Segmenting Areas of Potential Contamination for Adaptive Robotic Disinfection in Built Environments

Da Hu[a], Hai Zhong[b], Shuai Li[a,*], Jindong Tan[c], Qiang He[a]

[a] Department of Civil and Environmental Engineering, University of Tennessee, Knoxville, TN 37996 USA
[b] Department of Computer Science and Technology, University of Cambridge, Cambridge CB3 0FD UK
[c] Department of Mechanical, Aerospace and Biomedical Engineering, University of Tennessee, Knoxville, TN 37996 USA

Abstract

Mass-gathering built environments such as hospitals, schools, and airports can become hot spots for pathogen transmission and exposure. Disinfection is critical for reducing infection risks and preventing outbreaks of infectious diseases. However, cleaning and disinfection are labor-intensive, time-consuming, and health-undermining, particularly during the pandemic of the coronavirus disease in 2019. To address the challenge, a novel framework is proposed in this study to enable robotic disinfection in built environments to reduce pathogen transmission and exposure. First, a simultaneous localization and mapping technique is exploited for robot navigation in built environments. Second, a deep-learning method is developed to segment and map areas of potential contamination in three dimensions based on the object affordance concept. Third, with short-wavelength ultraviolet light, the trajectories of robotic disinfection are generated to adapt to the geometries of areas of potential contamination to ensure complete and safe disinfection. Both simulations and physical experiments were conducted to validate the proposed methods, which demonstrated the feasibility of intelligent robotic disinfection and highlighted the applicability in mass-gathering built environments.

Keywords

Robotic Disinfection; Built Environment; COVID-19; Infection Prevention; Deep Learning.

1. Introduction

Diseases caused by microbial pathogens have long plagued humanity, and are responsible for over 400 million years of potential life lost (a measure of premature mortality) annually across the globe [1]. Mass-gathering built environments such as hospitals, schools, and airports can become hot spots for microbial pathogen colonization, transmission, and exposure, spreading infectious diseases among people in communities, cities, nations, and worldwide. The outbreaks of infectious diseases impose huge burdens on our society. For instance, with more than 20 million people infected and 760,318 killed (as of August 14, 2020) [2], the pandemic of the coronavirus disease 2019 (COVID-19) continues to impose a staggering infection and death toll. In addition, the epidemic of flu costs the U.S. healthcare system an average of $11.2 billion each year [3]. During the 2019-2020 flu season, it was estimated that 24,000 to 62,000 people could die because of flu [4]. Each year, there are about 1.7 million hospital-acquired infections in the U.S., resulting in 99,000 related deaths [5]. The disastrous impacts of infections on the society and economy are enormous, highlighting the urgency for developing effective means to mitigate the spread of infectious pathogens in built environments.

Suggested by the World Health Organization (WHO) and the Centers for Disease Control and Prevention (CDC), frequent cleaning and disinfection are critical for preventing pathogen transmission and exposure to slow down the spread of infectious diseases. For instance, during the pandemic of COVID-19, 472 subway stations in New York City were disinfected overnight by workers after a second confirmed COVID-19 case in New York [6]. Deep cleanings are also conducted for school buildings during the closures [7]. Now disinfection is a routine and necessary for all mass-gathering facilities, including schools, airports, transit systems, and hospitals. However, manual process is labor-intensive, time-consuming, and health-undermining, limiting the effectiveness and efficiency of disinfection. First, pathogens can survive on a variety of surfaces for a long period of time. For example, norovirus and influenza A virus were found on objects with frequent human contacts in elementary school classrooms [8]. The coronavirus that causes severe acute respiratory syndrome (SARS) can persist on nonporous surfaces such as plastics for up to 9 days [9]. Second, pathogens spread very quickly within built environments. It was found that contamination of a single doorknob or tabletop can further contaminate commonly touched objects and infect 40-60% of people in the facilities [10]. Hence, the cleaning and disinfection workers are burdened by heavy workloads and subject to high infection risks. Third, workers could be harmed by the chemicals and devices used for disinfection. For instance, nurses who regularly clean surfaces with disinfectants were found to be at a higher risk of chronic obstructive pulmonary disease [11]. Exposure to disinfectants was also found to cause asthma [12]. Therefore, there is a critical need for an automated process for indoor disinfection to replace human workers from such labor-intensive and high-risk work.

To address this critical need, the objective of this study is to create and test a novel framework and new algorithms for a robotic manipulator to conduct automatic disinfection in indoor environments to reduce pathogen transmission and exposure, and thus potentially prevent outbreaks of infectious diseases. The contribution of this study is twofold. First, a deep-learning method is developed to detect and segment the areas of potential contamination. Using the visual simultaneous localization and mapping (SLAM) technique, the segmented areas of potential contamination are mapped in a three-dimensional (3D) space to guide the robotic disinfection process. Second, a new method is proposed to control a robot to move to the areas needing disinfection, and generate trajectories based on the geometries of areas of potential contamination and surrounding contexts. The adaptive motion will ensure disinfection quality and safety. The rest of the paper is organized as follows. Related studies are reviewed in Section 2 to reveal the knowledge gaps and technical barriers to be addressed in this study. Then, the framework and methods are elaborated in Section 3, followed by the experimentation and evaluation in Section 4. Section 5 concludes this study by discussing the applicability of robotic disinfection in built environments, and limitations, and future research directions. Table 1 presents a list of abbreviations used in this paper.

Table 1 List of abbreviations

Abbreviation	Definition
AP	Average Precision
CNN	Convolutional Neural Network
DSC	Dice Coefficient
DWA	Dynamic Window Approach
IK	Inverse Kinematics
IoU	Intersection over Union
LTM	Long Term Memory
mAP	Average of AP over all classes
mDSC	Average of DSC over all classes
mIoU	Average of IoU over all classes
ROS	Robot Operating System
SLAM	Simultaneous Localization and Mapping
STM	Short Term Memory

UV	Ultraviolet
WM	Working Memory

2. Literature Review

Cleaning and disinfection robots, such as the ones with ultraviolet (UV) lights, have been used in healthcare facilities [13,14] to prevent hospital-acquired infections. Perceived as a mobile UV light, the robot configuration only allows it to disinfect the room at an aggregate level. Precision disinfection is not considered feasible with this robot. During the COVID-19 pandemic, a new robot has been deployed to use vaporized hydrogen peroxide to clean and disinfect the stations and trains in Hong Kong [15]. Many studies focused on floor-cleaning robots. For example, the hTetro floor cleaning robot was developed [16–18], which can reconfigure its morphology to maximize its productivity based on the perceived environments. A novel method was proposed in [19] to express the situations of floor-cleaning robots to users. There are very few intelligent robot systems that can perform precision disinfection for various objects in different built environments. The absence of such intelligent robots stems from two knowledge gaps. First, there lacks a method to enable the robot to perceive and map the areas of potential contamination in the built environments, hindering the precision disinfection. Second, the robot needs to adapt its trajectories with respect to different areas of potential contamination for effective and safe disinfection. However, this capability has not been achieved by existing robot systems. Therefore, this study aims to address the two knowledge gaps and technical barriers. Next, a number of studies regarding fundamental robotic techniques are also reviewed.

SLAM is a fundamental technique that enables the robots to perceive the environment, localize itself, and build a map for subsequent applications. The SLAM techniques need to be compatible with the robot operating system (ROS) to allow robot navigation in built environments. GMapping [20] is a ROS default SLAM approach that uses a particle filter to create grid maps and estimate robot poses. GMapping and TinySLAM [21] can be used for localization and autonomous navigation. Using 2D light detection and ranging (LiDAR) with low computation resources, Hector SLAM [22] and ethzasl_icp_mapping [23] can provide real-time 2D occupancy mapping. Google Cartographer [24] is an efficient graph-based SLAM approach using portable laser ranger-finders. Maplab [25] and VINS-Mono [26] are graph-based SLAM approaches that fuse the information from an inertial measurement unit and a camera. The RTAB-Map [27] is a complete graph-based SLAM and has been incorporated in a ROS package for various applications. ORB-SLAM2 [28] is a popular feature-based visual SLAM approach that has been adapted to monocular, stereo, and RGB-D cameras. In contrast to feature-based algorithms, dense visual odometry DVO-SLAM [29] uses photometric and depth errors over all pixels of two consecutive RGB-D images to estimate camera motion.

Robot perception is important for deriving intelligent robot decisions and actions. In this application, robots need to perceive the areas of potential contamination on various objects for cleaning and disinfection. Detecting and segmenting the areas of potential contamination from images are related to object detection and semantic segmentation. In addition, the concept of object affordance is also relevant. Many studies have been conducted in object detection and recognition [30–34], scene classification [35,36], indoor scene understanding [37,38]. However, merely detecting the scene elements is not enough to make intelligent decisions. Particularly for this application, detecting a computer on an office desk does not mean that the computer needs to be disinfected. The existing object detection and segmentation techniques lack the capabilities to reason out which areas or under what circumstances, the object or the part of the object needs specific disinfection. Understanding how human interact with different objects will help determine the potential areas of contamination. For example, high-touch areas are considered to be contagious and should be disinfected based on WHO and CDC suggestions. Human interactions with objects have implications on how and which part of objects may be contaminated and could contaminate different parts of human body. In computer vision, studies have been conducted to predict affordance of whole objects [39,40]. In [41], a region proposal approach was integrated with convolutional neural network (CNN) feature-based recognition method to detect affordance. In [42], a method was developed to learn affordance segmentation using weakly supervised data. In [43], a number of studies of object affordance in computer vision and robotics were reviewed. Despite the advancements, there still remains a gap in linking the semantic segmentation and object affordance with the areas of potential contamination. This study aims to address this limitation.

There are many studies conducted on robot control and trajectory generation. A number of studies have been conducted in object grasping [44–46], which is closely related to our application. For instance, in [47], an "Objects Common Grasp Search" algorithm was developed to find grasp strategies that are suitable for various objects. The grasp strategies need to satisfy force-closure and quality measure criteria considering forces and torque applied to the objects. In [48], a robot grasping planning approach was proposed to extract grasp strategies from human demonstration. Human grasp intentions and constraints were also incorporated to improve the efficiency of grasp planning. In [49], a randomized physics-based motion planner was developed to allow complex multi-body dynamical robot-object interactions. The method can be adapted to cluttered and uncertain environments. In addition, in [50], an adaptive framework was proposed to allow construction robots to perceive and model the geometry of its workpieces using sensors and building information model data. In our application, after identifying the areas of potential contamination, the robot needs to move to an appropriate position and adapt its trajectory with respect to the geometry of the areas of potential contamination for complete disinfection and avoiding collision with adjacent objects. The methods developed in previous studies are not directly applicable, and thus a new method will be developed in this study to address this technical limitation.

3. Methods

Fig. 1 presents an overview of the proposed method that enables intelligent robotic disinfection in built environments. The robot is equipped with an RGB-D camera for SLAM and perception in built environments. RTAB-Map is used to provide pose estimation and generate 2D occupancy map. A deep learning method is developed to segment object affordance from the RGB-D images and map the segments to areas of potential contaminations in a 3D map. The high-touch areas can be automatically detected and segmented, which may be colonized by a variety of pathogens. The 3D semantic occupancy map and the locations of the areas of potential contamination are exploited for robot disinfection planning. The robots, with UV lights attached to end-effectors, will navigate to appropriate positions, and adapt its scanning trajectories to disinfect the objects. The framework and methods are detailed as follows.

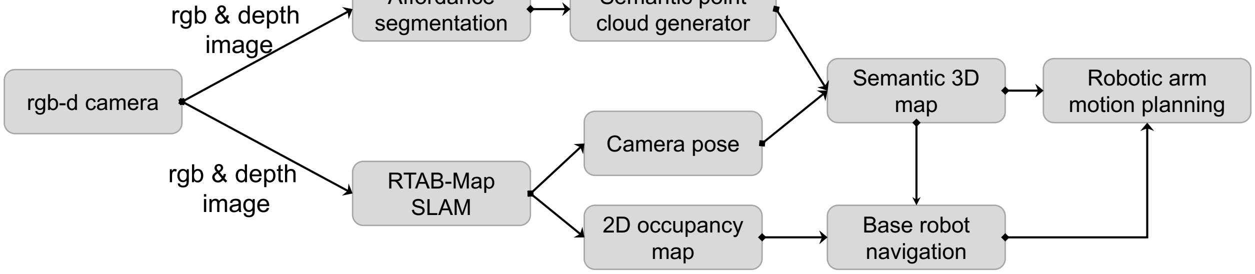


Fig. 1. Methodology overview

3.1 Localization and Mapping

The RTAB-Map SLAM method [27], a graph-based SLAM technique, is used in this study to locate the robot and produce the occupancy map for navigation, see Fig. 2. The structure of the map consists of nodes and links. Odometry nodes publish odometry information to estimate robot poses. Visual odometry obtained from ORB-SLAM2 [28] is used as odometry input in this study, because ORB-SLAM2 is fast and accurate. The short-time memory (STM) module is used to create nodes to memorize the odometry and rgb-d images, and calculate other information such as visual features and local occupancy grids. In order to limit the working memory (WM) size and reduce the time to update the graph, a weighting mechanism is used to determine which nodes in WM are transferred to long-term memory (LTM). Nodes in the LTM can be brought back to WM when a loop closure is detected. Links are used to store transformation information between two nodes. The neighbor and loop closure links are used as constraints for graph optimization and odometry drift reduction**.** The Bag of Words approach [51] is used for loop closure detection. The visual features extracted from local feature descriptor such as Oriented FAST and rotated BRIEF (ORB) [52] are quantized to a vocabulary for fast comparison. The outputs from RTAB-Map include camera pose and 2D occupancy grid, which are further used for semantic mapping and robot navigation. The rtabmap-ros package is available in ROS, which enables seamless integration with autonomous robots for this application. Because the settings of built environments do not change very frequently, maps of the built environments can be first produced and then used to locate and navigate the robots during the cleaning and disinfection process to improve the efficiency and reduce memory use.

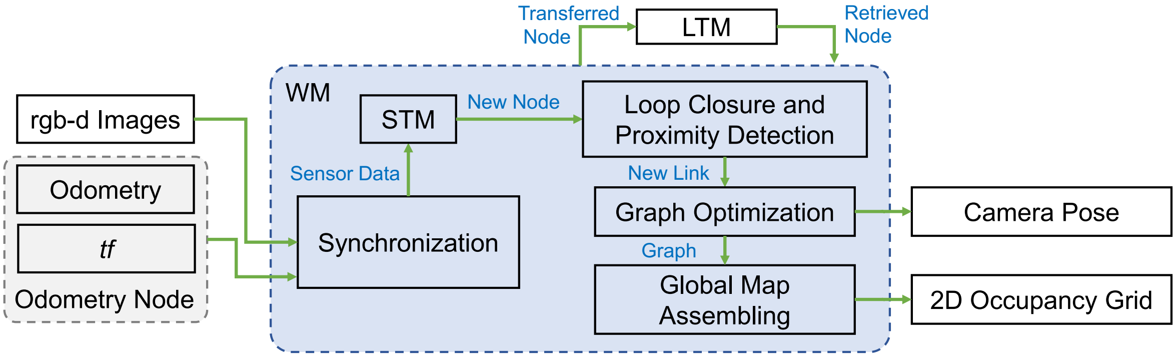
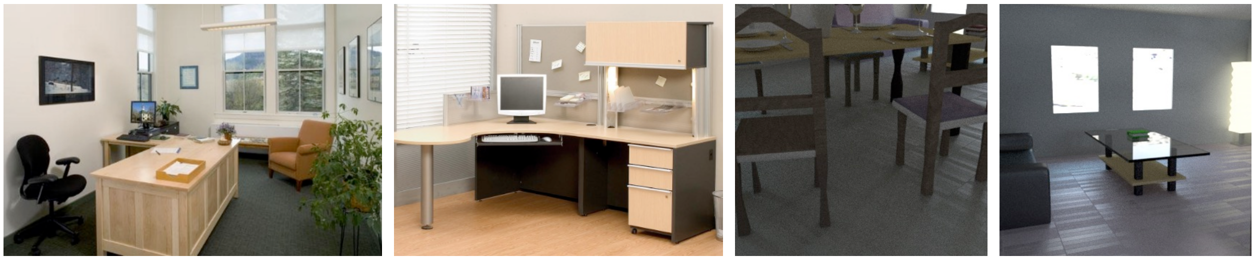


Fig. 2. RTAB-Map SLAM framework (adapted from [27])

3.2 3D Segmentation of Areas of Potential Contamination

The areas of potential contamination need to be automatically detected and mapped in 3D space to guide robotic disinfection. Particularly the object surfaces with frequent human contacts are the areas of potential contaminations requiring disinfection. Therefore, those areas need to be automatically detected and segmented from the RGB images, and thereafter projected to a 3D semantic map for robot navigations and actions. To this end, a deep learning method is developed based on the object affordance concept [40] and the approach proposed in [53] to segment the areas of potential contamination. It is necessary to label a surface that has interactions with different parts of human body. For example, the seating surface of a chair has contact with human hip, and the backrest has contact with human back, and the armrest has contact with human hand, posing different implications for distinction. Five object affordance labels are selected, including walk, grasp, pull, place, and sit, as these activities cover the most common interactions with inanimate objects in built environments. For example, the walkable areas indicate the places where the robot can move and conduct floor cleaning. The places where grasping, pulling, and placing occur represent potential high-touch areas that need to be frequently disinfected.

In this study, to train a deep learning method to segment the object surfaces as the areas of potential contamination, the ADE20K datasets [54] and simulated images [55] with appropriate labels are used. Fig. 3 presents some sample images of the training dataset. The ADE20K training dataset only labels objects and their parts. Similar to [56], a transfer table is defined to map 116 object labels to the corresponding five object affordance labels. Table 2 presents several examples. Each object or its part is associate with a five-dimensional vector, representing the five object affordance labels. The value 1 indicates that a specific object affordance is associated with an object or its part, and value 0 indicates that a specific object affordance is not associate with an object or its part. For example, “floor” is associated with “walk” affordance, “*/door/knob” is associated with “pull” affordance. If the correspondence between an object and the five affordance labels cannot be established, then the association will not be performed to ensure the reliability of predicting affordance from the trained network. Fig. 4 (a) presents an example of the label transformation. Using Table 2, annotated data from ADE20K can be transferred to affordance ground truth data. For instance, seat base is transferred to sit affordance. Fig. 4 (b) presents additional labelled simulated image for training. Affordances are directly annotated for the simulated dataset.

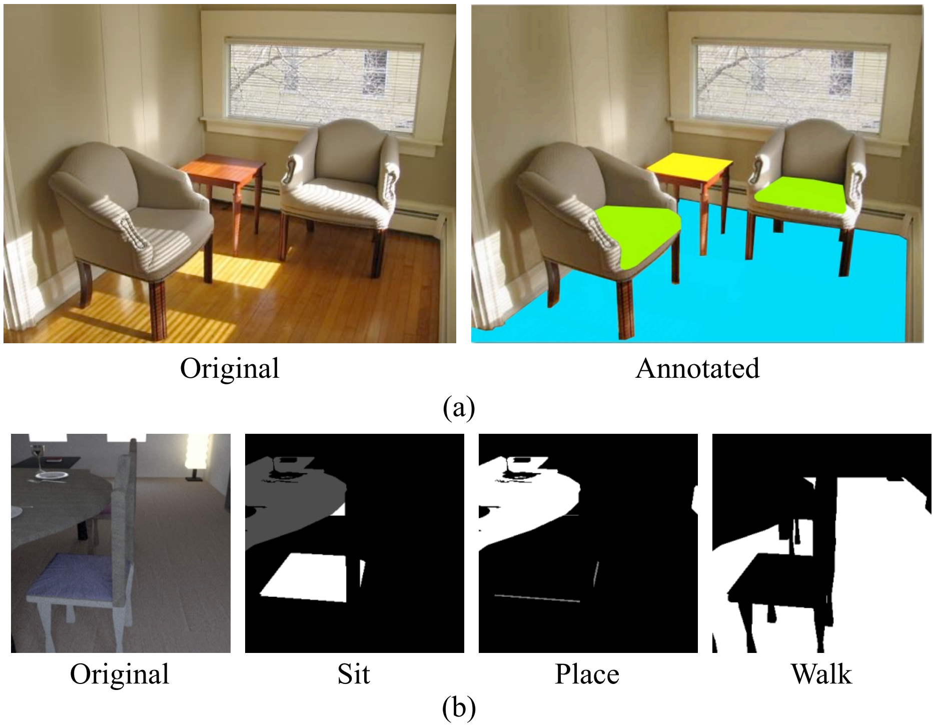


(a) (b) (c) (d)

Fig. 3. Sample images ((a)-(b) from ADE20K dataset [54], (c)-(d) from simulated images [55])

Table 2 Examples of the transfer table

Affordance	***Seat***	***Bottle***	***Floor***	****/door/knob***	***Countertop***	**…**
Walk (surfaces a human can walk)	0	0	1	0	0	…
Grasp (objects that can be grasped and moved by hands)	0	1	0	0	0	…
Pull (surfaces that can be pulled by hooking up fingers or by a pinch movement)	0	0	0	1	0	…
Place (elevated surfaces where objects can be placed on)	0	0	0	0	1	…
Sit (surfaces a human can sit)	1	0	0	0	0	…


Original

Annotated

(a)

Original

Sit

Place

Walk

(b)

Fig. 4 Example of annotated dataset. (a) using transfer table to transform original object labels in ADE20K dataset to affordance labels (green (seat base - sit); yellow (tabletop - place); blue (floor - walk)); b. annotated simulated image

The deep learning method is based on a convolutional neural network (CNN) following the U-Net architecture [57]. The encoder-decoder architecture is efficient for training and implementation when the input and output images are of similar sizes. Fig. 5 illustrates the U-Net architecture. The ResNet50 network [58] is used as the encoder. The architecture of ResNet50 includes basic block and bottleneck block, shown in Fig. 5. The basic block consists of convolution, batch normalization, ReLU, and max-pooling layers. An initial 7*7 convolution with a stride of 2 is first applied, followed by the batch normalization and ReLU activation layer. Thereafter, a max-pooling operation is conducted with a kernel size of 3 and a stride of 2. The two steps can reduce the spatial size, and thus reduce the computation cost and the number of parameters in the deep layers. In the bottleneck, the network has four connected blocks. As the network progresses from shallow to deep block, the spatial size of the input image reduces to half, and the channel number doubles.

For the decoder network, a refinement module is used to integrate low-level features and high-level semantic information from encoder network, and thus enhancing mask encoding [59]. First, the refinement module upsamples feature map size to be the same as that of skip connection from encoder network. The bilinear interpolation method [60] is used to perform upsampling. Then, skip feature map are concatenated with the local feature map. Last, convolution and batch normalization with ReLU activation are performed to compute the feature map of the next layer.

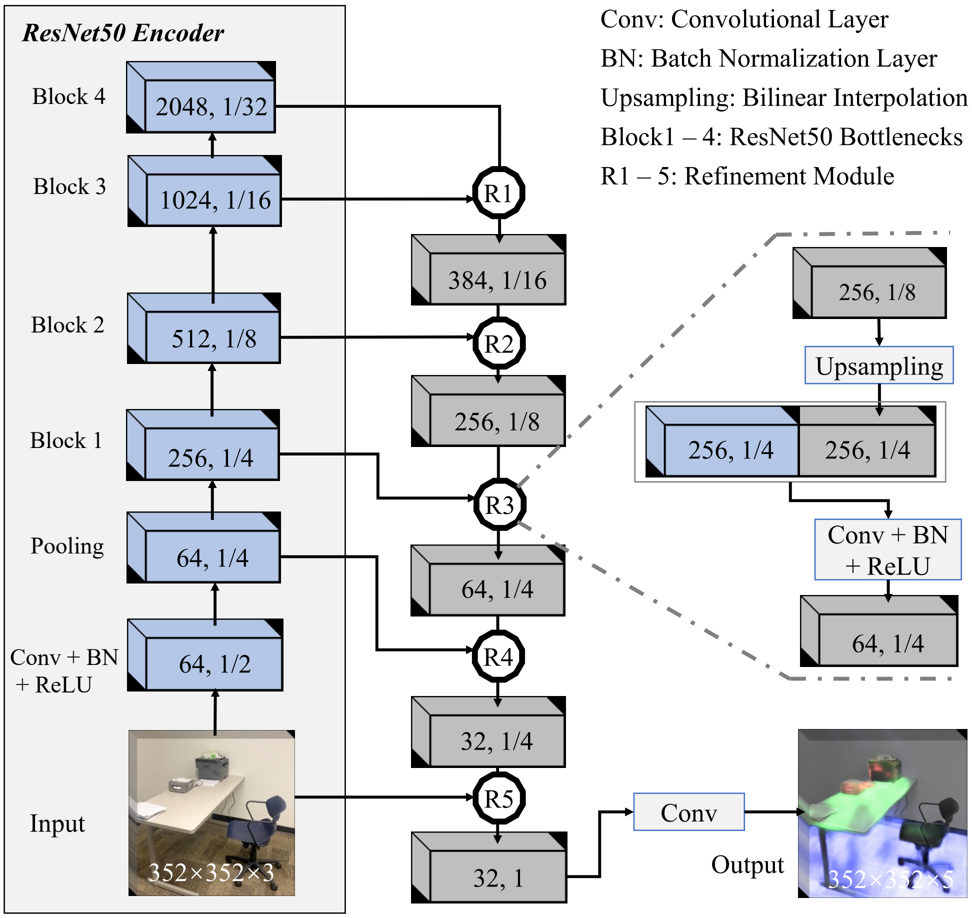
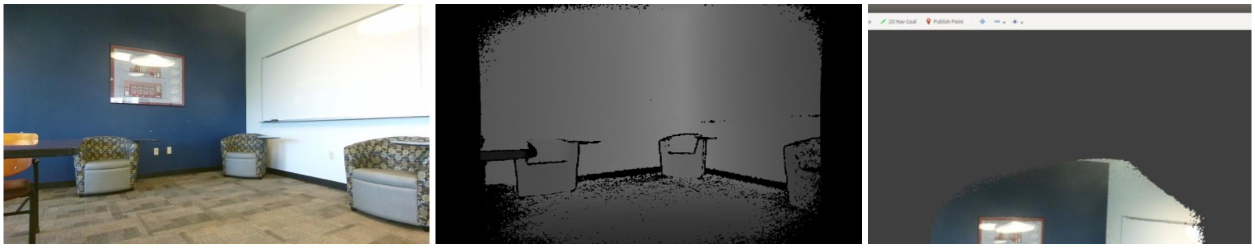


Fig. 5. Semantic segmentation with the U-Net architecture

After segmenting the object affordance from the 2D RGB images as the areas of potential contamination, it is necessary to project the 2D labels to a 3D grid map for guiding robot navigation and disinfection. As depth images are registered to the reference frame of RGB images, the first step is to use the classical pinhole camera model [61] to obtain the point cloud of the environment. Given a pixel (x, y) and its depth d, its world coordinate (X, Y, Z) is computed by Eq. (1), where f_x, f_y are the camera focal length in pixel units, (c_x, c_y) represents the principal point that is usually at the image center. Fig. 6 presents an example of the obtained point cloud. Each point stores information of world coordinates, label information, and its highest probability predicted by the network.

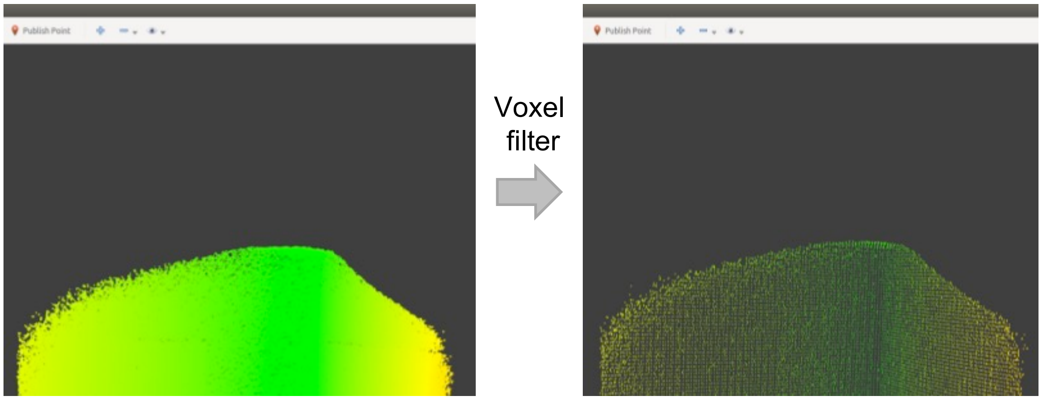
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x \\ y \\ d \end{bmatrix} \tag{1}$$



RGB image Depth image Point cloud

Fig. 6. An example of point cloud generation

Second, octomap library [62] is applied to generate a 3D occupancy grid map, using the obtained point cloud as input. A voxel filter is used to reduce the size of the point cloud to accelerate the mapping process. In each voxel space, only one point is stored as one point is adequate to update an octree node. The voxel filter resolution is set to the same resolution as that of the occupancy map. The resolution of the occupancy map is set as 4 cm, which can provide adequate details in indoor environments while maintaining processing efficiency. Fig. 7 presents an example of a 3D point cloud filtering. The image size is 960×540 and 518,400 points are generated for each frame. After using the voxel filter, the number of points reduces to 23,009 for the frame shown in Fig. 7. The number of filtered points could vary from frames due to noises in sensory data.



Original (518,400 points) Filtered (23,009 points)

Fig. 7. Using voxel filter for 3D mapping.

Since the camera is constantly moving, semantic information may continuously update. For instance, a small object may not be accurately segmented when the camera's view angle is not at a favorable position. Hence, semantic information at the pixel level from different frames are fused to deal with this situation, see Fig. 8. If two affordances are the same, the affordance will be kept, and the probability becomes the average of the two affordances. Otherwise, the affordance with higher confidence is kept and the probability is decreased to 0.9 of its original probability. This process can allow the occupancy map to update the semantic information with a new prediction of higher confidence. After the above steps, the areas of potential contamination are predicted and projected to the 3D occupancy map, which can further guide the robotic disinfection.

Pseudo-code for semantic fusion

```
function fusion ($sem_1$, $sem_2$)
   if $sem_1$.color is $sem_2$.color then
      $sem_{fusion}$.color = $sem_2$.color
      $sem_{fusion}$.probability = ($sem_1$.probability+ $sem_2$.probability)/2
   else
      if $sem_1$.probability < $sem_2$.probability then
         $sem_{fusion}$ = $sem_2$
      else
         $sem_{fusion}$ = $sem_1$
   $sem_{fusion}$.probability = 0.9 × $sem_{fusion}$.probability
return $sem_{fusion}$
```

Fig. 8. Semantic fusion of two different frames

3.3 Motion Planning for Robotic Disinfection

After mapping the areas of potential contamination, the next step is to generate robot motions to scan the areas with UV light for disinfection. The robot has a 3 degree of freedom base and 6 degree of freedom manipulator. First, the robot needs to move to the objects needing disinfection. A hierarchical planning approach is adopted, which consists of global and local path planning. Global path planning provides an optimal path from the start to the goal, and local path planning outputs a series of velocity commands for the robot. The A* algorithm [63] is used to find a globally optimal path for the robot. The heuristic function *h*(*n*) is used to guide the trajectory search toward a goal position. The A* algorithm can find the shortest path very efficiently. In this study, the Manhattan distance is used as the heuristic function that is defined in Eq. (2). This equation is used to calculate Manhattan distance from any node (n (x_n, y_n)), to the goal (g (x_g, y_g)) in the graph.

$$h\left(x_n, y_n\right) = \left|x_n - x_a\right| + \left|y_n - y_a\right| \tag{2}$$

The cost function is given in Eq. (3), where *g*(*n*) is the cost from starting point to node n, *f*(*n*) is the total cost. The objective is to minimize the total cost.

$$f\left(n\right) = g\left(n\right) + h\left(n\right) \tag{3}$$

Given a global path to follow, the local planner produces velocity commands for the robot. The Dynamic Window Approach (DWA) algorithm [64] serves as the local planner. The algorithm samples velocities in the robot’s control space discretely within a given time window. The samples intersect with obstacles will be recognized and eliminated. An optimal pair of (v, w) for the robot is determined by maximizing the objective function defined in Eq. (4), which is dependent on (1) proximity to the global path, (2) proximity to the goal, and (3) proximity to obstacles.

$$\mathrm{cost} = \alpha f_a\left(v, w\right) + \beta f_d\left(v, w\right) + \gamma f_c\left(v, w\right) \tag{4}$$

where $f_a(v, w)$ represents the distance between global path and the endpoint of the trajectory, $f_d(v, w)$ is the distance to the goal from the endpoint of the trajectory, $f_c(v, w)$ is the grid cell costs along the trajectory, α is the weight for how much the controller should stay close to the global path, β is the weight for how much the robot should attempt to reach the goal, and γ is the weight for how much the robot should attempt to avoid obstacles.

After moving to the vicinity of the objects, the scanning-based disinfection can be conducted. Because the UV light has been demonstrated to effectively reduce the bioburden of epidemiologically relevant pathogens, in this study, an UV disinfection device is integrated into the mobile manipulator as an end-effector. The continuous low doses of UV light can kill pathogens on various surfaces of the objects without harming human tissues. Human interventions can also be incorporated to guide the robot disinfection. For example, humans can issue commands to a robot to disinfect a particular area or object or schedule a fleet of robots to disinfect large facilities such as hospitals, schools, and airports. In addition, the human user could further adjust the autonomy or can force any decision of the robot regardless of what and how the onboard operation progresses. For instance, the user could stop the robot immediately in any unexpected safety-critical situations.

A list of waypoints is used as inputs to generate a trajectory for robotic disinfection. Waypoints are defined as individual points along the path followed by the end-effector. There are four steps to generate a trajectory from waypoints (see Fig. 9). First, sampling points are linearly interpolated between the first waypoint and the second waypoint. 1cm resolution is used to interpolate the points. The number of points between the two waypoints is the distance divided by the resolution. Second, TRAC-IK algorithm [65] is used as inverse kinematics (IK) solver to calculate joint solutions. This algorithm is a numerical IK solver that combines both pseudoinverse Jacobian and Sequential Quadratic Programming-based nonlinear optimization solvers. The solver will stop and return the best solution once either of the two solvers finishes with a solution. Third, taking the current position as the starting position and next waypoint as the goal, the first and second steps are repeated until all the waypoints are traversed. Finally, all the joint solutions are connected to generate a trajectory including velocity, acceleration, and duration.

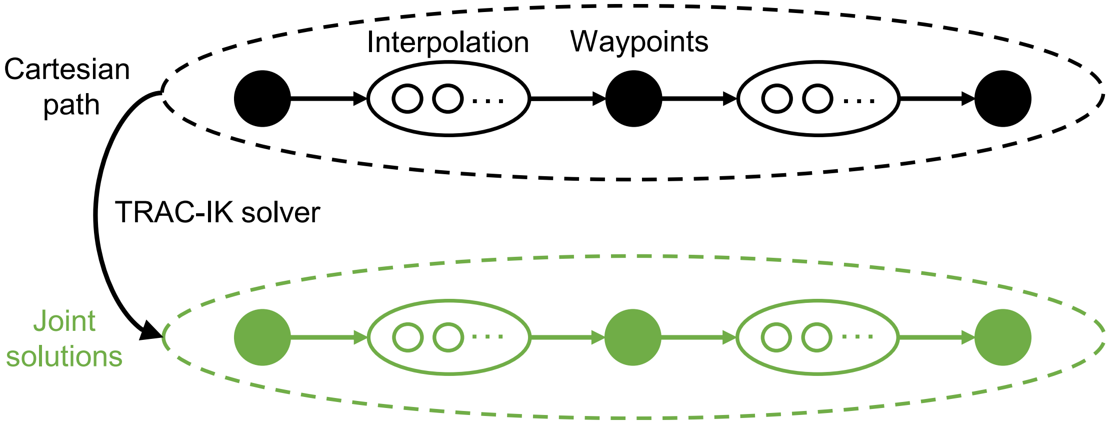


Fig. 9. Flowchart to generate a trajectory from waypoints

To make the disinfection process more efficient, the robotic arm is preprogrammed to adapt to objects with various geometries. As shown in Fig. 10, a plane scanning zone and a cylinder scanning zone are developed to generate scanning trajectories for different objects. Plane-scanning is suitable for objects that have plane surfaces such as office desk, keyboard, push bar of drinking fountain. Cylinder-scanning is suitable for objects that have large curvature such as a kettle. The two types of scanning zones can account for a variety of objects that are commonly seen in the built environments. The size of the scanning zone adapts to the segmented areas of potential contamination. Based on the 3D segments, a distance away from the surface will be maintained to generate the scanning zones and trajectories.

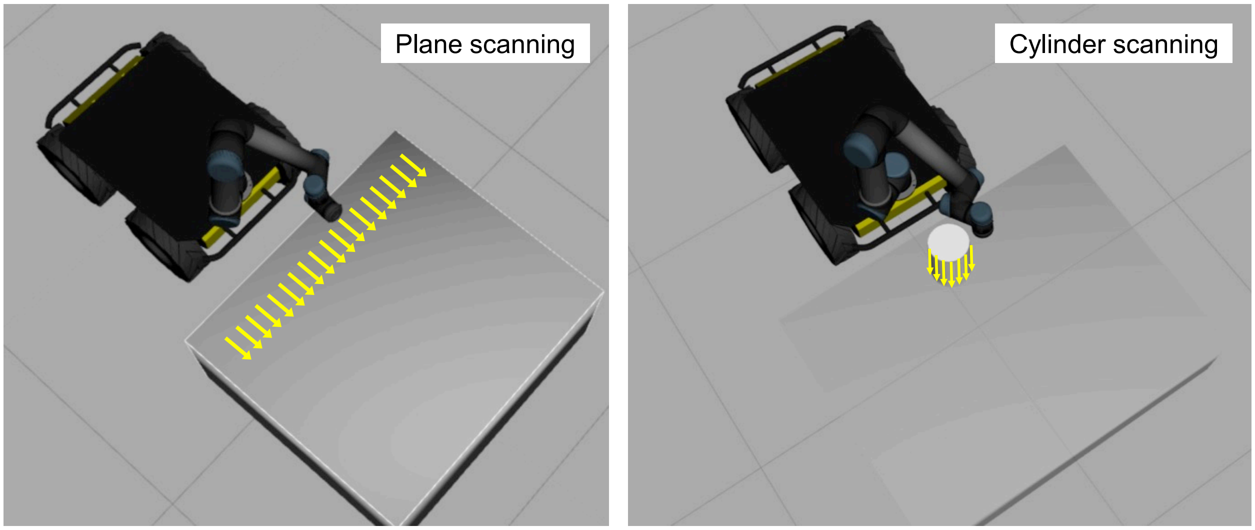


Fig. 10. Plane-scanning and cylinder-scanning zones. Yellow arrow represents waypoints and their corresponding poses along the trajectory.

4. Experiments and Results

Both simulations and physical experiments were conducted to validate the proposed methods. Segmentation and 3D mapping of potential areas of contamination were validated in indoor environments, including a dining room, a conference room, and a restroom in a university campus building. Motion planning for robotic disinfection was validated using a robot simulation platform and an AUBO-i5 robotic arm.

4.1 Segmentation and Mapping Evaluation

4.1.1 Evaluation datasets

The ADE20K dataset [54] and simulated dataset [55] were used to evaluate the performance of the network. The ADE20K dataset contains a total of 22,210 images with 20,210 training images and 2,000 validation images. The simulated dataset contains 2,530 synthetic images with 2,280 training images and 250 testing images. Hence, a total number of 22,490 images including both real and simulated images are used for training. The real and simulated images are first merged and then randomly mixed. Each mini batch for training can have samples from both datasets. In addition, data augmentation technique is used to increase its volume and variability of the training dataset. Training samples were augmented by cropping multiple image patches based on image quality and varying color and contrast of images to improve the capability and generalization of the trained model. Online augmentation method is used due to two reasons. First, as the model observes more samples in the training process, the model trained with online data augmentation can generalize better than the model trained with offline data augmentation [66]. Second, online augmentation does not need to store a large amount of augmented data in local disk. The validation set consists of 1,000 real images and 120 simulated images that are randomly split from the training dataset. The performance of the network was evaluated on 2,000 real images and 250 simulated images.

4.1.2 Metrics

To evaluate the performance of affordance segmentation, the metrics including the intersection over union (IoU), dice coefficient (DSC), and average precision (AP) were used to evaluate the network performance. These metrics have been widely used in evaluating the performance of semantic segmentation [67–69]. The IoU is the area of overlap between the predicted segmentation and the ground truth divided by the area of union between the predicted segmentation and the ground truth. The maximum IoU value is 1 representing perfect segmentation. The IoU metric is defined in Eq. (5), where $\mathbf{Y}_{ai}$ is the ground truth for affordance *a* at pixel $i \in I$, $\hat{\mathbf{Y}}_{ai}$ represents predicted affordance. The binarized prediction of the network is used to compute the IoU. The threshold values 0.2 and 0.5 are used in the experiment.

$$IoU_a = \frac{\sum_{i \in I}\left(\mathbf{Y}_{ai} = 1 \cap \hat{\mathbf{Y}}_{ai} = 1\right)}{\sum_{i \in I}\left(\mathbf{Y}_{ai} = 1 \cup \hat{\mathbf{Y}}_{ai} = 1\right)}$$
$$\hat{\mathbf{Y}}_{ai} = \begin{cases} 1 & \text{if p} > \text{threshold} \\ 0 & \text{else} \end{cases} \tag{5}$$

The DSC is similar to the IoU, which is another measure of overlap between prediction and ground truth. This measure ranges from 0 to 1, where a DSC of 1 denotes perfect and complete overlap. The DSC is defined in Eq. (6).

$$DSC_a = \frac{\sum_{i \in I} 2 \times \left(\mathbf{Y}_{ai} = 1 \cap \hat{\mathbf{Y}}_{ai} = 1\right)}{\sum_{i \in I}\left(\mathbf{Y}_{ai} = 1\right) + \left(\hat{\mathbf{Y}}_{ai} = 1\right)} \tag{6}$$

The AP metric summarizes a precision-recall curve as the weighted mean of precisions achieved at each threshold. AP is not dependent on a single threshold value since it averages over multiple levels. The AP is defined in Eq. (7), where P_n and R_n are the precision and recall at the nth threshold, and P_n is defined as precision at cut-off n in the list.

$$\text{AP} = \sum_n \left(R_n - R_{n-1}\right) P_n \tag{7}$$

4.1.3 Implementation

The models were trained on a workstation running Ubuntu 16.04 with Dual Intel Xeon Processor E5-2620 v4, 64GB RAM, and Dual NVIDIA Quadro P5000 with a pytorch backend [70]. The network was trained using RMSProp optimizer [71] with a learning rate of 0.0001 and batch size of 16. The ResNet50 encoder was initialized with weights pretrained on ImageNet [72]. The pretrained weight was further trained on the dataset without freezing any weights. The early stopping technique [73] was adopted to prevent overfitting. Specifically, the network is trained on the training set, and if the loss on the evaluation set does not decrease for 20 epochs, the training process will stop and the best model observed on the evaluation set will be saved. The performance of the network is evaluated on the testing dataset.

4.1.4 Results on Semantic Mapping

Fig. 11 presents the results of the affordance segmentation on the training, validation, and testing sets. The training set achieved the highest mAP, mIoU, and mDSC since the model is optimized using this set. The testing set #1 achieved the second-highest scores, and the difference of all the three metrics between the training set and testing set #1 is less than 0.1. However, testing set #2 achieved the smallest scores among the four datasets. This is because the training set contains both real and simulated images, while testing set #2 only contains real images. Synthetic images cannot reproduce richness and noise in the real ones, which may lead to the network trained on synthetic images performs undesirable on real images. Therefore, the network trained on both simulated and real images have a better performance on a testing set combines both samples.

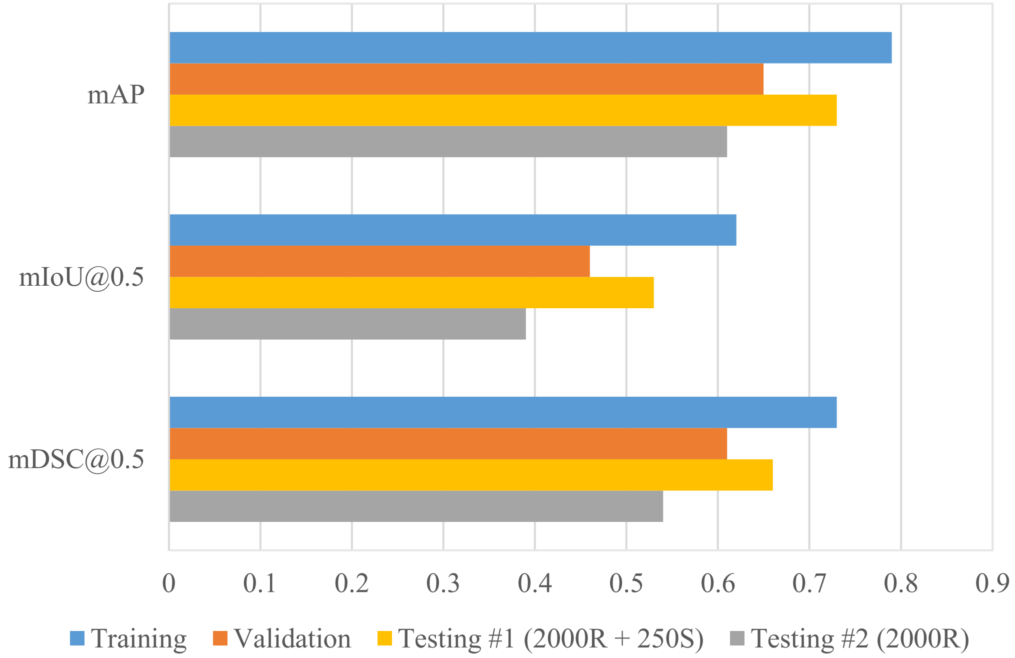


Fig. 11 The performance of the network on the training set, validation set, and two testing sets. (Testing #1 consists of 2000 real images and 250 simulated images and testing #2 only contains 2000 real images. mAP, mIoU, and mDSC are the average of AP, IoU, and DSC over all classes.)

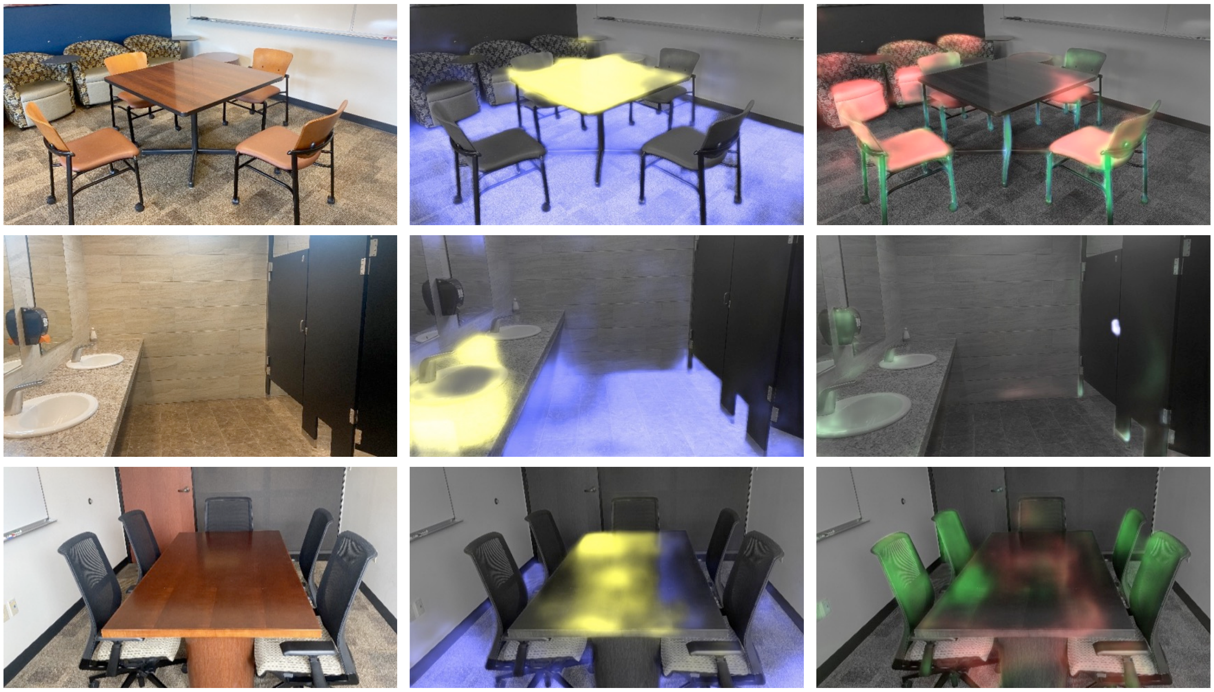
Table 3 presents the network performance for individual affordance on testing set #2. The results show a strong variation in performance for different affordances. For instance, affordance walk achieves the highest IoU and AP scores, which is attributed to a relatively large sample size compared to other affordances such as grasp and pull. In addition, walking surface often covers large areas in the scene. Pull has the lowest prediction accuracy among the five affordances. The pull affordance represents objects that can be pulled such as doorknob and cabinet handle. These objects are relatively small and have a small sample size in the dataset. The walk, grasp, place, and sit affordances achieved DSC and AP scores higher than 0.5, indicating the usability of the proposed method in built environments.

Table 3 Performance for individual affordance on testing set #2

Affordance	DSC@0.2	DSC@0.5	IoU@0.2	IoU@0.5	AP
Walk	0.84	0.87	0.72	0.77	0.94
Grasp	0.51	0.51	0.31	0.30	0.50
Pull	0.36	0.17	0.23	0.10	0.32
Place	0.57	0.56	0.38	0.37	0.62
Sit	0.64	0.60	0.47	0.42	0.67

The performance of the proposed method is also compared with two studies that achieved the best performance of affordance segmentation using RGB images captured in built environments. In [74], a Multi-scale CNN was developed to segment affordance in RGB images. Roy and Todorovic [74] achieved IoU scores 0.67 and 0.34 on walk and sit affordances, respectively. The proposed network achieved 0.76 and 0.42 for IoU scores for walk and sit affordances at the threshold 0.5. Lüddecke et al. [56] reported the best AP scores for sit, grasp, pull, place, and walk affordance are 0.54, 0.30, 0.02, 0.45, and 0.96. The proposed network achieved AP scores 0.67, 0.50, 0.32, 0.62, and 0.94 for sit, grasp, pull, place, and walk affordances on the same test set. Hence, it can be concluded that the proposed semantic segmentation method is at least comparable with the state-of-the-art.

A Kinect sensor is used to perform RTAB-Map SLAM and generate the semantic 3D occupancy map using the network. The frame size provided by the Kinect is 960*540 pixels. Fig. 12 shows the predicted affordances in images captured in the building. Walk, Grasp, Pull, Place, and Sit affordances are color-coded, and the color intensity represents their corresponding probabilities.



Original Place (yellow); Walk (blue) Sit (red); Grasp (green); Pull (purple)

Fig. 12. Results of affordance segmentation

Fig. 13 presents the results of 3D semantic occupancy mapping. Images were obtained to perform RTAB-Map SLAM to obtain camera poses. Thereafter, semantic reconstruction was conducted using recorded video and camera trajectory. At a resolution of 4cm, the indoor scene can be properly reconstructed. The results indicate that the proposed method can successfully segment affordances. The walk, place, sit, and grasp affordances are reasonably segmented. In Fig. 13 (a), small tablet arm of sofa on the left side is correctly segmented as place affordance. However, small objects like doorknob are not correctly recognized in the semantic map. In addition, a part of the table surface is wrongly identified as walk affordance. This is possibly due to the small size of the training data. The occupancy map can be continuously updated during the robot disinfection action to address the incorrect segmentation.

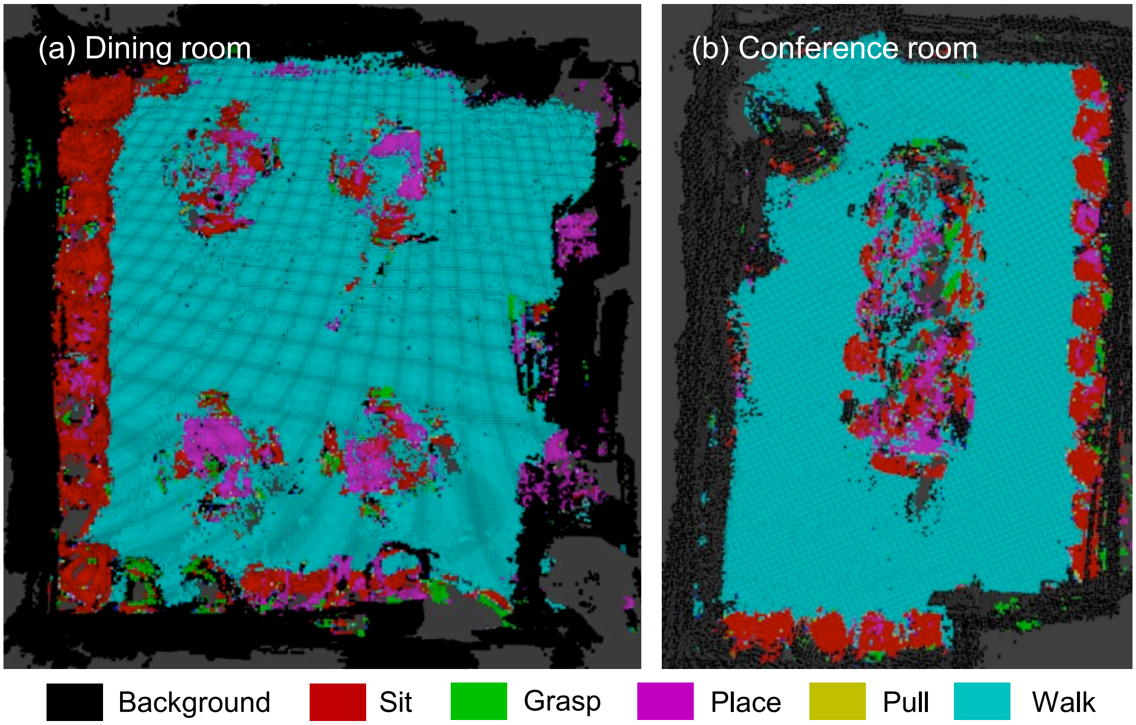


Fig. 13. Results of 3D semantic reconstruction

4.2 Processing Time

The processing time for each step was assessed in this study. Table 4 presents the average time spent on each processing stage. The occupancy map resolution is set as 4 cm. As shown in the table, the processing frequency of the entire system is about 3.2 Hz and 4.0 Hz for image size 960×540 and 512×424, respectively. The octomap update is the most time-consuming step in the system, since it requires raycasting to update occupancy map. The raycasting is used to clear all the voxels along the line between the origin and end point. The SLAM method achieves a high frame rate to track the camera in real time. Semantic segmentation and semantic point cloud generation are also run at a very high frame rate. Our system runs at 3.2 Hz for a high-resolution image streaming, which can be adapted to most indoor online applications.

Table 4 Average processing time for each step (Process with * and process with ** executed at the same time)

Step	**Consumed time**	
	960×540	512×424
SLAM *	50.2 ms	35.4 ms
Semantic segmentation **	25.1 ms	20.4 ms
Semantic point cloud generation **	28.3 ms	13.7 ms
Octomap update (resolution 4 cm) **	254.6 ms	215.7 ms
Total	308.0 ms	249.8 ms

Moreover, the occupancy map resolution significantly impacts the processing time. Fig. 14 presents the relationship between processing time and occupancy map resolution for image size 960×540 and 512×424. The result indicates that the processing time significantly decreases with decreasing map resolution. In addition, processing time increases as the image size increases under different occupancy map resolutions. When the resolution is 0.06 cm, the processing time can reduce to 206.5 ms and 134.2 ms for image size 960*540 and 512*524, respectively. However, a lower resolution may not capture detailed information, especially for small objects.

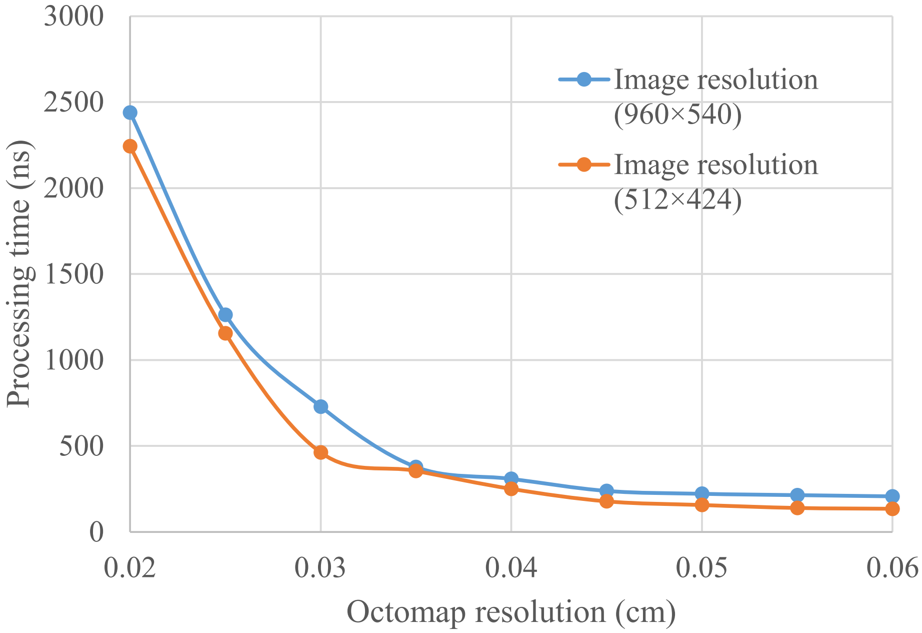


Fig. 14. Influence of image size and occupancy map resolution on processing time

4.3 Implementation of Robotic Disinfection

Physical and simulated experiments were performed to test the scanning-based disinfection. A Husky UR5 mobile manipulation robot is simulated in the gazebo. Husky is a mobile robot that can navigate in the ground plane. UR5 robotic arm can generate a designed path to scan the areas of contamination using UV light. The distribution of ROS is Kinetic, and the version of Gazebo is 7.16.0. The 3D occupancy map collected in the built environment was loaded into the simulation platform to test robot navigation. Table 5 shows the performance of path planning of the robot. The average computing time for 20 simulation experiments is low, and generated paths can successfully avoid collision with obstacles. The results demonstrate the efficiency and effectiveness of the robot path planning method. Fig. 15 presents two representative examples of the path planning of the robot. The robot moves to the proximity of objects needing disinfection.

Table 5 Performance of base robot path planning

Simulation case	Computing time (second)			Number of cases without collision
	Average	Minimum	Maximum	
20	0.231	0.2	0.375	20

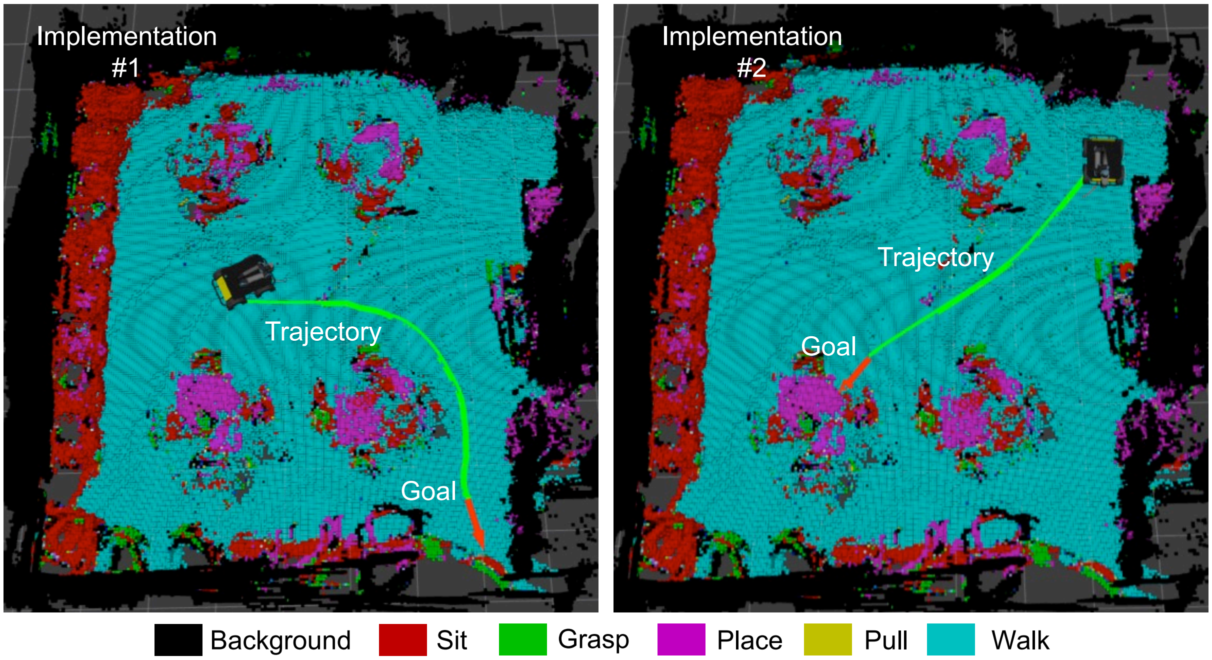
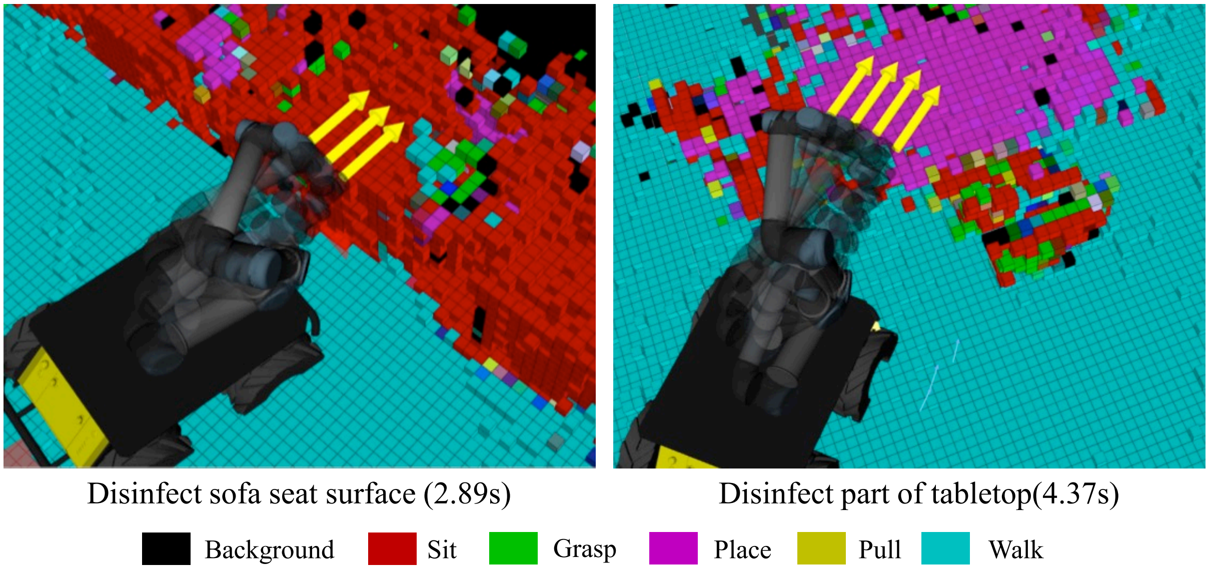


Fig. 15. Implementation of robot navigation. Red arrow is the pose of a goal point.

After navigating to the areas of potential contamination, trajectory will be generated to perform disinfection. Fig. 16 presents two examples of robot disinfection, where table surface and sofa seat surface were disinfected with plane scanning. The generated robotic arm trajectory can avoid collision with objects using the semantic occupancy map. The execution times are 2.89s and 4.37s for the sofa seat and part of tabletop, respectively. Additional scanning can be performed to ensure adequate exposure of UV light to eradicate pathogens. This study did not investigate the relationship between adequate UV light exposure and the effects of eradicating pathogens. The results demonstrate the feasibility of using robots to conduct disinfection.


Disinfect sofa seat surface (2.89s)

Disinfect part of tabletop(4.37s)

Fig. 16. Results of robotic arm motion planning. Yellow arrow represents waypoints and their corresponding pose along the trajectory.

In addition, a physical experiment was conducted using an AUBO-i5 robotic arm with a UV light attached as its end effector. The UV light will automatically turn on when it is close to the object surface requiring disinfection and shut off when moving away. As shown in Fig. 17, objects are correctly segmented to their corresponding affordances. Cabinet handle, the outside surface of tea kettle, and table surface are segmented as pull, grasp, and place, respectively. For cabinet handle and table surface, plane scanning is used to conduct disinfection. The outside surface of tea kettle is disinfected with cylinder scanning. The segmentation and mapping of areas of potential contamination can provide guidance for robot disinfection in the physical experiment, demonstrating the efficacy of the proposed method.

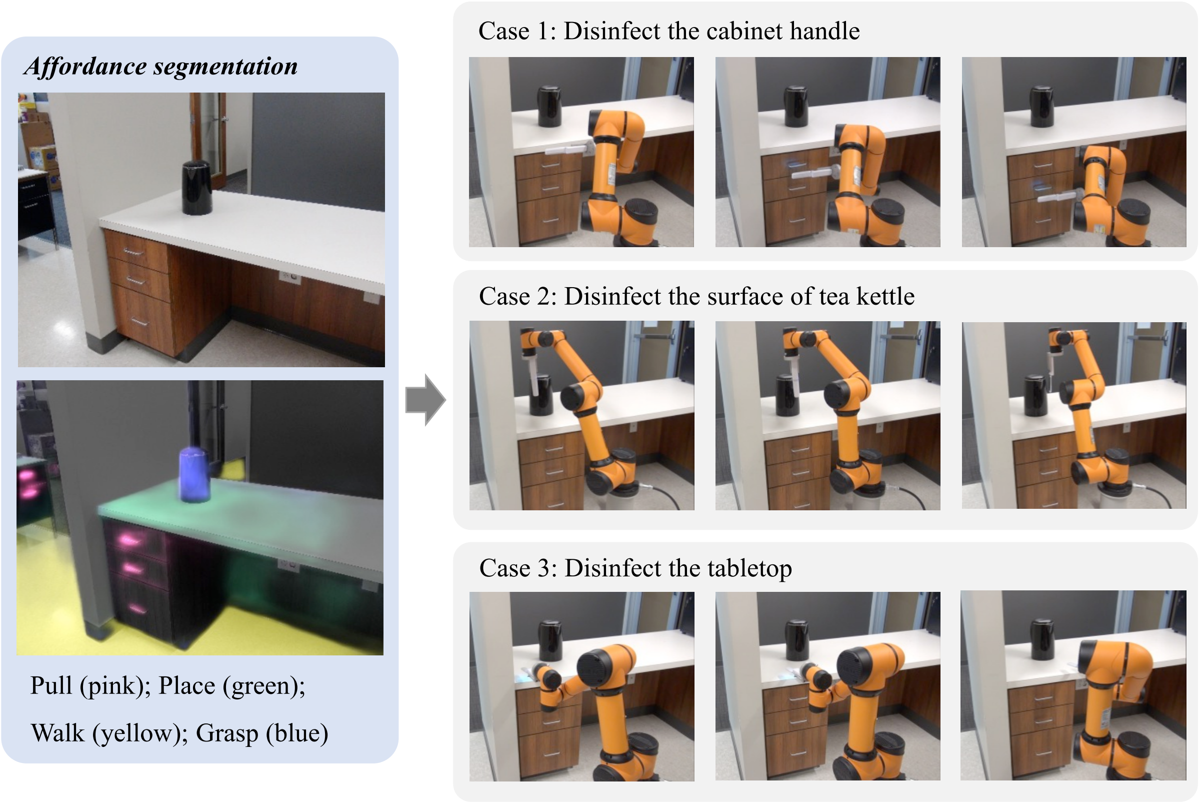

Affordance segmentation
Pull (pink); Place (green);
Walk (yellow); Grasp (blue)
Case 1: Disinfect the cabinet handle
Case 2: Disinfect the surface of tea kettle
Case 3: Disinfect the tabletop

Fig. 17. Results of robotic disinfection based on affordance map.

5. Conclusion

The mass-gathering built environments such as hospitals, schools, airports, and transit systems harbor a variety of pathogens that may cause diseases. Frequent disinfection is critical for preventing the outbreak of infectious diseases in built environments. However, manual disinfection process is labor-intensive, time-consuming, and health-undermining, highlighting the values of automated and robotic disinfection. To reduce the public health risks and alleviate the extensive labor efforts, this study presents a framework and algorithmic techniques to enable a robot to automatically use UV lights to disinfect the areas of potential contamination. Using SLAM technique, the robot is able to create occupancy map and estimate its pose. The areas of potential contamination are detected and segmented based on object affordance. The robot can then navigate to the areas needing disinfection and the robotic arm can generate an efficient and collision-free path to clean the area. The developed robots present a promising and safe solution to reduce the transmission and spread of microbial pathogens such as influenza and coronavirus.

Using the proposed method has at least two benefits in cleaning and disinfection practice. First, the robot platform can reduce the infection risk of cleaning workers by keeping them away from contaminated areas. Second, affordance information can guide the robot to focus on hot spots and thoroughly disinfect potentially contaminated areas. Thus, the developed methods will help reduce the seasonal epidemics, as well as pandemics of new virulent pathogens. The developed method achieved high accuracy in segmenting floors and high-touch surfaces as areas of potential contamination. Empirical evidence [75] suggests that floors can harbor a variety of pathogens including the SARS-CoV-2 for a long period. Human movements can lead to resuspensions of pathogens deposited on the floor, further contaminating other surfaces. Hence, to avoid reciprocal contamination and enhance disinfection efficiency, both floors and high-touch areas need to be disinfected. The performance in 3D segmentation and mapping achieved by this method demonstrated its applicability. With the developed perception capability, functionalities such as vacuum, spray, and mopping can be incorporated into the robot system for floor cleaning. In addition, comparing with overhead UV lights, the robotic disinfection proposed in this study can reach the places where conventional overhead lights cannot reach. Conventional germicidal UV lights can lead to skin cancer and cataracts, which pose a health threat to humans. The precision UV light scanning achieved by the adaptive robot motion can ensure the continuous and complete disinfection of high-touch areas, which is safer, and more efficient and effective than the overhead UV lights.

There are some limitations that need to be addressed in future studies. First, the network reported a low accuracy in segmenting the areas of potential contamination on small objects such as doorknobs and cabinet handles in unfavorable circumstances. The low accuracy stems from the scarcity of the available data. Future studies are needed to augment the dataset and develop more robust deep learning algorithm for 3D segmentation. Second, the developed robot system only employs UV lights to disinfect high-touch areas. However, other operation modes such as vacuum, spray, and swipe are needed to clean and disinfect a variety of surfaces, including floors. Advanced control techniques should be developed and parameters such as scanning time, spray dose, and swipe force should be calibrated for these modes to achieve optimal disinfection performance. Third, this study considers a single robot for disinfection at a room scale. A fleet of robots might be needed to disinfect a large facility such as a hospital or an airport. The planning and scheduling of multiple robots for coordinated disinfection will be an interesting and useful future study. Fourth, human presence and social context have not been considered in this research. The robot should never point the UV light to humans and should not spray disinfectant in vicinity of humans. Moreover, the robot should not interrupt ongoing human activities for disinfection. Future studies are needed to enable the robots to learn the rules and understand the contexts, which are important for the actual deployment in human-centric built environments.

Acknowledgement

This research was funded by the U.S. National Science Foundation (NSF) via Grant 2038967, 1952140, 2026719. The authors gratefully acknowledge NSF's support.

Reference

[1] M.C. Fitzpatrick, C.T. Bauch, J.P. Townsend, A.P. Galvani, Modelling microbial infection to address global health challenges, Nat. Microbiol. 4 (2019) 1612–1619. https://doi.org/10.1038/s41564-019-0565-8.

[2] E. Dong, H. Du, L. Gardner, An interactive web-based dashboard to track COVID-19 in real time, Lancet Infect. Dis. 20 (2020) 533–534. https://doi.org/10.1016/S1473-3099(20)30120-1.

[3] W.C.W.S. Putri, D.J. Muscatello, M.S. Stockwell, A.T. Newall, Economic burden of

seasonal influenza in the United States, Vaccine. 36 (2018) 3960–3966. https://doi.org/10.1016/j.vaccine.2018.05.057.

[4] CDC, 2019-2020 U.S. Flu Season: Preliminary Burden Estimates, (2020). https://www.cdc.gov/flu/about/burden/preliminary-in-season-estimates.htm (accessed August 17, 2020).

[5] E.J. Septimus, J. Moody, Prevention of device-related healthcare-associated infections, F1000Research. 5 (2016).

[6] C. Siemaszko, Coronavirus forces New York City subways, trains to clean up their act, NBC News. (2020). https://www.nbcnews.com/news/us-news/coronavirus-forces-new-york-city-subways-trains-clean-their-act-n1148231.

[7] H. Leone, Every public and private school in Illinois is closed because of the coronavirus. Here's what you need to know., Chicago Trib. (2020). https://www.chicagotribune.com/coronavirus/ct-cb-coronavirus-illinois-schools-closed-cps-parents-need-to-know-20200317-zrcim5cpcfcgnerkboyx7esg5y-story.html.

[8] K.R. Bright, S.A. Boone, C.P. Gerba, Occurrence of bacteria and viruses on elementary classroom surfaces and the potential role of classroom hygiene in the spread of infectious diseases, J. Sch. Nurs. 26 (2010) 33–41.

[9] G. Kampf, D. Todt, S. Pfaender, E. Steinmann, Persistence of coronaviruses on inanimate surfaces and its inactivation with biocidal agents, J. Hosp. Infect. (2020).

[10] A.S. for Microbiology, How quickly viruses can contaminate buildings -- from just a single doorknob, ScienceDaily. (2014). http://www.sciencedaily.com/releases/2014/09/140908093640.htm.

[11] O. Dumas, R. Varraso, K.M. Boggs, C. Quinot, J.-P. Zock, P.K. Henneberger, F.E. Speizer, N. Le Moual, C.A. Camargo, Association of Occupational Exposure to Disinfectants With Incidence of Chronic Obstructive Pulmonary Disease Among US Female Nurses, JAMA Netw. Open. 2 (2019) e1913563–e1913563.

[12] T. Weinmann, J. Gerlich, S. Heinrich, D. Nowak, E. Von Mutius, C. Vogelberg, J. Genuneit, S. Lanzinger, S. Al-Khadra, T. Lohse, Association of household cleaning agents and disinfectants with asthma in young German adults, Occup. Environ. Med. 74 (2017) 684–690.

[13] A. Begić, Application of Service Robots for Disinfection in Medical Institutions, in: Int. Symp. Innov. Interdiscip. Appl. Adv. Technol., Springer, 2017: pp. 1056–1065.

[14] D. Gibson, S. Kendrick, E. Simpson, D. Costello, R. Davis, A. Szetela, M. McCreary, J. Schriber, Implementation of Xenon Ultraviolet-C Disinfection Robot to Reduce Hospital Acquired Infections in Hematopoietic Stem Cell Transplant Population, Biol. Blood Marrow Transplant. 23 (2017) S472.

[15] M. Hui, Hong Kong's subway is sending robots to disinfect trains of coronavirus, Quartz. (2020). https://qz.com/1816762/coronavirus-hong-kongs-mtr-subway-uses-robot-to-disinfect-trains/.

[16] V. Prabakaran, M.R. Elara, T. Pathmakumar, S. Nansai, hTetro: A tetris inspired shape shifting floor cleaning robot, in: 2017 IEEE Int. Conf. Robot. Autom., IEEE, 2017: pp. 6105–6112.

[17] P. Veerajagadheswar, M.R. Elara, T. Pathmakumar, V. Ayyalusami, A tiling-theoretic approach to efficient area coverage in a tetris-inspired floor cleaning robot, IEEE Access. 6 (2018) 35260–35271.

[18] V. Prabakaran, M.R. Elara, T. Pathmakumar, S. Nansai, Floor cleaning robot with reconfigurable mechanism, Autom. Constr. 91 (2018) 155–165.

[19] M.A.V.J. Muthugala, A. Vengadesh, X. Wu, M.R. Elara, M. Iwase, L. Sun, J. Hao, Expressing attention requirement of a floor cleaning robot through interactive lights, Autom. Constr. 110 (2020) 103015.

[20] G. Grisetti, C. Stachniss, W. Burgard, Improved techniques for grid mapping with rao-

blackwellized particle filters, IEEE Trans. Robot. 23 (2007) 34–46.

[21] B. Steux, O.T. El Hamzaoui, A SLAM algorithm in less than 200 lines C-language program, Proc. Control Autom. Robot. Vis. (ICARCV), Singapore. (2010) 7–10.

[22] S. Kohlbrecher, J. Meyer, T. Graber, K. Petersen, U. Klingauf, O. Von Stryk, LNAI 8371 - Hector Open Source Modules for Autonomous Mapping and Navigation with Rescue Robots, 2013. http://www.gkmm.tu-darmstadt.de/rescue (accessed December 31, 2019).

[23] F. Pomerleau, F. Colas, R. Siegwart, S. Magnenat, Comparing ICP variants on real-world data sets, Auton. Robots. 34 (2013) 133–148.

[24] W. Hess, D. Kohler, H. Rapp, D. Andor, Real-time loop closure in 2D LIDAR SLAM, in: 2016 IEEE Int. Conf. Robot. Autom., IEEE, 2016: pp. 1271–1278.

[25] T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, R. Siegwart, maplab: An open framework for research in visual-inertial mapping and localization, IEEE Robot. Autom. Lett. 3 (2018) 1418–1425.

[26] Y. Lin, F. Gao, T. Qin, W. Gao, T. Liu, W. Wu, Z. Yang, S. Shen, Autonomous aerial navigation using monocular visual-inertial fusion, J. F. Robot. 35 (2018) 23–51.

[27] M. Labbé, F. Michaud, RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation, J. F. Robot. 36 (2019) 416–446.

[28] R. Mur-Artal, J.D. Tardós, Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras, IEEE Trans. Robot. 33 (2017) 1255–1262.

[29] C. Kerl, J. Sturm, D. Cremers, Dense visual SLAM for RGB-D cameras, in: 2013 IEEE/RSJ Int. Conf. Intell. Robot. Syst., IEEE, 2013: pp. 2100–2106.

[30] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (2009) 1627–1645.

[31] M. Liang, X. Hu, Recurrent convolutional neural network for object recognition, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015: pp. 3367–3375.

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, Imagenet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (2015) 211–252.

[33] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: Single shot multibox detector, in: Eur. Conf. Comput. Vis., Springer, 2016: pp. 21–37.

[34] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016: pp. 779–788.

[35] L. Zhang, X. Zhen, L. Shao, Learning object-to-class kernels for scene classification, IEEE Trans. Image Process. 23 (2014) 3241–3253.

[36] M. Dixit, S. Chen, D. Gao, N. Rasiwasia, N. Vasconcelos, Scene classification with semantic fisher vectors, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015: pp. 2974–2983.

[37] W. Choi, Y.-W. Chao, C. Pantofaru, S. Savarese, Indoor scene understanding with geometric and semantic contexts, Int. J. Comput. Vis. 112 (2015) 204–220.

[38] S. Gupta, P. Arbeláez, R. Girshick, J. Malik, Indoor scene understanding with rgb-d images: Bottom-up segmentation, object detection and semantic segmentation, Int. J. Comput. Vis. 112 (2015) 133–149.

[39] M. Stark, P. Lies, M. Zillich, J. Wyatt, B. Schiele, Functional object class detection based on learned affordance cues, in: Int. Conf. Comput. Vis. Syst., Springer, 2008: pp. 435–444.

[40] Y. Zhu, A. Fathi, L. Fei-Fei, Reasoning about object affordances in a knowledge base representation, in: Eur. Conf. Comput. Vis., Springer, 2014: pp. 408–424.

[41] C. Ye, Y. Yang, R. Mao, C. Fermüller, Y. Aloimonos, What can i do around here? deep functional scene understanding for cognitive robots, in: 2017 IEEE Int. Conf. Robot.

Autom., IEEE, 2017: pp. 4604–4611.

[42] J. Sawatzky, A. Srikantha, J. Gall, Weakly supervised affordance detection, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017: pp. 2795–2804.

[43] M. Hassanin, S. Khan, M. Tahtali, Visual affordance and function understanding: A survey, ArXiv Prepr. ArXiv1807.06775. (2018).

[44] Y. Golan, A. Shapiro, E.D. Rimon, A Variable-Structure Robot Hand That Uses the Environment to Achieve General Purpose Grasps, IEEE Robot. Autom. Lett. 5 (2020) 4804–4811. https://doi.org/10.1109/LRA.2020.3003885.

[45] Y. Golan, A. Shapiro, E. Rimon, Jamming-Free Immobilizing Grasps Using Dual-Friction Robotic Fingertips, IEEE Robot. Autom. Lett. 5 (2020) 2889–2896. https://doi.org/10.1109/LRA.2020.2972883.

[46] A. Sintov, A. Shapiro, Dynamic regrasping by in-hand orienting of grasped objects using non-dexterous robotic grippers, Robot. Comput. Integr. Manuf. 50 (2018) 114–131. https://doi.org/10.1016/j.rcim.2017.09.009.

[47] A. Sintov, R.J. Menassa, A. Shapiro, OCOG: A common grasp computation algorithm for a set of planar objects, Robot. Comput. Integr. Manuf. 30 (2014) 124–141. https://doi.org/10.1016/j.rcim.2013.09.004.

[48] Y. Lin, Y. Sun, Robot grasp planning based on demonstrated grasp strategies, Int. J. Rob. Res. 34 (2015) 26–42.

[49] M. Moll, L. Kavraki, J. Rosell, Randomized physics-based motion planning for grasping in cluttered and uncertain environments, IEEE Robot. Autom. Lett. 3 (2017) 712–719.

[50] K.M. Lundeen, V.R. Kamat, C.C. Menassa, W. McGee, Autonomous motion planning and task execution in geometrically adaptive robotized construction work, Autom. Constr. 100 (2019) 24–45. https://doi.org/10.1016/J.AUTCON.2018.12.020.

[51] N. Kejriwal, S. Kumar, T. Shibata, High performance loop closure detection using bag of word pairs, Rob. Auton. Syst. 77 (2016) 55–65. https://doi.org/10.1016/j.robot.2015.12.003.

[52] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: An efficient alternative to SIFT or SURF, in: 2011 Int. Conf. Comput. Vis., Ieee, 2011: pp. 2564–2571.

[53] T. Luddecke, F. Worgotter, Learning to Segment Affordances, 2017. https://doi.org/10.1109/ICCVW.2017.96.

[54] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, A. Torralba, Scene parsing through ADE20K dataset, 2017. https://doi.org/10.1109/CVPR.2017.544.

[55] T. Lüddecke, F. Wörgötter, Learning to Label Affordances from Simulated and Real Data, (2017). http://arxiv.org/abs/1709.08872 (accessed June 6, 2020).

[56] T. Lüddecke, T. Kulvicius, F. Wörgötter, Context-based affordance segmentation from 2D images for robot actions, Rob. Auton. Syst. 119 (2019) 92–107. https://doi.org/10.1016/j.robot.2019.05.005.

[57] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), Springer Verlag, 2015: pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28.

[58] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2016. https://doi.org/10.1109/CVPR.2016.90.

[59] P.O. Pinheiro, T.Y. Lin, R. Collobert, P. Dollár, Learning to refine object segments, in: Eur. Conf. Comput. Vis., 2016: pp. 75–91. https://doi.org/10.1007/978-3-319-46448-0_5.

[60] G. Ghiasi, C.C. Fowlkes, Laplacian pyramid reconstruction and refinement for semantic segmentation, in: Eur. Conf. Comput. Vis., Springer, 2016: pp. 519–534.

[61] G. Bradski, A. Kaehler, OpenCV, Dr. Dobb's J. Softw. Tools. 3 (2000).

[62] A. Hornung, K.M. Wurm, M. Bennewitz, C. Stachniss, W. Burgard, OctoMap: An efficient probabilistic 3D mapping framework based on octrees, Auton. Robots. 34 (2013) 189–

206. https://doi.org/10.1007/s10514-012-9321-0.

[63] P.E. Hart, N.J. Nilsson, B. Raphael, A Formal Basis for the Heuristic Determination of Minimum Cost Paths, IEEE Trans. Syst. Sci. Cybern. 4 (1968) 100–107. https://doi.org/10.1109/TSSC.1968.300136.

[64] D. Fox, W. Burgard, S. Thrun, The dynamic window approach to collision avoidance, IEEE Robot. Autom. Mag. 4 (1997) 23–33. https://doi.org/10.1109/100.580977.

[65] P. Beeson, B. Ames, TRAC-IK: An open-source library for improved solving of generic inverse kinematics, in: IEEE-RAS Int. Conf. Humanoid Robot., IEEE Computer Society, 2015: pp. 928–935. https://doi.org/10.1109/HUMANOIDS.2015.7363472.

[66] X. Peng, Z. Tang, F. Yang, R.S. Feris, D. Metaxas, Jointly optimize data augmentation and network training: Adversarial data augmentation in human pose estimation, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018: pp. 2226–2234.

[67] H. Noh, S. Hong, B. Han, Learning deconvolution network for semantic segmentation, in: Proc. IEEE Int. Conf. Comput. Vis., 2015: pp. 1520–1528.

[68] Y. Li, H. Qi, J. Dai, X. Ji, Y. Wei, Fully convolutional instance-aware semantic segmentation, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017: pp. 2359–2367.

[69] M. Kampffmeyer, A.-B. Salberg, R. Jenssen, Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Work., 2016: pp. 1–9.

[70] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch, in: NIPS 2017 Work., 2017.

[71] T. Tieleman, G. Hinton, Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, COURSERA Neural Networks Mach. Learn. 4 (2012) 26–31.

[72] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conf. Comput. Vis. Pattern Recognit., Ieee, 2009: pp. 248–255.

[73] C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, Understanding deep learning requires rethinking generalization, ArXiv Prepr. ArXiv1611.03530. (2016).

[74] A. Roy, S. Todorovic, A multi-scale cnn for affordance segmentation in rgb images, in: Eur. Conf. Comput. Vis., Springer, 2016: pp. 186–201.

[75] Z.D. Guo, Z.Y. Wang, S.F. Zhang, X. Li, L. Li, C. Li, Y. Cui, R. Bin Fu, Y.Z. Dong, X.Y. Chi, M.Y. Zhang, K. Liu, K. Liu, C. Cao, B. Liu, K. Zhang, Y.W. Gao, B. Lu, W. Chen, Aerosol and Surface Distribution of Severe Acute Respiratory Syndrome Coronavirus 2 in Hospital Wards, Wuhan, China, 2020, Emerg. Infect. Dis. 26 (2020) 1586–1591. https://doi.org/10.3201/eid2607.200885.

Preflight Results

Document Overview

Title:	Manuscript_BAE
Author:	Shuai Li
Creator:	Word
Producer:	macOS Version 10.15.7 (Build 19H2) Quartz PDFContext

Preflight Information

Profile:	Convert to PDF/A-2b
Version:	Qoppa jPDFPreflight v2020R2.01
Date:	May 13, 2021 2:12:47 PM

Legend: (X) - Can NOT be fixed by PDF/A-2b conversion.
(!X) - Could be fixed by PDF/A-2b conversion. User chose to be warned in PDF/A settings.

Page 5 Results
(X) Page uses transparency but does not have a device independent Blending Color Space

Page 19 Results
(X) Page uses transparency but does not have a device independent Blending Color Space