Autonomous Dirt Detection for Cleaning in Office Environments

Richard Bormann[1], Florian Weisshardt[1], Georg Arbeiter[1] and Jan Fischer[1]

***Abstract*— The advances of technologies for mobile robotics enable the application of robots to increasingly complex tasks. Cleaning office buildings on a daily basis is a problem that could be partially automatized with a cleaning robot that assists the cleaning professional yielding a higher cleaning capacity. A typical task in this domain is the selective cleaning, that is a focused cleaning effort to dirty spots, which speeds up the overall cleaning procedure significantly. To enable a robotic cleaner to accomplish this task, it is first necessary to distinguish dirty areas from the clean remainder. This paper discusses a vision-based dirt detection system for mobile cleaning robots that can be applied to any surface and dirt without previous training, that is fast enough to be executed on a mobile robot and which achieves high dirt recognition rates of 90% at an acceptable false positive rate of 45%. The paper also introduces a large database of real scenes which was used for the evaluation and is publicly available.**

I. Introduction

Cleaning of office buildings is a highly competitive market with high efforts regarding time and personnel costs. To lower efforts in this application case there are autonomous industrial cleaning machines available on the market. These machines are rather limited in their use in office environments because they need very structured environments with large free spaces to operate in, e.g. gymnasiums, airports and train stations. For the office market where a more unstructured environment is typical there is only limited use of such machines today.

To enable the use of autonomous cleaning machines in unstructured office environments a collaboration between the robot and human cleaning experts is needed. The robot can inspect and clean wherever it is able to clean by itself and notifies the human expert to do a focused cleaning if required. One core capability for the shared autonomy concept is that the robot is able to perceive dirt in its environment and use that information for planning the cleaning task. Within this paper an approach for a dirt detection algorithm is presented which can be used to locate spots in an office environment which need cleaning during an autonomous inspection by a robot. These spots become furthermore recorded in a dirt map that marks dirt which the robot can clean itself or that needs to be handled by a human cleaning professional.

As such a robotic system must be manageable by technical laypersons the requirements for the dirt detection algorithm are the simple transferability to places with new ground materials and the applicability to any kind of dirt. Especially, laborious teaching tasks on new kinds of dirt or floor materials should be avoided. Besides these practical considerations the algorithm needs to be fast enough to be computed on an autonomous mobile robot and should achieve high detection rates of dirt with a low number of false alarms. The dirt detection system that is presented in this paper fulfills many of these requirements as it has no need for a learning stage for neither dirt nor floor materials, works with any quite regularly structured floor and achieves a high dirt detection rate of 90% at a yet slightly too high false alarm rate of 45%. However, the last requirement is the least severe as cleaning a little bit more than necessary is acceptable for a robotic worker. In order to prove the function of the dirt detection system in real environments, we have recorded a large database consisting of 50 scenes which contain 65 different kinds of dirt. This database as well as a ready-to-use testing framework and the dirt detection system are publicly available at `http://www.ros.org/wiki/autopnp_dirt_detection`.

Fig. 1. Care-O-bot is localizing dirt at the ground.

In summary, the main contributions of this work are

1) a large, publicly available database of dirty ground floors which is the first of its kind to the best of the authors knowledge,
2) a testing framework which facilitates the simple usage of this database,

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

2013 IEEE International Conference on Robotics and Automation ICRA, 1252-1259

3) an improved algorithm for dirt detection that employs a camera perspective normalization and collects the detected dirt in a map and
4) a comprehensive analysis of the proposed dirt recognition system.

The remainder of this paper is structured as follows. After a brief discussion of relevant work in Section II the novel dirt database is introduced in Section III. Following the dirt detection algorithm is explain in Section IV and evaluated in Section V. The paper concludes with a summary and an outlook for the next research steps in Section VI.

II. Related Work

The problem of visual dirt recognition for multifunctional service robots is a completely new field of research that has been introduced with a preliminary detection system by Bormann et al. [1]. The idea of having robots clean the household is quite old and has been realized in several commercially available robotic vacuum cleaners like the iRobot Roomba or the LG Hom-Bot. This kind of cleaning robots is meant to clean the whole floor at once for one or several times a week in personal household environments. Because of the size of those robots available dirt sensing is quite restricted. Usually, piezo-acoustic sensors are employed to inform the robot about the degree of dirt so that it can increase the cleaning efforts where necessary. Professional cleaning, however, has quite different demands for an automatic cleaning system: the application environments are usually large office buildings which have to be cleaned on a daily basis. While there exist large human-operated machines for vacuum cleaning and wiping broad corridors, cleaning in more unstructured and diverse offices is a completely manual job today. Moreover, the style of cleaning inside offices can be characterized as an "only clean where necessary" approach. In order to cope with the amount of scheduled work the cleaning professional has to judge the degree of dirt and clean only at the polluted areas. With today's maturity of basic robot technologies like navigation, manipulation and perception, a mobile platform like Care-O-bot (see Figure 1) has the right size and equipment to automatize such kinds of cleaning tasks.

The first part of this selective cleaning approach is the detection of dirt spots. While other tasks like shopping [2] or cooking [3] have been demonstrated on human-size service robots, cleaning has only been tackled by a service robot at the Robocup@Home challenge by the team of robot Eraser in form of tidying up larger objects. The discovery of small particles of dirt is nevertheless a quite different problem. The search for undesired artifacts in movies has led to approaches that segment the image and remove segments with a different lifetime than their neighbors. However, as dirt does not disappear from the ground by itself in our case temporal filtering is not a viable solution. Moreover, segmentation or edge-based approaches are likely to fail on textured surfaces like carpets. Another idea would be to apply object detection algorithms like [4], [5], [6], [7] that have to learn the appearance of either the different types of dirt or the pattern of the clean floor. The first option has the drawback that a large variety of possible dirt has to be introduced to the system without having the guarantee that any new kind of dirt might be recognized as such. Vice versa, the drawback of the second variant is that each new floor must be input to the system. Although the latter option would allow to detect any dirt on known surfaces it has to overcome the problem of correct alignment of the clean pattern with the measurements.

The approach for the present dirt detection system is inspired by work on modeling the visual attention of humans. Typically, the regions attended first by humans are salient areas in the image that stand out by their color, shape, brightness or movement. Itti et al. [8] introduced a popular saliency detection system which works by means of the first three of these cues. Frintrop [9] extended this system with a top-down component which intensifies or weakens each salient region based on previously gathered object knowledge. A simpler attention algorithm was introduced by Hou and Zhang [10] who found that most images roughly share the piecewise linear shape of their log amplitude spectra. They associate this common shape with the image background whereas deviations from it are said to belong to exceptional foreground objects.

The dirt detection algorithm at hand utilizes a variant of the latter approach to filter the input color image for dirt region candidates. This approach has the advantages of working online on video streams, of being completely free of any learning and therefore working for unseen surfaces and kinds of dirt as well.

III. Dirt Database

In order to conduct a proper evaluation of the dirt detection algorithm that is presented in Section IV, a novel dirt database has been recorded and labeled with ground truth. The database provides 65 different kinds of dirt recorded at 5 floor materials. This section introduces the dirt database by explaining the recording conditions and the contents.

A. Setup for Recording

The whole database has been recorded with the service robot Care-O-bot 3 under realistic conditions. The Care-O-bot 3 platform is equipped with a mobile base, an arm for manipulation and a sensor head for perception as depicted in Figure 2(a). The integrated laser scanners are utilized to localize the robot within the environment during the recordings. This allows the detection software to relate dirt detections to real locations at the floor and furthermore enables the software to integrate detections in a map. In order to increase the reusability of the dirt detection algorithm the data was recorded with the general purpose RGB-D camera Microsoft Kinect mounted inside the robot's head. For the recordings, the flexible torso of Care-O-bot was bowed to the front to maximize the visible ground area close to the robot. This way the recorded data can be retained with the highest possible resolution and the robot is more likely being able to record every part of the ground surface. The camera

(a) Recording setup (b) Black carpet with street dirt

(c) Corridor with food (d) Kitchen with stones

(e) Linoleum with leaves (f) Office with office dirt

Fig. 2. Types of floors recorded for the database. The images are directly taken from the database and show the robot's perspective. Some exemplary dirt is displayed at each scene.

(a) Food (apple sticks, potato chips, peanuts, juice, coffee) (b) Fuzz (yellow and white rope, yellow and white fuzz, hair) (c) Leaves (green and brown leaves, fir needles, hazelnuts)

(d) Office (aluminum balls, styrofoam, laces, paper clips) (e) Paper1 (puncher snippets, paper balls) (f) Paper2 (white snippets of different shapes)

(g) Paper3 (colored paper triangles and rectangles) (h) Stones (bright and dark pebble stones, grit) (i) Street dirt (soccer shoe dirt, cement, sand)

Fig. 3. Types of dirt contained in the database. Each image contains a ruler for size comparison.

is mounted at a height of 1.26 m above the ground and is tilted downwards by 30°.

For capturing the dirt database various kinds of dirt have been distributed at the floor with a minimum distance of ca. 20 cm between the samples. Then the robot was manually driven through the room imitating the behavior of exploring the scene. This means that the robot was usually driven forwards into the dirty area and then backwards in conjunction with several turns. The data stream of the sensors has been recorded into ROS bag-files that contain the `/tf` topic, which allows for the conversion from camera to fixed world coordinates, `/cam3d/rgb/points`, which is basically the colored point cloud of the Microsoft Kinect sensor at a resolution of 640x480 pixels, and the `/cam3d/rgb/image_color` topic, which represents the color image of the Kinect sensor at a resolution of 1024x768 pixels. Given this selection of data, dirt recognition algorithms are enabled to segment the ground plane, to choose between the lower or higher color image resolution for dirt recognition, and to transform the respective coordinates between the camera and the fixed world coordinate systems. The next section details which kind of data has been recorded under the described conditions.

B. Database

The dirt database contains 5 different floor materials that were recorded with 9 groups of dirt and under clean conditions each, resulting in a total of 50 bag-files. The ground surfaces are depicted in Figure 2 and the 9 groups of dirt can be found in Figure 3. Each of these groups contains 6 to 10 different items totaling in 65 distinct kinds of dirt for the whole database. The captured sequences have a length of 12 s to 55 s and contain between 100 and 300 frames of point cloud and image data. The file sizes range between 1.18 GB and 4.86 GB. Every data sequence is accompanied by a ground truth file that labels the position and extent of the present dirt by means of a rotated rectangle as well as the type of dirt. The coordinates of the ground truth labels are provided in the fixed world frame which is possible because of the localization of the mobile robot. This proceeding has the advantage that each piece of dirt has to be labeled only once per bag-file and not for every frame. More specific directions for downloading and using the dirt database are available at `http://www.ros.org/wiki/autopnp_dirt_detection`.

IV. Detection Algorithm

The dirt detection algorithm bases on a spectral residual filtering approach that has been introduced by Hou and Zhang [10] for the computation of visual saliency. By applying several pre- and post-processing steps this approach can be transferred to the problem of dirt detection on surfaces

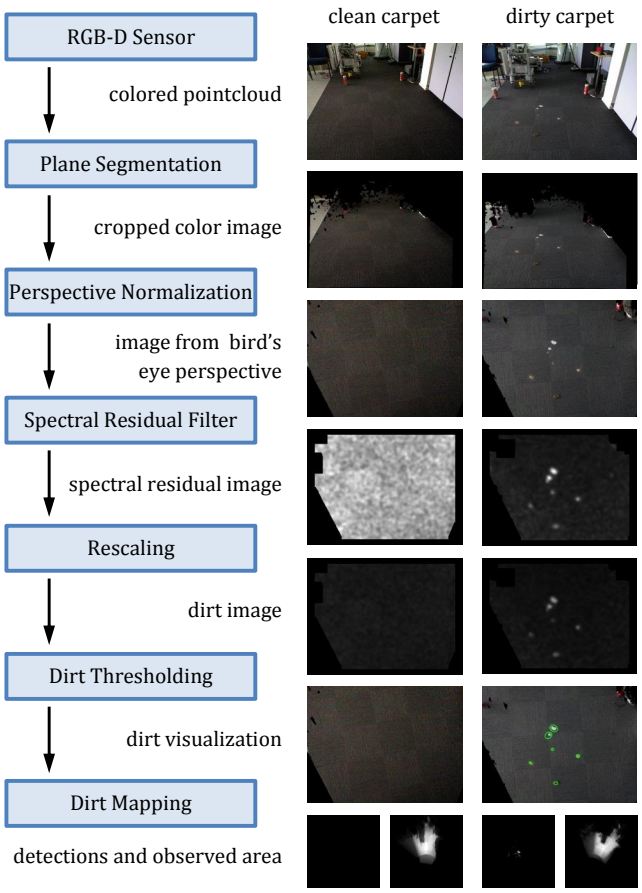


Fig. 4. Processing pipeline of the dirt detection algorithm.

with a more or less regularly structured pattern. An outline of the involved steps of the data processing pipeline is shown in Figure 4 and illustrated in the accompanying video. The individual stages are explained in the following subsections.

A. Ground Plane Extraction

The first step recognizes the ground plane within the point cloud delivered by the RGB-D sensor. This is facilitated by the application of RANSAC-based plane estimation as implemented in the PCL library [11]. The resulting plane equation is tested for two conditions: (i) the plane normal is supposed to direct parallel to the height axis of the world with a maximum deviation from that direction of 30° and (ii) the points of the plane are expected to appear at heights not larger than 50 cm above the ground modeled in the map. Although possible with the transformation tree of the robot model, we do not work with a pre-defined fixed ground plane because this would exclude slanted surfaces like ramps, would not allow to compensate for sensor noise and would eventually request more input from the operator than necessary. If the first estimate of the RANSAC algorithm delivers a different plane of the scene, those points are removed from the point cloud and another plane is estimated. This iteration is either stopped after a fixed number of trials or if the remaining points become too few. With the knowledge of the ground plane it is straightforward to mask all pixels that do not belong to the ground plane with black color in the color image of the scene.

B. Perspective Normalization

As the camera might not always operate in the same position above the ground, which happens at movements of Care-O-bot's torso or when the system is used on a different robot model, it is necessary to normalize the color image with respect to these degrees of freedom. Neglecting different camera poses would result in images with different spatial resolution of the ground dependent on the mounting height of the camera and distorted perspectives as a result of changing tilt angles of the camera. In order to make this variability of appearance transparent to the algorithm a homography $\mathbf{H}$ is estimated which transforms the ground plane as seen in the camera image into a plane seen from a bird's eye view. Before we can estimate this homography it is necessary to define proper coordinates for the normalized perspective. First, we construct a coordinate frame $\mathcal{P}$ whose $x_\mathcal{P}$- and $y_\mathcal{P}$-axis span the ground plane and whose $z_\mathcal{P}$-axis is represented by the plane normal. The axes can be computed by sampling two points $\mathbf{p_1}_{,\mathcal{C}}$ and $\mathbf{p_2}_{,\mathcal{C}}$ from the plane equation

$$ax_\mathcal{C} + by_\mathcal{C} + cz_\mathcal{C} + d = 0 \quad . \tag{1}$$

The index $\mathcal{C}$ at those coordinates refers to the coordinate system of the RGB-D camera, whose origin lies inside the camera and whose $z_\mathcal{C}$ axis points into the scene. The sampled points are used to construct coordinate frame $\mathcal{P}$ in the following way:

$$\begin{aligned}
\mathbf{n}_\mathcal{C} &= (a, b, c)^\mathrm{T} \\
\mathbf{d_1}_{,\mathcal{C}} &= \mathbf{p_2}_{,\mathcal{C}} - \mathbf{p_1}_{,\mathcal{C}} \\
\mathbf{d_2}_{,\mathcal{C}} &= \mathbf{n}_\mathcal{C} \times \mathbf{d_1}_{,\mathcal{C}} \\
(\mathbf{x}_\mathcal{P})_\mathcal{C} &= \frac{\mathbf{d_1}_{,\mathcal{C}}}{\|\mathbf{d_1}_{,\mathcal{C}}\|} && (2) \\
(\mathbf{y}_\mathcal{P})_\mathcal{C} &= \frac{\mathbf{d_2}_{,\mathcal{C}}}{\|\mathbf{d_2}_{,\mathcal{C}}\|} && (3) \\
(\mathbf{z}_\mathcal{P})_\mathcal{C} &= \frac{\mathbf{n}_\mathcal{C}}{\|\mathbf{n}_\mathcal{C}\|} && (4)
\end{aligned}$$

where the coordinates for the $x_\mathcal{P}$-, $y_\mathcal{P}$, and $z_\mathcal{P}$ axes are measured in the camera coordinate system $\mathcal{C}$. Using plane point $\mathbf{p_1}_{,\mathcal{C}}$ as the origin of frame $\mathcal{P}$ the conversion between camera frame $\mathcal{C}$ and plane frame $\mathcal{P}$ can be written as

$$\mathbf{p}_\mathcal{C} = \mathbf{R} \cdot \mathbf{p}_\mathcal{P} + \mathbf{t} \tag{5}$$

$$\mathbf{p}_\mathcal{P} = \mathbf{R}^\mathrm{T} \cdot \mathbf{p}_\mathcal{C} - \mathbf{R}^\mathrm{T} \cdot \mathbf{t} \tag{6}$$

$$\mathbf{R} = \begin{bmatrix} (x_\mathcal{P})_{\mathcal{C},x} & (y_\mathcal{P})_{\mathcal{C},x} & (z_\mathcal{P})_{\mathcal{C},x} \\ (x_\mathcal{P})_{\mathcal{C},y} & (y_\mathcal{P})_{\mathcal{C},y} & (z_\mathcal{P})_{\mathcal{C},y} \\ (x_\mathcal{P})_{\mathcal{C},z} & (y_\mathcal{P})_{\mathcal{C},z} & (z_\mathcal{P})_{\mathcal{C},z} \end{bmatrix} \tag{7}$$

$$\mathbf{t} = \mathbf{p_1}_{,\mathcal{C}} \tag{8}$$

With the $x_\mathcal{P}$- and $y_\mathcal{P}$-axes of the plane coordinate frame we can now define the coordinate system $\mathcal{B}$ for the virtual camera at the bird's eye perspective. Out of the visible ground plane in the original view, only the points with a

maximal distance d_{max} to camera C are transformed to the bird's eye perspective to retain a minimum quality of image resolution for all pixels. Camera $\mathcal{B}$ is placed above the center $\mathbf{m}_{\mathcal{P}}$ of this point set which is determined by

$$\mathbf{m}_{\mathcal{P}} = \begin{bmatrix} (\min p_{\mathcal{P},x} + \max p_{\mathcal{P},x})/2 \\ (\min p_{\mathcal{P},x} + \max p_{\mathcal{P},x})/2 \\ 0 \end{bmatrix} \quad . \tag{9}$$

Furthermore, the plane coordinates, which are measured in meters, have to be transformed into pixels for the projection into the virtual camera system $\mathcal{B}$. We define a fixed resolution of ϱ pixels per meter with which the plane shall be represented within the image plane of the normalized image. In summary, the projection from frame $\mathcal{P}$ into the image plane of the virtual camera $\mathcal{B}$ can be expressed in homogeneous coordinates as

$$\mathbf{P}_{\mathcal{B}} = \varrho \begin{bmatrix} p_{\mathcal{P},x} - m_{\mathcal{P},x} \\ p_{\mathcal{P},y} - m_{\mathcal{P},y} \\ 1/\varrho \end{bmatrix} \quad . \tag{10}$$

As each point of the estimated plane $\mathfrak{P}$ is measured by the RGB-D camera in image coordinates $\mathbf{P}_{\mathcal{C}}$ as well as metric coordinates $\mathbf{p}_{\mathcal{C}}$, which can be associated with image coordinates $\mathbf{P}_{\mathcal{B}}$ of virtual camera $\mathcal{B}$, we can estimate the homography connecting both plane representations from the point correspondences

$$\mathbf{P}_{\mathcal{B},\mathbf{i}} = \mathbf{H} \cdot \mathbf{P}_{\mathcal{C},\mathbf{i}} \ , \ \forall i \in \mathfrak{P} \quad . \tag{11}$$

The transformation into the bird's eye perspective using homography $\mathbf{H}$ has the advantage that all present dirt appears with the same extent and shape independent of its location in the image and that parallel lines are displayed parallel in the image. The first effect ensures that dirt at a far distance is displayed at the same size as it would have at a close distance so that all dirt detections occur at a rate that does not depend on the perceived size as a result of camera distance. The second effect helps the algorithm to recognize the regular surface pattern on tiled floors, which would be displayed with perspective distortion without normalization. Moreover, the conversion to a fixed resolution enables the algorithm to always perceive the same object with the same size independent of camera mounting position, resolution, object location and the type of ground surface.

C. Saliency Computation

After the determination of a normalized perspective onto the floor the dirt detection can be begun with. The first step in doing so is the computation of a spectral residual response that has been described in [10] for saliency calculation as well as in [1] where it was already used for a precursor of the current dirt detection system. Since the detailed procedure and the math can be reviewed in [1] we are only providing a brief intuition for the algorithm at this place.

The original RGB color image I is split into its three channels c_1, c_2, c_3 and each channel is processed with the spectral residual filter. The spectral residual filter computes the Fourier transform for each channel $c_i, i = 1, \ldots, 3$, and subtracts the smoothed logarithmized amplitude image from the original logarithmized amplitude image. The residual is supposed to encode the outstanding parts of the image, which are in our case the dirt spots. The residual is transformed back into the image domain, yielding a response $d_i, i = 1, \ldots, 3$, for each channel. Finally, all squared responses are added up to the spectral residual image D. In short, the computations are the following:

$$\mathcal{L}_i = \log\left(\Re\left\{\mathfrak{F}[c_i]\right\}\right) \quad , \quad i = 1, \ldots, 3, \tag{12}$$

$$d_i = \mathfrak{F}^{-1}[\exp(\mathcal{L}_i - h * \mathcal{L}_i) + \mathcal{P}_i] \quad , \quad i = 1, \ldots, 3, \tag{13}$$

$$D = d_1^2 + d_2^2 + d_3^3 \quad . \tag{14}$$

The spectral residual image D is finally smoothed with a Gaussian filter to suppress high frequency noise. Nevertheless, the spectral residual filter cannot distinguish between images with and without outstanding parts and therefore strongly amplifies noise as visible in the fourth row in Figure 4 for the clean floor.

D. Saliency Normalization

Hence, the spectral residual image needs to be scaled accordingly for a correct interpretation with respect to dirt detection. Again, the normalization procedure is described in detail in [1] so that a summary shall be sufficient here. As selecting the local maxima as dirt does not work because of occurring noise and fixed thresholds do not work for the case of a clean floor or for different floor types, the goal of the normalization is to rescale the filter response so that the presence of dirt can be simply judged by a fixed threshold. The rescaling approach therefore calibrates the filter response D against another filter response D_a to the same image which contains artificially added dirt. This dirt is represented by two black and two white dots of size 7x7 pixels for a 640x480 image. The rescaled filter response D_s is computed as

$$D_s(x, y) = \min\left\{s(D(x, y) - \min D), r\right\}, \tag{15}$$

$$s = \frac{r}{\max D_a - \min D_a}, \tag{16}$$

$$r = \min\left\{\frac{\max D_a}{M}, 1\right\} \quad . \tag{17}$$

As a result the values of D_s are bound to the range $[0, r]$ so that the application of a fixed threshold is now feasible. The calibration against standardized dirt ensures that the response for each part of the image is similar independent of the presence of dirt. This outcome is visible in the fifth row of Figure 4. The ratio r expresses to relation of the maximal filter response in D_a against a maximally expected filter response over all scenes M. The rescaled response D_s is limited to r in order to put the dirt responses from different floor types on a common scale. Consequently, a fixed dirt threshold is not only applicable to clean and dirty surfaces of the same floor material but also to other surfaces.

E. Dirt Thresholding

By virtue of the normalization of the filter response the dirt detection is limited to a thresholding of the rescaled

response D_s with a fixed dirt threshold T_d. Detected dirt is indicated by green ellipses in the images of the sixth row in Figure 4. As we have found that transitions between bright and dark surfaces generate numerous false detections an option was integrated to the system to exclude the pixels that belong to strong lines from the thresholding operation. For line detection a probabilistic Hough transform [12] is applied. Although the algorithm as described so far delivers quite accurate dirt detection results it is reasonable to exploit the localization information available on a mobile robot to increase the certainty of the detections. The accumulation of detected dirt is therefore discussed in the next subsection.

F. Dirt Mapping

The localization of the mobile robot suggests the collection of dirt detections within the same map. The advantages of this are threefold: first, the fusion of multiple observations of dirt from different perspectives strengthens the certainty of these detections. Second, the inspection and dirt removal tasks can be temporally separated which might be desirable. Third, the cleaning results can be directly verified and external help can be informed about the locations of persistent dirt where necessary. All these properties increase the autonomy and robustness of the cleaning robot. To put detected dirt into a common map the detection ellipses need to be transformed back from the bird's eye view $\mathcal{B}$ into image plane coordinates measured in camera frame $\mathcal{C}$. By applying transformation (10) in the opposite direction and afterwards transformation (5), each image point $\mathbf{P}_\mathcal{B}$ of a detection is associated with a metric plane point $\mathbf{p}_\mathcal{C}$. Using the transformation ${}^{\mathcal{W}}\mathbf{T}_\mathcal{C}$ from the transformation tree, which converts coordinates of the camera frame into the fixed `/map` frame, the points $\mathbf{p}_\mathcal{C}$ can be expressed in world coordinates

$$\mathbf{p}_\mathcal{W} = {}^{\mathcal{W}}\mathbf{T}_\mathcal{C} \cdot \mathbf{p}_\mathcal{C} \quad . \tag{18}$$

The dirt map is divided into cells of fixed size. Typically the side length of the cells is 5 cm which is approximately the uncertainty range of the localization in most cases. The transformed detections are matched with all grid cells they touch and each touched grid cell is incremented by one hit independent of the number of pixels that map into it from the same dirt spot. Besides the dirt detections we maintain a second grid which counts how often each cell was visible to the robot. This information is crucial to interpret the absolute numbers of dirt detections per cell correctly as will be detailed in the next section.

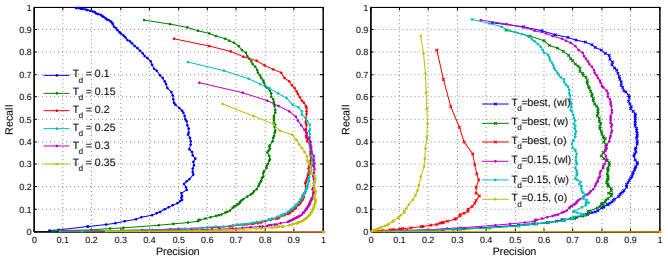
V. Evaluation

Together with the database and dirt detection software we provide the test framework at `http://www.ros.org/wiki/autopnp_dirt_detection` which was utilized for the following evaluation. Algorithms written by other developers can be easily integrated into the test framework enabling contributors to conduct the same experiments as will be discussed in this section.

The output of the test framework provides some freedom for the choice of evaluation measure. We define a grid cell as occupied by dirt if the number of dirt detections within that cell divided by the number of observations of the cell lies above a fixed threshold T_c. This choice has the advantage that dirt is not only judged by absolute numbers of detections but relative to the number of observations. Consequently, dirt detections are independent of observation time by the robot and false detections are not triggered while the robot is not moving as it would be the case if the choice was purely based on the number of detections.

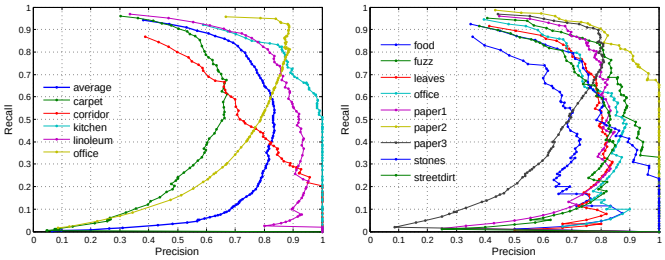
The evaluation of the presented dirt detection algorithm has been conducted over the whole 50 bag-files of the dirt database. First, we tested the algorithm with activated line removal option which is supposed to remove some outliers at strong lines in the image. The recall-precision curves for detecting dirt at different dirt thresholds T_d are displayed in Figure 5(a). Parameter of variation is the detection threshold T_c that decides which of the mapped detections should be judged as dirt. The dirt threshold T_d instead defines the minimum strength of the normalized filter response that is considered as a dirt spot and is applied on each image. As to expect, recall decreases if the threshold for finding dirt increases, the precision of the estimates instead becomes better. However, this pattern does not apply to the lowest part of the diagram yielding a quite uncommon behavior. Nevertheless, this observation can easily be explained with the fact that the true positive set decreases faster than the false positive set with increasing T_c. Since we are more interested in finding as much dirt as possible than in a very high precision of the predictions (attempting to clean a tidy spot is acceptable) dirt threshold $T_d = 0.15$ appears to be a reasonable choice. With this setting the algorithm can detect over 90% of present dirt at a precision of 55%. Detecting the missing 10% of dirt comes at a high price of only 15% precision. This result is not surprising, however, as the database contains several hard cases like dark dirt on dark ground or tiny paper clips that are hard to detect even for human observers of the image stream. Several dirt detection sequences are contained in the accompanying video.

Although a false positive rate of 45% is acceptable in the given application one might ask the question whether the assumption that all visually outstanding parts on the ground can be judged as dirt is too simplistic. The answer is twofold: on the one hand, office environments usually consist of rooms like offices, meeting rooms, corridors or kitchens that expose a relatively regular visual appearance and that are commonly rather tidy. Hence, small outstanding pieces that lie on the ground are very likely to represent dirt. On the other hand, yielding an unsupervised algorithm that is readily available for any kind of floor or dirt was one of the major design goals for the dirt detection system so that it is necessary to settle with this kind of simple assumptions. With respect to the high recall rates we see that this assumption is viable. With respect to the obtained precision we see some space for improvement. The current system already incorporates some simple heuristics to lower the false alarm rate, for example the exclusion of any larger 3D object or structure from saliency analysis or the



(a) (b)

Fig. 5. (a) Recall-precision curves of the present dirt detection system with active line removal on the database at different values for T_d. (b) Recall-precision curves for dirt detections with the current system with line removal (wl), the current system without line removal (w) and the old system (o). All curves were generated with fixed dirt threshold $T_d = 0.15$ as well as with optimized thresholds for each surface (best).



(a) (b)

Fig. 6. Recall-precision curves for (a) individual floor types and (b) individual dirt types obtained with the current dirt detection system with activated line removal and $T_d = 0.15$.

exclusion of longer straight lines. Experiences gained with the system recently propose to extend these measures in future work in two ways that mostly retain the unsupervised character of the algorithm. First, the response of common structures like wires or covers for power plugs in the floor that are frequently misinterpreted as dirt should be filtered by detecting them with additional simple heuristics on their appearance. Second, harder misclassifications that occur infrequently shall be learned through user interaction, i.e. the cleaning expert indicates false alarms in the map and the algorithm adaptively learns their appearance and optionally their typical location.

The current dirt detection system operates at a rate of approximately 3 Hz when running at one core of a mobile I7 M640 with 2.8 GHz and 6 GB RAM. The most time consuming step is the segmentation that typically needs around 160 ms of computation time depending whether the ground plane can be fitted to the data with the first or a later attempt. The whole dirt detection algorithm itself only needs 180 ms to compute. The overall computation time of 340 ms is smooth enough for a successful online application of the method on a mobile robot.

The next experiment demonstrates the significant superiority of the presented approach over the preliminary version of the dirt detection system [1]. The performance of the old version, which does not normalize the perspective into a bird's eye view, is displayed as the yellow curve in Figure 5(b) whereas the current system is associated with the purple curve. The turquoise curve represents the performance of the current system without line removal. All curves were computed with a dirt threshold of $T_d = 0.15$. However, the margin between the systems with normalized perspective and the old variant remain considerable when other thresholds are applied. Indeed can the old detection system obtain recall rates comparable to the new system but at much worse precision rates.

To provide the reader with a better impression of the variance in the results over different types of floor materials, Figure 6(a) presents the recall-precision curves attained with the system with line removal and $T_d = 0.15$ for each kind of floor individually. It shows that precision is significantly lower at the region of high recall rates for carpet and corridor compared to the other floor types. This effect is caused by the great amount of similarly dark colored dirt particles in the first case and by some additional false positives that originate from light reflections at small metal bars put into the ground in the latter case. Figure 6(b) details the detection results with respect to the kind of dirt. It can be observed that almost all kinds of dirt can be recognized at a similar level of confidence with the exception of food that is recognized with lower precision at constant recall rates. The best detection performance can be observed with white paper which is visually most outstanding in general.

Although the design goal of the algorithm to perform equally well on many kinds of floors can be achieved quite well, it is possible to compute optimal dirt thresholds for each floor type to improve the overall performance further as shown by the blue curve in Figure 5(b). The optimal threshold is defined as this one which lets the recall-precision curve pass closest to the ideal point (1,1). In this case, it shows that the current system is still significantly better than the old one but that the gain at the important high-recall area is not so large for the current system compared to using a fixed dirt threshold of $T_d = 0.15$.

The evaluation with the recorded database focuses on the detection rates of different kinds of dirt on different types of surfaces. The recorded scenes are therefore situated in a well-illuminated and fairly tidy environment as to expect during operation typical in offices. To judge the robustness of the dirt detection system with respect to more complex scenes we evaluated the performance additionally on scenes with varying illumination (e.g. direct sunlight, lights switched off), strong shadows, moving people, or heavy clutter recorded in our labs and during presentations at trade fairs. Figure 7 and the accompanying video show that the dirt detection system deals with all these kinds of disturbances in a satisfying way because any 3-dimensional clutter is excluded from dirt inspection beforehand and since the algorithm is rather insensitive to the absolute level of lighting. The method only fails if dim lighting prevents dirt from becoming visible in the image at all, even for a human observer.

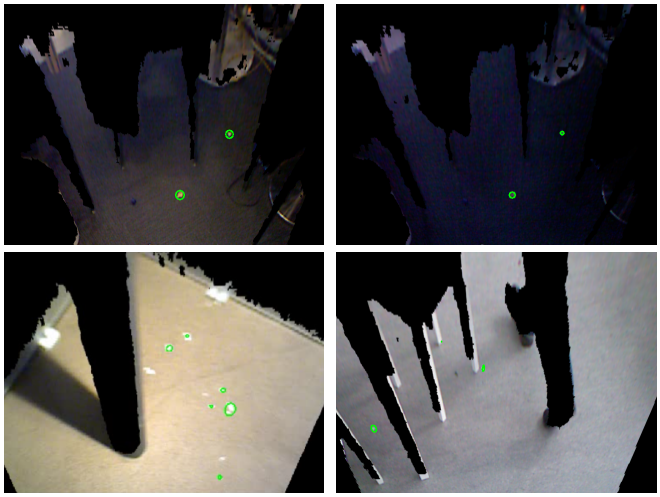


Fig. 7. Dirt detections (green circles) on complex real scenes like clutter below a desk, the same scene with varying lighting, a scene with strong shadows, or a scene with moving people.

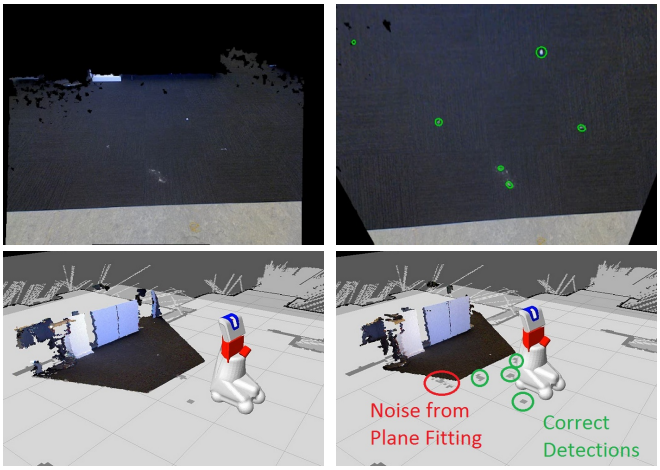


Fig. 8. Visualization of the dirt mapping process.

Finally, we like to visualize the result of mapping the detected dirt. Figure 8 therefore shows the robot's view of the scene and the normalized perspective with detected dirt in the upper row as well as the generated dirt map at two close points in time in the lower row. The visualization is taken from RViz which also displays the corresponding map of the environment, the colored point cloud data of the RGB-D sensor as well as the robot model. The images show that most of the present dirt spots are mapped but also some false positives find (temporary) entry into the dirt map. These false positives typically originate from bad ground plane estimates as depicted in the upper right image or from a bad estimate of the border to walls or objects placed at the floor.

VI. Conclusion and Outlook

In this paper, we have presented a novel, large and freely available database for the problem of dirt detection in office environments. Moreover, a corresponding testing framework has been made publicly available which facilitates the usage of this database. We hope that both items attract the focus of other researchers to this new and important field of application. Furthermore, a strongly improved version of a preliminary dirt detection system has been explained and discussed. This system is able to recognize 90% of present dirt with a false positive rate of only 45% and is fast enough to work on a mobile robot. The collection of dirt detections in a map of the environment has been shown to assist in distinguishing real dirt from false detections and is of great use for the interaction with the operator.

The next steps of the ongoing work include the automatic generation of inspection trajectories for the mobile robot within a mapped environment as well as the manipulation of a real vacuum cleaner to remove the found dirt. Afterwards, we plan to conduct a large scale evaluation of the whole system in one story of a complex office environment over a time of several days or weeks. The dirt detection system itself will be extended by a learning component that can store which items should not be considered as dirt or which areas are commonly misinterpreted as dirt in order to lower the false alarm rate during application. Furthermore, the benefit of using the images with the higher 1024x768 resolution, that are also part of the database, shall become evaluated.

Acknowledgments

The research leading to these results has received funding from the German Federal Ministry of Economics and Technology (BMWi) within the project AutoPnP (01MA11005).

References

[1] R. Bormann, J. Fischer, G. Arbeiter, F. Weißhardt, and A. Verl, "A Visual Dirt Detection System for Mobile Service Robots," in *Proc. of the German Conference on Robotics (ROBOTIK)*, Munich, Germany, May 2012.

[2] D. Pangercic, M. Koppany, Z.-C. Marton, L.-C. Goron, M.-S. Opris, M. Schuster, M. Tenorth, D. Jain, T. Ruehr, and M. Beetz, "A Robot that Shops for and Stores Groceries," AAAI Video Competition (AIVC 2011), San Francisco, CA, USA, 2011.

[3] M. Beetz, U. Klank, I. Kresse, A. Maldonado, L. Mösenlechner, D. Pangercic, T. Rühr, and M. Tenorth, "Robotic Roommates Making Pancakes," in *IEEE Int. Conference on Humanoid Robots*, 2011.

[4] J. Deng, A. C. Berg, K. Li, and L. Fei-Fei, "What Does Classifying More Than 10,000 Image Categories Tell Us?" in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010, pp. 71–84.

[5] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab, "Dominant Orientation Templates for Real-Time Detection of Texture-Less Objects," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2257–2264.

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

[7] J. Fischer, G. Arbeiter, R. Bormann, and A. Verl, "A Framework for Object Training and 6 DoF Pose Estimation," in *Proc. of the German Conference on Robotics (ROBOTIK)*, Munich, Germany, May 2012.

[8] 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.

[9] S. Frintrop, "Vocus: A Visual Attention System for Object Detection and Goaldirected Search," Ph.D. dissertation, Universität Bonn, 2006.

[10] X. Hou and L. Zhang, "Saliency Detection: A Spectral Residual Approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–8.

[11] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[12] J. Matas, C. Galambos, and J. Kittler, "Robust Detection of Lines Using the Progressive Probabilistic Hough Transform," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 119 – 137, 2000.