No More Heavy Lifting: Robotic Solutions to the Container Unloading Problem

Todor Stoyanov,[*] Narunas Vaskevicius[‡], Christian A. Mueller,[‡] Tobias Fromm,[‡] Robert Krug,[*] Vinicio Tincani,[†] Rasoul Mojtahedzadeh,[*] Stefan Kunaschk,[◇] Rafael Mortensen Ernits,[◇] Daniel R. Canelhas,[*] Manuel Bonilla,[†] Sören Schwertfeger,[‡] Marco Bonini,[∓] Harry Halfar,[∓] Kaustubh Pathak,[‡] Moritz Rohde,[◇] Gualtiero Fantoni,[†] Antonio Bicchi,[†] Andreas Birk,[‡] Achim J. Lilienthal,[*] Wolfgang Echelmeyer[∓]

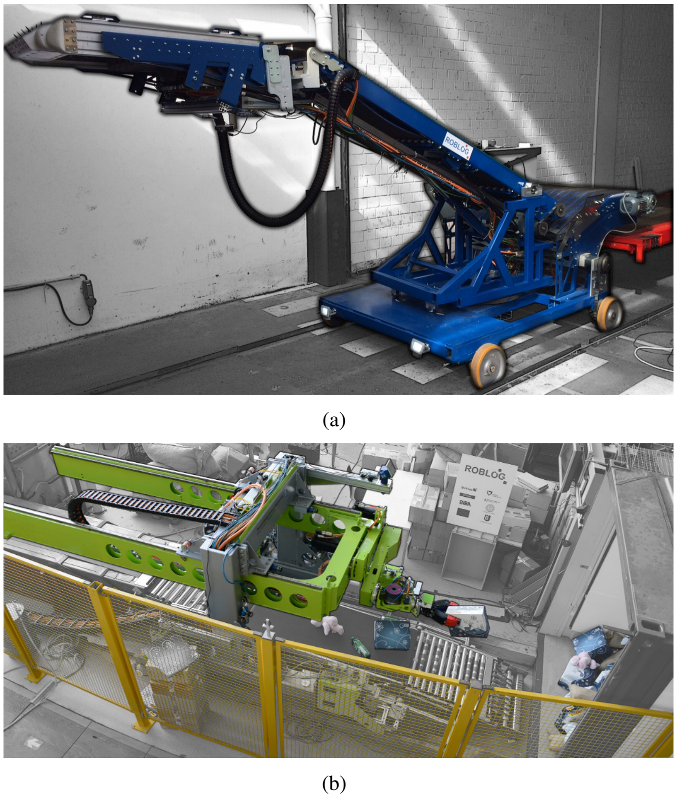
***Abstract*—This article discusses the scientifically and industrially important problem of automating the process of unloading goods from standard shipping containers. We outline some of the challenges barring further adoption of robotic solutions to this problem: ranging from handling a vast variety of shapes, sizes, weights, appearance and packing arrangement of the goods, through hard demands on unloading speed and reliability, to ensuring fragile goods are not damaged. We propose a modular and reconfigurable software framework in an attempt at efficiently addressing some of these challenges. We outline the general framework design, as well as the basic functionality of the core modules developed and present two instantiations of the software system on two different fully integrated demonstrators. While one is coping with an industrial scenario, namely the automated unloading of coffee sacks, with an already economically interesting performance, the other scenario is used to demonstrate the capabilities of our scientific and technological developments in the context of medium- to long-term prospects of automation in logistics. We performed evaluations which allow us to summarize several important lessons learned and to identify future directions of research on autonomous robots for handling of goods in logistics applications.**

I. Challenges in Autonomous Container Unloading

One of the seizable effects of globalization over the past decades is the steady increase of the volume of transported goods. Nowadays, most goods are shipped in containers and then transferred onto trucks for further transportation. Once the containers reach a target warehouse facility, they are typically unloaded manually — a strenuous task, since goods are often heavy and difficult to handle, resulting in significant health risks to the human workers. The lack of manpower willing to engage in such an exhausting and wearing labor, combined with an aging population and the increasingly strict labor law regulations make automated solutions for container unloading desirable, if not necessary.

A robotic system designed for unloading goods from containers has to deal with several major scientific and technological challenges. First and foremost, such a system needs to be able to handle a wide variety of goods of different size, shape, weight, texture and material. Individual goods need to be identified, grasped, transported to a drop-off location and finally placed securely, while maintaining human-like picking success rates and unloading times. Throughout all operations it is also imperative that goods are not dropped or otherwise damaged, as contents may be fragile. These constraints, coupled with the possibly chaotic and cluttered arrangement of goods in the container result in a set of difficult interleaved hardware design, perception, grasp- and motion planning problems. Efficient solutions to these challenges will have far-reaching applications to a number of domains involving autonomus manipulation in unstructured and uncontrolled environments.

This article delves deeper into the underlying challenges on


(a)

(b)

Fig. 1. The two autonomous container unloading systems described in this work: in (a) the Empticon II robot, designed for unloading of heavy sacks of coffee beans, shown deployed in a storage warehouse owned by Vollers GmbH; in (b) the Parcel Robot, targeted at unloading heterogeneous loose goods. The target application scenarios are further pictured in Fig. 7

[*] Center of Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden

[‡] Robotics, Computer Science and Electrical Engineering, Jacobs University Bremen, Bremen, Germany

[†] Interdepart. Research Center "E. Piaggio", University of Pisa, Pisa, Italy

[◇] Bremer Institut für Produktion und Logistik (BIBA), Bremen, Germany

[∓] ESB Business School, Reutlingen University, Reutlingen, Germany

This work was funded by the EU FP7 project ROBLOG (ICT-270350).

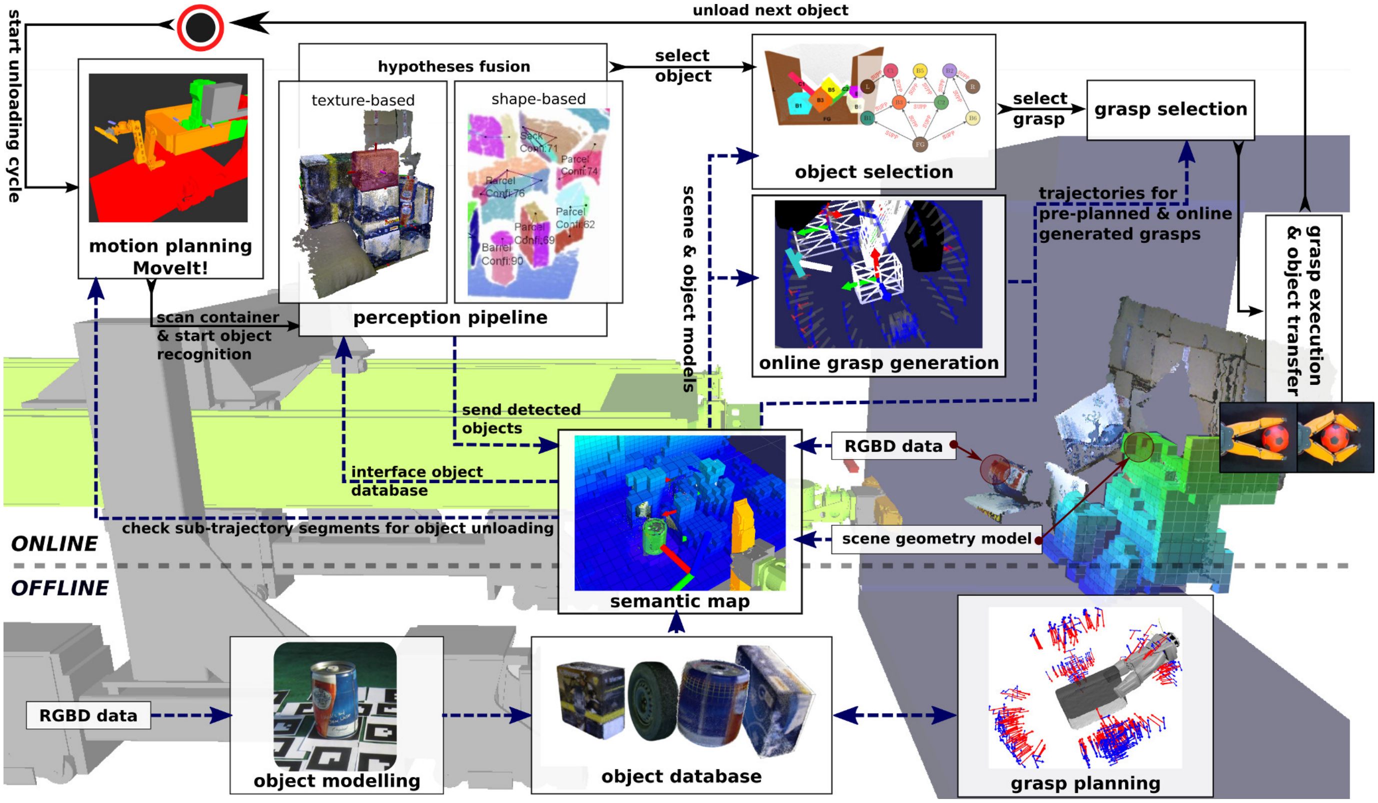


Fig. 2. Summary schematic of the software architecture applied to unloading of unstructured loose goods from containers. The modules of the system are configured in a similar fashion for the second target application (unloading of coffee sacks). In the figure, the solid black arrows signify changes in the current state of the robot's state machine over an unloading cycle, while the dashed blue arrows depict the information flow between different components.

the road towards fully autonomous container unloading robots, by examining two different sub-scenarios. The first is directly motivated by the real-world industrial use-case of unloading containers packed with 70 kg sacks of coffee. In contrast, the second sub-scenario is embedded in the scientifically more challenging domain of unloading containers packed with heterogeneous loose goods. While the latter scenario does not directly correspond to the current industrial practice of packaging goods in cardboard boxes prior to loading in a container, it serves as a scientific testbed for research on autonomous manipulation in complex cluttered environments which are very common in the context of logistics. Generalizing current approaches to such scenarios is at the forefront of research, and has direct applications to future robot aided logistics in, for example, autonomous commissioning systems (focus of the largely popular ICRA 2015 Amazon picking challenge[1]). We propose a common automation framework for handling the unloading automation task, and present two different instantiations deployed on two physically different robotic platforms: the Empticon II robot (Fig. 1(a)) targeted at coffee sack unloading; and the Parcel Robot (Fig. 1(b)) targeted at unloading heterogeneous loosely packed goods, thereby investigating the more general aspects of handling arbitrary goods in logistics applications. By examining two conceptually different scenarios, we evaluate both the readiness of the proposed automation framework to answer current industrial needs, as well as its suitability to addressing wider and more challenging future application domains.

II. Approach, Design and Implementation

A. Cognitive Approach

In order to meet the varying requirements of the two application domains, we opted for implementing two different robots with significantly different kinematic structures and gripping devices (see Fig. 1). However, as the chief objective of this work is to produce generally applicable solutions to the container unloading problem, we propose a modular and re-configurable framework, allowing to re-use common components and a unified approach to the two application domains. All components were implemented as modules in the Robot Operating System (ROS) [1].

The framework is designed around a classical sense-plan-act loop, embedded in a finite state machine. This approach allows for seamless transitioning between fully-autonomous unloading and operator-assisted autonomy, as well as for an easy means to substitute different module instances. At different states of the unloading process, the system uses a variety of processing paradigms: ranging from pipeline style processing, through centralized information aggregation, to high-level AI planning queries. A simplified illustration of our software framework is shown in Fig. 2. In a typical unloading scenario the system builds on several major components:

[1] http://amazonpickingchallenge.org/

- A customized motion planning pipeline based on the MoveIt framework[2] is used to plan trajectories (see Sec. II-H), which are subsequently executed by the industrial Programmable Logic Controllers (PLCs) of the two robots.
- A single RGBD camera (the Asus Xtion Pro Live) mounted close to the end effector is used as an input to a perception pipeline (Sec. II-C). The pipeline uses an offline-trained object database (Sec. II-D) to aid detection of known object instances, a shape-based approach for detecting known object classes, and a model-based approach tailored to coffee sack detection.
- Recognized objects and scene geometry models are fused into a semantic environment model (Sec. II-F) which serves as a centralized information repository of the system.
- A hybrid reasoning system (Sec. II-G) is used to select an appropriate target object for unloading.
- Two sets of candidate grasps for selected objects (pre-planned and online-generated) are tested for feasibility in the current scene and selected for execution (Sec. II-I).

B. Robot Hardware Design

1) Parcel Robot: To get the required flexibility to grasp different goods in various arrangements within the workspace of a standard shipping container, we used a robot previously developed for the purpose of unloading parcels — the Parcel Robot (Fig. 1(b)). The Parcel Robot features customized kinematics, which were developed to better cover the desired workspace. Typical industrial 6-axis articulated robot arms have a spherical work envelope, which requires that the robot has to be placed centrally in front of the container. This placement however leads to complicated trajectories from the grasping pose to a placing pose outside the container.

The design of the Parcel Robot kinematics — a five rotary degree of freedom (DoF) manipulator mounted on an linear portal axis, allows for shorter paths between grasping and unloading poses. The combination of linear and rotary axes describes a cylindrical work envelope that covers the full space inside the container. Unloaded goods are placed on an integrated conveyor belt and transported further along the handling chain.

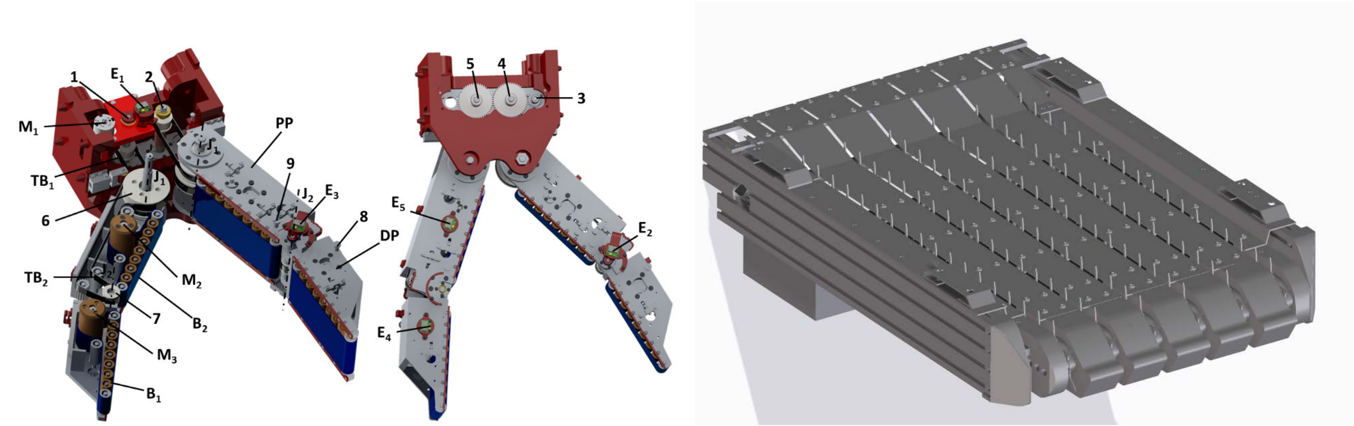
2) Empticon II Robot: Unloading coffee sacks implies the handling of tightly packed and heavy, yet deformable goods, which do not readily allow for stable top or side grasping. Thus, we opted for a manipulation solution which ensures that sacks are continuously supported from below during the unloading process. As standard industrial robots are not well suited for this task, we designed and implemented a special purpose robot: the Empticon II (see Fig. 1(a)). The robot design is optimized to minimize the amount of joint movements under typical unloading sequences, while maintaining sufficient freedom in the range of reachable target grasping poses. The implemented machine consists of a single DoF wheeled platform, on which a manipulator with five actuated joint axes (two pairs passively coupled) is mounted, resulting in a 4-DoF system. A conveyor belt runs along the whole manipulator chain, allowing for a smooth transition of sacks from the container onto the robot body and further along the material flow chain into the warehouse.

3) Velvet Gripper: In order to respond to the challenges of robust grasp acquisition of heterogeneous goods, we designed a new underactuated gripping device — the Velvet Fingers gripper [2], [3], shown in Fig. 3(a). Our application scenario demands a versatile device, which can reliably acquire grasps on objects of varying shape, material, weight and deformability, and does so under moderate object pose uncertainties. After analyzing the properties of typical goods in our scenario, we opted for a grasping principle which would allow both for embracing grasps with four contact areas for cylindrical and deformable goods, as well as grasps involving two opposing surfaces of parallelepipedic goods like boxes. To satisfy these requirements, we decided to endow the gripper with two fingers, each of which forms a planar rotary-rotary manipulator. In this fashion, all the links are involved in enveloping grasps of cylindrical and deformable goods, while for the grasping of boxes the proximal links are involved to a lesser extent. In order to keep the mechanical design conceptually simple, to allow for simpler control schemes and to keep the weight of the gripper low, we also opted for underactuation of the joint between the two links of each finger. Finally, the motor drives chosen for actuating the gripper were outfitted with back-driveable gear boxes, in order to allow for compliant behaviour via current control.

The design of the gripper is also affected by the environment in which it operates. As the containers in our scenario can feature a lot of clutter and closely packed goods, it is desirable that grasps can be acquired with a minimal gripper footprint. This requirement would however favor less stable fingertip grasps, instead of firm enveloping grasps. To solve this problem, the Velvet Fingers gripper is also endowed with in-hand manipulation capabilities through sliding surfaces (here implemented with controlled conveyor belts) on the inner side of the links. Analysis of the manipulability ellipsoids of the gripper [2] has demonstrated a clear improvement in terms of dexterity when using active sliding surfaces.

4) Needle Chain Gripper: A special-purpose gripping device — the Needle Chain gripper (Fig. 3(b)), was developed to address the coffee sack unloading task. The gripper is designed to grasp and pull a coffee sack from a pile onto its frame and further transport it to the conveyor belt of the Empticon II robot, thus facilitating a continuous material flow. For this purpose, six chains equipped with needles running synchronized and parallel along the transport direction of a grasped sack are activated by a servo electrical motor. The tip of the gripper constitutes the grasping zone where the needles pierce into the sack for the first time. The activated chains pull the sack on the gripper and subsequent pins support the grasping by continuing to pierce into the sack fabric. Proximity sensors on the gripper monitor the grasping process and stop the chains after a successful grasp.

[2] http://moveit.ros.org



(a)

(b)

Fig. 3. CAD design of the gripping devices developed: (a) the Velvet Fingers gripper design; (b) the Needle Chain gripper design.

5) Sensing Architecture: Both of the robots used were fitted with a low-cost Asus Xtion Pro Live[3] structured light camera, mounted close to the gripping system in an eye-in-hand configuration. Based on a comparative evaluation of the accuracy and quality of the sensor data [4], it was deemed sufficient to outfit the systems solely with one mobile Asus RGBD-camera each. The acquired range readings are fused into a world model, using the forward kinematic models of the robot-camera chain and the measured joint states. Extrinsic calibration of the camera pose with respect to the kinematic chain was performed using a combination of a grid-search method on a sum of pairwise L_2 distances between NDT models [5] of collected point clouds and an open-source calibration package[4]. To avoid synchronization issues between the camera and the joint state data, models are only updated when the robot is stationary for a set time duration.

C. Perception Framework

A flow diagram depicting the different building blocks of the perception pipeline is shown in Fig. 4. Some components of the pipeline are initially run offline – these include populating a database with models of objects to be recognized in the scene at run-time. The offline object-model acquisition is described in more detail in Sec. II-D. To provide an overview, the various *online* processing steps are briefly described.

Pre-Processing: Given the cluttered nature of typical containers, it is essential to reduce the noise present in the RGBD data. To do so, we use simple pixel-wise averaging of range data for a certain time-window, during which the sensor is kept at a stationary scene-observation pose.

Segmentation: The RGBD raster is divided into contiguous clusters (segments), which are homogeneous with respect to certain geometric and/or color-based criteria. We use two types of segmentation algorithms:

- **Type I**: *model-unaware* approaches do not utilize prior information about the known objects present in the object-database. In this work, an extended Mean-Shift [6] approach is used. The algorithm *over-segments* the scene and the resultant *atomic-patches* form the basis for downstream segmenters and Object Recognition (OR) modules.
- **Type II**: *model-aware* segmenters may combine neighboring atomic patches from the type I segmenters according to some application dependent heuristics, such as convexity.

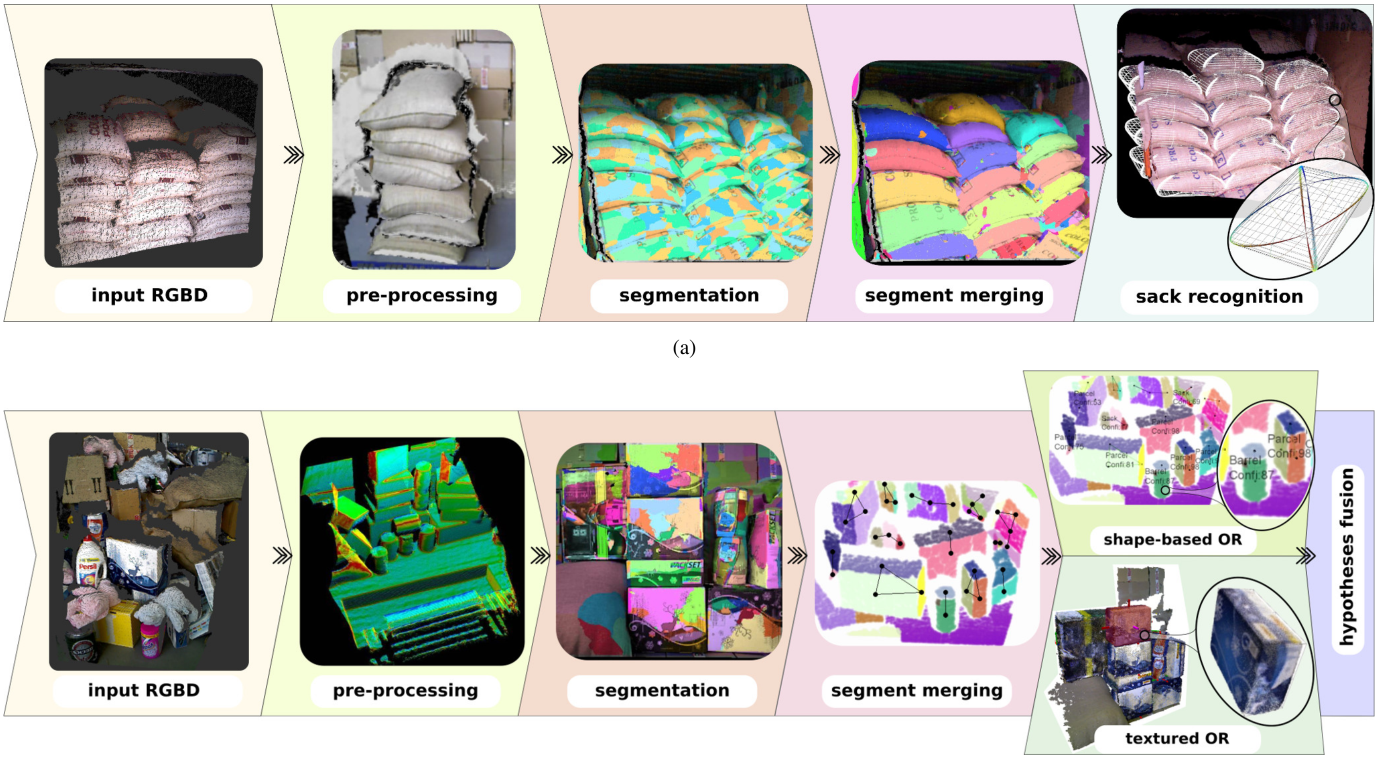
Finally, the atomic patches obtained during segmentation pass through a rough filtering step, which eliminates segments that cannot possibly be generated by unloadable objects (based on size and PCA analysis).

Object Recognition (OR) Modules: These modules analyze the resulting segments to recognize potentially unloadable objects and compute their 3D poses. Three recognition modules (see Sec. II-E) were developed to handle different modalities present in our target scenarios. The texture based recognition module (Sec. II-E1) is used to generate hypotheses about objects from the database. Hence it serves as an object instance recognizer and allows to use the additional information from the database to perform better informed manipulation in the later stages of the unloading cycle. Objects not present in the database are handled by the shape-based object recognition module (Sec. II-E2). In addition to segmenting object candidates, it provides shape category information which is later used for online grasp planning. The third recognition module was designed to handle specific perception challenges present in the coffee sack unloading scenario as described in more detail in Sec. II-E3. Each of these modules can, depending on the target application, be executed separately or in combination with the other modules. The module configurations used in this work are depicted in Fig. 4: the dedicated sack recognizer is used for the sack unloading scenario (Fig. 4(a)); a combination of the texture and shape based approaches is used in the heterogeneous loose goods scenario (Fig. 4(b).

Hypotheses Fusion: Results coming from different recognition modules might overlap, have missing information and more importantly contradict each other. The hypothesis fusion module uses complementary information available from OR modules to fuse candidate detections into consistent and more

[3] http://www.asus.com/Multimedia/Xtion_PRO_LIVE

[4] http://wiki.ros.org/ethzasl_extrinsic_calibration



(a)

(b)

Fig. 4. The two instances of our perception pipeline, consisting of the respective processing steps from data acquisition to the hypotheses fusion. The pipeline for coffee sack perception shown in (a) relies on a single, dedicated object recognition module. In contrast, the pipeline for heterogenous goods perception in (b) fuses the results of a shape-based and a texture-based recognition module.

confident hypotheses. To this end, correspondences between the estimated object locations are detected by considering the overlap of the segmentation patches associated with individual hypotheses. In case multiple hypotheses from different OR modules correspond to a single object location, an adaptive voting scheme is applied with respect to the object identity classifications of the conflicting hypotheses. In this scheme, hypotheses originating from OR modules that use a-priori known 3D object model knowledge from the database (see *e.g.* Sec. II-E1) are prioritized over those produced by the less informed sensor observation based OR modules (see *e.g.* Sec. II-E2). Finally, hypotheses specifics are also considered in case of the fusion of object pose information, where object pose estimations based on database models are prioritized over pose estimations based on partial object observations in the scene.

D. Object Database

Shipping containers or mixed pallets often contain a limited variety of types of goods. As each type of good has to be recognized and unloaded hundreds of times, substantial performance improvements can be achieved by maintaining a database containing models and properties of the unloaded objects. In addition, the recognition of known objects can be used for quality control and for accounting purposes, respectively it is of interest for logistics and mobile manipulation applications in general. Therefore, our system builds on an object database and provides tools allowing an untrained person to model goods with a low-cost sensor and without any costly infrastructure prior to automated unloading. Object models are acquired using a hand-held Asus Xtion camera and a set of Augmented Reality (AR) markers printed on paper and distributed in the environment (as shown in Fig. 5). The system tracks the pose of the markers and uses them in a global graph-optimization scheme to achieve high-quality sensor pose tracking. Individual RGBD views of the object are used to reconstruct point cloud and mesh models (see [7] for further details) and to automatically train the textured object recognition module (Sec. II-E1). The acquired object models are then also used as an input to an offline grasp planning stage (see Sec. II-I) and finally saved and referenced along with all relevant meta-information (*e.g.* visual features, planned grasps, physical properties).

E. Object Recognition Modules

1) Texture-based Object Recognition: An extended version of the texture-based object recognition system presented in [8] was integrated in the perception pipeline (see Fig. 4(b)) and employed as an object instance recognition module. For details of the initial approach, the reader is referred to [8]; here we only provide a short overview of the extended system. Our approach consists of two phases, described below: an offline training step, followed by an online recognition step.

Training: In this offline phase, we process all RGBD images used to create object models in the object database (Sec. II-D). We extract SIFT visual features on the object surface and store

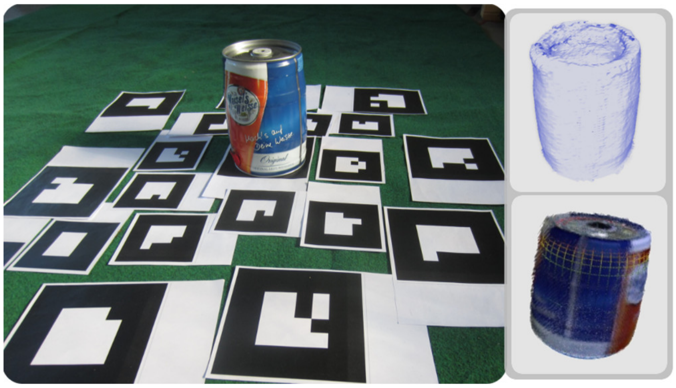


Fig. 5. Left: Database object acquisition setup. Right: Point-cloud and triangle mesh reconstructed models stored in the database.

the descriptors and their 3D positions relative to the object coordinate frame. The resulting geometrically-referenced visual cues are then stored as meta-data for each object in the database.

Recognition: The online recognition stage consists of bottom-up and top-down perception steps and combines texture information with geometric properties of the scene observed in an RGBD image. As an input, we take the filtered atomic patches obtained by the segmentation module, and use them to define regions of interest in the color image for which we extract visual features. Next, a RANSAC step is used to generate hypotheses for the most likely positions of database objects, while respecting 3D geometrical constraints between feature keypoints in the observation and the model. The candidate object poses computed by the matching algorithm are then used to re-project the models of the database objects back into the RGBD image plane. Patches from the over-segmentation, color and range information are then used to test the hypothesis for consistency and to filter out false positives. Objects with high consistency scores are considered to be recognized and their corresponding patches are removed from the input RGBD image. Detection is then re-iterated on the remaining segments to handle multiple object instances.

2) Shape-based Object Recognition: The shape-based recognition module illustrated in Fig. 4(b) focuses on object class learning using the hierarchical part-based shape categorization method for RGB-augmented 3D point clouds proposed in [9]. An unsupervised hierarchical learning procedure is applied, which allows to symbolically classify shape parts by different specificity levels of detail of their surface-structural appearance. Further, a hierarchical graphical model is learned that reflects the constellation of classified parts from the set of specificity levels learned in the previous step. Finally an energy minimization inference procedure is applied on the hierarchical graphical model to obtain the corresponding shape category of an object instance consisting of a set of shape parts. As demonstrated in [9], the additional evidence on different levels of shape detail contained in the proposed hierarchical graph constellation models is a major factor that leads to a more robust and accurate categorization compared to a flat approach [10].

3) Shape-based Object Recognition for Sack Unloading: Object recognition in the context of autonomous unloading of coffee sacks faces several challenges. Cargo is loaded in containers in a dense manner up to the container ceiling, therefore most of the sacks have to be perceived from the front with only a small part of the surface exposed to the sensor. The target objects are deformable, with little texture information and they are very tightly packed. Nonetheless, boundaries between the sacks have to be accurately perceived to facilitate reliable grasping and manipulation.

As described in [11], our perception strategy is based on modeling sacks by fitting superquadrics, using a model-aware, two step segmentation of an RGBD scan. The first step uses an extended version of the Mean-Shift algorithm to achieve over-segmentation of the scene with small, almost planar patches as shown in the segmentation component of Fig. 4(a). After the initial segmentation is complete, the results are passed to the second level segmentation. This model aware approach is based on the observation that transitions between patches are generally convex if they belong to the same sack, or conversely concave if they belong to different sacks. Therefore, a patch-graph is built and the edges between vertices are classified into convex and concave. The patch-graph is then analyzed and new segments are formed by grouping the small patches from the initial segmentation (see Fig. 4(a), segment merging component). These segments form a set of candidate sack objects on which further shape analysis is performed to filter out false positives. Finally, view-point aware superquadric fitting with a re-projection test is used to generate the final set of hypotheses, along with a set of sack pose estimates (see [11] for details).

F. Semantic Maps

To enable high-level autonomy, the system needs to maintain a consistent model of both the metric and the symbolic state of the shipping container during unloading operations. This functionality is provided by a semantic map — a centralized data structure which integrates the output from the object perception pipeline and the 3D map of the container, while providing interfaces for higher level modules. In addition to simply storing the list of recognized objects, the semantic map also models a temporal component of the scene by maintaining a history of world states by tracking significant changes in the map. The state of the scene is saved for further reference and can be accessed by other system components by referencing a time-stamp. Finally, to enable reasoning about spatially-sensitive information, entities in the map ($e.g.$ objects or obstacles) also encode the sensor pose from which they were observed.

To provide a consistent and time-synchronized world model to all system modules, all components which require information about the environment subscribe and obtain it from the semantic map, instead of retrieving it directly through sensors and/or other components. Thus, the Semantic Map works as an abstraction layer which responds to all queries on any information about the world, be it static attributes ($e.g.$ properties of database objects) or dynamic properties (object model, bounding box, grasp poses and gripper configurations, etc.). This central information-repository role of the semantic map is clearly visible in the framework illustration in Fig. 2: all exchange of information (the dashed arrows in the figure) goes through the semantic map. From an implementation perspective, this component provides interfaces to the underlying data structures used for storage ($e.g.$ the object database) or modeling ($e.g.$ the environment map) and encapsulates the results of perception and motion planning modules in a centralized and time-consistent manner.

G. High-level Planning

Once objects in the container have been recognized and the scene geometry has been modeled, the next task in the unloading cycle is the selection of an appropriate target good for grasping. In both scenarios considered here, it is desirable to select only objects which can be safely removed without disturbing the equilibrium of the other goods in the container. Planning safe unloading sequences can be accomplished using a heuristic approach in some simplified scenarios — $e.g.$, always selecting the top-most sack from the front layer in the coffee unloading scenario; however, in the more complicated object configurations typical for containers packed with heterogeneous goods, a more structured and complex approach is necessary. The according component takes a two-step approach to this problem: first, we create models that reflect how objects in the scene are physically interacting with each other, in order to identify which objects are supporting other objects; second, we use the created models to make an optimal decision regarding which object is the safest to remove.

The high-level unloading sequence planner was incrementally developed by progressively relaxing the assumptions on the created world models. In [12], we consider solid cuboid shape objects ($i.e.$, carton boxes) with known poses and models, and we propose a method based on geometrical reasoning and static equilibrium analysis to extract the gravitational support relations between objects and build a relational representation that can be further used with a high level AI reasoning module to select the safest object. In [13], we assume that only the shape and pose of a subset of objects composing the environment are available, and we propose a probabilistic decision making framework based on a principle of possible worlds representation and a machine learning approach for probabilistic estimation of the gravitational support relations. In [14], we present an extension of our method to address the problem of autonomously selecting the safest object from a pile considering objects with rigid convex polyhedron shapes, and examining the performance of different machine learning models for estimating the support relation probabilities.

H. Motion Planning

For facilitated motion planning, collision avoidance and manipulator control, we use a customized version of the *MoveIt!* motion planning framework. One of the main reasons for modifying the MoveIt framework is that it was originally developed under the assumption that successfully planned trajectories are directly executed by the robot. This assumption does not hold in our application scenarios, as we need to plan both approach and escape trajectories prior to execution, and may need to attempt a multitude of target grasp poses per object. These requirements in turn necessitate a consistent handling of the planning scene objects and a careful managing of the process of attaching and detaching object collision geometries to the kinematic chain. We handle this problem by employing a pre-planning process which includes a set of alternative approach/retrieve trajectories planned under different start/end pose constraints.

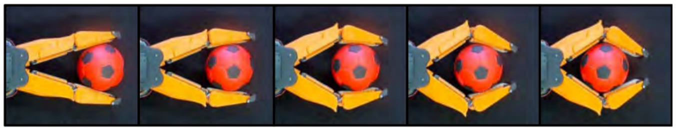


Fig. 6. Depicted is a sequence of intermediate grasp states where the belts of the gripper are used to pull the object towards its palm which results in a transition from a fingertip to an enveloping grasp.

I. Grasp Planning and Execution

Reliable grasp acquisition of heterogeneous goods from unstructured scenes is a challenging problem. The central idea of our approach, as presented in [15], [16], is to exploit the low pre-grasp pose sensitivity and the active surfaces of the Velvet Gripper in the grasping process. The experiments reported in [15] showed that in cluttered scenes fingertip grasps are more likely to be feasible than robust enveloping grasps, because the latter necessitate large opening angles resulting in bulky gripper silhouettes for which no collision free approach trajectories can be found. We employ a simple “pull-in” strategy (see Fig. 6), which exploits the underactuated nature and the conveyor belts on the grasping device to embrace the object in a firm envelope grasp by simultaneously squeezing it while actuating the belts inwards. The corresponding grasping controller was implemented by means of low-level current control of the gripper’s actuators, allowing for a simple compliant behavior. The grasping strategy is implemented in three steps: first, the fingers are closed with a low current setpoint until contact is detected; next the belts are actuated inward, while the gripper closing DoF is kept in a compliant low-current control mode, allowing the object to be pulled into the gripper; and finally if the phalanges have wrapped around the object a higher current setpoint is commanded to ensure a firm grasp.

To autonomously achieve a grasp on an object, the grasp planning problem ($i.e.$, finding an appropriate grasp configuration and corresponding joint trajectories) needs to be solved. We employ a data-driven solution where, in order to deal with the curse of dimensionality, the grasp synthesis problem ($i.e.$, finding a suitable palm pose and gripper joint configuration) is separated from the problem of planning collision free motions for the gripper-manipulator chain. In an offline stage, the database of known objects (Sec. II-D) is populated with a set of fingertip grasps, synthesized by following an approach similar to [17] — namely, by minimizing an energy function depending on the distance and the alignment of the object relative to pre-defined desired contact locations on the gripper's fingers. Additionally, grasping principles observed in humans (approach along an object surface normal and orientation of the hand's lateral axis normal to one of the object's principal components) are incorporated by imposing appropriate constraints to the underlying optimization problem. The approach was implemented in the GraspIt![5] framework, and subsequently used to plan a set of 400 fingertip grasps for each object in the database. As we can not employ this procedure for objects that are not in the database, we employ a two-step online approach to synthesize grasps for deformable and shape-categorized objects: (1) depending on the rough shape categorization of the cluster of patches belonging to an object, an according super-quadric model is fitted [11], (2) surface sampling with a chosen density is used to compute grasping-configurations (gripper pose and opening angle) around the super-quadric, while respecting the geometric constraints imposed by the shape model. In the online phase, grasps associated with recognized objects in the container are ranked according to their proximity to a desired template gripper orientation. The template is chosen to point into the container and at an angle of 20° towards the floor. This choice prioritizes grasps that would move the target object out and up, with the active surfaces acting against gravity to minimize friction between the object and its support. Finally, motion planning attempts are made in a feasible-first manner for the highest scored grasps of selected objects until a valid approach/escape trajectory of the manipulator is generated and selected for execution. Once a stable grasp has been achieved, the target good is transported to a predefined zone positioned at a set height above the conveyor belt and released. This strategy entails that goods are dropped from a height of several centimeters which may be sub-optimal. However, devising a safe grasp release procedure in the absence of an adequate force control interface for the manipulator is a difficult task. As force control is a critical requirement for future refinements of our system, we anticipate it would greatly simplify the design and implementation of a better grasp release strategy.

A different strategy is used for the online generation of grasps for unloading of coffee sacks with the needle chain gripper. To achieve a robust grasp, the tip of the gripper has to be aligned with the lower boundary of the target sack. To this end, we first plan for a grasp at a fixed offset from the estimated sack front boundary, and then refine the height of the gripper based on the estimated location of the sack's lower boundary. Sacks loaded close to the container floor can only be grasped with a tilted gripper. Therefore, as a final step, the height is used to choose between the straight and tilted gripper configurations. Once an appropriate grasping position and tilting state are chosen, a multi-step grasp acquisition procedure is initiated. First, the gripper is brought to the selected pre-grasp pose. Next, a combined movement of the two vertical rotary axes of the robot brings the gripper on a hooking trajectory under the sack to be grasped. As the gripper comes into contact with the sack, the chains are actuated in a low velocity / high torque mode, while the axes of the arm continue to lift and push into the sack. Once the sack passes over an infrared light proximity sensor on the gripper, the chain motion is stopped, while the gripper is retracted and lifted. Finally, the chains and conveyor are simultaneously activated, completing the transfer of the sack.

III. Performance and Evaluation

In this section, we focus on the overall system performance evaluation since even extensive testing of the sub-components in isolation, as it was previously reported in the respective scientific publications, is not sufficient to judge the general suitability of the presented approach for the task of automated container unloading, respectively autonomous manipulation in logistics applications in general. Our evaluation focused on measuring three main performance indicators — unloading cycle times per object, percentage of successfully unloaded goods, and amount of damage to the goods. These criteria stem directly from the requirements on a potential industrial product: such a system should perform at runtimes, success rates and damage rates close to those of human workers.

A. Unloading Coffee Sacks

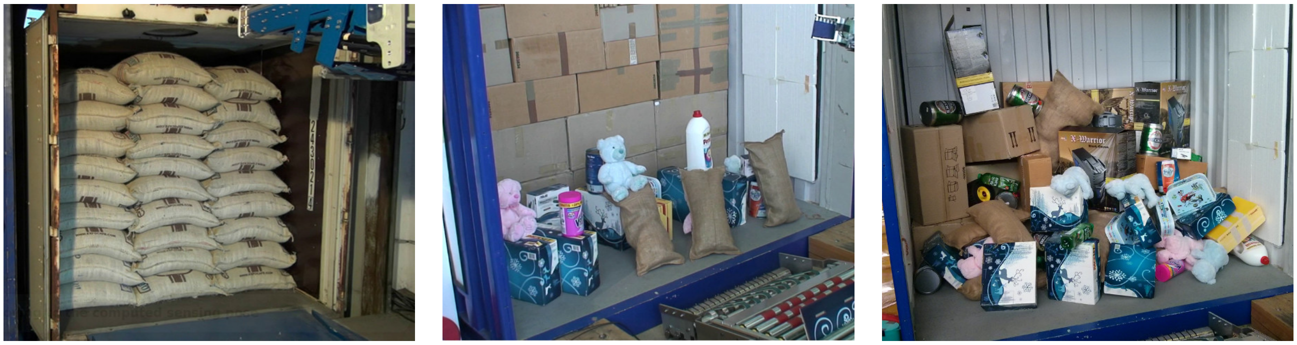
The coffee sack unloading system was tested in real-world conditions at a coffee storage warehouse owned by the logistics services provider Vollers[6]. Containers were packed with 70 kg coffee sacks (see Fig. 7(a)) in a manner consistent with typical shipping conditions. A video of the system performance can be found on the RobLog youtube channel[7]. The average cycle time for unloading a single coffee sack was 67 s $(\sigma = 8.2$ s, median 58.7 s), with all coffee sacks successfully unloaded in our tests ($N = 58$ for this evaluation). The damage to the goods in this scenario was judged by examining the tear pattern of the coffee sacks at the output side and marking the number of sacks that would not be re-usable for transportation. Throughout all the tests of the platform, seven percent of sacks were deemed unusable after unloading.

The current unloading practice at Vollers is to use four employees: two workers using two hooks each for unloading, a forklift driver and a weighing scale operator. A fully packed twenty-foot container can transport up to 24 tons of coffee, with roughly 300 to 400 sacks, and is usually unloaded in about 30-45 minutes on average. Thus, human picking rates are currently about 6 to 10 seconds per sack (with 4 workers), which is an order of magnitude faster than the autonomous system. According to our industrial partner, taking into consideration that the autonomous solution can be employed over longer shifts, the current system is already economically viable in some use cases. Furthermore, the automation of unloading is highly desirable to improve the working conditions in the handling of coffee and similar goods like cacao. Especially, labor regulations are worldwide becoming increasingly stricter with respect to the maximum amount of lifting allowed during the daily shift of a human worker. Coffee handling operations in the EU for example, where most of the coffee roasting and

[5] http://www.cs.columbia.edu/~cmatei/graspit

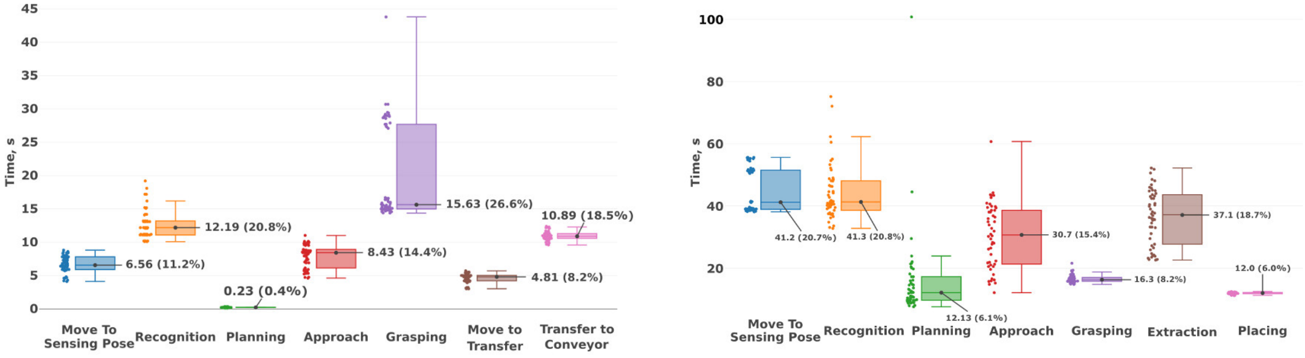
[6] http://www.vollers.com

[7] https://youtu.be/U7_lxH-qB5I



(a) (b) (c)

Fig. 7. Unloading of coffee sacks was evaluated on real-world scenarios at a warehouse owned by Vollers GmbH. A sample fully packed container is shown in Fig. 7(a). Unloading of loose goods was evaluated over many sample scenes (one instance shown in Fig. 7(b)). A more challenging cluttered evaluation scenario is shown in Fig. 7(c)



(a) (b)

Fig. 8. Breakdown of the runtimes of the two technology demonstrators. Fig. 8(a) shows the time taken by each system component in the coffee sack unloading scenario. Fig. 8(b) shows the corresponding component-wise time breakdown for the heterogeneous goods unloading scenario. Plots are shown as box-plots, where each box is centered at the average time value and extends to cover the 25th and 75th percentile, the thick line denotes the median, and the whiskers extend over the full range of inliers. Data samples distribution is shown as a scatter plot on the left of each box.

processing takes place, will be forced to switch to automation as soon as the currently practiced operations are no longer compatible with labor laws.

In order to identify how the system cycle times can be improved, we examine a breakdown of the runtime per component as shown in Fig. 8(a). Clearly, the system spends a large portion of the cycle time in grasp acquisition. Two separate clusters of grasp runtimes are also easily identifiable in this plot, with the more time consuming cluster corresponding to the more difficult grasps of sacks near the container floor. The second most time consuming module is object recognition, followed by the motion execution related components. Following these results, in order to achieve faster cycle times, future improvements of the system would primarily focus on optimizing the grasp acquisition procedure.

B. Unloading Loose Goods

As mentioned before, unloading of heterogeneous loose goods was selected to evaluate the system performance on scientifically more challenging problems. The relatively light (up to 5 kg) and loosely packed goods in the container would present little difficulty for a human worker, yet pose significant challenges to an autonomous system due to the complexity of the scenes. Therefore, in this scenario we do not aim to match the speed of manual container unloading, but rather concentrated on evaluating the robustness of the system to various object stacking configurations. The system was evaluated on a multitude of different container configurations, one of which is shown in Fig. 7(b). A video of a sample system run is available on the RobLog youtube channel[8].

Test scenes for this scenario contained on average between 20 and 25 objects from 7 to 9 different object classes, 5 of which were modeled in the object database, $i.e.$, the scenes also contained a substantial amount of unknown objects to be unloaded. On average, the system was capable of successfully unloading 80% of the target objects at a cycle time of 200.8 s (σ = 19.3 s, median, 198.9 s, N = 50), $i.e.$, about 3.5 minutes. Most of the unloading failures were due to failures to find collision-free grasping trajectories for objects placed in difficult configurations — $e.g.$, in proximity to the container walls, or in a tight packing with other objects. Only a handful of objects were not recognized by any of the modules in the recognition pipeline, and in very few cases this problem

[8] https://youtu.be/34ZXK6L1ixY

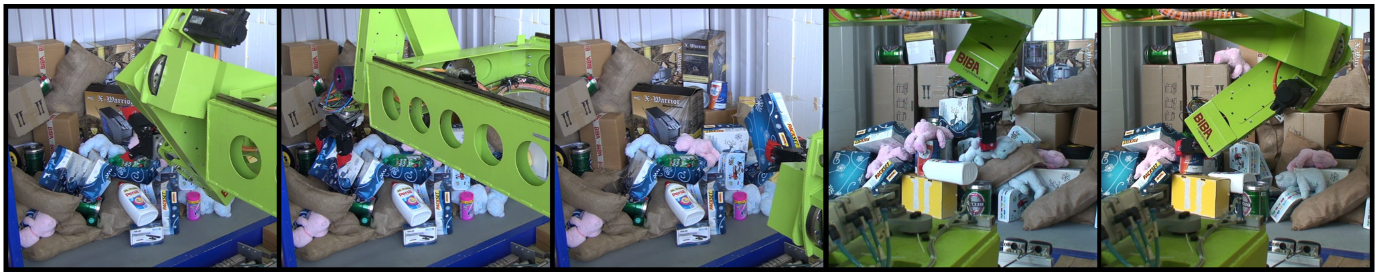


Fig. 9. Snapshots from an unloading sequence in a challenging cluttered scenario. Starting from a sensing pose, the robot approaches, grasps and extracts a parcel (frames one to three). Grasps acquired on a deformable teddy bear and a five liter barrel are shown in the next two frames.

persisted in between successive cycles. Fig. 8(b) shows the module-wise break down of runtimes in an unloading cycle. Other than in object recognition, the system also expended a lot of effort on motion planning, and subsequently on motion execution. One of the reasons for the long execution times is that, in the interest of safety, the robot was operated at a reduced speed (10% of the maximum drive capacity). The second, and often more prominent cause is linked to the poor quality of the motion plans obtained for some of the trajectories (see Sec. IV for discussion). Overall, the system performed within good time bounds and success rates for a research prototype, and did so with virtually no damage to the unloaded goods. To the best of our knowledge, no comprehensive results of a complete autonomous manipulation system, encompassing all steps from perception, through grasping and unloading, in similarly complex settings have been reported previously.

To stress test the system and explore the boundaries of the proposed approach, we also evaluated the performance on a significantly more challenging cluttered configuration of goods in the container as shown in Fig. 7(c). A sequence of still shots from various stages of unloading in this challenging setup are shown in Fig. 9. Overall, the system performance gracefully degrades, without suffering from a significant failure. As there are many objects in the scene, both recognition and grasp / motion planning take significantly longer, increasing the average cycle time to 313 s $(\sigma = 63$ s, median 300 s, $N = 26)$. Complete unloading of all objects in the container in this scenario is not always possible, as the likelihood of objects stacking in unreachable or collision-risky configurations increases significantly. As discussed in the next section, this limitation is a strong argument in favor of using compliant force-limited robots for unloading tasks in future applications.

IV. Lessons Learned

In this section we discuss some of the lessons learned over the four years spent in designing, implementing and testing the proposed automation framework and the two demonstrators. From one perspective, the design decisions taken proved to be sound, as evident by the performance on quite challenging and realistic test scenarios. On the other hand, over the course of development it was evident that some of the system components could not fully cope with the hard requirements posed by the container unloading problem.

Modular approach: Probably the main reason for the success of the two technology demonstrators is the modular system architecture. In addition to enabling easier integration between components, this approach was critical in fast migration of code between the two hardware platforms, as well as rapid deployment of new system components.

Perception and manipulation: Another main beneficial system feature was the synergy between perception and grasping / manipulation. On one side, the large tolerance to object pose uncertainty inherent in the Velvet Gripper eased the requirement on the perception system. On the other hand, the primitive shapes fit by the perception system greatly simplified the difficult problem of online grasp generation for unknown objects.

Compliance: A key to the successful grasping of goods in the loose goods scenario was provided for by the inherent compliance of the gripping device. While compliance in the grasping device was beneficial, the lack of compliance in the stiffly position controlled manipulator was one of the main drawbacks of our system. The container unloading scenario we explored in this project inherently requires that the robot comes into contact with the environment (in particular with the goods being unloaded). As the environment is unstructured and modeled based on noisy range measurements and uncertain object pose estimates, ensuring collision-free trajectories for a position controlled robot was one of the main challenges we had to address. Thus, we believe that a compliant control approach which sacrifices some precision for the sake of better interaction capabilities, would be preferable.

Grasp and motion planning: A somewhat related issue we identified was the planning of collision-free trajectories for the gripper-manipulator chain. Especially in the heterogeneous goods scenario, motion planning accounted for roughly 25% of the idle time of the robot. The motion planning problem is per se difficult, as the configuration space of the robot is high-dimensional and the distribution of valid collision-free states is potentially complex. In addition, we are forced to plan many possible trajectories until we find a valid grasp approach and object extraction plan. The reasons for this are two-fold: first, the high amount of clutter often causes grasp poses to be in collision; and second, placing and extracting the gripper through clutter causes a bottleneck in the configuration space, which in turn reduces the probability of our sampling-based planner to draw valid configurations. These issues, combined with the well known problems of sampling-based planners (unnatural and sub-optimal trajectories, probabilistic completeness, termination issues) and the particularities of handling the MoveIt scene collision models, resulted in major development difficulties for the motion planning module. Therefore, one of the main future research directions we identified is a better integration of grasp- and trajectory planning — possibly exploiting the large pose uncertainty tolerance of the gripper and a compliant manipulator.

V. Conclusions and Outlook

In this article we presented research on the automation of logistics processes in form of two autonomous container unloading systems. One is targeted at an industrial scenario in form of the automated unloading of coffee sacks and provides an already economically interesting performance. The other demonstrator deals with an advanced scenario, namely the unloading of various objects from a shipping container, and it is used to demonstrate the capabilities of our developments in the context of medium- to long-term prospects of automation in logistics. To meet the challenges posed by our application domain, we developed and evaluated an integrated software framework, with individual components contributing novel solutions to problems in perception, modeling, grasp- and motion planning, as well as high-level planning and autonomy. In order to ensure robust handling of the unloaded goods, we devised novel hardware solutions and endowed our autonomous systems with two customized grippers. Although the main focus of this work is the automation of container unloading, most of the developed techniques are relevant for a much broader set of robotics applications. In particular, the presented framework and its building blocks can be further utilized in a variety of other automated logistics problems: ranging from de-palletizing, through commissioning, to bin picking. It is our hope that the growing research community working on robotics for logistics applications will find inspiration in the presented system and in particular in the lessons learned identified in this article, in order to develop ever better and more capable robots.

References

[1] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, vol. 3, no. 3.2, 2009, p. 5.

[2] V. Tincani, M. G. Catalano, E. Farnioli, M. Garabini, G. Grioli, G. Fantoni, and A. Bicchi, "Velvet fingers: A dexterous gripper with active surfaces," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International conference on*. IEEE, 2012, pp. 1257–1263.

[3] V. Tincani, G. Grioli, M. G. Catalano, M. Garabini, S. Grechi, G. Fantoni, and A. Bicchi, "Implementation and control of the velvet fingers: a dexterous gripper with active surfaces," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2744–2750.

[4] T. Stoyanov, R. Mojtahedzadeh, H. Andreasson, and A. J. Lilienthal, "Comparative evaluation of range sensor accuracy for indoor mobile robotics and automated logistics applications," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1094–1105, 2013.

[5] T. Stoyanov, M. Magnusson, H. Andreasson, and A. J. Lilienthal, "Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations," *The International Journal of Robotics Research*, vol. 31, no. 12, pp. 1377–1393, 2012.

[6] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.

[7] R.-G. Mihalyi, K. Pathak, N. Vaskevicius, T. Fromm, and A. Birk, "Robust 3D Object Modeling with a Low-Cost RGBD-Sensor and AR-Markers for Applications with Untrained End-Users," *Robotics and Autonomous Systems*, vol. 66, April 2015.

[8] N. Vaskevicius, K. Pathak, A. Ichim, and A. Birk, "The Jacobs robotics approach to object recognition and localization in the context of the ICRA'11 Solutions in Perception Challenge," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3475–3481.

[9] C. A. Mueller, K. Pathak, and A. Birk, "Object shape categorization in rgbd images using hierarchical graph constellation models based on unsupervisedly learned shape parts described by a set of shape specificity levels," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 3053–3060.

[10] ——, "Object recognition in rgbd images of cluttered environments using graph-based categorization with unsupervised learning of shape parts," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 2248–2255.

[11] N. Vaskevicius, K. Pathak, and A. Birk, "Fitting superquadrics in noisy, partial views from a low-cost rgbd sensor for recognition and localization of sacks in autonomous unloading of shipping containers," in *Automation Science and Engineering (CASE), 2014 IEEE International Conference on*. IEEE, 2014, pp. 255–262.

[12] R. Mojtahedzadeh, A. Bouguerra, and A. J. Lilienthal, "Automatic relational scene representation for safe robotic manipulation tasks," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 1335–1340.

[13] R. Mojtahedzadeh, A. Bouguerra, E. Schaffernicht, and A. J. Lilienthal, "Probabilistic relational scene representation and decision making under incomplete information for robotic manipulation tasks," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5685–5690.

[14] ——, "Support relation analysis and decision making for safe robotic manipulation tasks," *Robotics and Autonomous Systems*, vol. 71, no. 0, pp. 99 – 117, 2015.

[15] R. Krug, T. Stoyanov, M. Bonilla, V. Tincani, N. Vaskevicius, G. Fantoni, A. Birk, A. Lilienthal, and A. Bicchi, "Velvet Fingers: grasp planning and execution for an underactuated gripper with active surfaces," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3669–3675.

[16] R. Krug, T. Stoyanov, M. Bonilla, V. Tincani, N. Vaskevicius, G. Fantoni, A. Birk, A. J. Lilienthal, and A. Bicchi, "Improving Grasp Robustness via In-Hand Manipulation with Active Surfaces," in *IEEE International Conference on Robotics and Automation (ICRA) – Workshop on Autonomous Grasping and Manipulation: An Open Challenge*, 2014.

[17] M. T. Ciocarlie and P. K. Allen, "Hand Posture Subspaces for Dexterous Robotic Grasping," *The International Journal of Robotics Research (IJRR)*, vol. 28, no. 7, pp. 851–867, 2009.