Towards Automated Order Picking Robots for Warehouses and Retail

Richard Bormann[1], Bruno Ferreira de Brito[2], Jochen Lindermayr[1], Marco Omainska[1], and Mayank Patel[1]

[1]Fraunhofer IPA, Robot and Assistive Systems, 70569 Stuttgart, Germany.
<first name>.<last name>@ipa.fraunhofer.de
https://www.ipa.fraunhofer.de

[2]Delft University of Technology, Department of Cognitive Robotics, The Netherlands.
bruno.debrito@tudelft.nl
https://www.tudelft.nl/en/

Abstract. Order picking is one of the most expensive tasks in warehouses nowadays and at the same time one of the hardest to automate. Technical progress in automation technologies however allowed for first robotic products on fully automated picking in certain applications. This paper presents a mobile order picking robot for retail store or warehouse order fulfillment on typical packaged retail store items. This task is especially challenging due to the variety of items which need to be recognized and manipulated by the robot. Besides providing a comprehensive system overview the paper discusses the chosen techniques for textured object detection and manipulation in greater detail. The paper concludes with a general evaluation of the complete system and elaborates various potential avenues of further improvement.

Keywords: Order Picking · Object Localization · Robot Vision

1 Introduction

Order picking is commonly considered a process that accounts for the largest share of manual labor and the highest costs of up to 55% of total warehouse operating expenses [16]. Operational Research has a long history in optimizing processes in order picking to lower costs, handling time, and reliability [10]. Besides assistive technical tools for the human worker like electronic guidance or virtual glasses, robotics opens up a new dimension of optimizing the order fulfillment process. Many new robotics companies emerged throughout the last years targeting different aspects in warehouse and retail shop automation. Kiva Robotics started with robots capable of relocating shelves inside the warehouse, e.g. for moving the right shelves to pick and pack stations. The Swisslog AutoStore system designs the warehouse as a large block of columns that can hold a stack of storage boxes each. Small robots operate on top of this rack accessing these boxes by pulling them up and transporting them to a pick and pack station likewise. In both cases, however, picking items from the shelves or boxes is still manual labor.

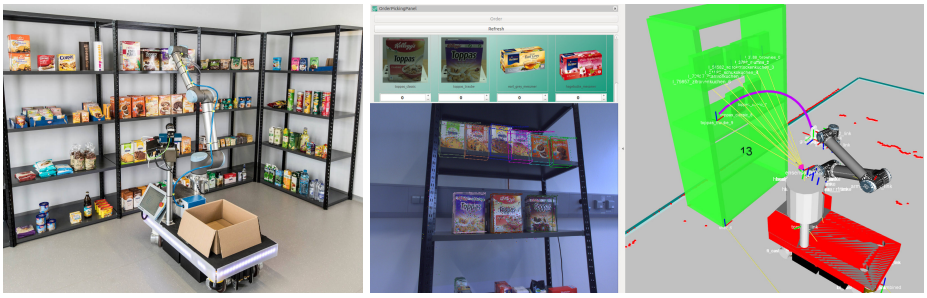

Fig. 1. Left: Automated order picking with a rob@work 3 platform. Right: RViz display of order picking with ordering GUI in upper left corner, detected objects with pose estimates in lower left corner and grasping trajectory to object "toppas_traube" with collision scene on the right side.

First order picking robots with specific applications have been introduced by Magazino. The Magazino TORU robot can handle cuboid items such as books or shoe boxes whereas the model SOTO can manipulate small load carriers. From Fetch Robotics customers can obtain a research platform for applications such as order picking. In order to accelerate research on automated order picking from nearly unstructured storage bins, Amazon initiated the Amazon Picking Challenge in 2015 [4,6] which has seen lively competition throughout the last years [12,20]. The system introduced in this paper is conceived as a mobile robot for automatic order picking tasks in a wide variety of applications in the retail and warehousing domains. Supermarkets and retail stores begin to offer their products at online shops and ship orders home to their clients. This paper focuses on such an application when our mobile robot is supposed to go shopping in a retail store or warehouse for collecting an online order. Another application directly emerging from a robot with such capabilities is the verification of stock levels and wrongly placed items in the shop.

This paper explains a complete order picking system for retail store or warehouse order fulfillment on textured retail items (Sec. 3). The employed vision system for object detection and localization (Sec. 4) as well as the motion planning procedures for object pick up (Sec. 5) are discussed and evaluated in detail. Eventually, we provide an evaluation on the overall system performance in Sec. 6 and discuss future improvements on system design in Sec. 7.

2 Related Work

Scientific interest in order picking from shelves filled in an unstructured way has been majorly raised by the Amazon Picking Challenge recently [6,4,12,20]. Participants of the Amazon Picking Challenge 2015 indicated that the most challenging components for order picking were vision and manipulation [4]. Consequently, we will discuss these parts of our system in greater detail than the other components.

2.1 Object Perception and Localization

In the past, object detection and localization approaches usually either focused on textured object recognition [8,21] or untextured object recognition [13,9]. The object recognition system utilized in this work is suited for textured objects, which applies to the majority of packaged retail articles, and it extends the local feature point based approach of [8] by the consideration of color data. A good survey on color extensions for feature point descriptors is given in [23]. The RGBSURF method favored in our system was inspired by [11,27]. In the Amazon Robotics Challenge, Convolutional Neural Network based (CNN) methods appear to be prevalent [12,20]. Sometimes the CNN is used to directly estimate the object pose [28] but often it is just employed for bounding box localization of the objects whereas the 3d pose is recovered with e.g. CAD model matching [12] or model-free grasping based on sensed surface geometry [20].

In contrast to those deep learning based methods, our procedure is based on object model matching. We see the major advantage in the simple extensibility of the object database which does not require any further training for adding a model. CNN-based methods struggle to learn new objects within short time periods and may do so even more if large amounts of new items have to be learned [20,2]. The object recording setup described in Sec. 3.6 is capable of physically recording a new object and adding its recognition model to the detection system within less than 2 minutes.

2.2 Motion Planning

Motion planners can be categorized into sampling-based, search-based, and optimization-based. The Open Motion Planning Library (OMPL) [25] is one of the most popular planner frameworks used with MoveIt! [3]. The library features several multi-query, single-query, and sampling-based planners. Multi-query planners build a roadmap of the entire environment that can be used for multiple queries. Single-query planners typically construct a graph of states connected by valid motions. On top of the above, sampling-based planners provide some level of optimization as well.

Optimization based planners like Covariant Hamiltonian Optimization for Motion Planning (CHOMP) [29] and Stochastic Optimization for Motion Planning (STOMP) [15] iteratively improve an initial trajectory while minimizing a cost function responsible for smoothness and obstacle avoidance. Both are highly dependent on the initial seed trajectory and hence sometimes fail at finding a valid motion. Dynamical Movement Primitives (DMP) [14] present a time-independent, scalable trajectory representation that allows start and end states to be changed while maintaining the dynamic characteristics of the motion encoded from demonstrations. DMPs either represent a motion in joint-space or Cartesian-space although rotations in the latter case require special attention [17,26]. For the order picking robot we developed an extension of the STOMP framework termed Guided Stochastic Optimization for Motion Planning (GSTOMP) which learns from demonstration to generate a better initial guess by using DMPs as seed generators.

3 System

This section explains the hardware setup and the employed software components. The application has been embedded in a warehouse or shop-like laboratory environment with typical supermarket items sorted into the shelves (see Fig. 1).

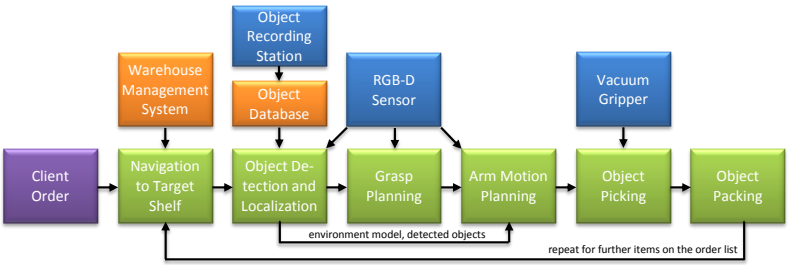

Fig. 2. System overview on the mobile order picking robot.

Robot Hardware As robotic platform the omnidirectional rob@work 3 manufactured by Fraunhofer IPA is used. It carries a Universal Robots UR 10 arm equipped with a Schmalz Cobot Pump vacuum gripper. The robot offers some payload area where a storage or shipping box can be placed. The camera system for object detection is a combination of an Ensenso N30 projection-assisted stereo camera paired with a 5 Megapixel IDS uEye color camera. The vision system is mounted on a pan-tilt unit next to the manipulator.

Robot Behavior Control The robot control program is implemented as a Python control script. The overall control diagram is depicted in Fig. 2. It is explained in the order of called components in the following paragraphs.

Order Placement A simple RViz-based ordering interface emulates the client order, as displayed in Fig. 1 (right). It transfers the types and quantities of ordered articles to the control script, which then queries the warehouse management system for the typical shelf locations where these objects are stored at. The script may choose to optimize the ordering sequence for a short traveling path through the warehouse, but may also consider further constraints such as which objects cannot be placed on top of others in the client box.

Mobile Navigation The navigation system drives the robot to the next target shelf location while avoiding dynamic obstacles on the way. Our mapping and localization system features multi-feature fusion and multi-robot mapping and localization [5]. Here we utilize the grid map together with line features, i.e. mostly wall segments. The navigation system seamlessly scales up to a fleet of robots with our multi-robot cloud navigation extension [1,19].

Object Detection and Localization At the shelf the cameras are adjusted to face the putatively correct section in the shelf as told by the warehouse management system. The object detection and localization procedure is outlined in Sec. 4. The script allows for multiple detection trials on failure and also inspects multiple shelf levels to extend the object search. The necessary object recognition models are retrieved from the system's object database, which can be easily extended with new models during operation.

Object Recording Station In practice, models of thousands of ever-changing retail items and their master data are required. We developed an industrial grade automatic capturing station together with the company Kaptura which can generate colored 3d point cloud and mesh models of arbitrarily textured and untextured objects within 45 s recording time and 45 s 3d modeling time. Required master data such as object weight and bounding box size are captured simultaneously. The station is designed in a modular way and allows for installing defined illumination and multiple cameras such that a reduction in recording time is easily achievable. The obtained 3d data can be used for generating object detection models as well as for showcasing a 3d model at a shop's web page. Fig. 3 shows the recording station and some captured 3d models.

Fig. 3. Kaptura object recording station and four 3d model samples.

Grasp Planning If the object of interest is available multiple times in the shelf, the script tries with the closest instance first and submits the 3d object pose data to the grasp planner. For cuboid packages and cylinders the planner just computes grasp points with a centered and aligned gripper at the 6 bounding box surface centers. These grasp proposals are usually sufficient in that case. However, on arbitrarily shaped objects we refine or discard the grasp poses based on local surface geometry using a surface-normals approach similar to the

method described in [20]. The grasp planner already conducts simple reachability checks based on the captured RGB-D scene data and removes inaccessible grasp pose candidates. Eventually, it outputs a ranked list of possible grasps on the desired object and leaves the motion planner with the choice whether to accept this ranking, revise it or remove grasps due to further scene constraints.

Motion Planning The motion planning component plans a collision-free path towards the grasp pose based on GSTOMP. This procedure is detailed in Sec. 5. The planner iteratively tries to find a path to any of the provided ranked grasp poses until one attempt is successful.

Object Picking The found trajectory is then executed by the arm and the vacuum gripper is activated close to the object. Once a contact can be measured by the vacuum system the arm is retracted along a suitable planned trajectory.

Object Packing Finally, the objects are packed into a storage box using a mixed palletizing method [24] for arranging the objects in a structured and space-saving manner.

If further objects are requested, the procedure continues with step 4 driving to the storage location of the next object on the order list.

4 Object Detection and Pose Estimation

This section explains the object detection and localization system and evaluates its performance on typical retail items.

4.1 Method

The utilized perception system is an extension of our feature-based object detection and localization system from previous work [8] which models objects on the basis of visual feature points which become accumulated into a 3d feature point model.

Due to model matching based on local feature points, this system finds as many objects of an individual kind as present and naturally handles occurring occlusions. Model matching also directly provides a complete 3d object pose estimate with localization in 3d position and angles.

We extended the approach of [8] by a refined model verification procedure and the incorporation of color cues. The original system established object model matches based on a fixed minimum threshold of matching features. This does not generalize well over a diverse set of objects since there is no single optimal number of minimum matching features. The new model verification approach seeks to match a certain percentage of visible model features. We explicitly analyze visibility and occlusions of the model localization hypothesis and only require that the necessary percentage of features is detected on the putative

object surface that would be visible from the given perspective. The dynamic feature matching threshold effects that detection is now more accurate in terms of false positives and adapts flexibly to the number of features of the object model, which may range from a few to several hundred features.

Our second enhancement of the object detection system is a color extension of the formerly gray-scale image-based SURF or sORB features. To this end we collected the following set of possible color extensions for feature descriptors.

SURF+HSV Adds a normalized 12-bin color histogram to the SURF descriptor which counts frequencies of surrounding pixels with white, gray, black or one of 9 basic colors. The neighborhood size is adapted with the descriptor's scale. The color assignment follows a fixed comparison scheme of the pixel's hue, saturation, and value in HSV color space.

SURF+OC Adds a 13-bin color histogram to the SURF descriptor, similar to SURF+HSV. The color definitions are derived from the Opponent Color space.

RGBSURF While the SURF feature points are determined from the gray-scale image, the RGBSURF descriptors are computed individually in the color image's R, G, and B-channels [23,27] yielding a descriptor of length $3 \cdot 64 = 192$.

rgSURF Computes the SURF descriptors on the two channels of the rg color space [23], similarly to RGBSURF.

HSVSURF Computes the SURF descriptors on the 3 channels of the HSV color space [23], similarly to RGBSURF.

OCSURF Computes the SURF descriptors on the three channels of the Opponent Color space (as described in [23,11]), similarly to RGBSURF.

LabSURF Computes the SURF descriptors on the three channels of the L*a*b* color space, similarly to RGBSURF.

4.2 Evaluation

We evaluated the object detection and localization system on a set of 71 textured supermarket articles[1] containing 16 paper boxes, 11 big paper bags (e.g. flour), 22 paper and plastic bags with limited flexibility (e.g. yeast, pudding), 10 plastic bags with medium flexibility (e.g. sultanas, almonds), 2 plastic bottles, 1 plastic can, 5 plastic wrappings, and 4 plastic blisters. Object sizes are well-distributed ranging from 5 cm to 25 cm side lengths. The data set contains 36 RGB-D perspectives per object captured under 3 conditions. First, the objects were recorded with normal indoor illumination from ca. 65 cm distance, similar to the model recording distance. In the second condition objects were captured from ca. 130 cm distance, which is the maximum camera to object distance in the real application, to evaluate scale invariance. Third, the recording distance was ca. 65 cm and images were captured against varying daylight falling into a window behind the objects (recorded on several consecutive days) and a spotlight illumination was installed at the right side to evaluate illumination invariance. A selection of these objects and recording conditions[2] is displayed in Fig. 4.

[1] The dataset is available from the authors upon request (> 500 GB).

[2] The data set was captured with a another but similar recording system before the professional Kaptura recording system was available.

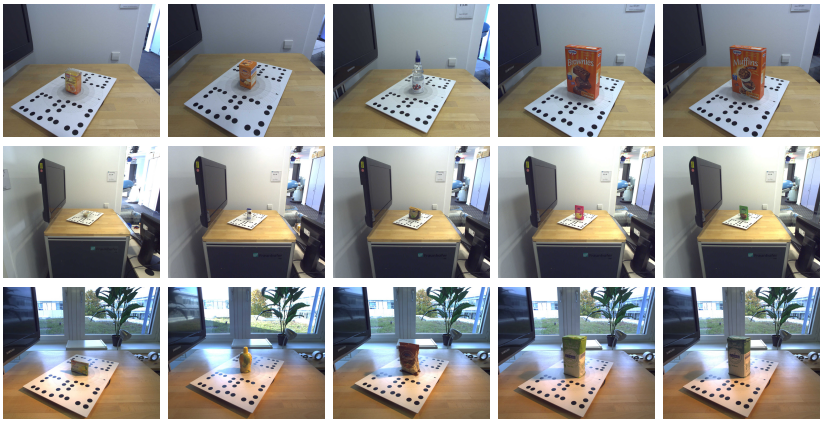

Fig. 4. Exemplary objects from the data set recorded at three conditions.

In the following evaluation we compare SURF and all color descriptor extensions against the SURF descriptor without model verification as described in [8]. For each descriptor, we tuned parameters to achieve the best performance possible. Tab. 1 provides several performance metrics on all 3 test cases. Reported values are averaged over the 36 views of all the 71 objects. ΔPos is the average translation error, and $\Delta\theta$ is the average angular error. Further we report macro recall, precision and f-score. Recall is the percentage of correctly identified objects throughout the database. For the best methods it is above 90% in test case 1. To interpret this number right, please notice that our database also includes side views of the objects. Almost half of the tested objects are quite flat for which these side views do not give any useful visual information. For those, we have at least 6 difficult side views out of 36 views. Hence, it is nearly impossible to detect the object on about 8.33% of the database images. Consequently, a recall above 90% indicates that nearly every object was correctly found whenever visually possible. Precision is the number of correct predictions divided by all predictions.

While all descriptors perform quite well at test case 1, SURF and RGBSURF are the only methods with high robustness to varying distance and illumination. According to the diagonal offset model [18,7], the SURF descriptor is invariant against uniform and chromatic multiplicative intensity changes and additive diffuse illumination shifts. Only RGBSURF retains the same illumination invari-

Table 1. Object detection and localization performance for the 3 test cases

	test case 1 (65 cm)					test case 2 (130 cm)					test case 3 (65 cm, illumination)				
Descriptor	ΔPos	$\Delta\theta$	recall	prec.	f	ΔPos	$\Delta\theta$	recall	prec.	f	ΔPos	$\Delta\theta$	recall	prec.	f
SURF+HSV	**5.1 mm**	**2.84°**	0.928	0.875	**0.901**	26.9 mm	6.58°	0.268	0.497	0.348	**8.2 mm**	**4.34°**	0.334	0.664	0.444
SURF+OC	5.3 mm	2.97°	**0.932**	0.849	0.888	27.2 mm	**6.43°**	0.360	0.559	0.438	**8.2 mm**	4.79°	0.408	0.704	0.517
RGBSURF	6.9 mm	5.13°	0.901	0.868	0.884	27.5 mm	8.00°	**0.546**	**0.815**	**0.654**	9.2 mm	6.23°	**0.665**	**0.798**	**0.725**
rgSURF	8.1 mm	5.70°	0.730	0.889	0.801	26.1 mm	7.39°	0.326	0.756	0.455	9.7 mm	6.79°	0.380	0.766	0.508
HSVSURF	9.4 mm	6.85°	0.757	0.841	0.797	**25.1 mm**	6.60°	0.126	0.445	0.196	12.9 mm	7.11°	0.129	0.642	0.209
OCSURF	7.9 mm	5.81°	0.824	0.882	0.852	26.7 mm	7.04°	0.370	0.801	0.506	9.6 mm	6.17°	0.347	0.744	0.473
LabSURF	7.6 mm	5.24°	0.834	**0.899**	0.865	27.0 mm	6.80°	0.328	0.682	0.443	10.1 mm	5.94°	0.315	0.640	0.422
SURF	6.4 mm	4.64°	0.925	0.838	0.880	27.2 mm	7.34°	**0.597**	**0.828**	**0.694**	8.3 mm	5.50°	**0.699**	**0.807**	**0.749**
SURF [8]	6.7 mm	4.69°	0.925	0.536	0.678	27.1 mm	7.04°	0.564	0.676	0.615	8.5 mm	5.54°	0.694	0.701	0.698

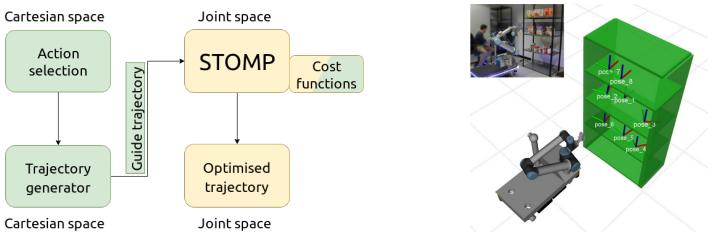

Fig. 5. The architecture of the GSTOMP motion planner (left) and the 8 target poses of the utilized demonstration trajectories in our experiments (right).

ance properties whereas all other extensions are only invariant against uniform intensity variation and/or diffuse shifts and hence perform significantly worse in test case 3. In test case 2, recall is around 55%-60% for the best methods RGBSURF and SURF because half of the database objects are just too small to be detected reliably with the available resolution at this distance (wide angle lens $74° \times 58°$). The real world experiments (Sec. 6) showed that detecting larger objects ($>$ 18 cm side length in longest dimension) worked reliably up to this distance.

The value of the model verification step becomes visible by the large improvement on precision between SURF with and without model verification [8]. Although SURF with model verification outperforms the best color extension RGBSURF slightly, our order picking robot was run with RGBSURF because it could still distinguish objects with same texture but different color design. This is especially relevant on partial occlusions which might hide distinguishing texture. Finally, we like to highlight that the reported detection quality is scene-independent in contrast to some CNN-based solutions [20]. We did not observe any influence on detection quality whether 1 or 20 objects were present in the scene.

5 Motion Planning

In this section we explain the motion planning approach GSTOMP which plans arm movements faster by starting from close-by, previously recorded motion primitives that just need to become refined accordingly to the current task.

5.1 Method

The architecture of GSTOMP is presented in Fig. 5 (left). A Dynamical Movement Primitives (DMP) collection is recorded beforehand, encoding for each pick&place task the necessary motion, such as pick, pull, push etc. In our approach we use Cartesian Dynamical Movement Primitives that is the combination of position and rotation component

$$\tau\ddot{y} = \alpha_z(\beta_z(g_p - y) - \dot{y}) + f_p \tag{1}$$

$$\tau\dot{\eta} = \alpha_z(\beta_z \cdot 2\log(g_o * \bar{q}) - \eta) + f_o \tag{2}$$

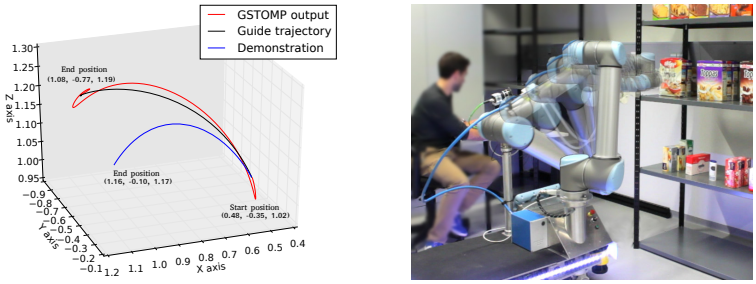

Fig. 6. Example of demonstrated, *guide*, and optimized trajectories (left) and the execution of an optimized trajectory (right).

where y is the current position, η represents the current orientation, τ is a scaling factor for time, $g = (g_p, g_o)$ is the goal pose, α_z and β_z are scaling terms, $q \in \mathbb{R}^4$ is a unit quaternion representation, $*$ denotes the quaternion multiplication and f_p, f_o are forcing terms for position and orientation defined as

$$f(x) = \frac{\sum_{i=1}^{N} \Psi_i(x) w_i}{\sum_{i=1}^{N} \Psi_i(x)} x(g - y_0) \tag{3}$$

where y_0 stands for the initial pose while g is the goal pose, w_i is a weighting for a given Lagrangian basis function Ψ_i, and x is a unit-free time equivalent, converging monotonically from 1 (initial pose) to 0 (target pose).

The action selection block determines which DMP to use at each planning phase of the task execution. The trajectory generator block takes the selected DMP and generates the initial guess, the *guide trajectory*, based on the previously demonstrated motions. To obtain an optimized GSTOMP trajectory as close as possible to the *guide trajectory* we implemented the Dynamic Time Warping (DTW) [22] method as cost function for the stochastic optimization framework of STOMP, introducing a similarity measure between the *guide* and optimized trajectories.

5.2 Evaluation

We evaluate most of the major state-of-the-art planners implemented in the state-of-the-art software for mobile manipulation, MoveIt! [3]: CHOMP, RRT-Connect provided by OMPL, STOMP, and GSTOMP. We specifically compare our method against all different interpolation methods used in STOMP as they are equivalent in purpose to our *guide trajectories*. For the experiments we recorded 8 demonstration trajectories, randomly sampled within the racks of the utilized shelf type (see Fig. 5, right). The DMPs generate *guide trajectories* for the queried arm motion which is in Cartesian space. An example of a *guide trajectory* generated from a demonstrated trajectory as well as the resulting optimized trajectory can be found in Fig. 6.

The planners are compared w.r.t. planning time, number of solver iterations, success rate, and smoothness. The smoothness value is the cumulative function

Table 2. Numerical results from 200 experiments with each motion planning algorithm

Algorithm	Planning time (s)		Smoothness		Success rate	Solver Iterations
	Median	Std dev	Median	Std dev		Mean
CHOMP	0.146	2.118	**0.038**	**0.004**	60%	7.6
RRTConnect	**0.071**	**0.049**	0.136	0.057	65%	-
STOMP (cubic)	0.579	2.711	0.185	0.027	70%	2.84
STOMP (linear)	0.575	1.634	0.172	0.023	67%	3.7
STOMP (min c cost)	0.548	3.049	0.188	0.024	58%	2.15
GSTOMP	3.430	1.838	0.049	0.038	**81%**	**1.16**

of the linear and angular accelerations. The accelerations are approximated using second order center difference formula. The results of 200 experiments with equally distributed random goal poses in all racks of the shelf are presented in Tab. 2. It shows that when a better initial guess is provided to the solver the number of iterations to find a solution is reduced, improving the solver performance, and so GSTOMP presents the best result. On the other hand, due to the recent implementation state of our planner the planning time is still suboptimal, resulting in a lower performance in comparison with the fastest planners, RRT-Connect and CHOMP. However, GSTOMP can still achieve reasonable planning times for a pick&place scenario.

In terms of smoothness, the GSTOMP planner is able to find smoother trajectories than all the other STOMP versions, achieving the same levels of performance as CHOMP. Since this smoothness is computed on the Cartesian path this result proves that GSTOMP manages to stay relatively close to an already smooth *guide trajectory*. Finally, the main advantage of the GSTOMP method is the increase of the success rate of planning by 10% compared to the second best, STOMP with cubic polynomial interpolation initial trajectory. This reduces the number of failed planning attempts significantly and hence increases the performance of any manipulation system. GSTOMP is superior to the other approaches since the *guide trajectories* provide a domain-adapted initial guess of the trajectory whereas the others only rely on unspecific heuristic initial guesses, e.g. STOMP with cubic polynomials.

6 System Evaluation

We evaluated the performance of the whole order picking system with 20 random orders, each containing 8-12 target objects out of the 71 database objects, in our supermarket lab environment with 6 different shelf locations. An RViz visualization on object detection and motion planning is provided in Fig. 1 (right). From these 200 picking tasks we counted the statistics as summarized in Tab. 3.

The objects that could not be found at all were very small objects with few texture. Sometimes, motion planning could not find valid plans when objects were obstructed by others and could not be removed withouth colliding. A complete pick excluding driving the robot to a shelf took 40 s on average (ranging

Table 3. Performance of the order picking system on 200 picks

number	% event
191	95.5% successful picks
5	2.5% object could not be found after 5 attempts
4	2.0% motion planning did not find a plan after 5 attempts

between 35 s and 45 s). Stereo and RGB-D processing took about 5 s, object detection further 3 s, grasp planning accounted for up to 2 s, motion planning for additional 3 s, the grasp execution took 12 s and the packing needed 15 s. Especially the manipulation times can be easily sped up by driving the arm with higher speeds. Likewise, the packing manoeuver is currently suboptimal with turning the arm around its first joint by a full rotation in total.

7 Conclusions and Future Work

In this paper we discussed a mobile order picking robot for order fulfillment applicable to any kind of retail store or warehouse environment. For the initial setup it just needs a navigation map and optionally collision models of the shelves. However, collision scenes for arm motion planning can also be generated online with the onboard 3d sensors. Additionally, a connection to a warehouse management system with knowledge about the usual object storage locations is necessary. The system can handle any textured objects up to 1 kg weight. The objects may be arranged in any ordered or chaotic way. Camera resolution and lens limit the maximum object detection distance. In our case, only objects larger than 18 cm could be detected reliably at 130 cm distance with the 5 MPixel color camera and wide angle lens ($74° \times 58°$). The workspace is only limited by the dexterity of the installed arm, in case of the UR10 we could only reach to objects in the front part for the lowest and topmost shelves. Although performance is already at a good level in a laboratory environment, an industrial application requires even higher dependability. For the system at hand we see the following major steps in achieving that goal.

For reaching to all storage locations in the top and bottom shelves, a different hardware setup with better dexterity is needed. For achieving maximum dexterity we are also working on a GSTOMP extension which can handle full body motion planning, i.e. for mobile platform and arm simultaneously.

We also consider using a gripper-mounted camera for object detection which would allow to get closer to objects of interest and facilitate detection of the smaller items. A light source should be added to the robot for avoiding extreme illumination variance. If needed by the application, the vision system could be complemented with a textured-less object detection method. Last but not least, the implemented error recovery behaviors are not sufficient for grasping obstructed objects. An extension of the order picking system with a reasoning module is necessary for clearing the occluding objects in a task-oriented way.

References

1. Abbenseth, J., Lopez, F.G., Henkel, C., Dörr, S.: Cloud-based cooperative navigation for mobile service robots in dynamic industrial environments. In: Proceedings of the Symposium on Applied Computing. pp. 283–288. ACM (2017)
2. Causo, A., Chong, Z.H., Ramamoorthy, L.: A robust robot design for item picking. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2018)
3. Chitta, S., Sucan, I., Cousins, S.: Moveit! Robotics & Automation Magazine **19**, 18–19 (March 2012)
4. 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 Transactions on Automation Science and Engineering **15**(1), 172–188 (Jan 2018). https://doi.org/10.1109/TASE.2016.2600527
5. Dörr, S., Barsch, P., Gruhler, M., Lopez, F.G.: Cooperative longterm SLAM for navigating mobile robots in industrial applications. In: 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI). pp. 297–303 (Sep 2016). https://doi.org/10.1109/MFI.2016.7849504
6. Eppner, C., Höfer, S., Jonschkowski, R., Martín-Martín, R., Sieverling, A., Wall, V., Brock, O.: Lessons from the amazon picking challenge: Four aspects of building robotic systems. In: Proceedings of Robotics: Science and Systems. AnnArbor, Michigan (June 2016). https://doi.org/10.15607/RSS.2016.XII.036
7. Finlayson, G.D., Hordley, S.D., Xu, R.: Convex programming colour constancy with a diagonal-offset model. In: Proceedings of the IEEE International Conference on Image Processing. vol. 3, pp. III–948–51 (Sep 2005). https://doi.org/10.1109/ICIP.2005.1530550
8. Fischer, J., Arbeiter, G., Bormann, R., Verl, A.: A framework for object training and 6 DoF pose estimation. In: Proceedings of the 7th German Conference on Robotics (ROBOTIK) (May 2012)
9. Fischer, J., Bormann, R., Arbeiter, G., Verl, A.: A feature descriptor for textureless object representation using 2d and 3d cues from rgb-d data. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). pp. 2104–2109 (2013)
10. Gu, J., Goetschalckx, M., McGinnis, L.F.: Research on warehouse design and performance evaluation: A comprehensive review. European Journal of Operational Research **203**(3), 539–549 (2010). https://doi.org/10.1016/j.ejor.2009.07.031
11. Henderson, C., Izquierdo, E.: Robust feature matching in the wild. In: Proceedings of Science and Information Conference (SAI). pp. 628–637 (Jul 2015). https://doi.org/10.1109/SAI.2015.7237208
12. Hernandez, C., Bharatheesha, M., Ko, W., Gaiser, H., Tan, J., Deurzen, K.v., Vries, M.d., Mil, B.V., Egmond, J.v., Burger, R., Morariu, M., Ju, J., Gerrmann, X., Ensing, R., Frankenhuyzen, J.V., Wisse, M.: Team Delft's Robot Winner of the Amazon Picking Challenge 2016. In: RoboCup 2016: Robot World Cup XX. pp. 613–624. Lecture Notes in Computer Science, Springer, Cham (Jun 2016)
13. Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., Lepetit, V.: Gradient response maps for real-time detection of textureless objects. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**, 876–888 (2012)
14. Ijspeert, A.J., Nakanishi, J., Hoffmann, H., Pastor, P., Schaal, S.: Dynamical movement primitives: Learning attractor models for motor behaviors. Neural Computation **25**(2), 328–373 (2013)

15. Kalakrishnan, M., Chitta, S., Theodorou, E., Pastor, P., Schaal, S.: STOMP: Stochastic trajectory optimization for motion planning. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). pp. 4569–4574 (2011). https://doi.org/10.1109/ICRA.2011.5980280
16. de Koster, R., Le-Duc, T., Roodbergen, K.J.: Design and control of warehouse order picking: A literature review. European Journal of Operational Research **182**(2), 481–501 (Oct 2007). https://doi.org/10.1016/j.ejor.2006.07.009
17. Kramberger, A., Gams, A., Nemec, B., Ude, A.: Generalization of orientational motion in unit quaternion space. In: Proceedings of the IEEE International Conference on Humanoid Robots. pp. 808–813 (2016). https://doi.org/10.1109/HUMANOIDS.2016.7803366
18. Kries, J.v.: Influence of Adaptation on the Effects Produced by Luminous Stimuli. In: Sources of Color Science, pp. 120–126. The MIT Press, Cambridge (Jun 1970)
19. Lopez, F.G., Abbenseth, J., Henkel, C., Dörr, S.: A predictive online path planning and optimization approach for cooperative mobile service robot navigation in industrial applications. In: Proceedings of the European Conference on Mobile Robots (ECMR). pp. 146–151 (2017)
20. Morrison, D., Tow, A.W., McTaggart, M., Smith, R., Kelly-Boxall, N., Wade-McCue, S., Erskine, J., Grinover, R., Gurman, A., Hunn, T., Lee, D., Milan, A., Pham, T., Rallos, G., Razjigaev, A., Rowntree, T., Vijay, K., Zhuang, Z., Lehnert, C., Reid, I., Corke, P., Leitner, J.: Cartman: The low-cost Cartesian Manipulator that won the Amazon Robotics Challenge. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (May 2018)
21. Romea, A.C., Torres, M.M., Srinivasa, S.: The moped framework: Object recognition and pose estimation for manipulation. International Journal of Robotics Research **30**(1), 1284–1306 (Sep 2011)
22. Sakoe, H., Chiba, S.: Dynamic Programming Algorithm Optimization for Spoken Word Recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing **26**(1), 43–49 (1978). https://doi.org/10.1109/TASSP.1978.1163055
23. Sande, K.v.d., Gevers, T., Snoek, C.: Evaluating Color Descriptors for Object and Scene Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **32**(9), 1582–1596 (Sep 2010). https://doi.org/10.1109/TPAMI.2009.154
24. Schuster, M., Bormann, R., Steidl, D., Reynolds-Haertle, S., Stilman, M.: Stable stacking for the distributor's pallet packing problem. In: Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS). pp. 3646–3651 (October 2010)
25. Sucan, I.A., Moll, M., Kavraki, L.E.: The open motion planning library. Robotics & Automation Magazine **19**, 72–82 (2012)
26. Ude, A.: Filtering in a unit quaternion space for model-based object tracking. Robotics and Autonomous Systems **28**(2), 163–172 (1999)
27. Wafy, M., Madbouly, A.M.M.: Increase Efficiency of SURF using RGB Color Space. International Journal of Advanced Computer Science and Applications (IJACSA) **6**(8) (2015). https://doi.org/10.14569/IJACSA.2015.060810
28. Zeng, A., Yu, K.T., Song, S., Suo, D., Jr., E.W., Rodriguez, A., Xiao, J.: Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2017)
29. Zucker, M., Ratliff, N., Dragan, A.D., Pivtoraiko, M., Klingensmith, M., Dellin, C.M., Bagnell, J.A., Srinivasa, S.S.: Chomp: Covariant hamiltonian optimization for motion planning. The International Journal of Robotics Research **32**(9-10), 1164–1193 (2013)