2015 IEEE International Conference on Robotics and Automation (ICRA)
Washington State Convention Center
Seattle, Washington, May 26-30, 2015

New Brooms Sweep Clean - An Autonomous Robotic Cleaning Assistant for Professional Office Cleaning

Richard Bormann, Joshua Hampp and Martin Hägele

Abstract— **Millions of office workplaces are cleaned by a surprisingly small group of cleaning workers every day, however, cleaning companies struggle to recruit enough personnel these days. One solution to this challenge is to schedule available professionals for demanding tasks while relieving them from simpler activities which are transferred to a robotic cleaning assistant. Two of such tasks are floor cleaning and waste disposal which account for 70% of the daily cleaning efforts. This paper presents the world's first autonomous cleaning robot prototype that masters both of these tasks and whose development was accompanied by the advice of a large cleaning company. Besides a detailed description of the overall system and its individual components an evaluation is provided based on real world experiments. The results indicate that both cleaning tasks can be solved at high quality but with potential for increased efficiency to meet the required performance. Hence, the paper concludes with a discussion on measures necessary for the development of a commercial prototype.**

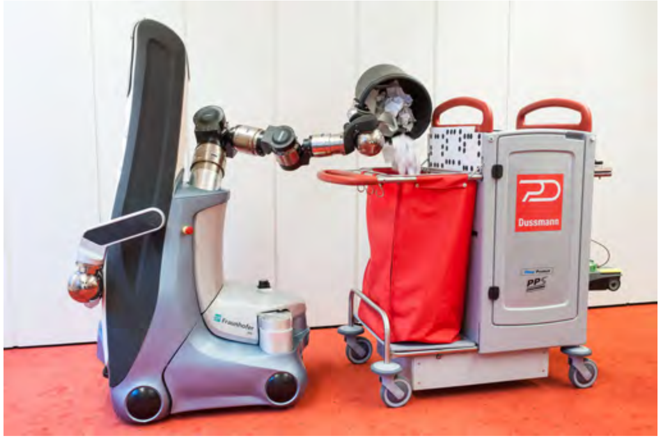

Fig. 1. The cleaning assistant robot clears a trash bin.

I. Introduction

Daily office cleaning is one of the largest service businesses that concerns millions of people, or better said their office workplaces, though many of us do not notice it happen because cleaning professionals usually start when the work of the office workers ends. Consequently, many internal matters of this sector remain a secret to most. One of those is the difficulty of cleaning companies to hire qualified staff nowadays, which intensified with the ongoing demographic change in western nations. In collaboration with one of the largest cleaning companies in Germany, which offers its service to over 3 million people every day and which informed us about this problem, a research project was initiated with the goal to evaluate the feasibility of an automated cleaning assistant for less complex cleaning tasks. 70% of daily office cleaning work is allotted to floor cleaning and waste disposal. As these tasks are less challenging the idea arose to mitigate the pressure on cleaning professionals by providing robotic assistance on such time intense activities while allocating more resources to the sophisticated jobs such as desk or sanitary cleaning. Moreover, robots could do their work all night, a time when professionals claim higher wages, and they are a sound alternative for cleaning in sensitive areas.

Although automatized cleaning is a topic of high interest in the commercial sector not many professional cleaning robots have been developed so far. The few existing industrial cleaning machines are meant to operate in structured and rather static environments with large free spaces, e.g. gymnasiums, airports, and supermarkets [1], while cleaning all surfaces equally. The application of cleaning robots in unstructured office environments is yet to be commercialized.

This paper introduces the world's first prototype autonomous robotic cleaning assistant which comprises professional floor cleaning and waste disposal exactly oriented to real world requirements. Specifically, the robot is supposed to enter all rooms of a working environment while identifying trash cans, clearing them into a waste container mounted on a movable cleaning trolley, searching for pollutions at the floor and cleaning those where necessary. The latter task is crucial for time and energy efficiency since floor cleaning is directly oriented on demand instead of blindly cleaning all ground surfaces on a daily basis. The main contributions of this work are

1) a detailed description of the autonomous robotic assistant for professional multi-purpose cleaning,
2) an evaluation of effectiveness and efficiency of the implemented solutions based on real world tests,
3) a discussion on further steps towards a commercially feasible prototype,
4) and examples and hints for realizing hardware and software plug and play in service robotics to increase the value and reusability of a development.

In the following, a general overview of the cleaning robot is given in Sec. II and the individual modules are discussed and evaluated in Sec. III. Subsequently, Sec. IV yields an assessment of the overall system performance. We conclude with an outlook in Sec. V.

This project has received funding from the German Federal Ministry of Economics and Technology (BMWi) through grant 01MA11005 (AutoPnP).

The authors are with the Institute for Manufacturing Engineering and Automation, Fraunhofer IPA, 70569 Stuttgart, Germany. <first name>.<last name>@ipa.fraunhofer.de

II. System Overview

The professional cleaning assistant prototype was realized on a Care-O-bot 3 platform which is a multi-purpose service robot equipped with a four wheel omni-directional base, three laser scanners for localization and navigation, a flexible torso and head that allow repositioning of the camera systems installed in the head as well as a 7-DOF manipulator [2]. The sensor head contains a stereo camera pair and an Asus Xtion Pro Live RGB-D sensor. Internally, the robot provides two Linux computers with an Intel Core 2 T7400 (2.16 GHz) processor and 4GB RAM each. These computers can operate the Care-O-bot completely autonomously within the battery runtime of up to four hours. To enable the robot for different handling or manipulation tasks the arm was upgraded with a tool changing interface from Schunk. The interface standardizes the mechanical and the electrical connection to mountable devices and can drive the lock between the arm attachment and its counterpart at the tool side by means of an electric motor. For the cleaning tasks we adapted two tools with the changing system, a three finger hand for general manipulation tasks and a vacuum cleaner. Obviously, the generic interface would easily allow attaching further cleaning equipment, e.g. a mop for wet cleaning. The hardware portfolio is complete with a tool trolley that carries containers for waste and fixtures for detached tools. The photograph in the upper left corner of Fig. 2 shows the Care-O-bot pulling the trolley with its attached hand while the vacuum cleaner is deposited at the back of the trolley.

The whole software project is embedded in the Robot Operating System (ROS) environment which is defining standards for the modularity of software and managing the communication between modules over Ethernet. To orchestrate single functionalities in a way that generates a useful system behavior, the cleaning robot is controlled by a SMACH state machine. Each state accesses capabilities of the robot and transitions to the next state depending on the outcomes. State machines are well-suited to model linear action sequences in a robot conducting a certain task that may not be interrupted frequently by a set of other activities before completion.

The cleaning application is designed to work out of the box within any arbitrary environment with the only prerequisite of having obtained a navigation map beforehand. The complete state machine of the cleaning task is depicted in Fig. 2. The sequence of actions was developed in close collaboration with an advisor from a large German cleaning company. It begins with the automatic division of the map into compact areas that represent rooms or parts of larger spaces (`ANALYZE_MAP`, Sec. III-A). Then each of these rooms is visited while the tool trolley is moved closer to the current working area if it is located too far away (`GO_TO_NEXT_UNPROCESSED_ROOM`). Inside each room the recognition modules for dirt and trash bins get activated and the robot examines the floor by driving systematically to different accessible places (`INSPECT_ROOM`, Sec. III-B and III-C). At this time, the robot is equipped with a hand so that it will directly clear all found trash bins of the room into the waste container of the tool trolley (`CLEAR_TRASH_BIN`, Sec. III-D). The dirt stains instead can only be cleaned with the vacuum cleaner but a tool change at each room would be less efficient than storing all polluted places in a map and cleaning them altogether at the end of the cleaning sequence. Hence, the robot visits the next room and repeats the inspection procedure. After all rooms have been processed the robot changes its hand against the vacuum cleaner automatically at the tool trolley (`CHANGE_TOOL`, Sec. III-E). Subsequently, the robot retrieves a list of polluted locations, moves to each of them (`GO_TO_NEXT_UNPROCESSED_DIRT_LOCATION`) and attempts to clean the spot with its vacuum cleaner (`CLEAN`, Sec. III-F). Afterwards, the robot verifies the cleanliness of the location and stores persistent stains (`VERIFY_CLEANING_SUCCESS`). The cleaning procedure repeats until all locations have been visited. Finally, the results (`PROCESS_CLEANING_VERIFICATION_RESULTS`) are presented to the operator on a graphical interface, who might clean remaining spots appropriately and inform the system about false alarms that will become better investigated during the next run.

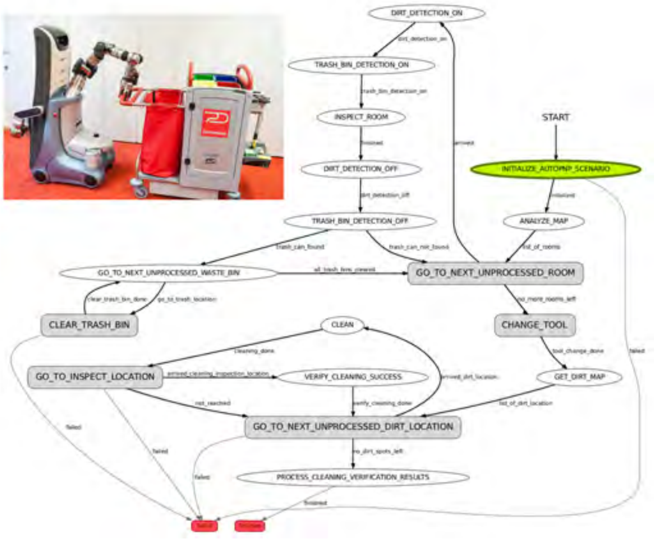

Fig. 2. State machine of the cleaning task. Boxes in dark shades of gray represent sub-state machines that contain further states. The photograph displays all hardware devices of the cleaning system.

III. Functional Modules

This section discusses a selection of the most important modules of the robotic cleaning assistant. Besides advancing a few scientifically new approaches it also sketches the amount of solid engineering necessary to solve the plethora of smaller issues that arise with systems that have to fulfill real tasks in real human environments.

A. Map Segmentation

Segmenting the map into compact room-like areas is an important prerequisite for efficient cleaning since it allows for planning a good trajectory through the environment as will be shown in Sec. III-B. An automatic room segmentation is desirable to minimize setup efforts.

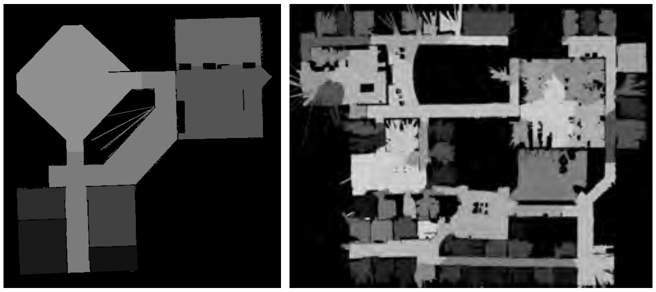

Fig. 3. Examples for map segmentations into room-like working areas. Larger spaces are separated into several regions.

1) Relevant Work: Room segmentation has been proposed in an interactive fashion by [3] as well as automatically, especially for creating topological maps. [4] create topological maps based on pixel-wise semantic classification with AdaBoost while nodes correspond well with rooms. [5] segments space with Voronoi diagrams, which do not constantly yield very room-like segments. The approach of [6], which is using morphological operators on fuzzy grid maps to divide regions with the watershed algorithm, shares some ideas of our very simple yet powerful algorithm.

2) Approach: The map segmentation algorithm works on a grid map M_1 whose pixels are initially labeled accessible at mapped accessible areas and inaccessible at walls and outer areas. The walls of M_1 are grown by one pixel with the morphological erosion operator. Then a connected component analysis determines whether the erosion results in separated accessible areas. If a separate region has a certain size between a lower and higher threshold, which represent the desired segment size dependent on the number of erosion steps, all its cells become labeled as an individual room r_i in a second map M_2, which is a copy of the original map M_1, and are set inaccessible in map M_1. This procedure repeats with the erosion until all accessible cells have been removed from map M_1. Following, the labeled areas in M_2 become extended into the accessible unlabeled space with wavefront propagation until all accessible cells have been labeled.

3) Results: The resulting segmentation yields compact areas that usually correspond with a room or a part of a larger area. The resulting sizes range between 15 and 80 m^2 which represents a good working unit for the given task. Fig. 3 displays two typical results for a smaller and a larger office environment. The former map represents the testing area of the system, the latter is a publicly available floor plan. In all our experiments the automatic segmentation generated a set of well-sized compact areas suitable for the robot to fulfill its cleaning tasks adequately.

B. Exploration Algorithm

Once a segmented map is available the robot plans its cleaning task by visiting one room after the other. For efficiency the sequence of entered rooms is optimized as a traveling salesman problem (TSP) since robot movement is one of the time consuming factors. Moreover, the robot has to operate near its tool trolley which is also used for waste disposal and whose displacement is taking a significant portion of time. Hence, a minimal set of trolley locations is determined, first, and then the TSP becomes solved at two levels, the sequence of visiting these global positions as well as the local sequences of entering each room. Inside a room, the robot needs to observe the accessible floor areas. Prior computation of an optimal path is not possible since dynamic and unmapped obstacles may repeal the optimality computed with the static map. For a systematic exploration behavior the room is overlaid with a regular grid of observation points and the robot approaches those points guided by an energy functional that rewards little movement and proximity to already visited locations.

1) Relevant Work: The TSP has seen numerous approaches for efficient solutions that are exact, e.g. linear programming [7] or branch-and-cut [8], or heuristic, like nearest neighbor, neural networks or genetic algorithms [9]. The trolley placement can be formulated as a set cover problem [10] and is solved here with an application-specific heuristic. The floor inspection task could be considered as an art gallery problem [11], however, overlap of observations strengthens the dirt detection results and is therefore desired. Trajectory planning for cleaning has been discussed by [12]. The robot navigation uses the ROS navigation stack [13].

2) Approach: First, a minimal number of trolley positions is computed given the maximally acceptable driving distance $d_{\max}$ between room center and trolley because robot and trolley movements consume much time. To determine such a set a graph G_t is constructed whose nodes represent the rooms. Between every pair of rooms an edge is present if the closest driving distance between both room centers on the static grid map is at most $2 \cdot d_{\max}$. The distances are computed along the path found by the A* algorithm with Euclidean metric. Let C_t be the set of all cliques within G_t then the minimum number of trolley locations is defined as the minimum set cover of G_t by the sets contained in C_t. Since the set cover problem is NP-hard [10] the following heuristic is applied which is tailored to the problem characteristics. Let a point in its maximum size clique be defined as an *end point* if it has no connection to another clique. This definition corresponds with rooms that are located at the end of a corridor. We pick one of those rooms or a random room if there are no end points and assign its maximum size clique to a trolley location. All participating nodes are then removed from G_t and C_t. Next, the set of end points is recomputed and the next clique is chosen as before. The algorithm terminates when all nodes have been assigned to a trolley location. The optimal metric trolley placement within a clique is determined by minimizing the driving distances between trolley and room centers. The visiting order of trolley locations is optimized as a TSP with the popular CONCORDE algorithm [8], which is also applied within the compact local groups of rooms around each trolley parking spot.

Once the robot has entered a room the following floor inspection strategy is applied: first, the room area is populated with a quadratic grid of observation points with spacing d_o.

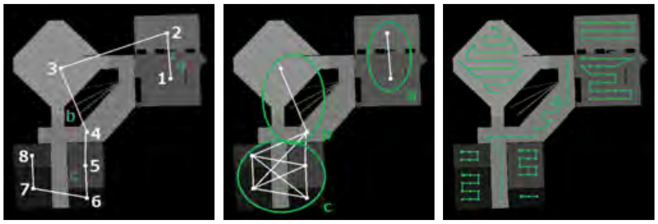

Fig. 4. Left: trolley placement and visiting order (green letters) and the sequence of visited rooms (white numbers). Center: graph G_t with determined cliques (green circles) at end points. Right: room exploration trajectories (starting at the green arrows).

Then the robot starts its inspection at a point in a corner of the room. On each arrival at a target point or when it is not accessible the respective entry is appended to a list L of processed locations. The next unseen target location is selected from the grid which minimizes the energy functional

$$E(l,n) = d_t(l,n) + d_r(l,n) + N(n) \tag{1}$$

$$d_t(l,n) = \frac{\sqrt{(l_x - n_x)^2 + (l_y - n_y)^2}}{d_o} \tag{2}$$

$$d_r(l,n) = \frac{\|l_\theta - n_\theta\|_{\text{angle}}}{\pi/2} \tag{3}$$

$$N(n) = 4 - \sum_{k \in \text{Nb}_8(n)} \frac{|k \cap L|}{2} \tag{4}$$

where l denotes the (x, y, θ) coordinates of the current location and n is the coordinate of the potential next location. The translational distance d_t is measured in units of d_o, the rotational distance d_r in units of $\pi/2$. Function $N(n)$ represents half the number of not yet visited locations among the 8 neighbors $\text{Nb}_8(n)$ around n. To avoid unnecessary and time-consuming attempts in approaching the selected location, the laser scanner readings are evaluated for dynamic or unmapped obstacles that may occlude the goal. If no accessible location can be found within the $\frac{d_o}{2}$ vicinity, the target is dropped and replaced by the next.

3) Results: A typical example for the trolley placement, the order of visiting those locations and the sequence of visited rooms is provided in the left image of Fig. 4. The corresponding graph G_t for this map is displayed in the center of Fig. 4. During all experiments the trolley placement and traveling routes gave very reasonable results. The floor inspection trajectories are shown in the right image of Fig. 4. Although not all paths are optimal they provided complete observability of the ground area at each testing environment.

C. Dirt Detection and Learning

In daily professional cleaning the ground must only be cleaned where necessary for the sake of efficiency. Consequently, the robot needs a method to judge the cleanliness of the floor while moving through the room. To increase the simplicity and generality of using the robotic cleaning system a visual dirt detection method is applied which judges cleanliness on a per image basis without any need for learning the patterns of the clean ground or the appearance of any pollution. Since high detection rates imply a number of false alarms for that approach, it is extended with a learning component that stores local appearance characteristics of false alarms identified by the operator together with their map coordinates for thorough comparison on later visits.

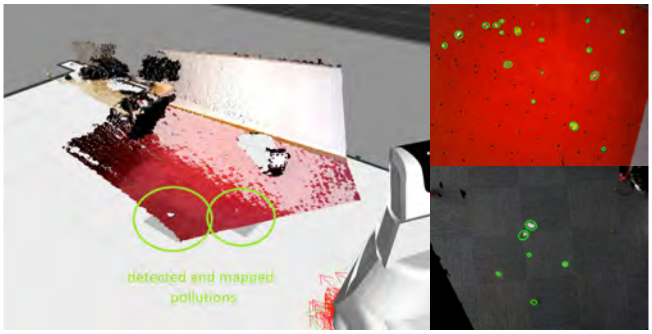

Fig. 5. Visualization of the robot's RGB-D perspective on the scene (left). The two gray map cells indicate the detection of the two dirt items in the foreground. Examples for dirt detection on different carpets (right).

1) Relevant Work: Dirt detection for selective cleaning is a quite new field of research. While tasks like image restoration make use of temporal segmentation methods [14] dirt detection for persistent spots employs visual attention algorithms [15], [16]. We extend our learning-free dirt detection algorithm [17] with image matching methods for the reduction of false alarms in this work.

2) Approach: The basic learning-free dirt detection method [17] requires an RGB-D sensor to segment the floor plane and mask pixels in the color image that do not belong to the ground. Following, the masked image is transformed into a virtual camera perspective perpendicular above the ground at a fixed height. This step allows the algorithm to compensate for different mounting positions of the sensor to keep it easily transferable to other systems. Then a spectral residual filter removes the common parts of the image emphasizing the outstanding remainder. It is implemented as the difference of the original and the smoothed logarithmic amplitude spectrum. The filter response is calibrated against the response of a modified scene image which contains artificially added standard pollution. Dirt is considered every peak above a threshold T_d in the calibrated dirt image. Through the localization of the robot it is furthermore possible to collate the detections in a common map and accumulate dirt responses from different perspectives as depicted in Fig. 5. Judging a location as polluted only if several positive responses are collected effectively filters sporadic misinterpretations [17].

Since that dirt detection method operates fully uninformed it regularly produces false alarms when tuned for high detection rates, e.g. at power plug covers at the floor. To mitigate the impact on performance the detection system has been enhanced with a learning module for the appearance of typical false alarms. After each cleaning attempt an image of the polluted spot is stored if the validation glimpse still detects dirt. The operator gets those spots displayed at his interface and may clean them appropriately or mark them as false alarm. Then the normalized view on that false alarm becomes stored permanently in an internal database together with its capturing perspective and map coordinates. If this location triggers a dirt detection at the next visit again, it will be compared to the stored normal appearance before asserting a pollution according to the following algorithm.

First, the perspective-normalized image of the potential pollution needs to be aligned with the stored reference view R. To compensate for localization inaccuracies an image patch P of the current view is selected with twice the side length of reference image R. Then weighted gradient direction histograms with bin width of 1° are constructed for both patches. The gradient directions are sampled from the Sobel-filtered patches and count with a spatially distributed Gaussian weight centered at each image patch and with $\sigma = 0.4 \cdot (0.5 \cdot (\text{width}(R) - 1) - 1) + 0.8$. The rotational offset $o_r = \tilde{o}_r + \delta o_r$ between both patches can be roughly estimated as $\tilde{o}_r$ from the given capturing perspectives and is optimized as δo_r within a $\pm 10^\circ$ range by matching the gradient histograms with a histogram intersection kernel. Following, the current view is rotated by o_r, yielding image P_r. The translational offset o_t between P_r and R is then determined by minimizing the sum of squared differences between the gradient magnitudes of both patches. The aligned patch $P_{r,t}$ is finally selected from P_r with the size of R. Using the gradients for alignment renders the alignment method invariant against changing illumination.

The aligned patches $P_{r,t}$ and R are compared for visual similarity by constructing a pyramid of local gradient magnitude histograms because pixel-wise comparisons turned out to be too unstable on minimal alignment flaws. The patches, which typically have a size of 100x100 pixels, are divided into a 10x10 and a 9x9 cell grid. For each cell, an 4-bin histogram over local gradient magnitudes is computed. All of the 181 corresponding 4-bin histograms are compared with a histogram intersection kernel between both patches. If one of them scores below 70% of the maximum the presence of real dirt is asserted. Otherwise the initial dirt detection is judged as a false alarm and not cleaned. Fig. 6 shows an example for the learning based dirt detection which aligns a potential false alarm to its reference image and compares histograms on the gradient response.

3) Results: Dirt detection operates on images of 640x480 pixels resolution from the built-in Asus Xtion Pro Live RGB-D camera mounted at 1.20 m height. These images are transformed into the virtual camera perspective facing the ground perpendicularly with a resolution of 300 pixels/m. Hence, the learning-free dirt detection module can detect spots that occupy an area with a diameter between 5 mm and 50 mm. Larger spots could be detected with pyramidal downsampling of the image. In [17] we report a dirt recognition rate of 90% at a false alarm rate of 45% for the learning-free method. On a set of 96 false alarms caused by structure we measured an alignment success rate of 85.4% with 8.3% rotational errors ($>1^\circ$) and 6.3% translation errors (>2 pixels). On half of these images we added real dirt close to the false alarm. Altogether, 70.2% of the real false alarms could be identified while only 13.7% of the real pollutions were missed.

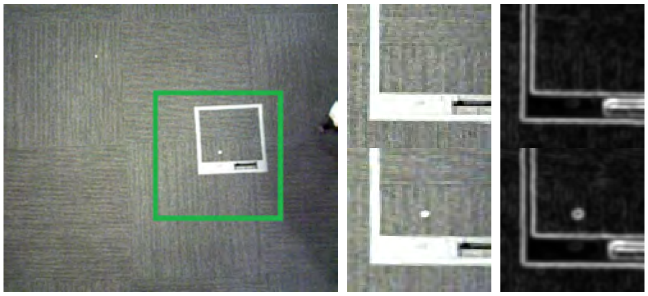

Fig. 6. Example for the reference image based dirt detection. The roughly located image patch (left image, green frame) becomes aligned to the reference image (center column) and the gradient images (right column) are compared.

D. Trash Can Clearing

During the room inspection behavior, the robot does not only search for dirt stains but also identifies trash bins. The detection of attached Pi-Tag markers proved most reliable and computationally economical in comparison with other approaches for object recognition or categorization and is acceptable for the real world usage. To handle the detected trash cans the robot is navigated into the proximity and grasps the bucket with its arm. After lifting it into a carrying position, the robot approaches the tool trolley, clears the trash bin and returns it to its original location.

1) Relevant Work: Detecting trash cans is an object detection or classification problem. Many methods are published on the detection of object instances [18]–[21] or categories [22]–[24] in the service robotics domain. However, to compensate for the variability of the environment those algorithms typically need to be so complex that they compute with 1 Hz or slower. Faster recognition algorithms can be used by modifying the environment slightly with specialized markers [25], [26]. For planning manipulation tasks the convenient MoveIt framework [27] was utilized.

2) Approach: Clearing trash bins starts with their detection in the environment. Attempts were made to utilize the object recognition framework of [20] which targets the indoor service robotics domain. Although yielding high success rates of 87% the method was computationally too demanding to run in parallel with the dirt detection algorithm on one computer. Pi-Tag markers [25] were then attached to the trash bins as a reliable and computationally tractable solution. These markers arrange 12 dots to encode two cross-ratios, which are constant even under perspective transformation so that only a calibrated color camera is needed for marker detection and 3d pose estimation.

Once a trash can marker is identified its pose is transformed to the map coordinate system and written into a local accumulator. After room inspection all local accumulators are approached for grasping. The robot is navigated to an accessible point at a given perimeter around the trash bin. The arm movement to grasp a trash bin is then planned with the MoveIt framework [27] that has been setup for the mounted Schunk LWA arm to avoid self collisions with the robot but has no complete knowledge about the surroundings. Hence, the arm movements are limited in their speed for

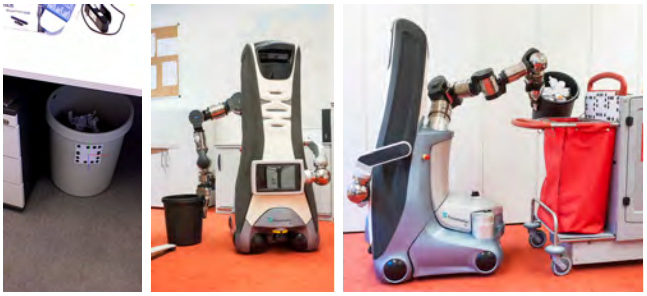

Fig. 7. Trash can detection via Pi-Tag markers, grasping and clearing.

safety at this early development stage. The first trajectory is planned to put the hand above the trash bin, open it, and finally move the hand inside at the rim. Then the fingers close and another trajectory brings the waste bin into a carrying position close to the robot at its backside. Next, navigation takes the robot to the last known position of the tool trolley, which is also covered by several Pi-Tag markers. After detecting the relevant markers, the robot corrects its relative pose to the trolley and starts with the clearing sequence. For clearing, the trash bin has to be lifted high before turning it upside down into the direction of the waste container. This is managed by splitting the sequence into two parts with the constraint of an upward facing trash bin for the first trajectory. The trolley is modeled as an obstacle at the known location so that no impacts can occur. Finally, the trash can is returned to its origin.

3) Results: The utilized Pi-Tag marker detection method could run in parallel with the dirt detection module at processing rates of 6 Hz for both methods. Markers of edge length 12 cm were attached to the trash can and the trolley allowing for a reliable detection with ±2 cm accuracy from up to 2.5 m distance and ca. 45° viewing angle at 640x480 pixels resolution (Fig. 7, left). A valid plan for grasping the trash can could always be found but since obstacles in the environment were neglected during planning the real success rate reached only 90% during tests in practice (Fig. 7, center). The lifting and grasping power of the employed hardware permitted to lift trash cans at several sizes up to 40 cm diameter with a maximum weight of 2 kg. The trolley could always be found and approached with an accuracy of ±2 cm so that cleared waste always hit the container (Fig. 7, right).

E. Tool Change

When the robot has finished its inspection and trash can clearance tasks it needs to vacuum clean the discovered pollutions. The hand can easily be exchanged with a vacuum cleaner attachment using the tool changing system. The change, however, requires some time since the tool changer must be aligned very precisely (±1 mm) via visual servoing.

1) Relevant Work: Automatic tool changing mechanisms are established in industrial robotics [28], [29] but rather exotic in service robotics [30]. The cleaning robot adopts an industrial interface for automated tool changing but the uncertainties of unstructured environments require visual servoing methods [31] for automatizing the full process.

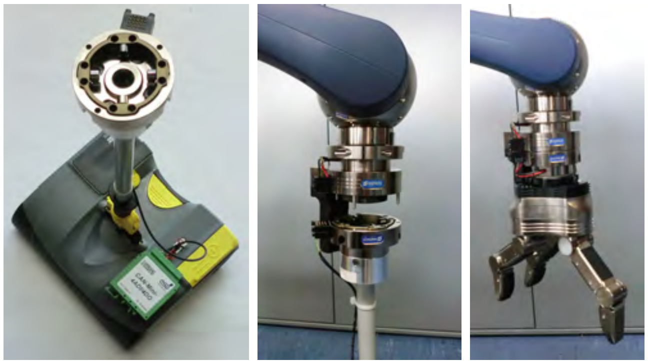

Fig. 8. Vacuum cleaner with tool changer adapter, tool changing mechanism and attached hand.

2) Approach: To avoid the troublesome operation of cleaning tools with the robot's three finger hand a tool changing system has been attached to the arm and its complements to the hand and the vacuum cleaner. Mechanically, the tool changer connects its counterpart by closing an inner rotational lock as illustrated in Fig 8. Opening or closing the lock can be commanded directly by the robot. The electric connection may be established with up to 20 pins of which 2 are used for power supply and 2 further pins are connected to the CAN bus interface. This setup yields an interface specification that may be utilized by a plethora of devices that communicate over CAN and boosts the plug and play idea behind that module. A master driver can query the CAN bus periodically for the ID of attached devices which lets the robot load the necessary device driver for the respective tool, i.e. the hand driver, that commands finger movements, or the vacuum cleaner driver, which can switch the device on or off.

The autonomous tool change positions the robot in close proximity to the tool trolley, releases the attached tool in one of the mountings and attaches a new tool from another mounting. This process demands high positioning accuracy in the range of ±1 mm which cannot be achieved by simply detecting the trolley's pose and moving the arm into the respective relative position. The latter process may achieve a precision of about ±5 mm for a well-calibrated system. To perform high precision positioning, visual servoing is used. It is based on the determination of the relative pose between the trolley reference coordinate system and a Pi-Tag marker attached to the last link of the arm. Their relative positioning error is reduced by commanding the respective metric arm movements in a control loop until the desired relative position is met with sufficient accuracy.

3) Results: During all tests the tool change worked technically reliably, releasing the old and fixing the new device properly and loading the respective driver for the type of detected module. The relative positioning accuracy with visual servoing attained a positioning accuracy of ±0.8 mm on average with a maximum displacement of ±1.7 mm. During 20 trials, uncoupling an attached device to different mountings worked reliably in 100% of the test cases whereas the coupling procedure, which needs the highest positioning accuracy, was successful in 80% of trials.

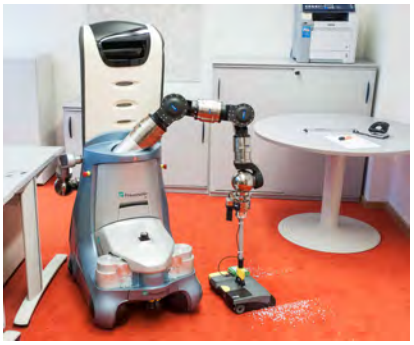

Fig. 9. The robot utilizes its vacuum cleaner to remove a pollution.

F. Vacuum Cleaning

After attaching the vacuum cleaner the robot retrieves a list of all dirty locations and verifies those locations marked as false alarms during earlier runs with the appearance comparison method explained in Sec. III-C. The remaining list of real pollutions is then checked for neighboring dirt cells which become agglomerated into one larger spot. Finally, each agglomerated spot is visited after the other for cleaning. Thereto, the robot is commanded to an accessible point on the perimeter of the dirt stain. Then the arm places the vacuum cleaner on the ground at the dirty location. If space allows the robot now drives back and forth or left and right to clean the spot. Within narrow areas respective arm movements are planned with the MoveIt framework. The cleaning activity is depicted in Fig. 9. After cleaning the vacuum cleaner is stored in a carry configuration and the robot moves to an observation position on a larger perimeter around the previously dirty location. The dirt detection module is activated again to verify the success of cleaning. If the robot failed to remove the pollution, which might be a persistent stain or a false alarm, its location and normalized image are stored and transmitted to the operator's interface. The professional may then judge the pollution as false alarm, which is remembered by the robot for later occurrences, or clean it with appropriate techniques, e.g. wiping.

IV. System Performance and Discussion

The cleaning assistant was tested in simulation and in real world experiments in two different office environments, one containing a long floor with offices at both sides and another one with the floor plan depicted in the left image of Fig. 3. Besides the encouraging results of the single modules (see Sec. III) the performance of the system as a whole proved to be surprisingly well for a first prototype. Ten test runs were conducted in the two office environments in total which contained dark carpet, linoleum and visually challenging red carpet with a regularly dotted black pattern as well as tiles. One trash bin was placed in each room with markers well-visible from several perspectives. Experiments showed that it is sufficient to put 3 to 4 markers on a trash can to remain independent of its pose. We furthermore dumped 1-2 visually perceivable dirt spots of various kinds (see [17]) in each room with a diameter of 5 mm and larger. Tab. I summarizes the measured average performance of these 10 test runs.

TABLE I
Experimental results (r=recall, p=precision).

task	success rate	time consumption
map segmentation	100%	1 s
room visiting	100%	1 min per room change
room exploration	100%	2 min per room (20 m^2)
dirt detection	r=95%, p=62%	included in exploration
vacuum cleaning	90%	2 min per spot
trash bin detection	100%	included in exploration
trash bin clearing	90%	3-4 min per trash bin
tool change	90%	3 min

As to expect, map segmentation, room visiting and room exploration always yielded good results. The tool change failed only once during the coupling procedure. Moreover, all trash cans could be detected by means of Pi-Tag markers, however, only 90% of them could be manipulated. The remainder is comprised of trash cans located completely below a table which could not be grasped because of occurring arm collisions or limited reach. Once grasped, a trash can could always be cleared with one exception when the robotic fingers clamped a very big piece of paper which prevented the waste from falling down. Pollutions could be found in 95% of the cases. The missed pollutions were majorly caused by insufficient imaging quality of the sensor with respect to resolution and contrast, or originate from too short observation periods and bad localization of the robot. The reported precision of 62% suffers from false alarms on structural or textural exceptions at the ground and does not include the utilization of the learning component. Using the learning algorithm improved precision to 89%. Vacuum cleaning the found spots succeeded in 90% of the cases whereat failures were mainly caused by dirt placed close to a corner with the navigation strategies failing to create collision-free paths. In summary, 90% of the trash bins could be cleared and 86% of the dirt was removed during the tests.

Table I also provides information on the average times consumed by the single subtasks which add up to a total of 10 minutes per room that contains one garbage can and two dirt stains. This yields a performance of approximately 120 m^2 per hour, which is 4 times less than for a professional who attains up to 450 m^2. Although the target performance is lower for the robot with 300 m^2 according to our professional consultant, because the robot can work significantly longer and during the night, this raises questions how time efficiency can be enhanced. The answer is two-fold; given the prototypical system realized on a multi-purpose robot the following measures are the most promising:

- Incremental 3d environment modeling [32] could add a static collision model in MoveIt for safer, hence faster, and more flexible arm movements. The sensor placement of Care-O-bot does not allow for online arm workspace monitoring for the tasks of this application.
- Avoiding to store the vacuum cleaner during each robot movement where unnecessary, e.g. in larger spaces.
- The addition of another camera at the side would accelerate the cleanliness verification.
- The usage of better computing hardware for higher processing rates of the dirt detection algorithm and hence higher exploration speeds.

Roughly estimated those improvements could reduce the vacuum cleaning time by 1 minute and the trash bin clearing effort by 1-2 minutes. Nevertheless, for a real commercial application the universal robot platform eventually needs to be replaced by a specialized construction to meet the target performance and price of ca. $50,000 which is considered cost-effective by the professional cleaning company. A performance of 300 m^2/h corresponds with cleaning 15 rooms per hour, what leaves 4 minutes each, e.g. 2 minutes for inspection and cleaning and 2 minutes for trash can clearance. Amongst the major enhancements we hence see:

- Building a smaller and more compact robot which could better approach trash cans below tables.
- Integrating all tools on-board would allow to clean a room directly after inspection saving the time for a second visit.
- Mounting all cameras at optimized positions for efficient dirt detection and cleaning verification without additional base movements.
- Using a simpler arm kinematic tailored for trash can clearing for faster handling and price reduction.
- Utilizing incremental, local 3d mapping for safer yet faster operation in general.

V. Conclusion

In summary, this paper has presented the first autonomous robotic cleaning system that combines the tasks of ground cleaning and trash can clearance. The system performance has been evaluated in many experiments that demonstrate the feasibility of automatizing basic cleaning tasks of this kind. The achievements of this early prototype were assessed as particularly promising by our consulting professional cleaning provider. While the general success rates are close to optimal the time efficiency of each task leaves space for improvement. Respective measures have been discussed for the major areas algorithm enhancement, safety measures and specialized robot construction.

Consequently, the next steps with the current system will be the optimization of single components, the integration of local 3d environment modeling for 3d obstacle avoidance and hence faster yet safer motion as well as the introduction of additional cameras for faster verification of cleanliness. In the long term, the construction of a task-specific platform is envisaged that incorporates the experiences learned from the current prototype.

References

[1] H. Endres, W. Feiten, and G. Lawitzky, "Field test of a navigation system: autonomous cleaning in supermarkets," in *Proc. of ICRA*, vol. 2, May 1998, pp. 1779–1781 vol.2.

[2] U. Reiser, C. Connette, J. Fischer, J. Kubacki, A. Bubeck, F. Weisshardt, T. Jacobs, C. Parlitz, M. Hägele, and A. Verl, "Care-O-bot 3 - Creating a product vision for service robot applications by integrating design and technology," in *Proc. of IROS*, 2009, pp. 1992–1997.

[3] A. Diosi, G. Taylor, and L. Kleeman, "Interactive SLAM using Laser and Advanced Sonar," in *Proc. of ICRA*, April 2005, pp. 1103–1108.

[4] O. M. Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard, "Supervised semantic labeling of places using information extracted from sensor data," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 391–402, May 2007.

[5] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation," *Artificial Intelligence*, vol. 99, no. 1, pp. 21 – 71, 1998.

[6] P. Buschka and A. Saffiotti, "A virtual sensor for room detection," in *Proc. of IROS*, vol. 1, 2002, pp. 637–642 vol.1.

[7] G. Dantzig, R. Fulkerson, and S. Johnson, "Solution of a large-scale traveling-salesman problem," *Operations Research*, vol. 2, pp. 393–410, 1954.

[8] D. Applegate, R. Bixby, V. Chvatal, and W. Cook, *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2006.

[9] D. Johnson and L. McGeoch, "The traveling salesman problem: A case study in local optimization," in *Local search in combinatorial optimization*. John Wiley and Sons, London, 1997, pp. 215–310.

[10] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, 5th ed. Springer, 2012.

[11] T. Shermer, "Recent results in art galleries," *Proceedings of the IEEE*, vol. 80, no. 9, pp. 1384–1399, Sep 1992.

[12] C. Hofner and G. Schmidt, "Path planning and guidance techniques for an autonomous mobile cleaning robot," *Robotics and Autonomous Systems*, vol. 14, no. 23, pp. 199 – 212, 1995.

[13] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," in *Proc. of ICRA*, 2010.

[14] J. Ren and T. Vlachos, "Segmentation-assisted dirt detection for the restoration of archived films," in *Proc. of BMVC*, 2005, pp. 359–368.

[15] X. Hou and L. Zhang, "Saliency Detection: A Spectral Residual Approach," in *Proc. of CVPR*, 2007, pp. 1–8.

[16] L. Itti, C. Koch, and E. Niebur, "A Model of Saliency-Based Visual Attention for Rapid Scene Analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[17] R. Bormann, F. Weisshardt, G. Arbeiter, and J. Fischer, "Autonomous dirt detection for cleaning in office environments," in *Proc. of ICRA*, 2013, pp. 1252–1259.

[18] A. Collet, M. Martinez, and S. Srinivasa, "The MOPED framework: Object Recognition and Pose Estimation for Manipulation," *The International Journal of Robotics Research*, 2011.

[19] R. Detry, N. Pugeault, and J. Piater, "A Probabilistic Framework for 3D Visual Object Representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 10, pp. 1790 –1803, Oct. 2009.

[20] J. Fischer, R. Bormann, G. Arbeiter, and A. Verl, "A feature descriptor for texture-less object representation using 2d and 3d cues from rgb-d data," in *Proc. of ICRA*, 2013, pp. 2104–2109.

[21] R. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram," in *IROS*, 2010.

[22] L. Bo, X. Ren, and D. Fox, "Depth Kernel Descriptors for Object Recognition," in *Proc. of IROS*, September 2011.

[23] R. Bormann, J. Fischer, G. Arbeiter, and A. Verl, "Adding rotational robustness to the surface-approximation polynomials descriptor," in *Proc. of Humanoids*, 2012, pp. 409–416.

[24] Z. Marton, D. Pangercic, N. Blodow, and M. Beetz, "Combined 2D-3D Categorization and Classification for Multimodal Perception Systems," *Int. J. of Robotics Research*, vol. 30, no. 11, pp. 1378–1402, Sep. 2011.

[25] F. Bergamasco, A. Albarelli, and A. Torsello, "Pi-tag: a fast image-space marker design based on projective invariants," *Machine Vision and Applications*, vol. 24, no. 6, pp. 1295–1310, 2013.

[26] S. Garrido-Jurado, R. Munoz-Salinas, F. Madrid-Cuevas, and M. Marin-Jimenez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, 2014.

[27] I. Sucan and S. Chitta, "Moveit!" Online, 2013. [Online]. Available: http://moveit.ros.org

[28] M. Gökler and M. Koc, "Design of an automatic tool changer with disc magazine for a CNC horizontal machining center," *Int. Journal of Machine Tools and Manufacture*, vol. 37, no. 3, pp. 277 – 286, 1997.

[29] B.-S. Ryuh, S. Park, and G. Pennock, "An automatic tool changer and integrated software for a robotic die polishing station," *Mechanism and Machine Theory*, vol. 41, no. 4, pp. 415 – 432, 2006.

[30] G. Bolmsjo, H. Neveryd, and H. Eftring, "Robotics in rehabilitation," *Trans. on Rehabilitation Engineering*, vol. 3, no. 1, pp. 77–83, 1995.

[31] F. Chaumette and S. Hutchinson, "Visual servo control, part I: Basic approaches," *IEEE RAM*, vol. 13, no. 4, pp. 82–90, 2006.

[32] G. Arbeiter, R. Bormann, J. Fischer, M. Hägele, and A. Verl, "Towards geometric mapping for semi-autonomous mobile robots," in *Proc. of Spatial Cognition VIII*, 2012, pp. 114–127.