



Construction waste recycling robot for nails and screws: Computer vision technology and neural network approach

Zeli Wang^a, Heng Li^{a,*}, Xiaoling Zhang^{b,c}

^a Department of Building and Real Estate, The Hong Kong Polytechnic University, Hong Kong

^b Department of Public Policy, City University of Hong Kong, Hong Kong

^c Shenzhen Research Institute, City University of Hong Kong, Shenzhen, PR China

ARTICLE INFO

Keywords:

Construction waste management
Robotics in construction sites
Mobile robot coverage
Neural network
Computer vision
Faster R-CNN

ABSTRACT

Waste management scene is in urgent need of robotic waste sorter. Nails and screws, as part of the construction waste scene, are hard to be found and can therefore, cause damage to the site's construction safety and increase the material loss. This paper presents a construction waste recycling robot. In order to complete the recycling tasks, robots are expected to inspect the entire working environment and identify the target objects. This research uses neural network technology to assist the robot patrol in an unknown work environment and to use faster R-CNN methods to find scattered nails and screws in real time, so that the robot can automatically recycle nails and screws. This study introduces computer vision technology and a full-coverage path-planning algorithm into the field of construction waste management and proposes a novel construction waste recycling approach. Based on this robot, we can continue our study of construction waste recycling robots that can automatically sort and recycle most construction waste in the future.

1. Introduction

Generally, construction waste is the surplus and damaged materials produced during the construction and demolition (C&D) process [1]. As a common human social activity, the construction industry makes many negative impacts on the natural environment, and construction waste is one of them [2]. Many countries and regions are suffering from construction waste, such as China, the United States, South Korea and Kuwait [3]. Since the main treatment of construction waste is landfill, the huge amount of construction waste consumes many of land sources and leads to a large amount of GHG emissions [4]. Hong Kong, for instance, is running out of existing landfill sites [5]. Shenzhen, one of China's first-tier cities, is also facing the same problem [6]. This problem not only occurs in developing countries but also in developed countries. 98 million tons of construction waste was landfilled in 2003 in the United States [7]. Therefore, when faced with difficulties reducing construction waste at source, attention should be diverted to re-use and recycling [8]. By on-site recycling of construction waste, we can increase the proportion of reuse and recycling of construction materials, reduce waste transportation and disposal costs, prolong the service life of landfills, and lessen the pollution generated by construction waste [9]. Of all kinds of construction waste, nails and screws are common objects on construction sites and are hard to find. Being hurt by

construction materials can also result in workers having to take time off, medical costs, being maimed and suffering fractures [10]. Construction workers are also facing the risk of stepping on a nail or screw [11], which may cause such serious infection as tetanus. Therefore, recycling nails and screws can reduce the risks of injury on construction sites as well as saving money. However, workers usually ignore and nails and screws lying around construction sites as being too small and insignificant to look for.

Although automation technologies have been widely applied in the construction industry, much of the on-site construction waste cleaning and recycling work is still processed manually, which is inefficient and costly [12]. Autonomous nail and screw recycling robots are expected to increase the efficiency and possibility of their recycling [13]. However, the unstructured nature of on-site construction environments, typically reflected in the absence of a predefined fixed location or a predefined fixed working path, makes the design and implementation of automated recycling robots more difficult [14].

This paper presents a prototype nail and screw recycling robot based on computer vision technology and a complete coverage path planning (CCPP) algorithm. The prototype contains a crawler chassis, a four-degree-of-freedom paw robot arm, detection system with camera and omnibearing laser scanner, multi-cell storage box, and control system. The robot uses computer vision technology to identify nails and screws

* Corresponding author.

E-mail addresses: jerry.wang@connect.polyu.hk (Z. Wang), bshengli@polyu.edu.hk (H. Li), xiaoling.zhang@cityu.edu.hk (X. Zhang).

and put them in specified storage boxes and is expected to search the entire workspace without any previous knowledge of the location of obstacles and waste, with the working path being determined solely by the information provided by the laser scanner. To do this involved the creation of a dedicated training dataset for Faster R-CNN training, to enable the control system to convert environmental information into neural activities and guarantee the time efficiency of coverage.

The rest of this paper is organized as follows: Section 2 reviews the current literature relating to CAPP algorithms for cleaning robots and technologies used in target detection. Section 3 illustrates the design of the prototype, including the mechanics, moving control algorithms, and detection approach. Section 4 evaluates the performance of the prototype, followed by the Section 5, which discusses the limitations of the study and potential future work.

2. Literature review

This section reviews the technology and state-of-the-art in relation to full-coverage path planning and computer vision real-time detection. We briefly introduce various methods, their applications, and advantages and disadvantages. Previous studies of coverage path planning show that it is difficult to find the optimal path in the location environment, and instead, the ‘greedy algorithm’ is used to select the current optimal solution. For the target detection algorithm, the emergence of ‘faster R-CNN’ means the target detection technology can be widely used in real-time detection related applications.

2.1. Path planning algorithm

Many coverage path planning algorithms are widely applied to the automation control of terrain scanner robots, demining robots, floor cleaning robots, painter robots, lawn mowers, automated harvesters, window cleaners, inspection of complex underwater structures, etc. [15–19]. Since information relating to the entire workspace is given before path planning, the algorithms used are multitudinous off-line planning algorithms, such as classical exact cellular decomposition methods (trapezoidal decomposition [20], boustrophedon decomposition [21,22], Morse decomposition [23] and distance transforms algorithms [24]). However, there are many uncertain factors on construction sites. Although the border of the entire workspace can be accurately represented by the construction drawings, it cannot authentically reflect the details of workspaces due to the existence of dismantling, rebuilding and fitting works. Therefore, using CAPP algorithms for uncertain environments could provide a feasible approach to assist a floor-tiling robot, for instance. Many methods have been applied to path planning in uncertain environments during the last two decades, such as disk covering [25], bioinspired neural networks [26], cellular decomposition [27,28], sensor-based coverage approaches [21,29] and finite state machines (FSM) [30].

Obstacle avoidance by the disk covering algorithm was first proposed in 2004, as an early trial of the CAPP algorithm. This solution assumes that the working area can be represented by an arbitrary closed curve. Therefore, the entire workspace can be encased by a minimum-area rectangle using an optimized method based on a method proposed in 1945 [31]. They can decompose the quadrature workspace into several circular regions using the method proposed by Kershner [41] and ensure the minimum number of disks at the same time. After the decomposition work, the patrolling robot is expected to go through every center of disks from the boundaries of the entire workspace. Though the decomposition method of this solution is convincing, we did not consider the obstacle avoidance method for the unknown barriers inside the workspace as it is yet far from practical application.

The second solution uses neural network technology. The main principle of this algorithm is to make the uncleaned area global to attract the cleaning robot while making the barriers local to reject the robot. In contrast with other solutions, this CAPP algorithm represents

the workspace using the triangular cell decomposition method proposed by Joon Seep Oh et al. [48] to expand the potential directions of the robot. The neural network algorithm then determines the coverage priority of robot. However, the complexity of neural networks makes the computational complexity of this algorithm higher than other methods. A similar approach has been used in underwater vehicles [32], which extends the 2-D path planning algorithm to 3-D underwater environments. Similarly, Guo and Balakrishnan proposed a coverage solution for nonholonomic mobile robots by integrating neural network technology and the circular region decomposition method [33].

Boustrophedon decomposition is one of the cellular decomposition methods that have been widely used in CAPP research both for certain and uncertain environments. Batsaikhan et al. proposed the method to decompose the workspace into easily covered rectangles using the detected characteristic bindery of obstacles and cover the single region with the traditional zig-zag path, which can work well in covering uncertain workspace [21]. Although this method was used in a certain workspace by Choi et al. in given environments, there are many differences between certain and uncertain environments. More importantly, this algorithm may cause dead zones in extreme situations, and the overlap rate of coverage path is primary determined by the location of obstacles. Recently, there is another method based on boustrophedon motion, which combined with advanced point-to-point path planning algorithms to reduce the distance of the backtracking path [22]. This method performs a lower overlap rate when comparing with other CAPP algorithms based on boustrophedon motion. However, due to the disadvantage of boustrophedon motion, the algorithm needs to find the backtracking path multiple times in a complex environment, which makes the robot still waste more time and computational power.

Another CAPP algorithm proposed by Caihong Li et al. claimed that the algorithm based on the Finite State Machine (FSM) approach and rolling windows approach could cover unknown environments completely without causing too much overlap [30]. Their result shows the algorithm will cause an 11.98% overlap rate, which is about 40% of the random CAPP approach when supposing the coverage rate is about 98%. This online path-planning algorithm is a step-by-step method for coverage planning. The robot detects adjacent environmental information and updates the rolling window every time it shifts to another area. The size of the rolling window is very small and makes the computational complexity of this algorithm relatively low. Hence, the robot decides the direction of its next motion using ‘greedy strategy’, which means that the robot will visit the grid with the highest possibility of reaching another unvisited grid. This algorithm successfully proved that the FSM approach, rolling windows approach and greedy strategy could be effective and efficient in solving CAPP problems in uncertain environments. Therefore, it is used with some modifications and improvements to fit the requirements of a floor tiling robot. This research provided reliable proof of the effectiveness of the FSM approach, rolling windows approach and the greedy strategy while solving path-planning problems in complicated unknown environments. The characteristics of this algorithm are in good agreement with such requirements of floor tiling works as low repetition rate, orderly work, and controllable coverage rate. Therefore, the path-planning algorithm described in the present paper learns from this algorithm and improves its shortcomings.

2.2. Target detection

Ross Girshick et al. proposed a CNN-based target detection algorithm named *Regions* with CNN features (R-CNN) [34]. The process is divided into four steps: (1) *Extract region proposals*, where the Selective Search method is used to generate 1 K–2 K candidate areas in a picture; (2) *Feature Extraction*, in which, for each candidate region, the feature is extracted using a depth convolutional network; (3) *Test-time detection*, in which the feature is sent to each class of SVM classifier to determine if it belongs to that class; and (4) *Bounding box regression*, using this

simple method to reduce the localization error. R-CNN was the first to use CNN technology in object detection, which greatly enhanced the effectiveness of target detection and laid the foundation for future research in this field. However, R-CNN is not fast enough in real-time detection because the algorithm needs to process the CNN operation for each object proposal.

Fast R-CNN is a method based on R-CNN that classifies object proposals more quickly and efficiently [35]. Fast R-CNN is superior to both R-CNN and SPPnet in training, testing, and mAP. In short, compared with R-CNN, fast R-CNN completes the repeat computations of R-CNN in one operation, using a multi-task loss.

Since fast R-CNN successfully reduces the time to detect objects using CNN, more possibilities for optimizing the detection time exist in the region proposal algorithms. Faster R-CNN came into being because of this request. Faster R-CNN proposes a Regional Proposal Network (RPN), which shares convolutional features with the down-stream detection network to enable nearly cost-free region proposals. Meanwhile, the learned RPN can also improve the accuracy of object detection. The efficiency of faster R-CNN makes real-time detection a truly widespread reality.

Faster R-CNN has been widely used in various fields. Some previous studies focused on the detection of small objects. One improved the faster R-CNN algorithm and enabled faster R-CNN to be used in lane marking detection by combining such other methods as fast multi-level combination, context cues, and a new anchor generating method [36]. A recent study has improved faster R-CNN and used it in the field of face recognition [37]. For construction sites, Fang and other researchers have used this method in the field of safety inspection and supervision, providing important opportunities for real-time on-site monitoring and improved safety management [38]. Other construction safety management research also mentions this method [39], in the use of computer vision technology to detect objects as dangerous or not, depending on their location.

3. Methodology

A nail and screw recycling robot mainly faces three challenges. Firstly, the unstructured working environments require robots to move without any environmental information. Secondly, robots need to identify nails and screws in real-time. Finally, robots are expected to return automatically to avoid affecting other work. To face these challenges, recycling robots need a meticulous structural and algorithmic design. There are some structural similarities with cleaning robots since their tasks are similar. Learning from a traditional cleaning robot [40], the structure of a nail and screw recycling robot can be broadly divided into a motion unit, collection unit, storage unit, sensing unit, and control system. However, unlike a floor-cleaning robot, it needs to identify and classify nails and screws in complex environments, which increases the task's complexity.

3.1. Robot design

The nail and screw recycling robot consists of four main units: the motion unit, the robot arm unit, and two detection units. Recycling robots for construction sites are usually required to move and operate on rough and uneven terrain because most of the time the ground is not flat. Numerous studies have shown that crawler robots can adapt well to complex terrestrial environments and help robots work in harsh working conditions. Wang et al. [49] proved that the tracked robot could complete such tasks as moving on slopes, stepping over ditches, and climbing over obstacles. Therefore, as shown in Fig. 1(a), the movement unit of this autonomous nail and screw recycling robot is a crawler mobile unit with a specially-designed passive suspension that can go through obstacles without violent vibration of the robot body. This is very important for robot mobility and agility when moving on the rough ground. A robot arm of 5 DoF, lightweight and high loading

capacity for flexible manipulation was developed for picking and sorting nails and screws. In addition, the robot has two environment perception devices using an LIDAR scanner and video camera using computer vision technology to identify barriers and target objects respectively. The prototype is shown in Fig. 1(b).

3.2. Complete coverage searching

The proposed algorithm is expected to build a neural network architecture with dynamic neural activities that can represent a dynamic workspace and assist the recycling robot in patrolling through every sub-region in an authorized area. We firstly divide the specified workspace into a minimum number of sub-regions, with each sub-region treated as a neuron in the neural network. The membrane potential of each neuron is determined by internal neural connections, which makes unclean areas and obstacles attract and exclude robots respectively. The robot path is planned on-line based on the membrane potential of adjacent neurons, and the membrane potential of each neuron is updated by a rolling windows approach so that the robot will go through all regions autonomously.

3.2.1. Environments and discretization

The detection device of the robot is a video camera. The camera is placed at the front of the robot at a fixed angle and scans the ground ahead of the robot in real time. As Fig. 2 shows, the ground being scanned is trapezoidal due to the way the camera works and how it is placed.

The number of nodes in the neural network algorithm directly affects its computational complexity. Therefore, increasing the area of a single neuron can reduce the number of neurons in the neural network and reduce the computational burden. Hence, we designed the robot scanning process as two steps. In the first step, the robot shifts from the initial region to the next region and the camera sweeps through a rectangular area. Then, the robot rotates in place and sweeps the camera across an annular area. As shown in Fig. 3, the entire circular area can be cleaned by the two steps.

3.2.2. Map representation

According to the initial solution of decomposing a rectangle by circles [41], the algorithm encases a limited area by a minimum-area rectangle [31] and, as presented by Yi and Zhihua [42], the map of workspace can be easily represented by a minimum number of circles whose area is determined by the detection distance of the video camera.

Firstly, this study encases the given workspace using a minimum encasing rectangle (MER) - defined as the rectangle of minimum area that can enclose the given area. Although the MER is calculated by various methods, the one used here is based on the theorem that the MER must be collinear with a side of the given polygon [31]. Since the common boundaries of the construction sites will not be complicated curves, the computational complexity will not be too high. A typical example is shown in Fig. 4.

Therefore, an irregular polygon workspace can be decomposed into circular areas. Due to the theorem proposed by Kershner [41], the distance between adjacent circles is $\sqrt{3}R$, which determines the position of the circles. The result is shown in Fig. 5, where R is the detection distance mentioned above.

Based on the structure of the barrier detection area recognition area, the network planning of node i is represented in Fig. 6, where each node has six adjacent nodes.

Moreover, as a common standard, we assume that the circular center coordinates of the lower left corner are $(0.5R, 0)$, which has been proven valid in a previous study [42]. The workspace represented by the circular areas is shown in Fig. 7. Obviously, over 95% of the workspace will be covered if the robot goes through every node inside the workspace.

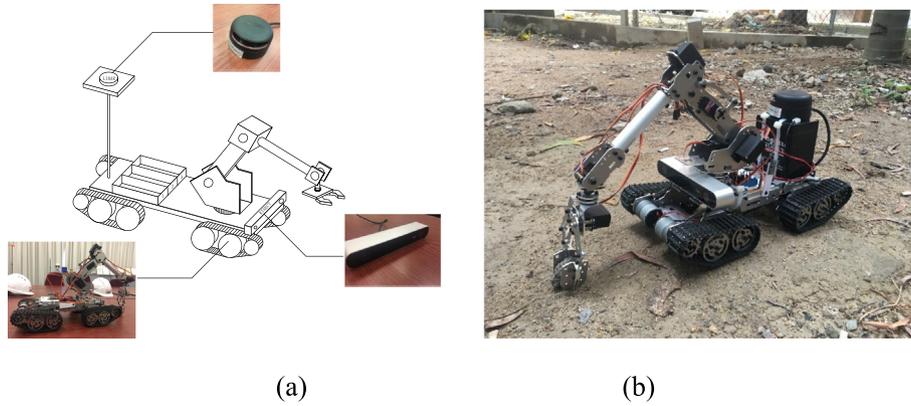


Fig. 1. Design drawing and prototype of the construction-recycling robot.

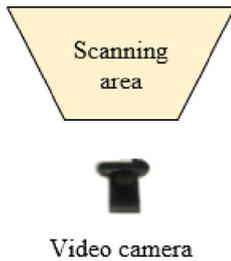


Fig. 2. Video camera working range.

3.2.3. Model construction

An improved neural network model was proposed by Luo and Yang [26], in which the dynamics of each neuron is calculated by the shunting equation, derived from Hodgkin and Huxley's membrane equation [43]. In contrast with Luo and Yang's approach, we integrate the spiral filling motion because of its robustness, as proved in previous studies [27,28,44,45]. The basic idea of this neural network approach is to make the neurons, which are surrounded by more obstacle and cleaned areas, be more attractive to the robot while the obstacles are excluded from the robot to avoid collision through the dynamic neural network landscape.

Since we want to encourage robots scanning from the boundary to the center, which reduces the return distance at the completion of all tasks, the shunting equation is defined as:

$$dx_i/dt = -Ax_i + (B - x_i) \left\{ [I_i]^+ - \sum_{j=1}^{j=k} \omega_{ij} [x_j]^+ \right\} - (D + x_i)[I_i]^- \quad (1)$$

where i and j are the number of neurons; I_i is the external input and ω_{ij}

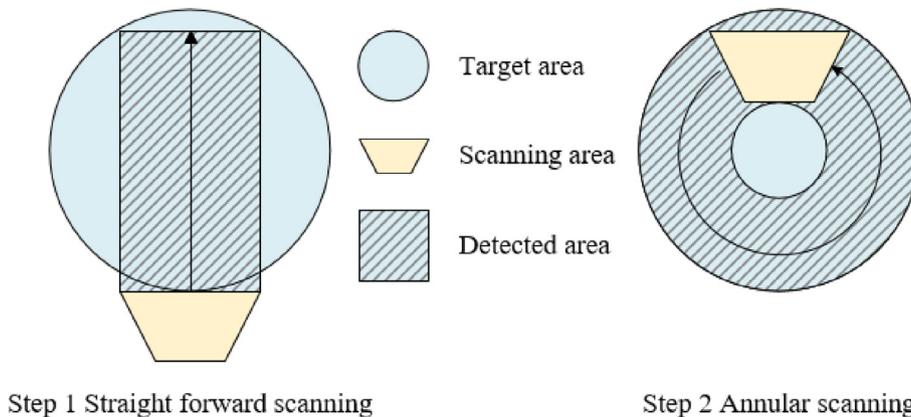


Fig. 3. Scanning process of a single neuron.

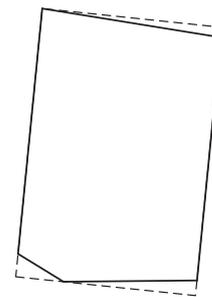


Fig. 4. Encasing the given workspace using a minimum encasing rectangle.

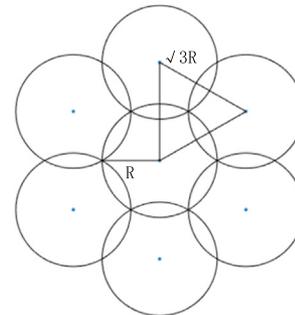


Fig. 5. Pattern of decomposing a rectangle by circles.

is the connection weight between neuron i and j; A, B and D represent nonnegative constants describing the passive decay rate and upper and lower bounds respectively. Affected by this equation, neural activity of

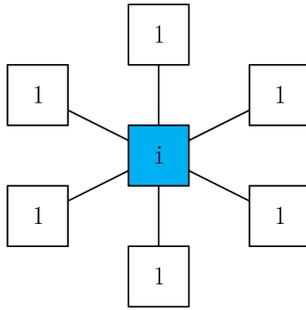


Fig. 6. Adjacent nodes to node i.

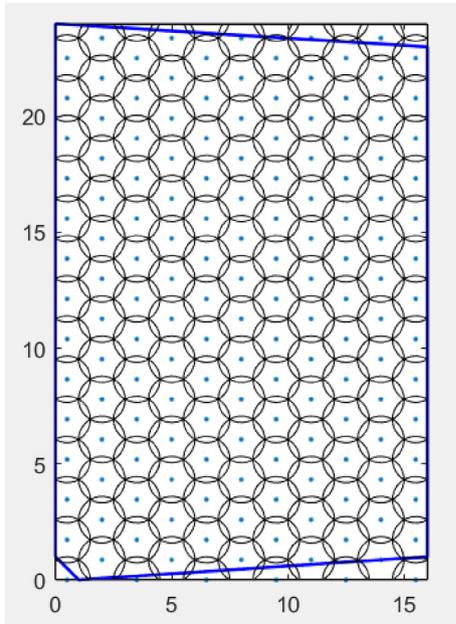


Fig. 7. Workspace represented by circular areas.

the obstacle will be less than zero and neural activity of the area near the obstacle and cleaned area will be slightly higher than the area away from the obstacle.

The value of the external input is determined by whether the node is occupied or not. In order to avoid collision, the value of I_i is set to $-E$ when the neuron is occupied. In contrast, the value of I_i is set to E when the node is free. The value of I_i is set to 0 when the node is cleaned to make sure the cleaned area neither excludes nor attracts the robot. The equation is

$$I_i = \begin{cases} -E & \text{obstacle} \\ E & \text{free} \\ 0 & \text{cleaned} \end{cases} \quad (2)$$

The connection weight ω_{ij} is determined by the Manhattan distance of two regions. In this case, the Manhattan distance of an adjacent neuron is set to 1. Therefore, the connection weight can be calculated by Eq. (3), where d_{ij} is the Manhattan distance of neuron i and neuron j - the value of ω_{ij} being is inversely proportional to d_{ij} . In order to decrease computational complexity, farther nodes will be ignored. This is controlled by variable α .

$$\omega_{ij} = \begin{cases} \mu/d_{ij} & 0 \leq d_{ij} \leq \alpha \\ 0 & d_{ij} > \alpha \end{cases} \quad (3)$$

Similarly, when the robot performs point-to-point motion, the neuron activities are calculated by

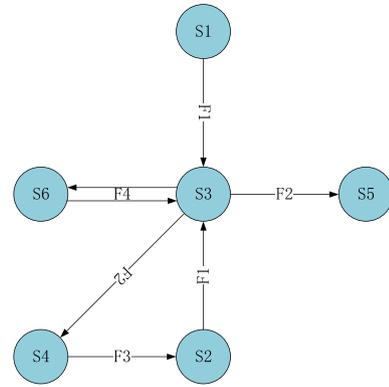


Fig. 8. Structure of the Finite State Machine (FSM).

$$\frac{dx_i}{dt} = -AX_i + (B - x_i) \left\{ [I_i]^+ + \sum_{j=1}^{j=k} \omega_{ij} [x_j]^+ + \varphi \cos \delta_{it} \right\} - (D + x_i) [I_i]^- \quad (4)$$

In contrast with shunting Eq. (1), this equation adds a parameter $\varphi \cos \delta_{ij}$, which guarantees that the robot tends to approach the target neuron. φ is a positive coefficient and δ_{it} is the angle between the vector from robot to neuron i and the vector from robot to target neuron t . Meanwhile, the parameter $\sum_{j=1}^{j=k} \omega_{ij} [x_j]^+$ ensures the robot moving path will be far away from the obstacles.

3.2.4. Path planning

We divide the robot into five states in FSM as shown in Fig. 8. S1 is the initial state, S2 is the state of existing uncovered free grids adjacent to the robot, S3 is the state of without uncovered free grids adjacent to the robot, S4 is the state the nearest uncovered grid has been found, and S5 is the end of the entire process. S6 is the state that the robot went to the appointed place when lacking power or with a full load. The strategies marked F1, F2, and F3 are illustrated in Fig. 8. F1 is the strategy that the robot is searching for the neuron with maximum neural activity. F2 is the strategy that the robot is searching for the nearest uncovered free grid using the Dijkstra algorithm, and F3 is the strategy that the robot is shifting to the target uncovered free grid.

In the F1 strategy, the path-planning algorithm decides the next working grid of the robot based on the rolling windows as Fig. 9 shows. The degree of attraction of the robot's peripheral nerve nodes to the robot can be calculated by the previously mentioned shunting equations. In order to reduce the energy and time costs of the robot, the coverage pattern is generated from the neural network model and the previous situation of the robot. The following movement direction is firstly determined by the neural activity and secondly determined by the degree of swerving. Therefore, the next neuron N_{next} is obtained by

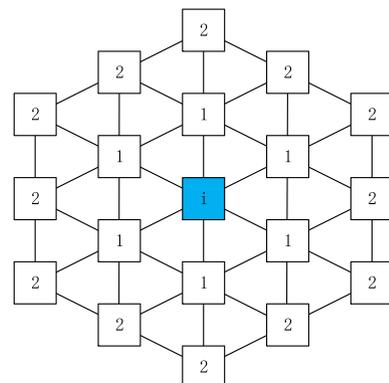


Fig. 9. Network planning of node i.

$$N_{next} \leftarrow x_{N_{next}} = \text{MAX}(x_j + \beta \times \text{COS}\theta, j = 1, 2, \dots, k) \quad (5)$$

where β is a small positive constant and k is the number of adjacent neurons of the current neuron; $\theta \in [0, \pi]$ is defined as the turning angle between the current orientation and the next moving direction.

When the grids in the first layer are either visited or occupied, the robot is in the S3 state. In this state, the control system uses the Dijkstra algorithm to search for the nearest uncovered free grid. The grid information record in F1 strategy will be used. With the Dijkstra algorithm, the system searches the nearest unvisited neuron in the neural network topology from near to far with regard to the central neural, where the robot is located. The robot can subsequently move close to the nearest unvisited point as soon as possible using Eq. (5).

Strategy S5 means that the robotic package is full and needs to be moved to the item collection point. Similarly, the robot uses Eq. (5) to determine its own movement path. When no grid is detected that has not been visited, the system enters the S6 state. In this state, the robot is expected to return to its initial position.

3.3. Detection and recycling

Faster R-CNN, as the most reliable computer vision object detection technology, is used here to detect the location of objects. This section illustrates the training, testing, and application process of the model. A large amount of diverse training and testing data has been established to ensure the reliability and stability of the model. After the model passes the test, we use it for an instance of real-time video exploration to ensure the model is viable.

Three common types of nails and screws are used as prototype recycling targets. The practical application experiment is completed by a fixed robot to verify the reliability of the model.

3.3.1. Training

We placed the different kinds of nails and screws separately on the cement floor, sand, and marble and collected the required data through many cameras. The data covers a variety of possible scenarios and has ample dataset size, including various backgrounds and perspectives.

After collecting the data, we manually annotated the data through the graphical image annotation tool Labeling [46]. The preprocessing of the prepared data set is shown in Fig. 10. Then we saved the file in VOC2007 format in the specified directory and trained the ZF model using faster R-CNN in CAFFE [47]. To ensure the reliability of the model, we set the first and second stage of RPN and the first and second stage of faster R-CNN to 40,000 and 20,000 iterations respectively. The result shows that the model's mean average precision (AP) for nails and screws is 0.891.



Fig. 10. Preprocessing the training dataset.

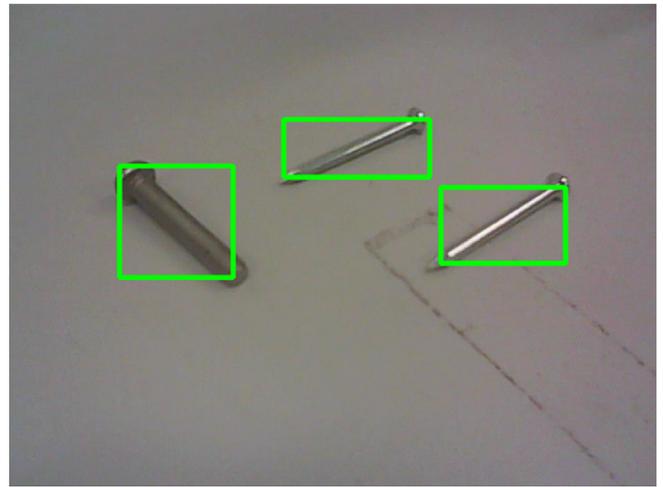


Fig. 11. Object detection testing result

3.3.2. Testing

We tested the model repeatedly with video and images. Some results are shown in Fig. 11.

Using the faster R-CNN technology, the camera can accurately identify the target object and provide an accurate relative position. Hence, we can convert the relative position into the absolute position of the object and control the robot to pick it up. Since the position of the camera is fixed and we assume that the prototype is on the same plane as the detected object, the coordinates in the video can easily be converted to the difference in position between the detected object and the prototype in the working plane.

4. Evaluation

4.1. Detection and picking-up system

In order to evaluate the detection unit, we applied the trained model to different backgrounds and observed the results. As shown in Fig. 12, the nails and screws can be detected in different backgrounds.

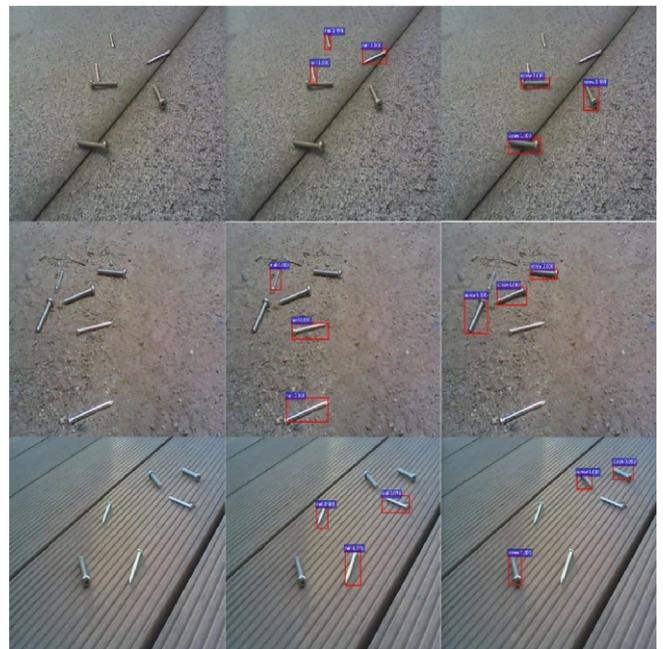


Fig. 12. Video frame and detection results in different working environments.

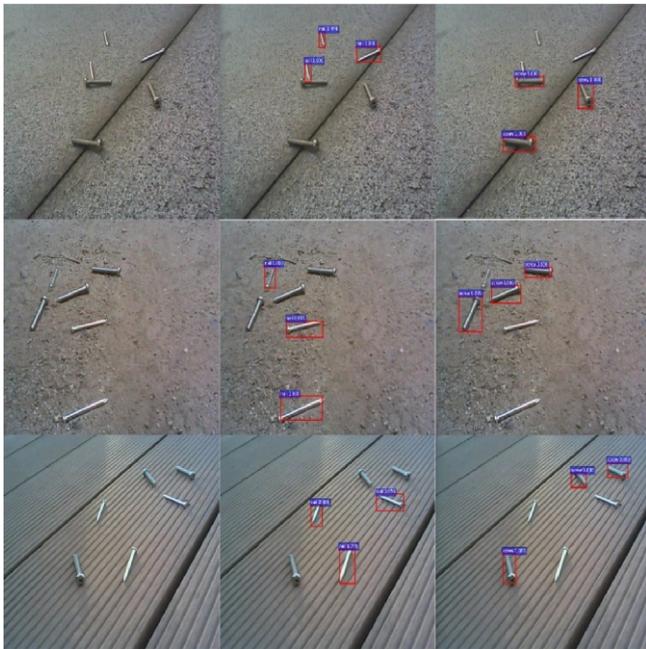


Fig. 13. The picking up and sorting process.

Then, we tested the model with the real nail and screw recycling robot. As shown in Fig. 13, the robot not only can successfully pick up the nails and screws, but also put them in the correct box.

Since the experiment is supported by video camera, some of the video frames gathered by the camera are shown in Fig. 13. With the faster R-CNN model, the robot can recognize the target object with high probability, which is shown in Fig. 14.

4.2. Prototype movement

The proposed neural network approach is capable of planning a complete coverage path for cleaning robots without any human intervention. In this section, the model is applied in a circular cell decomposition workspace without using any previously known environmental information, although the boundary of the entire workspace is known.

For comparison, the model was applied to a completely unknown outdoor environment. The environment of the entire workspace is assumed totally unknown except the boundary of the construction site, which can be easily obtained from a drawing and is usually static. The robot can only sense a limited range with a sensor named LIDAR



Fig. 14. Video frame and detection results.

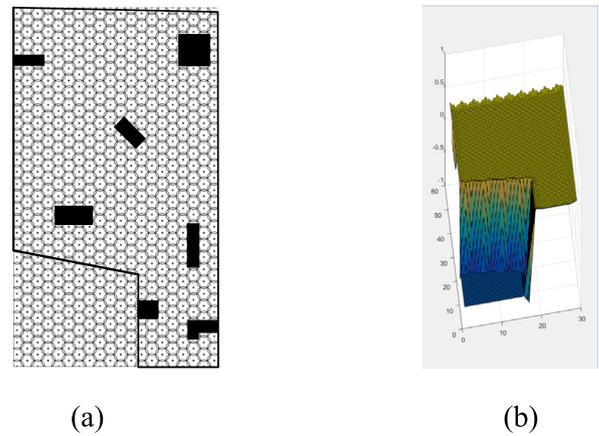


Fig. 15. Top view and activity landscape of the neural network upon commencement.

Scanner. The neural network includes 20×30 discretely and topologically organized neurons, where all the neural activities are initialized to zero. The model parameters are set as $A = 50$, $B = 1$, and $D = 1$ for the shunting equation; $\mu = 0.05$ and $\alpha = 1$ for the lateral connections; and $E = 100$ for the external inputs. The robot is initially set in $S(1,11)$ which is the left bottom of the workspace. The boundary of the entire workspace is obviously illustrated by the neural activity landscape, where the unknown area is regarded as an uncleaned area (Fig. 15).

When the robot meets the first deadlock in $L(15,3)$, as shown in Fig. 16(a), the neural activity of the entire workspace is represented in Fig. 16(b). The neural activity of the unknown environment and uncleaned regions is high, while the neural activity of the obstacle area is low, and approximate to zero for the cleaned area.

When the entire task is finished, the neural network turns into a static situation, in which the neuron activities of all unoccupied neurons are equal to zero, as illustrated in Fig. 17.

Whether it is for deadlock off or return mode, point-to-point path planning is essential. We assume that the robot is currently in $E(16,24)$ and the coordinate of the target neuron is $T(1,11)$. Therefore, the landscape of the neural network is generated as shown in Fig. 18. The neural activity of neurons away from the target neuron is much lower than that close to the target neuron. Meanwhile, the neurons near obstacles have a lower degree of neural activity. The neural landscape determines that the movement of the robot will be as straight as possible towards the target point while avoiding obstacles.

The result shows that in this case study, the coverage rate is 100% with a 0.88% repetitive rate. In this case, we assume that the radius of a circle is 2m. Therefore, the robot has passed 1828m and turned

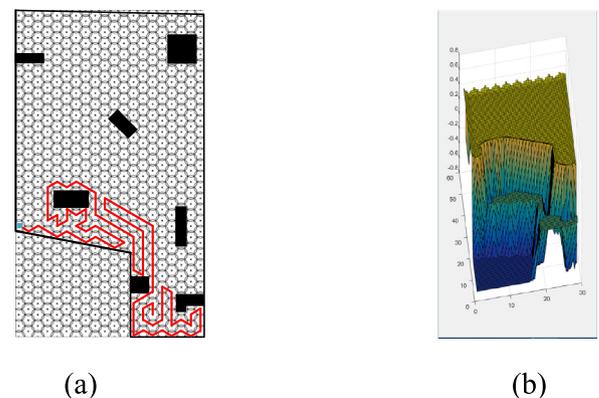


Fig. 16. Top view and activity landscape of the neural network when meeting the first deadlock.

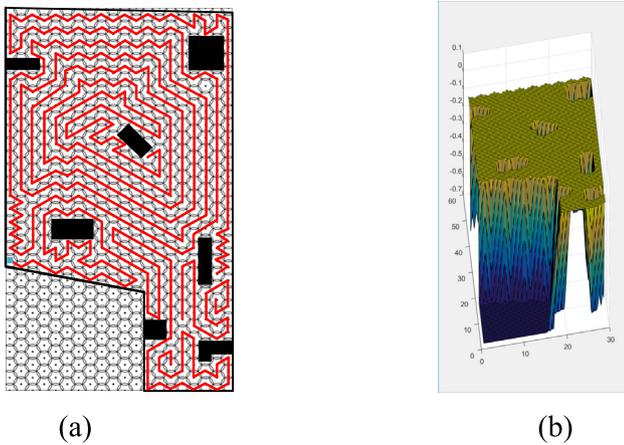


Fig. 17. Top view and activity landscape of the neural network on completion.

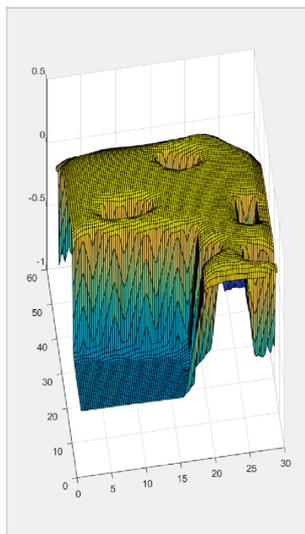


Fig. 18. Activity landscape of the neural network for the point-to-point case.

17,700° through the entire process.

In order to demonstrate the advancement of this algorithm, the performance of a recent algorithm is used for comparison. Due to the different ways in which maps are expressed, we improved the algorithms developed by Amna et al. [22] and tested them on the same map. The result is shown in Fig. 19. Although the algorithm reduces the length of the backtracking path compared with the traditional boustrophedon motion algorithm, it still has more backtracking and has a higher repetitive rate. In this case, the coverage is 100% and the repetition rate is over 12%. The total turning angle exceeds 8040° and the total distance is more than 2036 m by using this algorithm. In this study, the speed of the robot is 0.5 m per second, and it takes 4 s to rotate one revolution. Therefore, the algorithm proposed in this study reduces the coverage time by nearly 7.4%. Meanwhile, the efficiency of the algorithm will be more significant in a larger working environment.

5. Conclusion

This paper describes the development and testing of a prototype nail and screw recycling robot. Its structure mainly comprises two parts, one for moving, and the other for recycling. The moving unit is a crawler robot chassis with a CCP algorithm to assist it in going through every part of the construction site, while the recycling unit, which contains a video camera and 5 DoF robot arm, collects and classifies the target objects. Computation complexity and work efficiency are traded off by

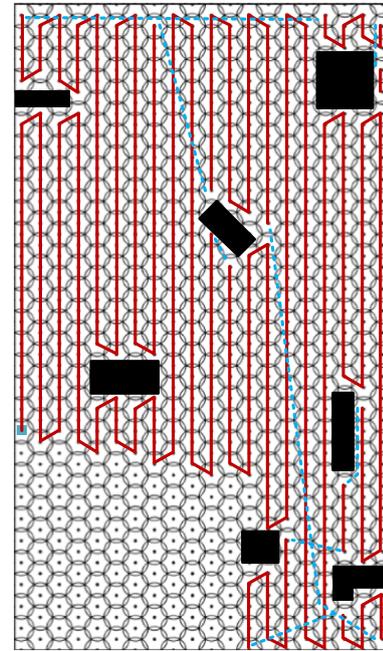


Fig. 19. The performance of path planning using two-way proximity search.

using circular sub-regions to represent the entire workspace, a neural network is used to ensure the efficiency of coverage, and the faster R-CNN algorithm is applied in detecting target objects. The feasibility and efficiency of the proposed prototype are discussed and illustrated through experiments.

The combination of computer vision technology and a full-coverage path planning algorithm for recycling is a novel development in the field of construction waste management and, based on this robot, we will continue to study construction waste recycling robots that can automatically sort and recover most of the construction waste in our future work. However, while the proposed prototype provides an effective solution for automatic waste recycling, some improvements in efficiency and general availability are still needed. In order to avoid causing more repetition rates, the algorithm generates a lot of corners on the path. In future works, we will further improve coverage efficiency by adding parameters that control the degree of turning in the shunting equation and setting reasonable weights. The Omnidirectional Multi-Camera System, for instance, which can detect surroundings without robot rotation, may be a better choice for a recycling robot as the efficiency of cleaning one region could be increased. However, this requires higher accuracy of Omnidirectional Multi-Camera System and higher computation complexity in object detection. Setting up a larger computer vision dataset will make also it possible to identify more types of construction waste in different circumstances. Furthermore, we verified the performance of the faster R-CNN algorithm in non-complex environments, while the current prototype expectedly is less effective in more complex situations. The use of other computer vision technologies and larger datasets, such as 'big data', are likely to increase the prospects for the development of recycling robots for real construction sites in the future.

Acknowledgements

We are thankful for the financial support of 1) The Research Grants Council, University Grants Committee of Hong Kong grant entitled "Proactively Monitoring Construction Progress by Integrating 3D Laser-scanning and BIM" (PolyU 152093/14E); and 2) Environment and Conservation Fund entitled "Construction Sorting and Recycling Robot" (grant no. ECF 70/2017).

References

- [1] Z. Chen, H. Li, C.T.C. Wong, An application of bar-code system for reducing construction wastes, *Autom. Constr.* 11 (5) (2002) 521–533, [https://doi.org/10.1016/S0926-5805\(01\)00063-2](https://doi.org/10.1016/S0926-5805(01)00063-2).
- [2] W. Lu, C. Webster, K. Chen, et al., Computational Building Information Modelling for construction waste management: Moving from rhetoric to reality, *Renew. Sust. Energ. Rev.* 68 (2017) 587–595, <https://doi.org/10.1016/j.rser.2016.10.029>.
- [3] Tao Ding, Jianzhuang Xiao, Estimation of building-related construction and demolition waste in Shanghai, *Waste Manag.* 34 (11) (2014) 2327–2334, <https://doi.org/10.1016/j.wasman.2014.07.029>.
- [4] Md Uzzal Hossain, Zezhou Wu, Chi Sun Poon, Comparative environmental evaluation of construction waste management through different waste sorting systems in Hong Kong, *Waste Manag.* 69 (2017) 325–335, <https://doi.org/10.1016/j.wasman.2017.07.043>.
- [5] Environmental protection department an overview on challenges for waste reduction and management in Hong Kong, https://www.epd.gov.hk/epd/english/environmentinhk/waste/waste_maincontent.html, (2015), Accessed date: 19 September 2018.
- [6] Zhikun Ding, et al., A system dynamics-based environmental performance simulation of construction waste reduction management in China, *Waste Manag.* 51 (2016) 130–141, <https://doi.org/10.1016/j.wasman.2016.03.001>.
- [7] T.W. Ann, et al., Impact of construction waste disposal charging scheme on work practices at construction sites in Hong Kong, *Waste Manag.* 33 (1) (2013) 138–146, <https://doi.org/10.1016/j.wasman.2012.09.023>.
- [8] O.O. Faniran, G. Caban, Minimizing waste on construction project sites, *Eng. Constr. Archit. Manag.* 5 (2) (1998) 182–188, <https://doi.org/10.1108/eb021073>.
- [9] Jiayuan Wang, et al., Critical success factors for on-site sorting of construction waste: a China study, *Resour. Conserv. Recycl.* 54 (11) (2010) 931–936, <https://doi.org/10.1016/j.resconrec.2010.01.012>.
- [10] Tore J. Larsson, Brian Field, The distribution of occupational injury risks in the Victorian construction industry, *Saf. Sci.* 40 (5) (2002) 439–456, [https://doi.org/10.1016/S0925-7535\(01\)00015-7](https://doi.org/10.1016/S0925-7535(01)00015-7).
- [11] Toivo Niskanen, Olli Saarsalmi, Accident analysis in the construction of buildings, *J. Occup. Accid.* 5 (2) (1983) 89–98, [https://doi.org/10.1016/0376-6349\(83\)90014-7](https://doi.org/10.1016/0376-6349(83)90014-7).
- [12] L.Y. Shen, et al., Mapping approach for examining waste management on construction sites, *J. Constr. Eng. Manag.* 130 (4) (2004) 472–481, [https://doi.org/10.1061/\(ASCE\)0733-9364\(2004\)130:4\(472\)](https://doi.org/10.1061/(ASCE)0733-9364(2004)130:4(472)).
- [13] Carlos Balaguer, Nowadays trends in robotics and automation in construction industry: Transition from hard to soft robotics, *Proc. 21st Int. Symp. Automation and Robotics in Construction*, 2004 <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.462.5812&rep=rep1&type=pdf>, Accessed date: 19 September 2018.
- [14] Chen Feng, et al., Vision guided autonomous robotic assembly and as-built scanning on unstructured construction sites, *Autom. Constr.* 59 (2015) 128–138, <https://doi.org/10.1016/j.autcon.2015.06.002>.
- [15] Mesut Sifyan, Haluk Rahmi Topcuoglu, Murat Ermis, Path planning for mobile sensor platforms on a 3-D terrain using hybrid evolutionary algorithms, *Evolutionary Computation (CEC)*, 2010 IEEE Congress on. IEEE, 2010, <https://doi.org/10.1109/CEC.2010.5586378>.
- [16] Ercan U. Acar, et al., Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods, *Int. J. Robot. Res.* 22 (7–8) (2003) 441–466, <https://doi.org/10.1177/02783649030227002>.
- [17] Timo Oksanen, Arto Visala, Coverage path planning algorithms for agricultural field machines, *J. Field Rob.* 26 (8) (2009) 651–668, <https://doi.org/10.1002/rob.20300>.
- [18] Adiyabaatar Janchiv, et al., Time-efficient and complete coverage path planning based on flow networks for multi-robots, *Int. J. Control. Autom. Syst.* 11 (2) (2013) 369–376, <https://doi.org/10.1007/s12555-011-0184-5>.
- [19] Enric Galceran, Marc Carreras, A survey on coverage path planning for robotics, *Robot. Auton. Syst.* 61 (12) (2013) 1258–1276, <https://doi.org/10.1016/j.robot.2013.09.004>.
- [20] Jerome Barraquand, Jean-Claude Latombe, Robot motion planning: A distributed representation approach, *Int. J. Robot. Res.* 10 (6) (1991) 628–649, <https://doi.org/10.1177/027836499101000604>.
- [21] Dugarjav Batsaikhan, Adiyabaatar Janchiv, Soon-Geul Lee, Sensor-based incremental boustrophedon decomposition for coverage path planning of a mobile robot, *Intelligent Autonomous Systems 12*, Springer, Berlin, Heidelberg, 2013, pp. 621–628, https://doi.org/10.1007/978-3-642-33926-4_59.
- [22] Amna Khan, et al., Online complete coverage path planning using two-way proximity search, *Intell. Serv. Robot.* 10 (3) (2017) 229–240, <https://doi.org/10.1007/s11370-017-0223-z>.
- [23] Ercan U. Acar, et al., Morse decompositions for coverage tasks, *Int. J. Robot. Res.* 21 (4) (2002) 331–344, <https://doi.org/10.1177/027836402320556359>.
- [24] Alexander Zelinsky, et al., Planning paths of complete coverage of an unstructured environment by a mobile robot, *Proceedings of International Conference on Advanced Robotics*, Vol. 13 1993 <http://pinkwink.kr/attachment/cfile3.uf@1354654A4E8945BD13FE77.pdf>, Accessed date: 19 September 2018.
- [25] Yi Guo, Qu. Zhihua, Coverage control for a mobile robot patrolling a dynamic and uncertain environment, *Intelligent Control and Automation*, 2004. WCICA 2004. Fifth World Congress on, Vol. 6 IEEE, 2004, <https://doi.org/10.1109/WCICA.2004.1343643>.
- [26] Chaomin Luo, Simon X. Yang, A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments, *IEEE Trans. Neural Netw.* 19 (7) (2008) 1279–1298, <https://doi.org/10.1109/TNN.2008.2000394>.
- [27] Young-Ho Choi, et al., Online complete coverage path planning for mobile robots based on linked spiral paths using constrained inverse distance transform, *Intelligent Robots and Systems*, 2009. IROS 2009. IEEE/RSJ International Conference on, IEEE, 2009, <https://doi.org/10.1109/IROS.2009.5354499>.
- [28] Tae-Kyeong Lee, et al., Complete coverage algorithm based on linked smooth spiral paths for mobile robots, *Control Automation Robotics & Vision (ICARCV)*, 2010 11th International Conference on, IEEE, 2010, <https://doi.org/10.1109/ICARCV.2010.5707264>.
- [29] Aydin Sipahioglu, et al., Energy constrained multi-robot sensor-based coverage path planning using capacitated arc routing approach, *Robot. Auton. Syst.* 58 (5) (2010) 529–538, <https://doi.org/10.1016/j.robot.2010.01.005>.
- [30] Li Caihong, et al., A complete coverage path planning algorithm for mobile robot based on FSM and rolling window approach in unknown environment, 2015 34th Chinese Control Conference (CCC), 2015, <https://doi.org/10.1109/ChiCC.2015.7260560>.
- [31] Herbert Freeman, Ruth Shapira, Determining the minimum-area encasing rectangle for an arbitrary closed curve, *Commun. ACM* 18 (7) (1975) 409–413, <https://doi.org/10.1145/360881.360919>.
- [32] Jianjun Ni, et al., A dynamic bioinspired neural network based real-time path planning method for autonomous underwater vehicles, *Comput. Intell. Neurosci.* 2017 (2017), <https://doi.org/10.1155/2017/9269742>.
- [33] Yi Guo, Mohanakrishnan Balakrishnan, Complete coverage control for non-holonomic mobile robots in dynamic environments, *Robotics and Automation*, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on, IEEE, 2006, <https://doi.org/10.1109/ROBOT.2006.1641952>.
- [34] Ross Girshick, et al., Rich feature hierarchies for accurate object detection and semantic segmentation, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014 https://www.cv-foundation.org/openaccess/content_cvpr_2014/papers/Girshick_Rich_Feature_Hierarchies_2014_CVPR_paper.pdf, Accessed date: 19 September 2018.
- [35] Ross Girshick, Fast r-cnn, *Proceedings of the IEEE International Conference on Computer Vision*, 2015 https://www.cv-foundation.org/openaccess/content_iccv_2015/papers/Girshick_Fast_R-CNN_ICCV_2015_paper.pdf, Accessed date: 19 September 2018.
- [36] Yan Tian, et al., Lane marking detection via deep convolutional neural network, *Neurocomputing* 280 (2018) 46–55, <https://doi.org/10.1016/j.neucom.2017.09.098>.
- [37] Xudong Sun, Pengcheng Wu, Steven C.H. Hoi, Face detection using deep learning: an improved faster RCNN approach, *Neurocomputing* 299 (2018) 42–50, <https://doi.org/10.1016/j.neucom.2018.03.030>.
- [38] Qi Fang, et al., Detecting non-hardhat-use by a deep learning method from far-field surveillance videos, *Autom. Constr.* 85 (2018) 1–9, <https://doi.org/10.1016/j.autcon.2017.09.018>.
- [39] Sean McMahon, et al., Multi-Modal Trip Hazard Affordance Detection on Construction Sites, arXiv preprint arXiv:1706.06718 (2017), <https://doi.org/10.1109/LRA.2017.2719763>.
- [40] Jordi Palacin, et al., Building a mobile robot for a floor-cleaning operation in domestic environments, *IEEE Trans. Instrum. Meas.* 53 (5) (2004) 1418–1424, <https://doi.org/10.1109/TIM.2004.834093>.
- [41] Richard Kershner, The number of circles covering a set, *Am. J. Math.* 61 (3) (1939) 665–671, <https://doi.org/10.2307/2371320>.
- [42] Yi Guo, Qu. Zhihua, Coverage control for a mobile robot patrolling a dynamic and uncertain environment, *Intelligent Control and Automation*, 2004. WCICA 2004. Fifth World Congress on, Vol. 6 IEEE, 2004, <https://doi.org/10.1109/WCICA.2004.1343643>.
- [43] Alan L. Hodgkin, Andrew F. Huxley, A quantitative description of membrane current and its application to conduction and excitation in nerve, *J. Physiol.* 117 (4) (1952) 500–544, <https://doi.org/10.1113/jphysiol.1952.sp004764>.
- [44] Yoav Gabriely, Elon Rimon, Spiral-stc: An on-line coverage algorithm of grid environments by a mobile robot, *Robotics and Automation*, 2002. Proceedings. ICRA'02. IEEE International Conference on, Vol. 1 IEEE, 2002, <https://doi.org/10.1109/ROBOT.2002.1013479>.
- [45] Enrique Gonzalez, et al., BSA: a complete coverage algorithm, *Robotics and Automation*, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, IEEE, 2005, <https://doi.org/10.1109/ROBOT.2005.1570413>.
- [46] Tzutalin, LabelImg. Git code, <https://github.com/tzutalin/labelimg>, (2015).
- [47] Yangqing Jia, et al., Caffe: Convolutional architecture for fast feature embedding, *Proceedings of the 22nd ACM international conference on Multimedia*, ACM, 2014, <https://dl.acm.org/citation.cfm?id=2654889>.
- [48] J.S. Oh, Y.H. Choi, J.B. Park, et al., Complete coverage navigation of cleaning robots using triangular-cell-based map[J], *IEEE Trans. Ind. Electron.* 51 (3) (2004) 718–726, <https://doi.org/10.1109/TIE.2004.825197>.
- [49] W. Wang, Z. Du, L. Sun, Dynamic load effect on tracked robot obstacle performance [C]/*Mechatronics, ICM2007 4th IEEE International Conference on IEEE*, 2007, pp. 1–6, <https://doi.org/10.1109/ICMECH.2007.4280041>.