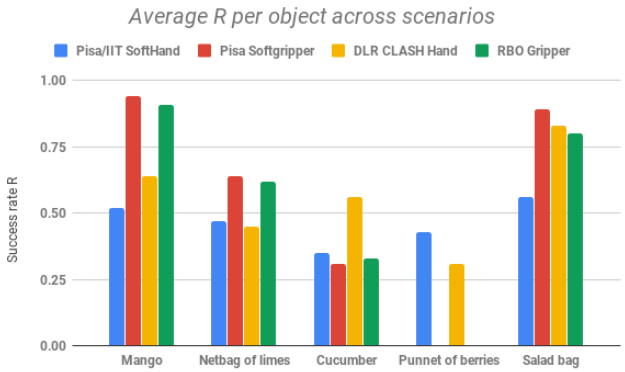

Fig. 5. Success rate R per object averaged across scenarios.

to cage the first object in the storage container and the fact that the configuration of the scene cannot be inadvertently changed (as there is no free space for the objects to move) mean that a system without highly intelligent manipulation strategies and/or hardware design will fail. That is what occurs on scenario P3 for all systems, scenario M3 for System A and scenario C3 for Systems A, B and D, where we have tightly packed, fairly rigid objects. On the other hand, this does not happen for scenarios L3 and S3 because of the deformability of the objects. All in all, the evaluation results confirm our hypothesis presented in Sec. II-C; the complexity of the manipulation task increases with the level of clutter.

B. System Introspection Results

An analysis per phase and component can guide system development by offering insights on which system component or task phase should the system developer focus on to increase system performance. For our tests, Table II shows the per-phase result, the components-per-phase breakdown is shown in Table III, and the type of failures averaged across all scenarios is illustrated in Fig. 6.

For the systems evaluated in this paper, the only input to the task planning component is visual perception of the environment, more specifically the output of the *object segmentation*, the *object pose estimation* and the *containers' pose estimation* components (see Table III). For this reason, the performance of these three components (which can be defined as the accuracy of the information they provide to the task planning component) is crucial to the success of the pick-and-place task and affects all its phases. Unfortunately, especially for the *object pose estimation* component, quantitative evaluation of its performance during the execution of the pick-and-place task is very difficult due to practical problems for obtaining the ground truth (i.e. actual pose of the objects in the storage container). Also, in some systems the evaluation of the perception might be hard to detach from the evaluation of the whole system, as it typically happens in learning-based approaches for picking tasks [13]. The performance of individual components such as *object segmentation* and *object pose estimation* can be evaluated using existing benchmarks [14], [15], but as the context (objects and environment) is different, only a coarse performance estimate can be acquired that way.

Nevertheless, using the results of Table II one can evaluate the effect that a change in any component has on the system's performance. More specifically, by performing the benchmark before and after a change in a certain component and with all the other components remaining the same, one can see

TABLE II
SYSTEM INTROSPECTION RESULTS

Object	Mango			Netbag of limes			Cucumber			Punnet of berries			Salad bag		
Scenarios	M1	M2	M3	L1	L2	L3	C1	C2	C3	P1	P2	P3	S1	S2	S3
System A (Pisa/IIT SoftHand)															
Pre-Grasp fail	0 %	0 %	n/a	0 %	0 %	0 %	8 %	6 %	n/a	45 %	62 %	n/a	0 %	40 %	0%
Grasp fail	100 %	92 %	n/a	86 %	80 %	60 %	84 %	88 %	n/a	45 %	38 %	n/a	0 %	20 %	78 %
Transport fail	0 %	8 %	n/a	14 %	13 %	26 %	8 %	6 %	n/a	10 %	0 %	n/a	100 %	40 %	22%
Placement fail	0 %	0 %	n/a	0 %	7 %	7 %	0 %	0 %	n/a	0 %	0 %	n/a	0 %	0 %	0%
System fail	0 %	0 %	n/a	0 %	0 %	7 %	0 %	0 %	n/a	0 %	0 %	n/a	0 %	0 %	0%
System B (Pisa Softgripper)															
Pre-Grasp fail	0 %	10 %	0 %	0 %	0 %	0 %	7 %	6 %	n/a	n/a	n/a	n/a	0 %	0 %	0 %
Grasp fail	100 %	80 %	94 %	50 %	67 %	60 %	93 %	88 %	n/a	n/a	n/a	n/a	0 %	0 %	50 %
Transport fail	0 %	10 %	6 %	25 %	11 %	13 %	0 %	6 %	n/a	n/a	n/a	n/a	0 %	0 %	0 %
Placement fail	0 %	0 %	0 %	0 %	11 %	7 %	0 %	0 %	n/a	n/a	n/a	n/a	0 %	0 %	0 %
System fail	0 %	0 %	0 %	25 %	11 %	20 %	0 %	0 %	n/a	n/a	n/a	n/a	100 %	0 %	50 %
System C (DLR CLASH Hand)															
Pre-Grasp fail	0 %	0 %	0 %	0 %	0 %	0 %	0 %	15 %	0 %	0 %	0 %	n/a	0 %	0 %	62%
Grasp fail	100 %	62 %	67 %	33 %	67 %	50 %	100 %	76 %	77 %	89 %	83 %	n/a	0 %	0 %	25%
Transport fail	0 %	25 %	13 %	56 %	0 %	21 %	0 %	9 %	23 %	11%	17%	n/a	0 %	0 %	0%
Placement fail	0 %	0 %	7 %	11 %	0 %	0 %	0 %	0 %	0 %	0 %	0 %	n/a	0 %	0 %	13%
System fail	0 %	13 %	13 %	0 %	33 %	29 %	0 %	0 %	0 %	0 %	0 %	n/a	100 %	100 %	0%
System D (RBO Gripper)															
Pre-Grasp fail	56%	50 %	6 %	17 %	62 %	8 %	78 %	35 %	n/a	n/a	n/a	n/a	50 %	50 %	0%
Grasp fail	22 %	17 %	38 %	66 %	22 %	62 %	22 %	55 %	n/a	n/a	n/a	n/a	50 %	0 %	0%
Transport fail	22 %	33 %	25 %	0 %	8 %	14 %	0 %	5 %	n/a	n/a	n/a	n/a	0 %	0 %	0%
Placement fail	0 %	0 %	0 %	0 %	0 %	8 %	0 %	0 %	n/a	n/a	n/a	n/a	0 %	0 %	0%
System fail	0 %	0 %	31 %	17 %	8 %	8 %	0 %	5 %	n/a	n/a	n/a	n/a	0 %	50 %	0%

TABLE III
ACTIVE COMPONENTS PER TASK PHASE

Components	Pre-Grasping	Grasping	Transport	Placement
Object segmentation	✓	-	-	-
Object pose estimation	✓	-	-	-
Containers' pose estimation	✓	-	-	-
Motion planning	✓	-	✓	-
Robot control	✓	✓	✓	✓
End-effector control	✓	✓	✓	✓
Task planning	✓	-	-	-

what the effect is both on overall task success and on phase success. The latter is extremely useful, as a change to a certain component might be beneficial, but the components used later on might dampen the overall performance gain (e.g. a change in the *object segmentation* component can reduce grasp failures, but if the *robot control* component used during transport leads to all the objects being dropped, no change is observed in overall system performance).

In what follows, we will give some examples of how our empirical observations as designers/users of the benchmarked systems are reflected in Table II. The plans produced by the task planner in the benchmarked systems lack pre-grasp manipulation actions. This can be problematic in cluttered environments (where object singulation might be required prior to grasping) or in the presence of ECs that prevent the end-effector from approaching some objects from certain directions. Systems A, B and C mitigate the effects of this issue by employing strategies that exploit ECs instead of avoiding them. However, some of these strategies are not available for system D and for the punnet of blueberries for system A (see scenarios P1 & P2), thus leading to a high

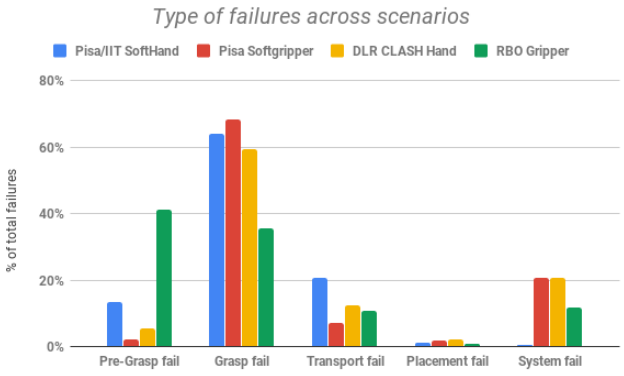

Fig. 6. Type of failures averaged across all scenarios.

percentage of failures in the pre-grasping phase (Fig. 6).

For the benchmarked systems, the way an object has been grasped is not taken into account when computing the transport motion of the robotic arm. At best, this means that the robot might not execute a trajectory that is optimized for preserv-

ing the grasp. At worst, the robot's motion might introduce additional perturbations to the grasp. As a consequence, there is a significant percentage of transport failures in cases where the objects are small and deformable (scenarios L1-L3 for all systems) or at the limit of the end-effector's aperture (see scenarios M1-M3 for system D). This indicates the importance of monitoring the object/end-effector interaction after the grasping phase is completed, and developing intelligent transport controllers that prevent object slippage or collision of the object with the environment. As far as the placement phase is concerned, there is a very small percentage of failures because of its low complexity.

Finally, it is worth noting that there are also some system failures that are not phase-related. These are robot controller failures due to joint limits, or motion planning failures due to not taking into account the grasped object when computing collision-free trajectories. Dealing with these failures is important so that one can instead focus on task-specific problems.

V. Conclusions and Future work

In this paper we proposed a benchmarking framework (protocol and benchmark) for system-level evaluation of robotic pick-and-place systems. The protocol is based on a well-defined task and an easy-to-replicate experimental setup. The use of a standard object set allows for reproducible and comparable experiments. Also, the object set could easily be extended to cover domains other than the fruit and vegetables case covered here. We proposed a set of measures qualifying system performance on a number of standardized scenarios with different degrees of difficulty. Finally, we broke down the pick-and-place task into phases to allow pinpointing and reporting per-phase and system failures as part of our benchmark; this helps to map the problem domain more efficiently compared to masking possibly critical failures behind a single success rate.

As a working example, we used the proposed framework to evaluate four prototype systems developed within the SOMA project, based on three grippers and one anthropomorphic hand. We compared the overall performance of the systems across five object categories and demonstrated the use of the framework as a tool for system introspection and redesign.

Future iterations of the proposed framework will focus mainly on i) revisiting placement requirements, and ii) including a damage metric. As far as placement is concerned, our current requirements (i.e. being above the delivery container with the object in hand) constitute the absolute minimum if a system is to achieve more complex placement objectives (e.g. place the objects at specific locations in the delivery container). We intend to look for objectives that balance generality with realism/complexity. Concerning damage, delivering products intact is a necessity for a pick-and-place system (especially in the case of fruits and vegetables). We have already done preliminary work in this direction during the SOMA project. However, equipping end-effectors with sensors that offer repeatability and comparability of results is still an open problem that hinders adoption of such metrics.

References

[1] F. Bonsignorio and A. P. del Pobil, "Toward replicable and measurable robotics research," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 32–35, 2015.

[2] N. Correll, K. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. Romano, and P. Wurman, "Analysis and observations from the first Amazon Picking Challenge," *IEEE Trans. Automation Science and Engineering*, vol. 15, no. 1, pp. 172–188, 2018.

[3] Y. Sun, J. Falco, N. Cheng, H. Chosi, E. Engeberg, N. Pollard, M. Roa, and Z. Xia, "Robotic Grasping and Manipulation competition: task pool," in *Robotic grasping and manipulation*, Y. Sun and J. Falco, Eds. Springer-Verlag, 2018, pp. 1–18.

[4] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: Using the Yale-CMU-Berkeley object and model set," *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 36–52, 2015.

[5] J. Leitner, A. W. Tow, N. Sünderhauf, J. E. Dean, J. W. Durham, M. Cooper, M. Eich, C. Lehnert, R. Mangels, C. McCool, P. T. Kujala, L. Nicholson, T. Pham, J. Sergeant, L. Wu, F. Zhang, B. Upcroft, and P. Corke, "The ACRV picking benchmark: A robotic shelf picking benchmark to foster reproducible research," in *Proc. IEEE Int. Conf. Robotics and Automation - ICRA*, 2017, pp. 4705–4712.

[6] P. Triantafyllou, H. Mnyusiwalla, P. Sotiropoulos, M. A. Roa, D. Russell, and G. Deacon, "A benchmarking framework for systematic evaluation of robotic pick-and-place systems in an industrial grocery setting," in *Proc. IEEE Int. Conf. Robotics and Automation - ICRA*, 2019, pp. 6692–6698.

[7] M. Catalano, G. Grioli, E. Farnioli, A. Serio, C. Piazza, and A. Bicchi, "Adaptive synergies for the design and control of the Pisa/IIT SoftHand," *Int. J. Robotics Research*, vol. 33, no. 5, pp. 768–782, 2014.

[8] W. Friedl, H. Höppner, F. Schmidt, M. A. Roa, and M. Grebenstein, "CLASH: Compliant low cost antagonistic servo hands," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems - IROS*, 2018, pp. 6469–6476.

[9] R. Deimel and O. Brock, "A novel type of compliant and underactuated robotic hand for dexterous grasping," *Int. J. Robotics Research*, vol. 35, no. 1-3, pp. 161–185, 2016.

[10] J. Mahler, R. Platt, A. Rodriguez, M. Ciocarlie, A. Dollar, R. Detry, M. A. Roa, H. Yanco, A. Norton, J. Falco, K. van Wyk, E. Messina, J. Leitner, D. Morrison, M. Mason, O. Brock, L. Odhner, A. Kurenkov, M. Matl, and K. Goldberg, "Guest editorial open discussion of robot grasping benchmarks, protocols and metrics," *IEEE Trans. Automation Science and Engineering*, vol. 15, no. 4, pp. 1440–1442, 2018.

[11] M. Grebenstein, A. Albu-Schäffer, T. Bahls, M. Chalon, O. Eiberger, W. Friedl, R. Gruber, S. Haddadin, U. Hagn, R. Haslinger, H. Höppner, S. Jörg, M. Nickl, A. Nothhelfer, F. Petit, J. Reill, N. Seitz, T. Wimböck, S. Wolf, T. Wüsthoff, and G. Hirzinger, "The DLR hand arm system," in *Proc. IEEE Int. Conf. Robotics and Automation - ICRA*, 2011, pp. 3175–3182.

[12] C. Eppner, R. Deimel, J. Alvarez, M. Maertens, and O. Brock, "Exploitation of environmental constraints in human and robotic grasping," *Int. J. Robotics Research*, vol. 34, no. 7, pp. 1021–1038, 2015.

[13] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, 2019.

[14] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Europ. Conf. Computer Vision - ECCV*, 2014, pp. 740–755.

[15] T. Hodan, F. Michel, E. Brachmann, W. Kehl, A. G. Buch, D. Kraft, B. Drost, J. Vidal, S. Ihrke, X. Zabulis, C. Sahin, F. Manhardt, F. Tombari, T. Kim, J. Matas, and C. Rother, "BOP: benchmark for 6D object pose estimation," in *Proc. Europ. Conf. Computer Vision - ECCV*, 2018, pp. 19–35.