

**REGULAR ARTICLE**

# An autonomous strawberry-harvesting robot: Design, development, integration, and field evaluation

Ya Xiong  | Yuanyue Ge | Lars Grimstad | Pål J. From

Faculty of Science and Technology, Norwegian University of Life Sciences, Ås, Norway

**Correspondence**

Ya Xiong, Faculty of Science and Technology, Norwegian University of Life Sciences, Ås 1433, Norway.

Email: yaxio@nmbu.no

**Funding information**

Norges Miljø- og Biovitenskapelige Universitet; Norwegian University of Life Sciences, Norway

**Abstract**

This paper presents an autonomous robot capable of picking strawberries continuously in polytunnels. Robotic harvesting in cluttered and unstructured environment remains a challenge. A novel obstacle-separation algorithm was proposed to enable the harvesting system to pick strawberries that are located in clusters. The algorithm uses the gripper to push aside surrounding leaves, strawberries, and other obstacles. We present the theoretical method to generate pushing paths based on the surrounding obstacles. In addition to manipulation, an improved vision system is more resilient to lighting variations, which was developed based on the modeling of color against light intensity. Further, a low-cost dual-arm system was developed with an optimized harvesting sequence that increases its efficiency and minimizes the risk of collision. Improvements were also made to the existing gripper to enable the robot to pick directly into a market punnet, thereby eliminating the need for repacking. During tests on a strawberry farm, the robots first-attempt success rate for picking partially surrounded or isolated strawberries ranged from 50% to 97.1%, depending on the growth situations. Upon an additional attempt, the pick success rate increased to a range of 75–100%. In the field tests, the system was not able to pick a target that was entirely surrounded by obstacles. This failure was attributed to limitations in the vision system as well as insufficient dexterity in the grippers. However, the picking speed improved upon previous systems, taking just 6.1 s for manipulation operation in the one-arm mode and 4.6 s in the two-arm mode.

**KEYWORDS**

active obstacle separation, agricultural robotics, cable-driven gripper, field evaluation, strawberry-harvesting robot

## 1 | INTRODUCTION

Strawberries (*Fragaria × ananassa* Duch.) are farmed extensively in most parts of the world, growing either outdoors in open fields or in controlled environments, like greenhouses or polytunnels. In 2016,

according to market research company IndexBox, the global strawberry market amounted to 9.2 million tons, increasing by 5% against the previous year. Strawberry production is heavily reliant on human labor, especially for harvesting (Xiong, Peng, Grimstad, From, & Isler, 2019). It was reported that 25% of all working hours in Japan

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2019 The Authors. *Journal of Field Robotics* Published by Wiley Periodicals, Inc.

are consumed by harvesting operations (Yamamoto, Hayashi, Yoshida, & Kobayashi, 2014). Strawberry producers in the Western world, particularly the United Kingdom and United States, are similarly concerned about the future availability of labor for picking, as well as about inflation in the cost of labor. In the United Kingdom, for example, the need is especially significant in the soft fruit sector, which uses 29,000 seasonal pickers to generate over 160,000 tons of fruit every year (British summer fruits seasonal labor report, 2017). In California, the cost of manual harvesting cost could be as much as 60% of production costs for fresh market strawberries (Anjom, Vougioukas, & Slaughter, 2018), which concurs with research conducted in Norway (Xiong et al., 2019). These dual labor challenges of shortages and high costs are, therefore, advancing developments in the automation of fruit harvesting operations.

Despite several attempts to develop a robotic solution for harvesting strawberries and many other crops, a fully viable commercial system has yet to be established (Silwal et al., 2017). One of the major challenges is that the robots need to be able to operate equally efficiently within diverse, unconstrained environments and crop variations with a variety of features (Bac, Hemming, & Van Henten, 2013; Silwal et al., 2017). A harvesting robot is generally a tightly integrated system, incorporating advanced features and functionalities from numerous fields, including navigation, perception, motion planning, and manipulation (Lehnert, McCool, Sa, & Perez, 2018). These robots are also required to operate at high speed, with high accuracy and robustness and at a low cost, all features that are especially challenging in unstructured environments, such as the strawberry farm utilized for testing in this paper.

Fruit harvesting offers significant opportunities for the field of agricultural robotics and has, thus, gained much attention in recent decades. Several robots have been developed for harvesting fruits and vegetables, including those for apples, sweet peppers, cucumbers, tomatoes, litchis, and strawberries. An apple robotic harvester was designed and evaluated with an overall success rate of 84% and an average picking time of 6.0 s per fruit; however, they encountered challenges, such as obstacle detection and avoidance (Silwal et al., 2017). A sweet pepper-harvesting robot achieved success rates of between 26% and 33% in a modified environment and a cycle time of 94 s for a full harvesting operation (Bac et al., 2017). Similarly, another sweet pepper-harvesting robot, named Harvey, achieved a 46% success rate for unmodified crops and 58% for modified crops, with average picking times of 35–40 s (Silwal et al., 2017). They reported that the most common detachment failure was that of the cutter missing either side of the peduncle. This team subsequently presented an improved version of Harvey, with a higher success rate of 76.5% in a modified scenario (Lehnert, McCool, et al., 2018). A harvesting robot was developed for greenhouse tomatoes, with a success rate of 86% and a picking speed of approximately 15 s per tomato (Lili et al., 2017); however, the literature provides no in-depth analysis of their failure cases. A study of cherry tomato harvesting robot reported a success rate of 83%, with an average 1.4 attempts for each successful picking and a time cost of 8 s for a single

successful harvesting excluding the time cost of moving between targets (Feng, Zou, Fan, Zhang, & Wang, 2018). The main failure found in the tests was collisions between the end-effector and the plant stems (Feng et al., 2018).

An increasing number of robots for autonomous strawberry picking have also been developed in recent few years. Japanese researchers developed and evaluated a strawberry-harvesting robot with a scissor-like cutter, which had a success rate of 34.9% and 41.3% when picking with suction and without suction, respectively (Hayashi et al., 2010). Their harvesting time for single fruit was 11.5 s. They concluded that a suction end-effector did not greatly contribute to picking performance and further reported that their failures were incorrect peduncle detection (Hayashi et al., 2010). The groups subsequent version of this strawberry-harvesting robot achieved a success rate of 54.9%. Another strawberry-harvesting robot using a 3D Cartesian-type arm was tested by its detection of the peduncle before picking target strawberries laid out on a laboratory surface (Cui, Gejima, Kobayashi, Hiyoshi, & Nagata, 2013). The system achieved a successful detection rate of 70.8% with a successful picking cycle time of 16.6 s per fruit, and the authors reported the main challenge for their work as peduncle detection (Cui et al., 2013). Unlike the abovementioned selective harvesting robots, researchers also proposed a strawberry harvester that shook the plants to detach fruits (Vakilian, Jafari, & Zarafshan, 2015). The focus of this study was mainly on the dynamics modeling and control. Aside from research in academia, a number of start-up companies have also recently developed several strawberry-harvesting robots, none of which have successfully commercialized. These include AGROBOT (Huelva, Spain), who used 24 independent picking systems mounted on a mobile base to increase efficiency, OCTINION (Leuven, Belgium), who designed a force-limit soft gripper in an attempt to avoid damage while grasping, and Harvest CROO (Florida) who designed a rotation apparatus that includes several grippers for picking strawberries on the ground. Generally, strawberry harvesting in cropping environment is very challenging. First, ripe strawberries are easily damaged and bruised (Dimeas, Sako, Moulitanitis, & Aspragathos, 2015; Hayashi et al., 2014; Xiong et al., 2019). This feature requires gentle handling during manipulation procedures. Noncontact picking might be an acceptable solution to avoid damage. Second, strawberries are small in size and tend to grow in clusters, which makes it difficult to identify and pick individual strawberries (Xiong et al., 2019; Yamamoto et al., 2014). Picking in clusters with dense obstacles is one of the main challenges for strawberry harvesting (Xiong et al., 2019; Yamamoto et al., 2014) as well as for many other crop harvesting systems, such as tomato harvesting (Yaguchi, Nagahama, Hasegawa, & Inaba, 2016) and sweet pepper robot (Bac et al., 2016).

In this paper, we address some of the challenges of working in unstructured farming environment. The main contributions of this paper are as follows:

- (1) A novel active obstacle-separation path-planning algorithm for cluster picking: The gripper can actively push aside the bottom

obstacles before swallowing and separate the top obstacles during swallowing. The pushing vectors are derived based on the surrounding obstacles that are calculated using downsampled blocks of 3D point cloud.

- (2) Improvements to the vision system, the gripper, the arm, and the control: An adaptive color thresholding for adaption of ambient changing light, a new feature of the gripper that can pick a market punnet and harvest berries straight into the container and a low-cost dual-arm system with optimized harvesting order.
- (3) A fully integrated harvesting system: The robot is able to pick strawberries continuously in polytunnels. Field evaluation shows the robot's first-attempt success rate for picking partially surrounded or isolated strawberries ranged from 50% to 97.1%, depending on the growth situations.

## 2 | RELATED WORK

### 2.1 | Fruit identification

Machine vision is an essential component for agricultural robots, enabling them to detect and localize the target crop. When the 3D position of a target is obtained, its coordinates can be further utilized to instruct the movements of the manipulation. For strawberry detection, image processing based on color thresholding is a frequently applied method in research papers (Hayashi et al., 2014; Yamamoto et al., 2014), primarily due to the significant differences of color among ripe strawberries, green strawberries, and green plants. Peduncle detection is another widely researched harvesting step (Cui et al., 2013; Hayashi et al., 2010; Huang, Wane, & Parsons, 2017; Shiigi et al., 2008). Color-based image processing methods were used to detect the strawberry first and then set a certain region above the strawberry for peduncle detection, with the accuracy influenced by the results of preprocessing and complexity of the environment. Other researchers have explored feature learning methods to analyze strawberry fruit shapes (Ishikawa et al., 2018). Recently, extensive work used deep learning as an approach for fruit detection. Deep learning, which can autonomously extract fruit features, has shown results in strawberry detection (Habaragamuwa et al., 2018). In addition to strawberries, deep learning, especially the Faster RCNN network, has been widely used for detection of many other fruits, including sweet pepper, mango, apple, almond, and kiwifruit (Fu et al., 2018; Mai, Zhang, & Meng, 2018; Sa et al., 2016; Zhang et al., 2019). All these systems used detection networks to generate bounding boxes around the target fruits.

Unstructured growing conditions, including variable clustering, occlusions, and varying lighting conditions, have been considered as the common challenges for fruit detection in farm environments (Silwal et al., 2017). Consequently, the focus of much ongoing research is novel ways to resolve these situations. One study proposed a color-based adaptive thresholding method for sweet pepper detection that can deal with changing illumination conditions (Vitzrabin & Edan, 2016), for example, while another proposed a

visual servoing-based method accurately localizes sweet peppers in occlusion situations (Lehnert, Tsai, Eriksson, & McCool, 2018). Deep learning is a promising method to deal with the lighting variations and the general idea is to capture and train images under different lighting conditions (Bargoti & Underwood, 2017; Fu et al., 2018). However, this method may require additional hardware (GPU) and a large data set as well as intensive work on image annotations, thus increasing the cost and power consumption.

### 2.2 | Mobile platform and navigation

Over the years, mobile platforms have been developed for a range of agricultural applications, from weeding (McCool et al., 2018), to high throughput phenotyping (Vijayarangan et al., 2017), to transportation (Ye et al., 2017). Some mobile robots are task-specific, meaning that they are specially designed for one particular application. Several task-specific mobile bases can be found in literature including the sweet pepper-harvesting robot (Lehnert, English, McCool, Tow, & Perez, 2017) and robots for phenotyping (Mueller-Sim, Jenkins, Abel, & Kantor, 2017). Task-specific mobile bases can also be found in various commercial projects, for example, the weeding robots created by companies like ecoRobotix and Franklin Robotics, and harvesting robots being developed by companies like AGROBOT or Harvest CROO Robotics. Other mobile robots are generic, designed to work with multiple, interchangeable implements, and can thus be used in several different applications. Examples include Bonirob by Bangert et al. (2013), and Robotti by commercial company Agointelli.

Most agricultural robots rely on a mobile base, that is, specifically designed for one type of environment. A mobile base designed for driving in tractor-sized tracks in open fields, for example, will normally not fit in a greenhouse. There is a lot of variation found in agriculture, and there may be large differences between farms, even if they grow the same crop. The mobile platform used in the current work is the Thorvald robot (Grimstad, Skattum, Solberg, Loureiro, & From, 2017). It is created from modules that may easily be reconfigured into robots of different sizes and shapes for different environments. A slim robot configured for greenhouses and polytunnels, such as the one used in this study, may quickly and easily be resized wide enough to fit within tractor tracks. The robots navigation system is different depending on project and application. In previous work we used techniques based on light detection and ranging (LIDAR) and cameras (Grimstad, Zakaria, Le, & From, 2018) as well as RTK-GPS (Grimstad et al., 2017). The navigation system used in polytunnels in current work is based on well-established techniques of probabilistic localization (Thrun, Burgard, & Fox, 2005) as well as the use of topological maps (Fentanes, Lacerda, Krajník, Hawes, & Hanheide, 2015). The navigation setup is briefly described in Section 4.

### 2.3 | End-effector and manipulation

Various end-effectors have been developed for strawberry-harvesting robots. The most widely used is the scissor-like end-effector for

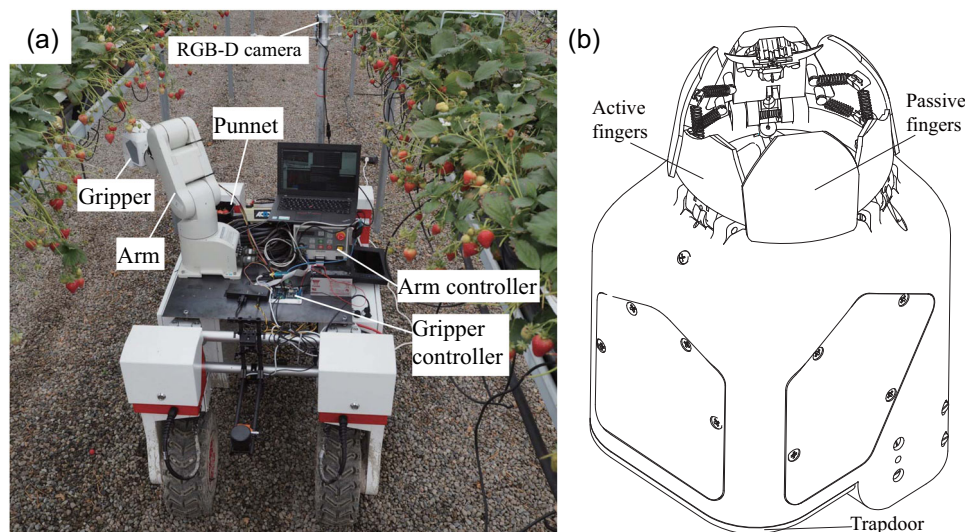
fruit detachment purpose only (Cui et al., 2013; Hayashi et al., 2014; Yamamoto et al., 2010; AGROBOT Ltd.; Dogtooth Technologies Ltd.). With an additional suction device, the scissor-like end-effector might be able to hold fruit (Feng, Wang, Zheng, Qiu, & Jiang, 2012; Hayashi et al., 2010). Contact grasping grippers are also common to see, such as the three-finger clamps with force-limit function (Dimeas et al., 2015) and two or more fingers with rotational motion to break peduncles (Yamamoto et al., 2014; OCTINION Ltd.). The scissor-like gripper requires more advanced vision system to detect the peduncle position and might unintentionally cut surrounding plants in clusters (Hayashi et al., 2010; Xiong et al., 2019). The grasping contact type grippers might easily bruise fragile strawberries (Hayashi et al., 2010).

Due to the uncertain environment, such as the presence of obstacles and clusters of fruits, manipulation is considered one of the main challenges in getting harvesting robots to become a reality (Lehnert, McCool, et al., 2018; Silwal et al., 2017). Cluster picking is difficult since the surrounding fruits, leaves, stems and other obstacles are difficult to separate from the target, both in detection and in manipulation. Similar to many other picking systems (Cui et al., 2013; Hayashi et al., 2014), our previous system used a point-to-point path-planning method to move the arm from a start point to a point underneath the target. However, with this method, it was difficult for the gripper to avoid swallowing below-hanging or surrounding berries, leaves, or stems along with its target berry. To avoid occlusions, a “3D-move-to-see” method was proposed to find the best view with less occlusions (Lehnert, Tsai, et al., 2018). To avoid obstacles, a method for cucumber picking was developed that uses a search algorithm to explore the search space for a feasible trajectory, in which each step of the trajectory is checked by a collision detector (Van Henten et al., 2002). Another work used a randomized path planner to generate a random path tree and then tested each path with a local path planner to determine the collision-free one for pruning grape vines (Botterill et al., 2017). Furthermore,

to avoid the arms self-collision or collision with obstacles, they incorporated a collision detector based on geometric primitives. Most of the methods found in the literature are passive obstacle avoidance methods, in which the aim is to avoid existing obstacles without changing the environment. However, obstacles are not always avoidable, especially when picking small-size fruits in clusters, where the obstacles may be extremely close to the targets.

## 2.4 | Previous work and challenges

In 2017, we developed the first version of a strawberry-harvesting robot and implemented a set of field experiments for performance evaluation (Xiong et al., 2019). As shown in Figure 1a, the robot hardware comprised four modules: (a) a cable-driven gripper attached to (b) a Mitsubishi five-degrees-of-freedom (5-DOF) serial arm, mounted on (c) the Thorvald platform (Grimstad et al., 2017) and (d) a stationary RGB-D camera facing one side of table-top grown strawberries. The fingers of the novel cable-driven gripper (Figure 1b) were able to separate surrounding berries out of the way and could open to form a closed space in which to swallow a target strawberry (Xiong, From, & Isler, 2018). Equipped with three internal infrared (IR) sensors, the gripper could sense and correct for positional errors. An integrated container was used for collecting picked strawberries, which reduced picking time significantly; however, this system necessitated repacking the strawberries into punnets for market. The vision system of this version used a color thresholding-based algorithm for object detection and localization; however, the thresholds needed to be changed manually according to the changing sunlight intensity. Furthermore, while the industrial arm was robust and convenient, it was not suitable for use in small working spaces, which limited its picking operation, and the systems significantly low baud rate made it unsuitable for closed-loop control. The robot could pick strawberries continuously without being integrated into the platform, which was moved with a joystick.



**FIGURE 1** The previous version of our strawberry picking robot: (a) The first version robot in a strawberry tunnel and (b) the cable-driven gripper with perception capabilities. Source: Xiong et al., 2018 [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

The evaluation results showed that the robot was capable of a 96.8% success rate when picking isolated strawberries, but it struggled when picking in clusters, resulting in a low success rate (53.6% without damage) in farm setting.

### 3 | OVERVIEW OF THE NEW SYSTEM

The autonomous strawberry-harvesting system described in this paper incorporates the lessons and addresses the challenges of the original robot, described above. The images and captions in Figure 2 provide an overview of the new robot, photographed during field testing on an English strawberry farm. As with the previous system, the hardware consists primarily of four modules, namely, an Intel R200 RGB-D depth camera, a newly developed single-rail dual-arm manipulator, two improved patented grippers, and a previously developed Thorvald platform. A Hokuyo LIDAR is mounted on the front of the robot for navigation sensing. The arm module is mounted horizontally on the platform for picking strawberries along one side of the table-top trays. The stationary RGB-D camera faces the same side for strawberry detection and localization. An additional one-axis punnet station, attached to the left side of the platform, lifts up to enable the grippers to pick or release the punnets and returns to its lower position once the operation is complete. The punnet station uses the same motor and control system as the arm system. Electronics are placed on the rear of the robot. These include a gripper controller, a CAN to USB convertor for the arm, a DC 48 to 12 V power convertor, and a power switch. All power is provided by the Thorvald battery, which supports approximately

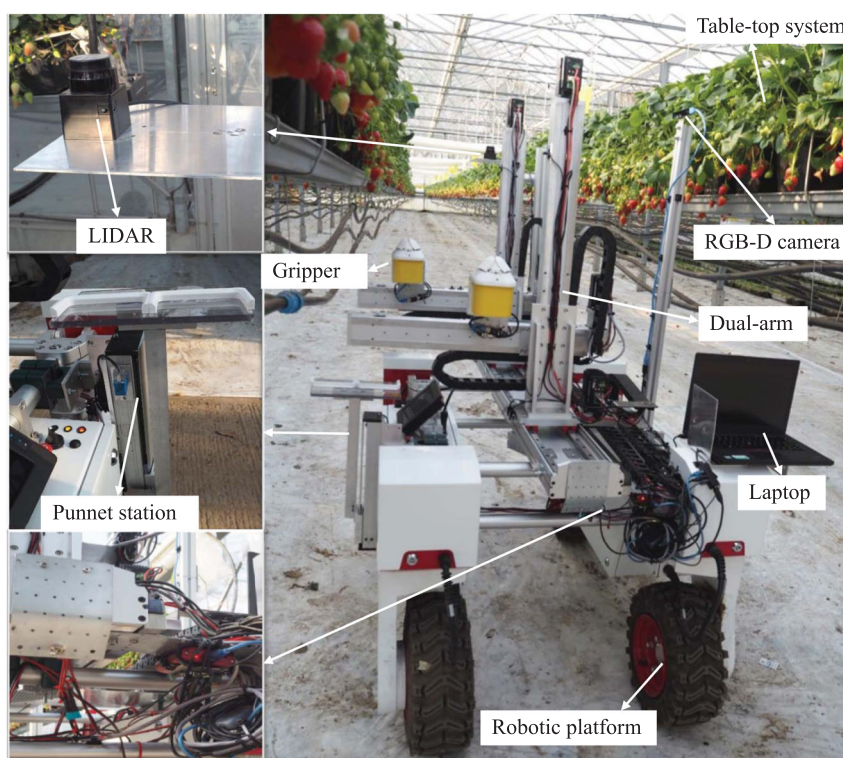
48 hr of continuous picking. All of the components are connected to a laptop (Intel i5-6700 CPU and 16 GB RAM), including the robotic platform, thus simplifying communication. The entire system is fully integrated into the robot operating system (ROS).

### 4 | NAVIGATION IN TABLE-TOP FARMS

Table-top systems are commonly used in polytunnels and greenhouses. Several different systems exist, and there is therefore a great deal of variation between farms in terms of infrastructure. Some farms have tables mounted on poles in the ground, while others suspend their tables from the ceiling. The spacing between rows, as well as the overall layout of the tunnels or greenhouses, also varies between farms. As the navigation system for the mobile base used in this study was tested in a polytunnel with table-tops mounted on poles in the ground, it is this type of environment that is discussed here.

The mobile robot was assembled using modules from the previously developed Thorvald II modular system, described above. The robot has four-wheel drive and four-wheel steering, which enables it to move in any direction, and also turn in place, thus substantially increasing its ability to navigate tight spaces. The system is fitted with a Hokuyo UTM-30LX-EW 2D LIDAR and an Xsens MTi-30 IMU; however, the latter was not used for the purposes of this paper.

In addition to tens, or even hundreds of polytunnels on a strawberry farm, there are several other points of interest for a robot, including charging stations and cold storage units for harvested fruit. Therefore, to simplify the task of navigating this



**FIGURE 2** Hardware assembly of the new strawberry-harvesting robot in a strawberry greenhouse: The robot consists mainly of a RGB-D camera, a single-rail dual-arm manipulator, two grippers, and a mobile platform [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

type of complex environment, a topological navigation system is employed (Fentanes et al., 2015). Here, a metric map is used together with a graph. Nodes in the graph represent goals or gateways found in the polytunnel, while edges represent navigable paths between two nodes. The robot can only move between connected nodes. When provided with a goal, the robot will find a connected set of nodes to the goal node, and move through these nodes to reach its target. Different actions for moving the mobile robot can be defined for the different edges. For example, if the robot needs to dock at a charging station, a special action for accurate docking may be required. Moreover, different behavior may be required when the robot is moving in an open space, compared to when it is driving inside a tunnel row where movement is far more constrained. Edges between two nodes can be defined as either unidirectional or bidirectional, enabling operators to enforce one-way traffic where necessary, specifying that a robot may, for example, move from Node A to Node B, but not from Node B to Node A.

Encoder-based velocity estimates are used together with data from the 2D LIDAR to create a map using the GMapping simultaneous localization and mapping (SLAM) technique (Grisetti, Stachniss, & Burgard, 2007). During this process the robot is teleoperated. The resulting map is stored and used by the robot during autonomous operation. The robot uses the map, LIDAR data, and encoder-based odometry to localize in the tunnel. A copy of the map is altered to mark out areas where the robot is not allowed to drive and the robots global costmap is generated from this no-go map. This prevents the robot from planning paths through certain areas, such as underneath the table trays between rows.

The robots navigation system was tested in a polytunnel at a research farm. As such, the size of the topological map presented here (Figure 3) is somewhat limited; however, the principles are equally applicable to larger polytunnel environments. First, we defined the topological nodes on either sides of four rows in the tunnel, as well as intermediate nodes inside these rows. A node representing the robots charging station and a few gateway nodes between the charging station and the tunnel rows were further defined. Unidirectional edges (for one-way driving) was defined for inside two of the rows, and the remaining two rows were defined as bidirectional edges. Possible actions for moving the robot along the edges were specified as simple waypoint navigation, with either forward drive, sideways drive, or reverse drive, as well as a dynamic window approach for navigation around unforeseen

obstacles. For edges inside the rows, only simple waypoint navigation was used, with no planning around unforeseen obstacles (if an obstacle appears, the robot will simply stop and wait until the obstacle is moved). A reverse action was specified for the edge going to the charging station, while the robot would use forward drive along the edge moving away from the charging station. Edges between rows were specified as either forward drive or sideways drive.

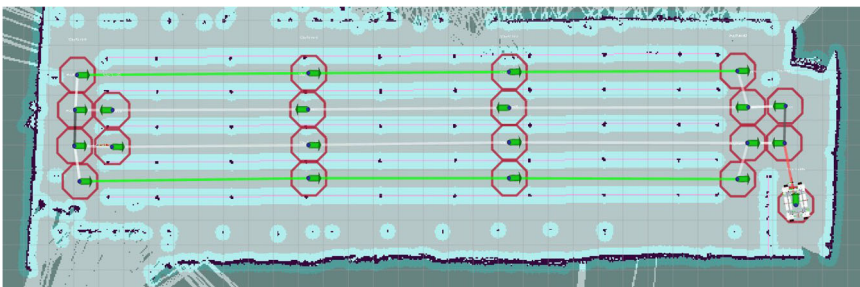
Using this system, the mobile robot was able to successfully navigate the somewhat cramped environment inside the polytunnel. In the supplementary materials, the robot can be seen navigating the tunnel, starting at the node representing the charging station and then driving once through all four rows before returning to park at the charging station.

## 5 | ENVIRONMENT ADAPTIVE MACHINE VISION

### 5.1 | Motivation

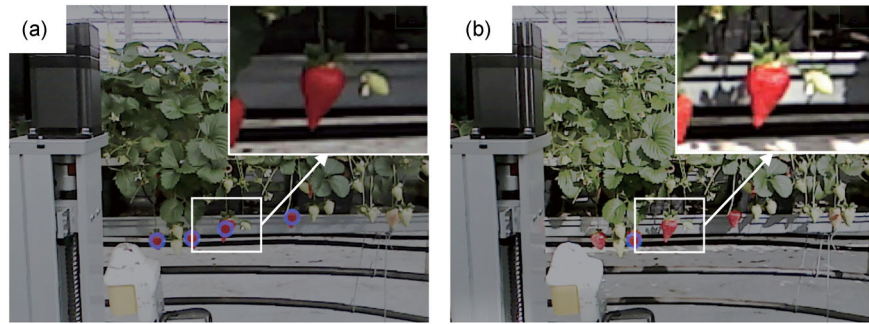
As with many other field machines (Bac et al., 2017; Hayashi et al., 2014), a color-based algorithm was utilized in this system to take advantage of color differences and retain a fast processing speed. Hue saturation value (HSV) images are transformed from the RGB images and used for image processing. The aim with this machine vision subsystem is to detect and localize ripe strawberries and to pass the detected strawberry bounding boxes to the other subsystems.

Changing ambient illumination in the field is a challenge for image processing. During the experiments, it was found that changes in the available sunlight significantly influenced the detection results. As shown in Figure 4, Figure 4a displays situations with weak light intensity, while Figure 4b shows much stronger light intensity in the same place. As a result, four strawberries were detected in Figure 4a (blue circles) but only one in Figure 4b, with the same thresholds. This problem was also pointed out by Hayashi et al. (2014), who subsequently adjusted the thresholds on their system manually on the farm, as with our previous system. Raja et al. (1998) proposed a statistical approach, in which light intensity was estimated over time, while other researchers have investigated how robots can learn to adapt to various lighting conditions (Sridharan & Stone, 2007). In this paper, we propose a modeling-based technique for automatic updating thresholds by using the grippers IR sensor.



**FIGURE 3** Topological map for driving in a polytunnel. Here, the robot can be seen parked in the lower right corner [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

**FIGURE 4** Two set of images capturing the same area with different light intensities: (a) shows low sunlight intensity, in which four strawberries were detected (in blue circles) and (b) shows high sunlight intensity, in which only one berry was identified, despite having the same threshold as (a) [Color figure can be viewed at wileyonlinelibrary.com]

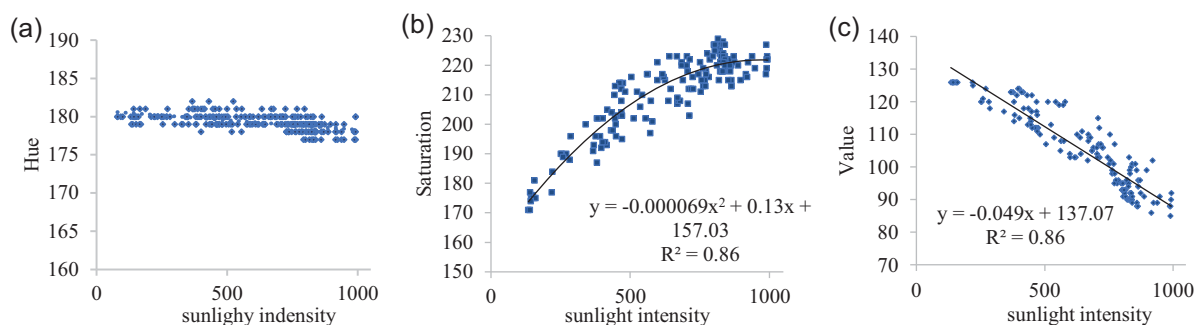


## 5.2 | Light intensity modeling and adaptive color thresholding

The robots gripper has internal IR sensors that can sense pure sunlight IR light intensity when the emitter light-emitting diodes are turned off in a mouth-open configuration (Figure 1b). Therefore, the HSV in the region of a ripe strawberry and the gripper IR value were recorded for almost an entire day. A sufficient amount of data about the range of sunlight intensities and corresponding HSV values is essential to ascertain the relationship between them. In total, 243 sets of data were recorded with various sunlight intensities and the corresponding HSV values of the ripe strawberry. To determine the connection between the values, the HSV data were analyzed independently, as can be seen in Figure 5. Here, hue range is from 0 to 179 (circular) in OpenCV and the value for pure red is 0. To clearly see the dependent relationship in the coordinate system, data around 0 were added by 179 to obtain Figure 5a. As the variations in the range of hue are relatively small, at around 5, the interaction is not significant and, therefore, it can be concluded that light density has a low influence on the hue for strawberry detection. However, based on the data in Figure 5b,c, it is clear that saturation and value change regularly with the variances in sunlight intensity. The correlation equations of saturation-sunlight intensity and value-sunlight intensity can thus be concluded as follows:

$$y_s = -0.00069x^2 + 0.13x + 157.03 (R^2 = 0.86), \quad (1)$$

$$y_{value} = -0.049x + 137.07 (R^2 = 0.86). \quad (2)$$



**FIGURE 5** Modeling of HSV and sunlight intensity: (a) Hue to sunlight intensity, no significant interactions; (b) saturation to sunlight intensity, significant quadratic relationship; and (c) value to sunlight intensity, significant linear relationship. HSV, hue saturation value [Color figure can be viewed at wileyonlinelibrary.com]

In application, the above models would be recorded in the codes. The gripper would detect the real-time sunlight intensity at the beginning of each image frame for every picking circle. The image processing algorithm would then set the saturation and value thresholds within ranges based on the detected sunlight intensity according to the correlation equations, thus forming an adaptive color threshold. After the basic color-thresholding process, the strawberry image would go through a series of postprocessing based on erosion and dilation, as previously described by the authors (Xiong, Ge, Liang, & Blackmore, 2017). During this processing, two commonly connected strawberries can be segmented. Once all the strawberries have been detected, their coordinates would be transferred to the gripper frame according the calibrated extrinsic parameters.

## 6 | SINGLE-RAIL MULTIPLE CARTESIAN ARMS

In the authors' previous strawberry-harvesting system, a Mitsubishi serial arm (RV-2AJ) with 5-DOF was employed, which was robust in terms of control and communication (Xiong et al., 2018). However, the high cost of the industrial arm is not appropriate for application in commercial farming robots, especially when multiple manipulators are required to optimize the harvesting efficiency. Moreover, in the previous system, the orientation of the 5-DOF arm was locked to keep the gripper horizontal, which also made its working space small,

however, in the system, the gripper is designed so that its workspace is strictly Cartesian, with no rotations needed, and, therefore, a 3-DOF Cartesian arm is sufficient to generate this motion. The Cartesian arm is widely used due to its simplicity and low cost. Moreover, unlike the serial arm, it has no singularity problem and it has a wider working area if no rotations are required. In their development of harvesting robotics, researchers have developed a 3-DOF Cartesian-type arm for strawberry picking (Cui et al., 2013) as well as an algorithm to plan the movements of multiple (Zion et al., 2014), independently functioning 3-DOF Cartesian arms for crop harvesting, mounted in backward–forward positions on the platform. In this current system, to mitigate cost and complexity, a low-cost single-rail-based Cartesian-type multiarm system was developed.

## 6.1 | Arm design and hardware

Figure 6 shows the concept design and the prototype of the proposed arm. In Figure 6a, the three arms have independent y-axis and z-axis rails, mounted on a common x-axis rail. The vertical z-axis rail uses ball-screw transmission for lifting heavy loads, while the y-axis uses a belt transmission for fast movement. The pinion-rack helical gear transmission between the x-axis rail and the y/z-axis rails enables the arms to have independent movement on the x-axis. Compared to a system with several independent arms, the single-rail multiarm system has three key advantages for harvesting robots: (a) two or more arms can be mounted on the same rail so that the transformation between the arm frame and the camera frame need only be calibrated once; (b) there is no unreachable space between the arms; (c) the cost is reduced as fewer parts are required and the time required for platform mounting is also reduced.

Figure 6b shows the assembly prototype of the proposed arm, which has two arms mounted on the x-axis rail. The arm rails

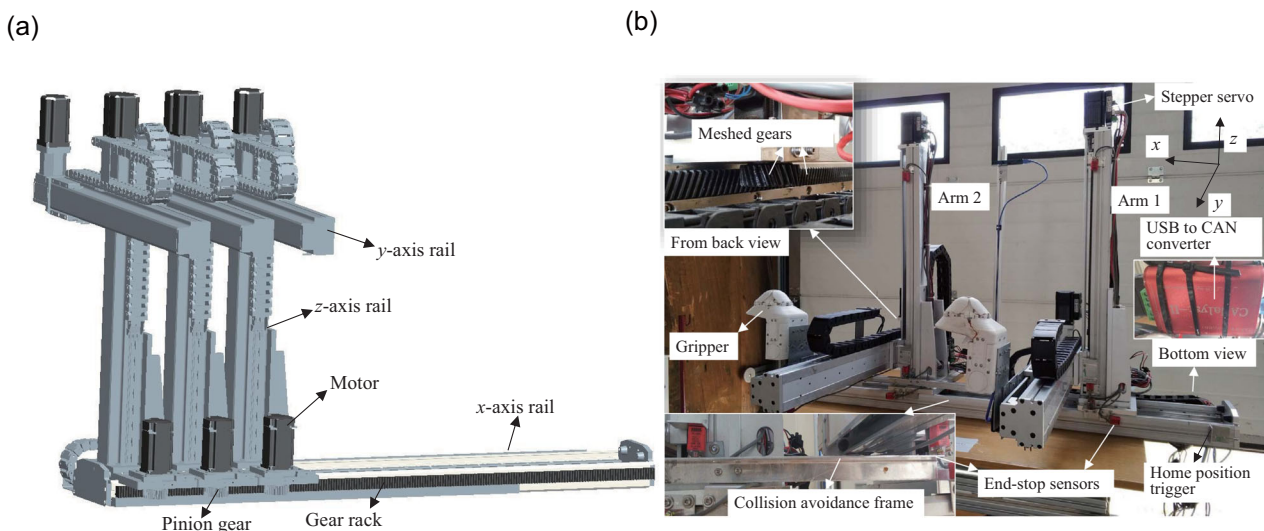
**TABLE 1** Key specifications of the single-rail dual Cartesian arms

Parameters	Value
Dimensions (mm)	1,400 × 900 × (930–1,080)
x-axis stroke (mm)	1,200
y-axis stroke (mm)	500
z-axis stroke (mm)	500
Max velocity (mm/s)	600
Approximate payload (kg)	10
Communication type	CANbus
Operating system	Linux
Control method	ROS topics
Input voltage (V)	48
Rated power (W)	900
Approximate weight (kg)	120

Abbreviation: ROS: robot operating system.

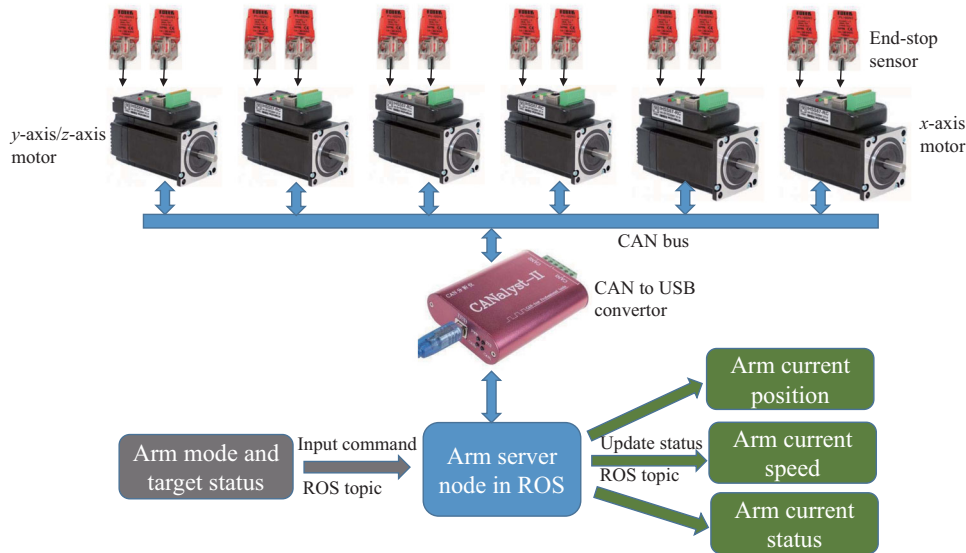
structure was manufactured by the GaoGong Intelligence Mechanical Drive Co., Ltd., China. PL-05N/2 inductive proximity sensors were used as end stops for homing the arms and limiting their movement range. A collision avoidance frame was mounted on Arm 2 that will trigger the end-stop sensor on Arm 1 when the arms are close, so as to avoid any mechanical collision. Stepper servo motors (Shenzhen Just Motion Co., Ltd., China) were selected as they are low in cost and deliver precise position control. These motors have integrated encoders and controllers and can be easily communicated via a CANbus network. A CAN to USB converter was used to enable the computer to access the CANbus network.

Table 1 describes the key specifications of the developed arms. The axis strokes and dimensions were determined by estimating the required picking space in the strawberry tunnels.



**FIGURE 6** Single-rail Cartesian-type multiarm: (a) 3D model shows that the three Cartesian arms move on a single rail (x-axis) using pinion-rack gear transmission; the single-rail (x-axis) could be mounted with two or more arms; (b) the prototype of a dual-arm system; the single-rail multiarm only needs to be calibrated once to identify the transformation between the arm frame and the camera frame [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]



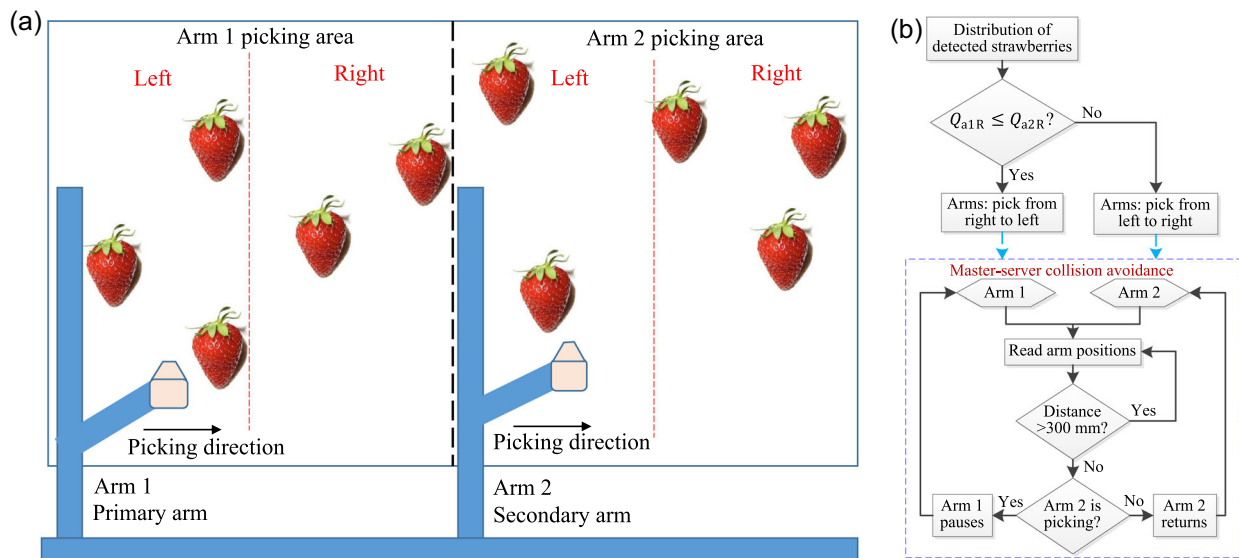


**FIGURE 7** Control architecture of the single-rail dual Cartesian arms: All the stepper servos are connected to the CANbus network; a ROS servo node is built to receive target mode and status commands from the user nodes and then control the motors by using CiA 402 protocol; the servo node can also get motor information and update the arm position, speed and status in 40 Hz, which can be used for feedback control. ROS, robot operating system [Color figure can be viewed at wileyonlinelibrary.com]

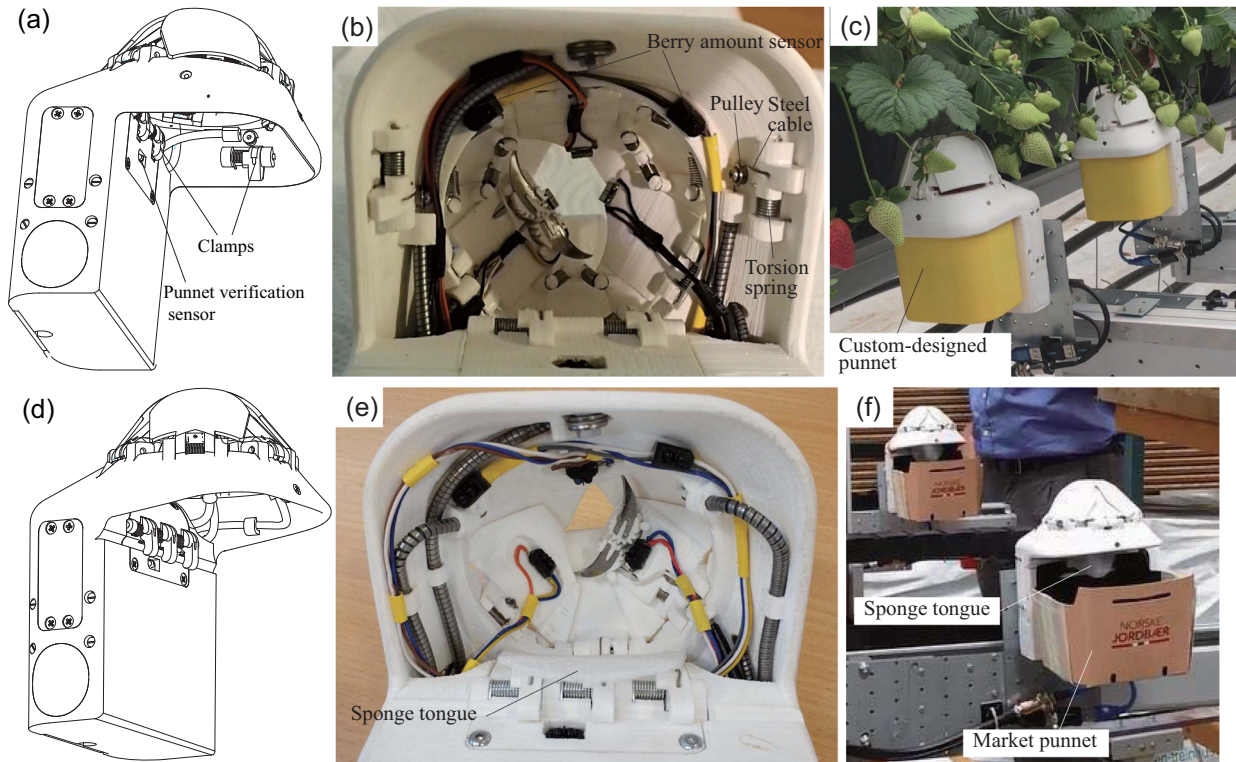
**6.2 | Arm control**

Figure 7 indicates the control architecture of the single-rail dual Cartesian arms. The stepper servo is written with CiA (CAN in Automation) 402 motion control protocol. All motors are connected to the host computer through a CAN to USB converter via the CANbus network. To modularize the arm system, an arm server node in ROS was built as a coordinator between user nodes and arms. For input, user nodes can simply send an arm target mode together with the status command to the server node via the ROS topics.

The server node will then decode and encode these commands to control the individual motors. Six modes were established on the basis of the harvesting robots requirements: home, pause, continue, position control, unblock, and reset. The status command includes the arm target positions and the moving speed. Acceleration and deceleration can be automatically adjusted according to the required speed in the server node. Furthermore, the server node can also output the arms status as ROS topics in 40 Hz by reading the motor status. The output data topics include the arms current position, speed, and status, which can be used as feedback control information.



**FIGURE 8** Harvesting order planning: (a) Schematic of harvesting order planning, with the picking area divided into two sections, namely, for Arm 1 and Arm 2; the picking area of each has been further separated into two subsections (left and right); if strawberries are uniformly distributed, the two arms systematically pick strawberries from left to right; if they are not uniformly distributed, strategies must be specified to increase picking efficiency and avoid collision; (b) the control algorithm for planning the harvesting order as well as for collision avoidance [Color figure can be viewed at wileyonlinelibrary.com]



**FIGURE 9** Improved gripper design: (a) Sketch of the version 2.1 gripper; (b) bottom view of version 2.1 prototype; three additional IR sensors are used to detect the punnet and estimate the amount of strawberries in it; (c) attaching 3D-printed punnets during picking; (d) sketch of version 2.2 gripper; (e) bottom view of version 2.2 prototype; a sponge tongue is used to reduce impact; (f) market punnets are attached for picking. IR, infrared [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

The arm current status topic publishes information such as whether the arm has reached the target position or not and its current running status (normal or faulty).

### 6.3 | Collision avoidance and harvesting order planning

One challenge that the single-rail dual-arm system presents is the effective cooperation between the two arms to increase picking efficiency and avoid collision with each other. In this harvester, as illustrated in Figure 8a, the picking area in the cameras view is divided equally into sections for Arm 1 (the primary arm) and Arm 2 (the secondary arm). Each section is then further separated into two subsections: left and right. Within these, one requirement is that the robot should pick berries from the bottom of the tray working upwards to the top. This is because the gripper is designed to pick from below. If the upper strawberry is picked first, the gripper and arm might touch and move the lower ripe strawberries, which may result in failure when picking the lower targets.

Figure 8b shows the control algorithm for planning the harvesting order as well as for collision avoidance. In the process of inputting the detected strawberries to the algorithm, the first aim is to determine the picking sequence for the arms to maximize the simultaneous picking period of both arms and avoid their possible collisions. The default picking sequence for both arms is from the left

subsection to the right subsection. However, if the quantity of strawberries in the right subsection of Arm 1 ( $Q_{a1R}$ ) is equal to or less than that in the right subsection of Arm 2 ( $Q_{a2R}$ ), it is better that both arms pick strawberries from right to left, since Arm 1 will finish picking in its right subsection and move across to the left subsection earlier than Arm 2 will complete its operation. Similarly, when the quantity of strawberries in the Arm 1 left subsection ( $Q_{a1L}$ ) is equal to or more than that in the Arm 2 left subsection ( $Q_{a2L}$ ), the distance between the two arms is always larger than the safety distance if they all pick from left to right. Theoretically, then, in this particular case, the distance between the two arms will always be greater than one subsection width (normally around 300 mm, which is equal to the safety distance between the two arms); however this has not been tested in other situations and, therefore, the distance between the two arms on this system cannot be guaranteed to be within the safety range.

In Figure 8b, a primary-secondary method is proposed to control the arms within a closed loop to avoid collisions. Arm 1 is the primary arm, and has picking priority, while Arm 2 acts as the secondary arm. The distance between the two arms is compared in real-time so that, should they come within the safety distance, the secondary arm will return to a safety position provided it is not in picking status and will wait until the primary arm finishes and moves away. However, if the secondary arm is in picking status and close to the primary arms target, the primary arm will not interrupt current picking but will wait

until the secondary arm completes its operation. Thereafter, the picking priority of the primary arm will be restored, so the secondary arm will return to a safety position if the next target is not within safety range.

## 7 | IMPROVEMENTS ON GRIPPER DESIGN

### 7.1 | Punnet picking and releasing

The previous version of our gripper included an integrated container for collecting picked strawberries, a feature that could reduce picking time as the arm does not need to move to store each picked strawberry. However, collecting with the container and repacking to the punnets could increase the risk of damage and, therefore, to avoid repacking, the gripper in the new system is designed to pick a punnet and harvest directly into it, as shown in Figure 9. Figure 9a–c shows the design, prototype, and field application of the version 2.1 gripper that was used in the field tests. In this version, instead of a container, the gripper has a hollow space under its fingers to attach a custom-designed punnet, which it picks from the punnet station (Figure 2). Four cable-driven clamps, distributed on three sides of the gripper, are used for picking and holding the punnet. These clamps are opened simultaneously by a servo motor and closed by torsion springs. In addition, an IR sensor placed under the front-side clamps is used to verify the attachment of the punnet. Another two IR sensors, mounted on the bottom of the fingers, are used to estimate the amount of strawberries in the punnet. The IR sensors detect the distance between the obstacle (strawberries in the punnet) and the sensor, which is changed during collection. Once the desired amount of strawberries has filled the punnet, the gripper returns it to the same location on the punnet station. A punnet transportation system is required for stocking empty punnets and collecting full punnets.

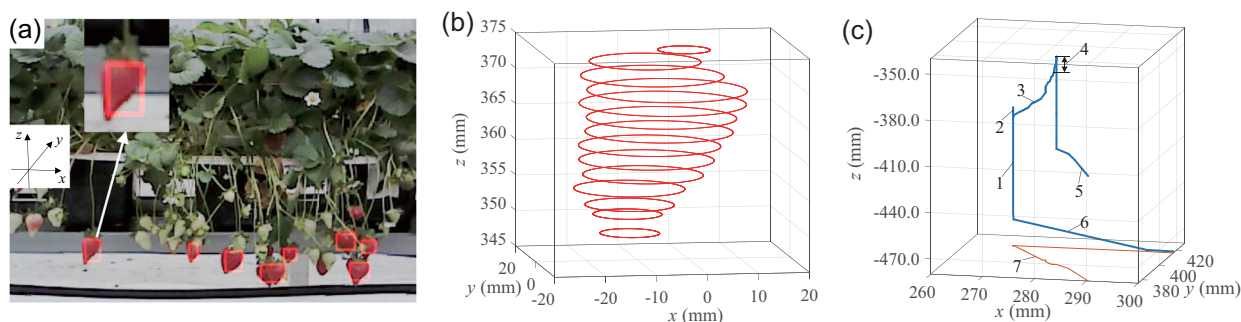
A further improved version (2.2) of the gripper was also subsequently developed, as shown in Figure 9d–f. This version of the gripper can pick a market punnet directly, as shown in Figure 9f, in which a new Norwegian standard strawberry punnet is attached. Instead of having four clamps on three sides, this version only has

three clamps on the front side so that the gripper can successfully pick various punnet shapes. Under the clamps, there is a groove for fitting the convex edge of some market punnets. Moreover, a sponge tongue is mounted on the top of the clamps, which can reduce impact significantly. The clamp module is independent of the gripper body, so it can be easily mounted and replaced.

### 7.2 | Scanning control

In the previous system, the gripper could control the arm using internal IR sensors to correct any positional errors. When targeting a strawberry, the gripper moved to the bottom of the target and used a slow lifting speed in search of the strawberry. Once detected, the arm stopped lifting and moved horizontally to place the gripper at the optimal cutting position based on one located section of the strawberry. This method works well when strawberries are growing vertical to the ground, however, if a strawberry is inclined towards the ground, as per the example shown in Figure 10a (enlarged red berry), one section located on the lower part of the strawberry might be different to another, upper section. This would affect the systems estimation of the location of the peduncle, which it requires for picking. To overcome this challenge, a scanning control method was used in the current system, in which the arm lifts and simultaneously moves the gripper in a horizontal plan to scan the shape of the strawberry. The gripper uses IR sensors to estimate the diameter and centroid of a circular section of an object. Without the gripper moving horizontally, the scanner can even reconstruct the shape and orientation of the strawberry, as is shown in the scan examples in Figure 10a,b.

Figure 10c shows the scanning control path of the gripper for picking the target enlarged in Figure 10a. First, the arm moves quickly to the bottom of the target and lifts the gripper slowly to search for the strawberry. Due to inertia, the arm is not able to come to an abrupt stop once the target has been located, but it will return to the first detection point, so there is an overshoot path. The gripper then uses the scanning control method to control the arm path according to the target strawberry's shape. When the strawberry is



**FIGURE 10** Scanning control: (a) An example of detected strawberries, in which the left strawberry (enlarged in image) is inclined to the ground; (b) reconstruction of the shape and orientation information from the grippers scan of the enlarged strawberry in Figure 8a; (c) scanning control path of the gripper for picking the enlarged target in Figure 8a; number represents: 1—searching path, 2—overshoot path, 3—scanning control path, 4—peduncle length adjustment, 5—return path, 6—forward path, and 7—trajectory projection [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

out of gripper sensing range, the arm moves up quickly with a desired peduncle length compensation value to control the peduncle length that remains on the strawberry after picking. Compared to the previous system, which used properties in the strawberry shape to estimate the peduncle length, this method is robust to the differences in shape and variety, however it does require an increased computation resource.

## 8 | A NOVEL ACTIVE OBSTACLE-SEPARATION PATH-PLANNING STRATEGY FOR CLUSTER PICKING

The previous system achieved a high success rate for picking isolated strawberries, however, it struggled with cluster picking. In this current work, inspired by human pickers who usually use their hands to push and separate surrounding obstacles during picking, we propose a novel active obstacle-separation path-planning strategy, using the gripper to push obstacles in the clusters.

### 8.1 | Algorithm

#### 8.1.1 | Scanning control

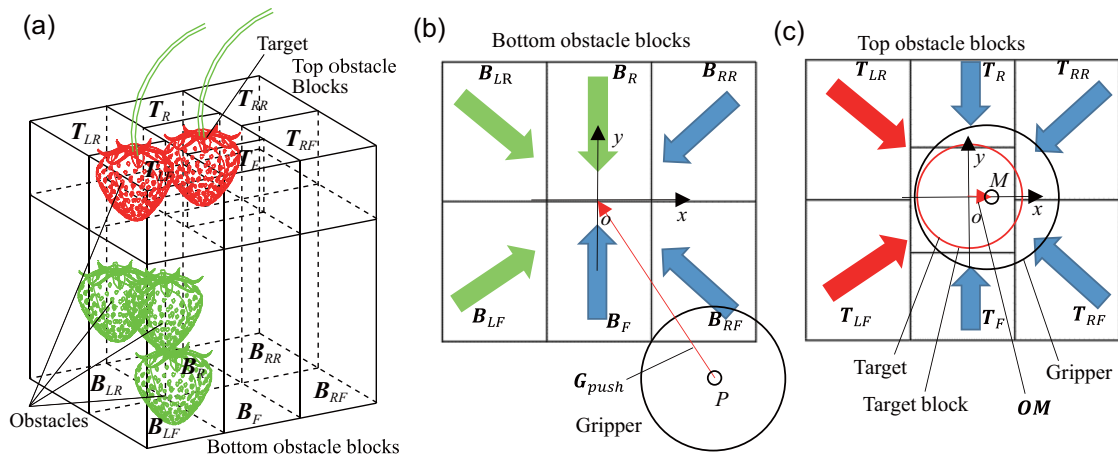
The first step for obstacle separation or avoidance is to obtain a 3D image of the area. In the current system, a single-view image is used to create the 3D point clouds based on a combination of depth and RGB images. To reduce computation costs, this system extracts and focuses only on the obstacles closest to the target, or region of interest (ROI) obstacles. As illustrated in Figure 11a, the ROI obstacles are those that are located on the bottom or at the same height as the target. The obstacles above the target are irrelevant since the gripper picks from below and, when it is swallowing the target strawberry, they will not affect fruit detachment. Unlike Bac et al. (2013) who classified obstacles into hard and soft types with more efforts from the vision side but did not get significant results, we simply use the quantity of 3D points within the

ROI to determine obstacles without further classification. Our goal is to gently separate all pushable obstacles, similar to human picking. The only nonmovable obstacle is the table-top system, which can be avoided by screening of distance.

ROI obstacles are divided into two main sections: top obstacle blocks and bottom obstacle blocks. Both top and bottom obstacle sections have been further separated into six subsections, based on their directions: left front, left rear, front, rear, right front, and right rear, respectively. The bottom blocks are used to guide the gripper when pushing obstacles aside before reaching its target while the top blocks assist the gripper in avoiding neighboring obstacles. The block size is mainly determined by the bounding box size of the detected target in the vision system. Among the top blocks, the front and rear have the same dimensions as the target block, while the length of the four left and right blocks is 1.5 times that of the target block. The length and width of the bottom blocks are the same as the top left and right blocks, but their height is three times that of the target block.

The two obstacle-separation actions can be described as either pushing aside the bottom obstacles before swallowing or pushing aside the top obstacles during swallowing. The operations are as follows: First, the gripper travels from a start point,  $S$ , to an intermediate point,  $P$ , that is next to the bottom blocks with the same height, as shown in Figure 11b. The gripper then uses the outside of its fingers to push the bottom obstacles from  $P$  to the origin,  $O$ , of the Oxy frame in the bottom blocks. This pushing path can be defined as a vector,  $\mathbf{G}_{push}$ . Block vectors ( $\mathbf{B}_{LF}$ , ...,  $\mathbf{B}_{LR}$ ) are used to describe the obstacles and calculate the pushing vector,  $\mathbf{G}_{push}$ . If obstacles are founded within a block, the vector in the block is labeled a unit vector; otherwise the empty block has a zero vector. Currently, the threshold of 3D point quantity for being an obstacle or an empty block is 50. The direction of the vector is determined by the block location. They all face towards the origin of the coordinate system, either vertical or at 45 degree to the  $x$  axis.  $\mathbf{G}_{push}$  can be expressed as

$$\mathbf{A} = \mathbf{B}_{LF} + \mathbf{B}_{LR} + \mathbf{B}_F + \mathbf{B}_R + \mathbf{B}_{RF} + \mathbf{B}_{RR}, \quad (3)$$



**FIGURE 11** Active obstacle-separation algorithm: (a) Schematic of ROI obstacle blocks; (b) top view of bottom obstacle blocks; and (c) top view of top obstacle blocks. ROI, region of interest [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

$$j = |\mathbf{B}_{LF}| + |\mathbf{B}_{LR}| + |\mathbf{B}_F| + |\mathbf{B}_R| + |\mathbf{B}_{RF}| + |\mathbf{B}_{RR}|, \quad (4)$$

$$\begin{aligned} \mathbf{C} = & (2 - |\mathbf{B}_{LF}| - |\mathbf{B}_{RR}|)(\mathbf{B}_{LF}' - \mathbf{B}_{RR}') + (2 - |\mathbf{B}_F| - |\mathbf{B}_R|)(\mathbf{B}_F' - \mathbf{B}_R') \\ & + (2 - |\mathbf{B}_{RF}| - |\mathbf{B}_{LR}|)(\mathbf{B}_{RF}' - \mathbf{B}_{LR}'), \end{aligned} \quad (5)$$

$$\mathbf{G}_{push} = \begin{cases} -r\mathbf{A} / |\mathbf{A}|, & \mathbf{A} \neq 0, \\ \pm r\mathbf{C} / |\mathbf{C}|, & \mathbf{A} = 0 \text{ and } j \neq 0, \\ 0, & \mathbf{A} = 0 \text{ and } j = 0, \end{cases} \quad (6)$$

where  $\mathbf{B}_{LF}'$ , ...,  $\mathbf{B}_{LR}'$  represents the unit vector that has the same direction as  $\mathbf{B}_{LF}$ , ...,  $\mathbf{B}_{LR}$ .  $r$  is the parameter used to scale the  $\mathbf{G}_{push}$  norm, which is 50 mm for the current system. Two opposite vectors, for example  $\mathbf{B}_{LF}$  and  $\mathbf{B}_{RR}$ , will cancel each other out if they both represent obstacles. If not all vectors are opposite and cancel each other ( $\mathbf{A} \neq 0$ ), the  $\mathbf{G}_{push}$  only needs to be obtained using  $\mathbf{A}$  and  $r$ .  $j$  is thus used to determine whether all of the vectors cancel each other out. In the case of  $j \neq 0$  and  $\mathbf{A} = 0$  (e.g., if only  $\mathbf{B}_F$  and  $\mathbf{B}_R$  have obstacles), the direction of  $\mathbf{G}_{push}$  is then decided by the distance between  $S$  and two possible points  $P$  (left or right of  $\mathbf{B}_F$  and  $\mathbf{B}_R$ ). The smaller distance point  $P$  is selected as it is the shorter traveling time for the gripper. If no obstacles are detected, the gripper has no pushing action at this stage. The intuition of  $\mathbf{G}_{push}$  is that the gripper moves from the side of the empty blocks towards the origin  $O$  to push the obstacle blocks. The empty blocks can be regarded as the entrance for gripper pushing. Figure 11b demonstrates the obstacle-separation direction for Figure 11a where the left-front, left-rear, and rear blocks (marked as green arrows in Figure 11b) do contain obstacles, so the gripper would come from the bottom right corner to push aside the obstacles.

After clearing aside the bottom obstacles, the gripper will swallow the target and separate it from the top obstacles. As shown in Figure 11a, if an obstacle (left) is next to the target, it is better for the gripper to move an opposite offset (right) at point  $M$  before lifting up to swallow the target. In such a way, the gripper can avoid

swallowing the neighboring obstacles. Similarly, the offset vector is calculated  $\mathbf{OM}$  by using top block vectors  $\mathbf{T}_{LF}$ , ...,  $\mathbf{T}_{LR}$ :

$$\mathbf{K} = \mathbf{T}_{LF} + \mathbf{T}_{LR} + \mathbf{T}_F + \mathbf{T}_R + \mathbf{T}_{RF} + \mathbf{T}_{RR}, \quad (7)$$

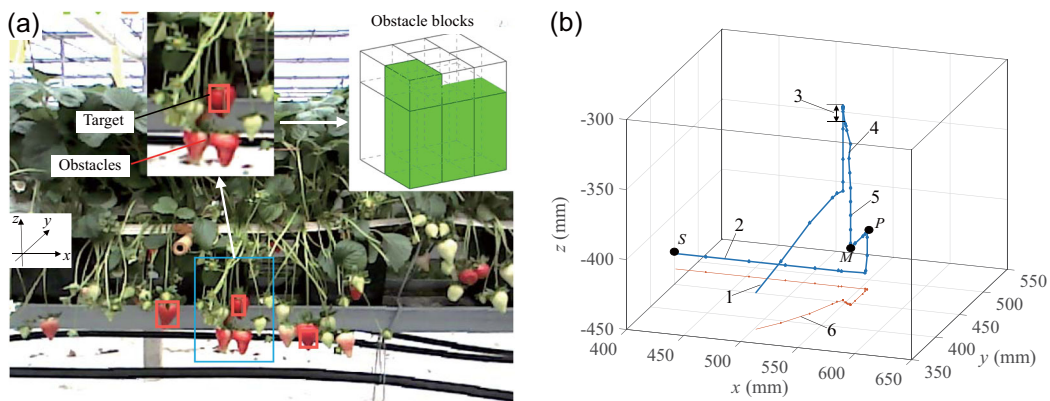
$$\mathbf{OM} = \frac{R\mathbf{K}}{|\mathbf{K}|}, \quad (8)$$

where  $R$  is the norm of  $\mathbf{OM}$ , 3 mm in the system. In the situation in 11a, in which a red obstacle is situated to the left of the target, within the left-front and left-rear blocks, the gripper will move 3 mm from the target origin  $O$  to the right point  $M$  (11b) before moving up to swallow the target. If both bottom and top blocks contain obstacles, the gripper moves directly from  $P$  to  $M$  without transferring at  $O$ .

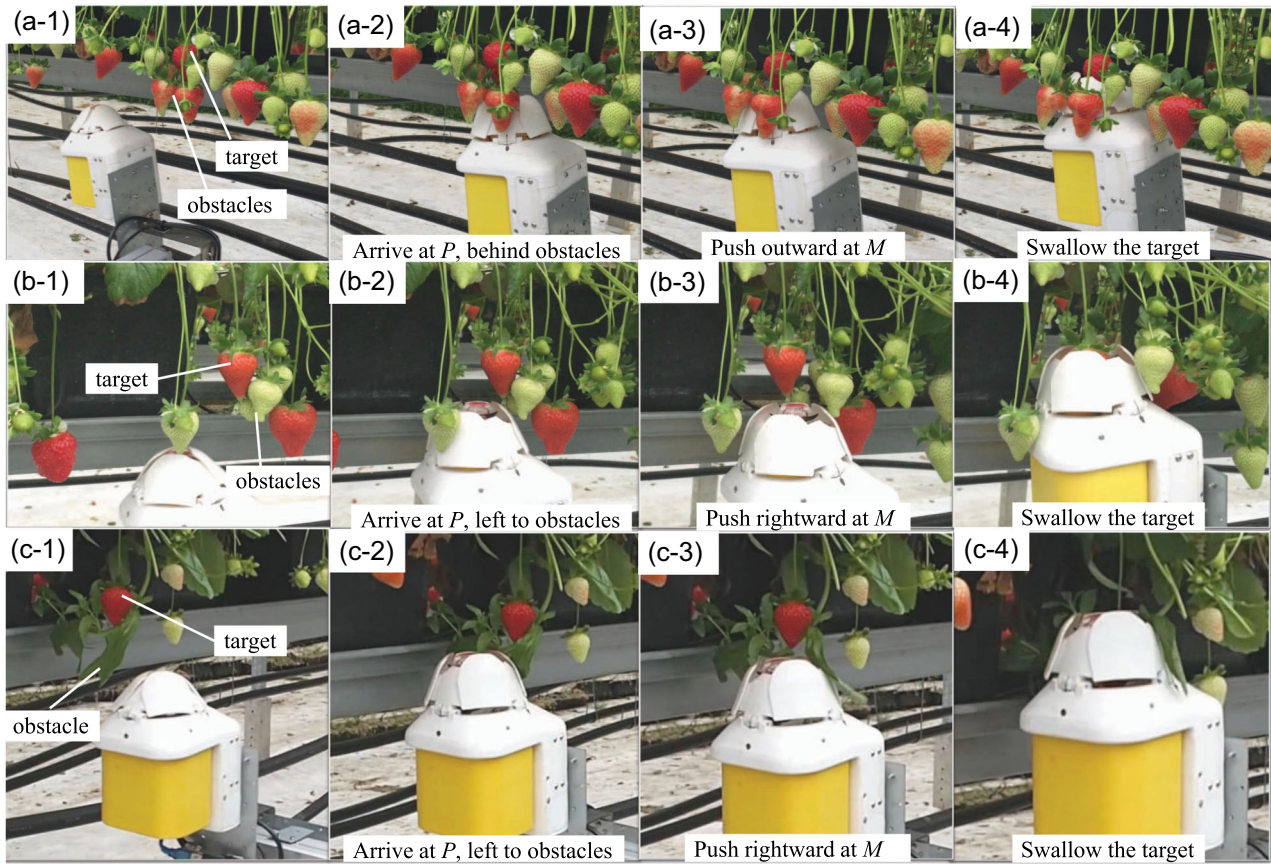
### 8.1.2 | Application examples

Figure 12 shows an example of the robot actively separating obstacles by using the proposed algorithm. In Figure 12a, a target ripe strawberry has been detected. The right-top corner figure displays the obstacle blocks around the target. The vision system detected obstacles (marked as green) in three bottom front blocks ( $\mathbf{B}_{LF}$ ,  $\mathbf{B}_F$ , and  $\mathbf{B}_{RF}$ ) and the top front-front block ( $\mathbf{T}_{RF}$ ). After path-planning, as shown in Figure 13a-1, the gripper moves to the intermediate point  $P$  which is behind the obstacles. Then it moves outward to push the front obstacles arriving at point  $M$ . After pushing, the gripper is going up to swallow the target. The path of the gripper can be seen in Figure 12b, which is recorded from field test. Without the obstacle-separation algorithm, the below-picking gripper might swallow the surrounding obstacles during lifting.

Figure 13b,c demonstrates two more examples of the active obstacle-separation algorithm in different situations. In Figure 13b-1, a ripe strawberry is detected together with several green strawberries surrounding on the right ( $\mathbf{T}_F$  and  $\mathbf{T}_R$ ) and right-bottom ( $\mathbf{B}_{RF}$  and  $\mathbf{B}_R$ ) sides. Hence, the gripper first move to the left of the obstacles (point  $P$ ; Figure 13b-2) and then it pushes the obstacles rightward to point  $M$  (Figure 13b-3) for better swallowing. Similarly, as shown in



**FIGURE 12** An example of active obstacle-separation in the field test: (a) Detection of target strawberries and obstacles; (b) path of the gripper for picking the target using the obstacle-separation algorithm; number represents: 1—return path, 2—forward path, 3—peduncle length adjustment, 4—scanning control path, 5—searching path, and 6—trajectory projection [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]



**FIGURE 13** Action sequence of active obstacle separation in the field: (a-1 to a-4) picking sequence of the example in Figure 10 to separate the front obstacles; (b-1 to b-4) example of pushing the right-bottom green berries; and (c-1 to c-4) example of removing bottom leaves [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

Figure 13c-1, a huge leaf is on the bottom of a target. To avoid getting stuck by the leaf (the gripper mouth may be sheltered by the leaf), the gripper moves to the left of the leaf (Figure 12c-2) and pushes the leaf to the right side of the target (Figure 13c-3) before picking.

Figure 12 shows an example of the robot actively separating obstacles by using the proposed algorithm. In Figure 12a, a target ripe strawberry has been detected. The right-top corner figure displays the obstacle blocks around the target: the vision system detected obstacles (marked as green) in three bottom front blocks ( $B_{LF}$ ,  $B_F$ , and  $B_{RF}$ ) and the top right-front block ( $T_{RF}$ ). After path-planning, as shown in Figure 13a-1, the gripper moves to the intermediate point  $P$ , which is behind the obstacles. Then it moves outward to push aside the front obstacles before arriving at point  $M$ . After pushing, the gripper moves up to swallow the target. The path of the gripper can be seen in Figure 12b, recorded from field test. Without this obstacle-separation algorithm, the below-picking gripper is at risk of swallowing the surrounding obstacles during lifting.

Figure 13b,c further demonstrates two more examples of the active obstacle-separation algorithm in different situations. In Figure 13b-1, a ripe strawberry has been detected together with several green strawberries surrounding it on the right ( $T_F$  and  $T_R$ ) and

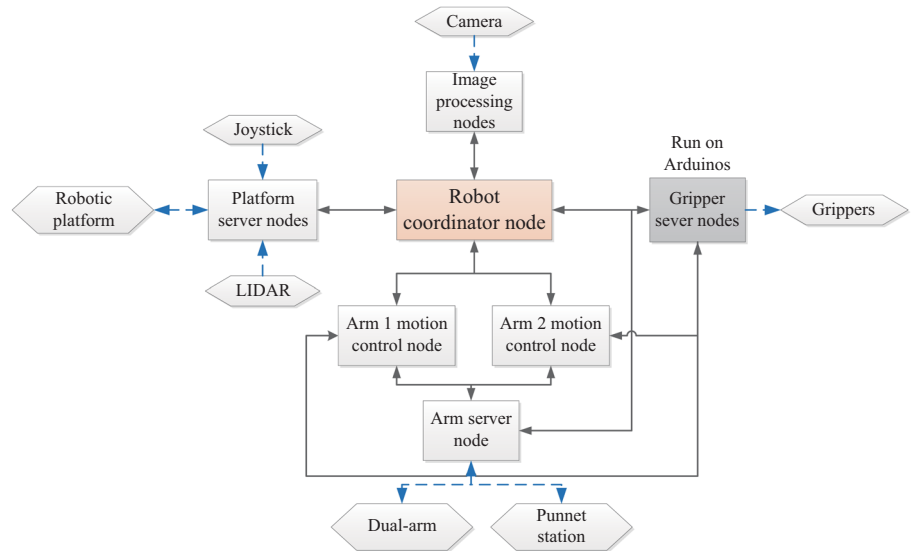
right-bottom ( $B_{RF}$  and  $B_{RR}$ ) sides. Hence, the gripper first moves to the left of the obstacles (point  $P$ ; Figure 13b-2) and then it pushes the obstacles rightward to point  $M$  (Figure 13b-3) for better swallowing. Similarly, as shown in Figure 13c-1, leaves have been detected on the bottom of a target. To avoid getting stuck (as the gripper's mouth may be sheltered by the leaves), the gripper moves to the left of the leaf (Figure 13c-2) and pushes it to the right side of the target (Figure 13c-3) before picking.

## 9 | SYSTEM INTEGRATION AND CONTROL

### 9.1 | System architecture

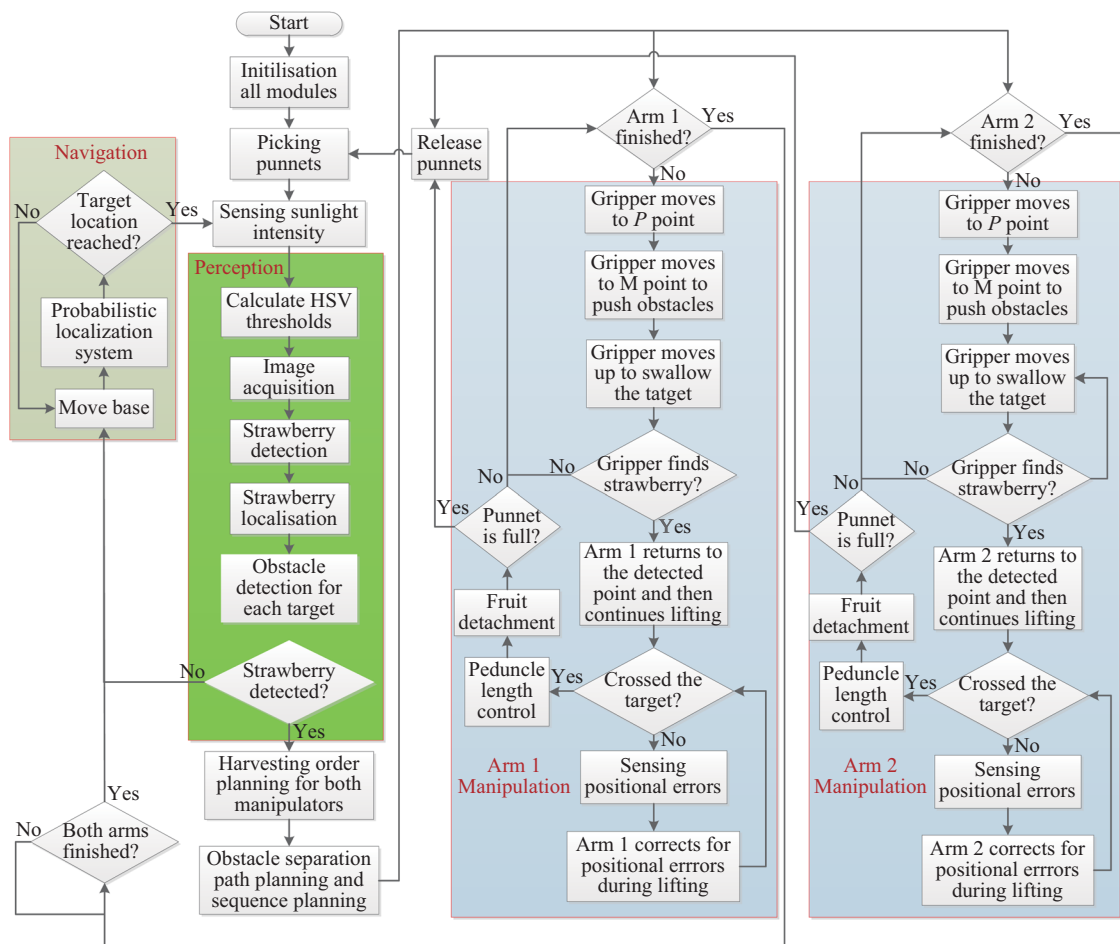
The system's full integration enables the robot to harvest continuously along the strawberry rows. The overall sequence is termed static strawberry harvesting, because the platform will stop, carry out picking operation and then move on when picking is finished, which is similar to the sequences of other agricultural robots (Xiong et al., 2017). The hardware and software architecture of the robot is shown in Figure 14, in which the outside hexagons represent the hardware modules while the inside rectangles are the software functions. Compared to the previous system (Xiong et al., 2019), the main software updates can be summarized as follows: new function

**FIGURE 14** Hardware and software architecture of the robot: The hexagons represent the hardware modules, while the inside rectangles are the software functions [Color figure can be viewed at wileyonlinelibrary.com]



of adaptive color thresholding, integration of the platform module, handling with parallel manipulator harvesting, path-planning of the active obstacle separation, punnet picking and releasing, and full autonomous integration.

A robot coordinator node was created to manage and synchronize the information flow for all the modules. The gripper server nodes comprise a servo control node and an IR sensor feedback node, which are running ROS nodes on two Arduino controllers. The arm



**FIGURE 15** Flowchart of the control system: The entire loop shows the top-level sequential control while the four colored modules represent mid-level control loops, in which navigation, Arm 1 manipulation, and Arm 2 manipulation are continuous closed-loop control systems [Color figure can be viewed at wileyonlinelibrary.com]

motion control nodes are used for manipulation. They communicate with the gripper server nodes and the arm server nodes, receive IR sensor feedback, arm positions status and publish the goal mode and status of the arms, as well as the gripper actions. Each arm has an independent motion control node to ensure that the two manipulators pick in parallel. The platform server nodes are used to navigate the mobile base, based on the feedback from the wheel encoders and the LIDAR scanner. The navigation module can also be manually operated with the joystick in case of emergency.

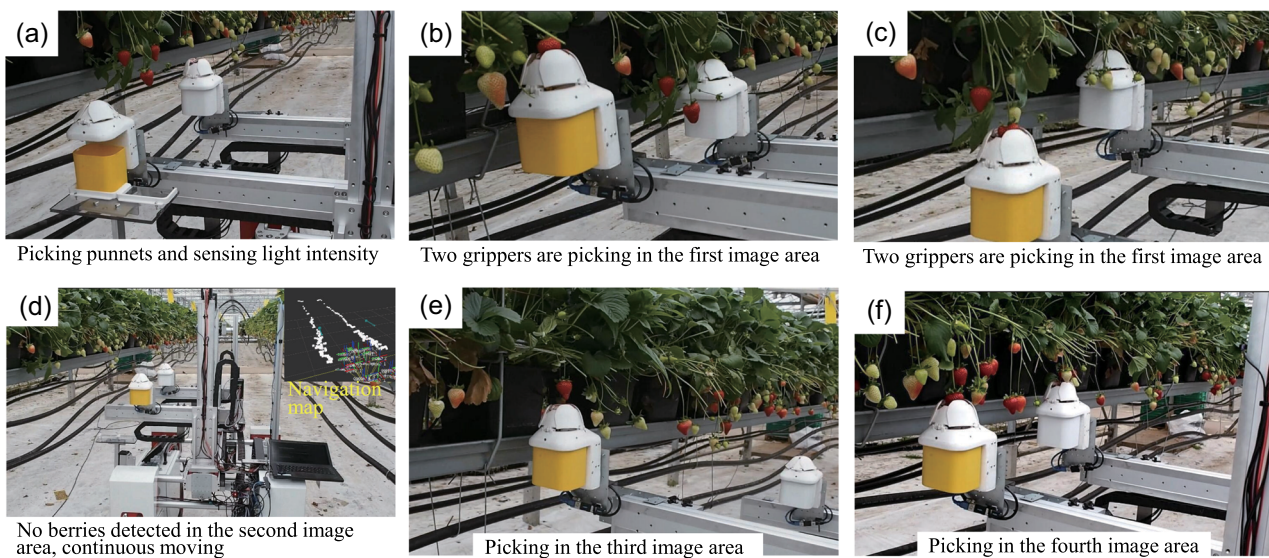
## 9.2 | Harvesting sequence and system control

The flowchart in Figure 15 illustrates the complete sequence and control strategy for the harvesting robot. The whole system consists of several control loops, which can be classified as three levels: top level, mid-level, and lowest level. The top level is an open-loop sequential control system, which is used to trigger a series of operations in the correct sequence, from perception through manipulation to navigation. In the top-level loop, there is one mid-level open-loop module (perception) and three mid-level continuous closed-loop control systems (Arm 1 manipulation, Arm 2 manipulation, and navigation). The two arm manipulation loops are the same, using a multithread method for computing. The arm manipulation module consists of an open-loop obstacle-separation action, a closed-loop collision avoidance function (see Section 6.3) and a closed-loop scanning control for picking (Section 7.2). The scanning control function closes the loop between the arm and the gripper. The gripper continuously senses each targets location with respect to itself, while the arm uses the feedback from the gripper to control its position and correct positional errors using proportional-integral-derivative (PID) controllers. The detailed control method is introduced in Section 7.2 and our previous work (Xiong et al., 2019). The

navigation module controls the mobile base steering by using a proportional controller based on the feedback from the wheel encoders and the LIDAR scanner. Further, among these mid-level loops, a single actuator makes up a lowest level loop, such as servo stepper motors for the dual-arm and servo motors for the grippers and the mobile base. All of these actuators are continuous closed-loop control systems, using PID controllers.

As shown in Figure 15, with the exception of the four colored rectangle modules, the uncolored processes are all executed by the robot coordinator node. Figure 16 shows an example of the harvesting sequence in the farm. After initialization, the robot first picks punnets from the punnet station with verification from the gripper sensors (Figure 16a). Meanwhile, the perception module draws a light intensity value from the gripper server nodes to update color thresholds. The perception module outputs the detected strawberry bounding boxes to the coordinator, together with obstacle block vectors. If no strawberries are detected at this stage, the platform will move forward to the next image area using the navigation control module. The coordinator node sorts all the input targets and determines the harvesting order for both arms according to the algorithm in Section 6.3. In addition, the coordinator creates a path plan to separate obstacles based on the methods in Section 8 and, finally, sends a full path of arms and gripper actions to the arm manipulation modules.

Once the target coordinates have been obtained, the two arms are actuated to pick strawberries, here shown in the first image area (Figure 16b,c). The arm moves the gripper to separate obstacles before swallowing the target strawberry based on the method describes in Section 8. When the gripper detects the presence of the target during the swallowing searching procedure, the arm will return to the detected point and then use the scanning control method (Section 7.2) to correct for positional errors while passing the target body. If the



**FIGURE 16** Continuous harvesting in the strawberry farm: (a) Picking punnets and sensing light intensity; (b) two grippers are picking in the first image area; (c) two grippers are picking in the first image area; (d) no berries detected in the second image area, continuous moving; (e) picking in the third image area; and (f) picking in the fourth image area [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]



gripper is not able to detect the target in this procedure, the arm will move to the next target directly. After that, the cutter is actuated to detach the target strawberry with verification from the IR sensors. After each picking, the gripper will estimate the amount of picked strawberries in the punnet using berry amount sensors (Section 7.1). If the punnet is full, the arm will move the gripper to pick a new punnet.

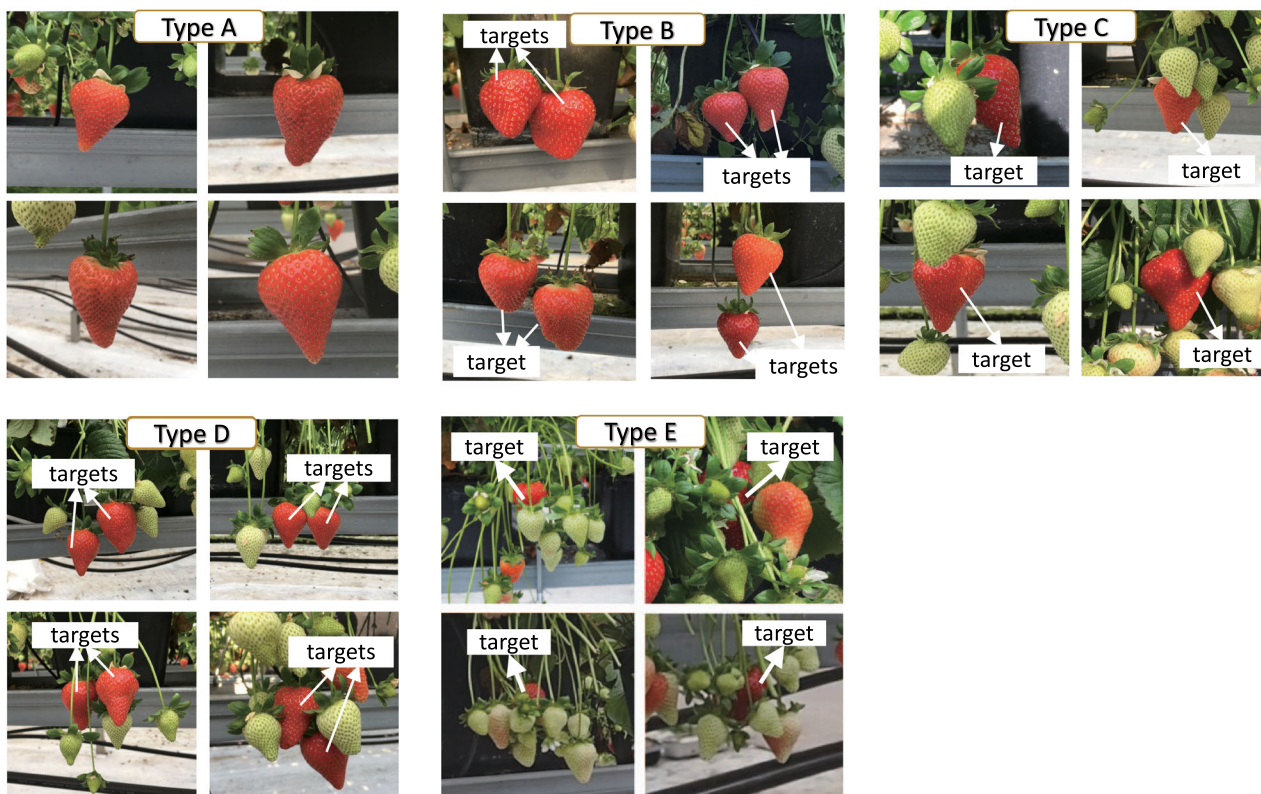
When the picking is finished, the arm returns a signal to the coordinator node. As a manager, the coordinator node collects the signal and commands the platform to move to a new image area when both arms finish. As shown in Figure 16d, after it has finished picking in the first image area, the robot moves to the second image area, in which no ripe berries are detected so it continues to move forward. Then the robot is continuously picking in the third and fourth image areas (Figure 16e,f).

## 10 | FIELD EXPERIMENT SETUP

The experiments were conducted in the Boxford Suffolk Farms (England), which utilizes a table-top strawberry growing system in the greenhouse. The tests were carried out on a variety of strawberries called “Lusa” (Driscolls Ltd.), which is productive in the greenhouse annually from March to July. This variety of

strawberries has long peduncles, making the fruit easy for both humans and robots to pick. Unlike our previous work, which defined all growth situations of strawberries as the natural environment, in this paper the strawberry growing distributions are classified into five types for better evaluation of the robots performance, as illustrated in Figure 17. Based on our observations at the farm, not all strawberry distributions can be classified as a specific type; however, they can all be said to have derived from these five specific types, each of which was evident on the farm and influenced the test results. The five types are defined as follows:

- (1) Type A: One isolated ripe strawberry with no other strawberries around it. This is the simplest situation but also common in this strawberry variety.
- (2) Type B: Two ripe strawberries growing very close to each other but with no other strawberries around. Their distribution may be left-right, front-rear, or top-bottom and so on.
- (3) Type C: One ripe strawberry partially surrounded by unripe strawberries. There are spaces through which the gripper can access the ripe berry. This situation is also common in this variety.
- (4) Type D: Two ripe strawberries partially surrounded by unripe strawberries. This situation is similar to type B and type C but more complicated.



**FIGURE 17** Definitions of five strawberry growing types for the picking experiments: Type A: isolated ripe strawberry; Type B: two connected ripe strawberries; Type C: one ripe strawberry with surrounding (not fully) raw strawberries; Type D: two ripe strawberries with surrounding (not fully) raw strawberries; and Type E: one ripe strawberry that is fully surrounded by raw strawberries. Five different types of strawberry growth, as defined for the picking experiments. Type A: isolated ripe strawberry; Type B: two connected ripe strawberries; Type C: one ripe strawberry partially surrounded by unripe strawberries; Type D: two ripe strawberries partially surrounded by unripe strawberries; and Type E: one ripe strawberry fully surrounded by unripe strawberries [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

- (5) Type E: One ripe strawberry that is fully surrounded by unripe strawberries. This is the most challenging growing situation but was not commonly seen in our experiments with the variety "Lusa."

## 11 | RESULTS

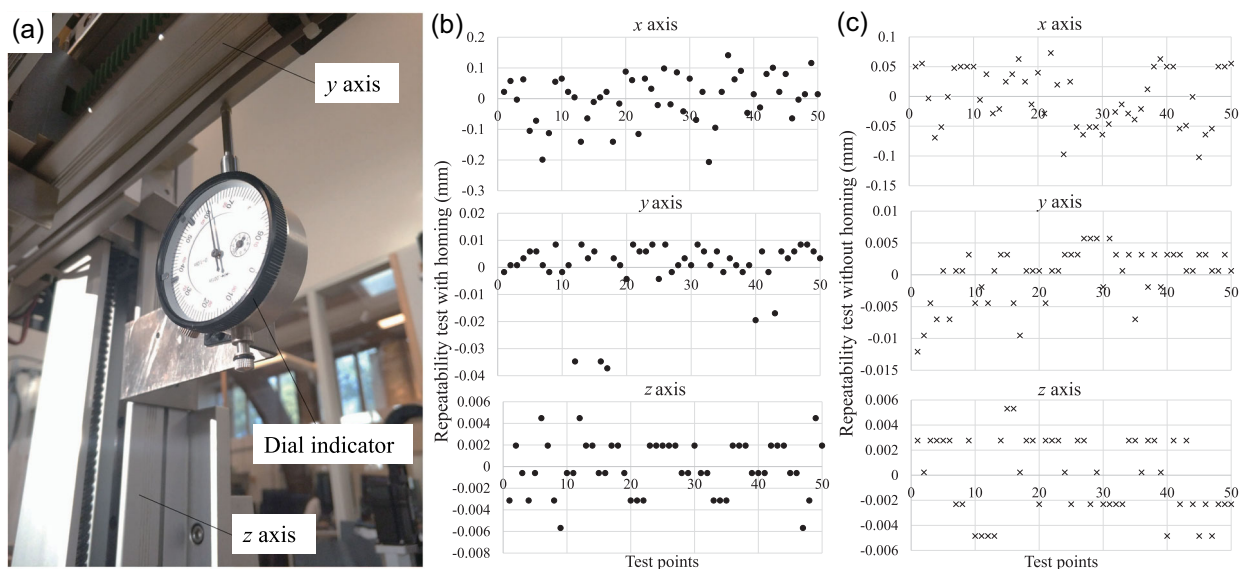
### 11.1 | Arm repeatability test

To evaluate the arm performance, a repeatability test on the dual-arm system was conducted, which tested each axis independently. As shown in Figure 18a, a dial indicator is attached to the arm z axis, and the y axis will touch the indicator tip when z axis has an up-down movement. Two sets of experiments were performed: with homing and without homing. With homing the arm during each trail, the precision is also influenced by the end-stop sensors. This is meaningful to the nonabsolute motor encoder, as the arms need to be homed every time after restarting and the main positional error is due to robot zeroing (Conrad et al., 2000). With homing option, Figure 18b illustrates the repeatability test results after zero-mean normalization of 50 trials at 200 mm/s traveling speed. Following the ISO 9283 standard on arm repeatability calculation, the repetition precisions of x, y, and z axes are 0.209, 0.032, and 0.006 mm, respectively. Similarly, without homing, as shown in Figure 18c, the repetition precisions of x, y, and z axes are 0.109, 0.011, and 0.007 mm, respectively. The variance of the precision among axes is mainly due to the different transmission type and gear ratio, but all of these precisions are high enough for our harvesting application. To evaluate the performance of the arms in this new strawberry harvester, a repeatability test was conducted on the dual-arm system, with each axis tested independently. As shown in Figure 18a, a dial indicator was attached to the arms z axis, the tip of which

was touched by the y axis during the up-down movement of the z axis. Two sets of experiments were performed, namely one with homing and one without homing. When homing the arm during a trial, the precision is also influenced by the end-stop sensors. This is meaningful to the nonabsolute motor encoder, as the arms require homing after every restart and any positional error is mainly due to robot zeroing (Conrad et al., 2000). Here, Figure 18b illustrates the repeatability test results of the homing experiments after zero-mean normalization of 50 trials at 200 mm/s traveling speed. Following the ISO 9283 standard on arm repeatability calculation, the repetition precisions of the x, y, and z axes were measured at 0.209, 0.032, and 0.006 mm, respectively. Similarly, in the experiments without homing, shown in Figure 18c, the repetition precisions of the x, y, and z axes were 0.109, 0.011, and 0.007 mm, respectively. The variance of the precision among axes is considered to be mainly due to the different transmission types and gear ratios; however, these precisions are all of a sufficiently high standard for this harvesting application.

### 11.2 | Success rate, failure cases, and cycle times for different types

The performance tests conducted on this new strawberry-harvesting robot provide valuable information on current state and identify the limitations and challenges to the system, which are important for future improvements. The evaluation tests were implemented from April 8, 2018 to April 10, 2018, following the completion of the system integration. Two main indicators were used to evaluate the robots performance, namely success rate, and picking cycle time, representing harvesting accuracy and speed, respectively. The failure cases were recorded and analyzed to identify the challenges, which may be attributed to a variety of



**FIGURE 18** Arm repeatability test results: (a) Testing setup for z axis where a dial indicator was mounted on the z axis and the y axis was used to touch the dial indicator tip during the up-down movement of the z axis; (b) repeatability test results for each axis with homing; and (c) repeatability test results for each axis without homing [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

**TABLE 2** Harvesting success rate of the robot in different growing types on the “Lusa” variety of strawberries

Strawberry distribution type	Quantity of target fruit	Success on the first attempt	Success with two attempts	Unpicked two attempts	Success rate on the first attempt (%)	Success rate with two attempts (%)
Type A	34	33	34	0	97.1	100.0
Type B	44	32	40	4	72.7	90.9
Type C	37	28	31	6	75.7	83.8
Type D	40	20	30	10	50.0	75.0
Type E	20	1	4	16	5.0	20.0

factors ranging from the subsystems of the robot to environmental factors or even the strawberry itself. Table 2 shows the rates of harvesting success for the five growing types, while the failure rates are listed in Table 3. In each trial, the robot attempted to pick a second time if the first attempt was a failure. More than two attempts are considered unsuitable since multifailure attempts might damage the fruit, especially fragile strawberries. Some of the reasons for picking failure were found to be common for all growing types; however, there were others in which the robot encountered new problems when the growing environment changed. The total number of failures listed in Table 3 includes both attempts. Therefore, if a strawberry was unreachable, the number of failures listed under common reasons (5) is 2. Additionally, several failure cases can appear in one attempt. For

example, if two ripe strawberries were not segmented (Type B (2)), their localization is listed as incorrect (common reasons (1)).

In general, picking success was seen to decrease gradually from Type A to Type E, as the growing situations became increasingly complex. For Type A, the robot was tested on 34 targets with only one failure at the first attempt, which was because the size of the target strawberry was almost at the maximum limit (diameter 45 mm) that the gripper can swallow. After changing the swallowing position, the second attempt was successful. For Type B, 22 pairs of targets were tested for a total of 44 fruits. On the first attempt, 12 of the 44 picks failed while another 8 were successful in the second attempt, representing a first attempt success rate of 72.7% and 90.9% for the two attempts. For Type B, the most frequent failure was caused by swinging (Type B (3) in Table 3). If two strawberries

**TABLE 3** Harvesting failures of the robot in different growing types on the “Lusa” variety of strawberries

Type	Failure reasons	Failure times	Happening rates among all failures (%)
Common reasons	1. Localization error		
	2. Target strawberry not detected		
	3. Target strawberry diameter too big (diameter over 45 mm) for gripper swallowing		
	4. Failure to cut the peduncle of strawberry		
	5. Target locations unreachable, either too high, too low or too far		
Type A	Common reasons (3)	1	100.0
Type B	1. Common reasons (1), (2), (3), and (4)	4, 4, 2, 1	21.1, 21.1, 10.5, 5.3
	2. Two connected ripe strawberries were not segmented during image processing	3	15.7
	3. The second strawberry was swinging after picking the first one, resulting in large positional error	5	26.3
Type C	1. Common reasons (1), (2), (3), and (5)	2, 4, 2, 2	13.3, 26.7, 13.3, 13.3
	2. One or more surrounding immature strawberries were picked together with the ripe strawberry	3	20.0
	3. Peduncle was connected to nearby immature strawberries, stems, or leaves, thus pushing the ripe strawberry together with these obstacles	2	13.3
Type D	1. Common reasons (1), (2), (3), and (4)	7, 10, 2, 1	19.4, 27.8, 5.6, 2.8
	2. Reasons as per Type B (2) and (3)	4, 3	11.1, 8.3
	3. Reasons as per Type C (2) and (3)	5, 4	13.9, 11.1
Type E	1. Common reasons (1) and (2)	16, 22	28.1, 38.6
	2. Reasons as per Type C (2) and (3)	10, 9	17.5, 15.8

are connected with static force between them, picking of the first strawberry could change the position of the second strawberry or even make it swing dynamically. Thus, when the robot is picking the second strawberry, the previously obtained position might be incorrect. This problem can be overcome by incorporating visual servoing or other real-time detection techniques for closed-loop control. In the tests, most of the swinging strawberries were successfully picked on the second attempt, after the image processing results were updated. Moreover, the vision algorithm was sometimes not able to segment the connected strawberries, which meant that the robot would go to the center of the two targets, regarding them as one strawberry. If one of these strawberries was picked in the first attempt, the second attempt was regarded as a Type A situation.

In Type C, new problems appeared because of surrounding immature strawberries. Without segmentation and swinging issues, the first-attempt success rate for Type C was slightly higher than that of Type B (75.7%), but the success rate of the two attempts (83.8%) was lower than in Type B. This is because the second attempt in Type C was on the same target with fewer environment changes, which is markedly different from the circumstances of Type B. If surrounding small immature strawberries are growing too close to the target, they are at risk of being swallowed together, which would not only decrease future yields but would also mix immature strawberries into the punnets with the ripe fruit. Another issue (Type C (3)) is that the peduncle of the target might be connected to nearby immature berries, stems or leaves, so that the ripe strawberry is pushed up together with these obstacles that should not be swallowed. Type D can be regarded as a combination of Types B and C, so their failure cases may also happen for Type D, making it a more challenging growing situation than the others. In the tests, only half of the targets were picked successfully on the first attempt and, with two attempts, the rate increased to 75.0%. Many of the complex surrounding berries, leaves or stems were not detected by the vision system. Finally, Type E presented an almost impossibly complicated situation for our system, resulting in a mere 5% first-attempt success rate, in which one pick was attributed to luck, and increasing to 20% on the second attempt. Detecting a strawberry with many others in front of it was a challenge and, in fact, 11 berries could not be detected at all. Three of the successful picks during the second attempt were achieved only because the first attempt had cut some of the surrounding obstacles, making the second attempt easier.

To assess the picking speed of the robot, the picking times for both the one-arm and dual-arm modes were calculated from video recordings of the movement. Researchers proposed a definition for cycle harvesting time, which includes perception operation, manipulation of a fruit, placement of the detached fruit, and also the arm traveling time to the next fruit (Bac et al., 2013, 2017). Due to the variation in robots and crops, similar metrics have been used by other works but with some differences, for example, without counting the time for the arm traveling to the next fruit (Lehnert et al., 2017) or without adding the perception time (Silwal et al., 2017). Nevertheless, platform moving time has not been taken into account by

most of the reports (Bac et al., 2017; Lehnert et al., 2017; Silwal et al., 2017). In our system, most of the time taken is in the manipulation process, since the top-level control is open loop so the robot only need to sense an image area once and then the two arms are actuated to pick all the targets in this image without further perception needed. Laboratory tests with fake strawberry plants (6–12 ripe strawberries) indicated that the average time for our perception system is 0.11 s (i5-6200 CPU, 16 GB RAM), including image acquisition, detection, obstacle calculation, and path-planning. The cycle time including perception is varied if the number of strawberries in each image area is different. Also, the gripper can collect strawberries during picking, so the time taken for the manipulator to drop individual fruit does not exit. Therefore, similar to apple harvesting (Silwal et al., 2017), we report the harvesting time on manipulation time only, including the picking time and arm traveling time, excluding the time taken to move platforms and pick punnets. On average, the time in which one arm successfully picked one target and traveled on to the next was 6.1 s, as shown in Table 4. This picking speed is faster than that of our previous versions average of 7.49 s, and is attributed to the increased speed of the arms in both movement and communication, as well as the new scanning control method. When using two arms, one berry was picked in 4.6 s, which is more than half the time taken by the single arm. This is because of delays while one arm waited for the other during picking to avoid collision or while the platform was moved.

## 12 | DISCUSSION AND LESSONS LEARNED

Results show the new autonomous strawberry-harvesting robot is more accurate and faster than the previous version. These improvements are the combined result of tight system integration, adaptive machine vision, cooperative dual arms, an improved gripper, and intelligent obstacle separation. However, along with these improvements, the new system still faces numerous challenges.

First, the adaptive color-thresholding method in the machine vision subsystem shows the ability to adapt to the changing sunlight. Color thresholding on 2D images is a simple and fast algorithm. It is

**TABLE 4** Manipulation time on successful picking with one arm or two arms configurations

Number of arms	Test no.	Number of picked strawberries	Manipulation execution time per strawberry	Average(s)
One arm	1	2	6.8	6.1
	2	3	6.1	
	3	3	6.0	
	4	5	5.4	
Two arms	1	5	4.4	4.6
	2	5	5.2	
	3	7	4.6	
	4	8	4.2	

effective and efficient when the environment is simple. However, during the experiments on the farm, most of the localization errors came from the image processing. For example, two connected strawberries could not be segmented or one strawberry was segmented into two parts because there was a stem in front of it. This problem may be considered commonplace for traditional image processing. The current alternative which is to use more advanced technologies like 3D image processing and deep learning, is likely to solve many of these problems, especially as processing speed is unlikely to remain a problem as new hardware is developed. The whole system was evaluated in the field as listed in Table 3, including all aspects of failures, but lacked specific evaluation of individual system. Our future work will consider to standardize the metrics and data set to compare the adaptive color thresholding with other detection methods. In addition, the current gripper is unable to distinguish between correct (true positive) and incorrect detection (false positive) of mature strawberries, thus once receiving a target location, the gripper will pick it anyway regardless of the actual targets. However if nothing is detected by the gripper, the arm will skip it immediately and move to the next target (Section 9.2). Therefore, future improvements will be to use an additional hand-eye camera for final verification. Additional hand-eye cameras can also be used for closed-loop vision-based manipulation, because the current stationary camera is easily occluded by the arms or grippers during picking operations.

Second, the proposed active obstacle-separation algorithm proved to be effective in field applications, thus improving the harvesting ability of the robot. As long as there was an entrance (empty blocks) within the bottom blocks, the gripper was able to find a way to push aside the surrounding obstacles. However, it was still unable to pick targets that were fully surrounding by obstacles (leaving no entrance). It was also still not robust and revealed some limitations, especially from the vision side. The first limitation is the insufficient view and point cloud. In the current system, only a single view was used to get the 3D scenario. However, because of occlusions and the straight projection of the camera, the rear obstacles were not easily detected, such as the case in Figure 12a when the bottom left-rear obstacle was not detected. Therefore, future work should make use of multi-view images and reconstruct more accurate scenes. The second vision problem is that of inaccurate localization. As the obstacle block size is dependent on the target bounding box, inaccurate localization of ripe strawberries might result in the gripper pushing the target when separating surrounding obstacles. In addition to the vision system limitations, closed-loop control between perception and manipulation may be able to improve the performance of obstacle separation. From a mechanical perspective, an additional manipulator may also help to separate the obstacles, like human manipulation in cluttered environment. Also, in some cases, the gripper size was found to be too large to separate the obstacles delicately, however, a small-sized gripper may not be able to swallow large fruits, so this gap remains to be solved.

Finally, strawberry variety is an important factor that can influence how the robot, especially the gripper, is designed, as well as the picking success rate. Based on field observation, varieties like "Lusa" are easier

for picking than others, such as "Rumba," which has lots of clusters with short peduncles growing on one stem. This feature makes it difficult for robots to separate obstacles. This suggests that the automation of the agriculture industry requires more efforts from horticulture technology in breeding new varieties and developing new growing systems to simplify the environment for robots.

## 13 | CONCLUSIONS

This paper presents a fully integrated strawberry-harvesting system capable of picking strawberries in clusters. While several harvesters that can cope with isolated strawberries have been developed, those growing in complex clusters remain a challenge. The main scientific contribution of this paper is the novel obstacle-separation path-planning algorithm, which allows the successful harvest of strawberries that are surrounded by other strawberries, as well as by leaves and other obstacles. The algorithm uses the gripper to push surrounding obstacles from an entrance, thus clearing the way for it to swallow the target strawberry. The separation actions consist of pushing aside the bottom obstacles before swallowing and pushing aside the top obstacles during swallowing. The pushing vectors are derived based on the surrounding obstacles that are calculated using downsampled blocks of 3D point cloud. This technique might be applicable to other fruit harvesting systems.

In addition to obstacle separation, improvements were made to the gripper, the vision system, and the control. For adaptation to the field environment, a vision system that could automatically change color thresholds was developed based modeling of color against sunlight intensity, making it robust to variations in lighting. Furthermore, a low-cost single-rail two Cartesian arm system was developed, which makes it suitable for agricultural robot application. The harvesting sequence for the dual-arm was studied to optimize harvesting efficiency and avoid collision. This study also presents an improved gripper design that enables the robot to pick a market punnet and harvest berries directly into the punnet, thus eliminating the cost and time for repacking.

Finally, we show the full integration and control algorithm of the whole system, which enables the robot to harvest continuously along the polytunnels. The system was tested in the field on a strawberry farm. Results revealed that the robot was capable of picking partially surrounded strawberries, with success rates ranging from 50.0% to 97.1% on the first attempt, depending on the different type settings. This rate rose to between 75.0% and 100.0% on the second attempt. However, the system was not able to pick a target that was fully surrounded by obstacles, recording a first-attempt success rate of just 5.0%. The picking speed in the one-arm mode increased to 6.1 s, including both picking and the arm's travel time to the next target, while, for the dual-arm mode, the average picking time was recorded as 4.6 s per strawberry. Failures in this new system were caused mainly by the vision system and insufficient dexterity in the grippers, which will be addressed in future developments of the harvester.

## ACKNOWLEDGMENTS

This study was supported by the Norwegian University of Life Sciences, Norway. We thank Mr. Robert England and Mr. Mihail Marita from the Boxford Suffolk Farms (Colchester, UK) for providing the strawberry greenhouse and accommodation to conduct field experiments.

## ORCID

Ya Xiong  <http://orcid.org/0000-0001-5593-8440>

## REFERENCES

- Anjom, F. K., Vougioukas, S. G., & Slaughter, D. C. (2018). Development of a linear mixed model to predict the picking time in strawberry harvesting processes. *Biosystems Engineering*, *166*, 76–89.
- Bac, C., Hemming, J., & VanHenten, E. (2013). Robust pixel-based classification of obstacles for robotic harvesting of sweet-pepper. *Computers and Electronics in Agriculture*, *96*, 148–162.
- Bac, C. W., Hemming, J., vanTuijl, B., Barth, R., Wais, E., & vanHenten, E. J. (2017). Performance evaluation of a harvesting robot for sweet pepper. *Journal of Field Robotics*, *34*(6), 1123–1139.
- Bac, C. W., Roorda, T., Reshef, R., Berman, S., Hemming, J., & vanHenten, E. J. (2016). Analysis of a motion planning problem for sweet-pepper harvesting in a dense obstacle environment. *Biosystems Engineering*, *146*, 85–97.
- Bangert, W., Kielhorn, A., Rahe, F., Albert, A., Biber, P., Grzonka, S., & Haug, S. (2013). Field-robot-based agriculture: "RemoteFarming. 1" and "BoniRob-Apps". 71th conference LAND. TECHNIK-AgEng 2013, Hannover, Germany, 2193, 439–446.
- Bargoti, S., & Underwood, J. P. (2017). Image segmentation for fruit detection and yield estimation in apple orchards. *Journal of Field Robotics*, *34*(6), 1039–1060.
- Botterill, T., Paulin, S., Green, R., Williams, S., Lin, J., Saxton, V., & Corbett-Davies, S. (2017). A robot system for pruning grape vines. *Journal of Field Robotics*, *34*(6), 1100–1122.
- Conrad, K. L., Shiakolas, P. S., & Yih, T. C. (2000). Robotic calibration issues: Accuracy, repeatability and calibration. In Proceedings of the 8th Mediterranean Conference on Control and Automation (MED2000), Rio, Greece, 1719, 1–6.
- Cui, Y., Gejima, Y., Kobayashi, T., Hiyoshi, K., & Nagata, M. (2013). Study on cartesian-type strawberry-harvesting robot. *Sensor Letters*, *11*(6-7), 1223–1228.
- Dimeas, F., Sako, D. V., Moulianitis, V. C., & Aspragathos, N. A. (2015). Design and fuzzy control of a robotic gripper for efficient strawberry harvesting. *Robotica*, *33*(5), 1085–1098.
- Feng, Q., Wang, X., Zheng, W., Qiu, Q., & Jiang, K. (2012). New strawberry harvesting robot for elevated-trough culture. *International Journal of Agricultural and Biological Engineering*, *5*(2), 1–8.
- Feng, Q., Zou, W., Fan, P., Zhang, C., & Wang, X. (2018). Design and test of robotic harvesting system for cherry tomato. *International Journal of Agricultural and Biological Engineering*, *11*(1), 96–100.
- Fentanes, J. P., Lacerda, B., Krajnik, T., Hawes, N., & Hanheide, M. (2015). Now or later? predicting and maximising success of navigation actions from long-term experience. 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE, Seattle, USA, 1112–1117.
- Fu, L., Feng, Y., Majeed, Y., Zhang, X., Zhang, J., Karkee, M., & Zhang, Q. (2018). Kiwifruit detection in field images using faster r-cnn with zfnet. *IFAC-PapersOnLine*, *51*(17), 45–50.
- Grimstad, L., Skattum, K., Solberg, E., Loureiro, G. D. S. M., & From, P. J. (2017). Thorvald II configuration for wheat phenotyping. IROS Workshop on Agri-Food Robotics: Learning from Industry, Vancouver, Canada, vol. 4.
- Grimstad, L., Zakaria, R., Le, T. D., & From, P. J. (2018). A novel autonomous robot for greenhouse applications. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–9.
- Grisetti, G., Stachniss, C., & Burgard, W. (2007). Improved techniques for grid mapping. *IEEE Transactions on Robotics*, *23*(1), 34–46.
- Habaragamuwa, H., Ogawa, Y., Suzuki, T., Shiigi, T., Ono, M., & Kondo, N. (2018). Detecting greenhouse strawberries (mature and immature), using deep convolutional neural network. *Engineering in Agriculture, Environment and Food*, *11*(3), 127–138.
- Hayashi, S., Shigematsu, K., Yamamoto, S., Kobayashi, K., Kohno, Y., Kamata, J., & Kurita, M. (2010). Evaluation of a strawberry-harvesting robot in a field test. *Biosystems Engineering*, *105*(2), 160–171.
- Hayashi, S., Yamamoto, S., Saito, S., Ochiai, Y., Kamata, J., Kurita, M., & Yamamoto, K. (2014). Field operation of a movable strawberry-harvesting robot using a travel platform. *Japan Agricultural Research Quarterly*, *48*(3), 307–316.
- Huang, Z., Wane, S., & Parsons, S. (2017). Towards automated strawberry harvesting: Identifying the picking point. In Gao, Y., Fallah, S., Jin, Y., & Lekakou, C. (Eds.), *Conference Towards Autonomous Robotic Systems* (pp. 222–236). Guildford, UK: Springer.
- Ishikawa, T., Hayashi, A., Nagamatsu, S., Kyutoku, Y., Dan, I., Wada, T. ... Oku, K. (2018). Classification of strawberry fruit shape by machine learning. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *42*, 2.
- Lehnert, C., McCool, C., Sa, I., & Perez, T. (2018). A sweet pepper harvesting robot for protected cropping environments. Retrieved from <https://arxiv.org/abs/1810.11920>
- Lehnert, C., Tsai, D., Eriksson, A., & McCool, C. (2018). 3d move to see: Multi-perspective visual servoing for improving object views with semantic segmentation. Retrieved from <https://arxiv.org/abs/1809.07896>
- Lehnert, C. F., English, A., McCool, C., Tow, A. W., & Perez, T. (2017). Autonomous sweet pepper harvesting for protected cropping systems. *IEEE Robotics and Automation Letters*, *2*(2), 872–879.
- Lili, W., Bo, Z., Jinwei, F., Xiaoan, H., Shu, W., Yashuo, L., & Chongfeng, W. (2017). Development of a tomato harvesting robot used in greenhouse. *International Journal of Agricultural and Biological Engineering*, *10*(4), 140–149.
- Mai, X., Zhang, H., & Meng, M. Q.-H. (2018). Faster r-cnn with classifier fusion for small fruit detection. 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, Brisbane, Australia, 7166–7172.
- McCool, C., Beattie, J., Firn, J., Lehnert, C., Kulk, J., Bawden, O., & Perez, T. (2018). Efficacy of mechanical weeding tools: A study into alternative weed management strategies enabled by robotics. *IEEE Robotics and Automation Letters*, *3*(2), 1184–1190.
- Mueller-Sim, T., Jenkins, M., Abel, J., & Kantor, G. (2017). The robotanist: A ground-based agricultural robot for high-throughput crop phenotyping. IEEE International Conference on Robotics and Automation (ICRA), Singapore, 3634–3639.
- Raja, Y., McKenna, S. J., & Gong, S. (1998). Tracking and segmenting people in varying lighting conditions using colour. Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 228–233.
- Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & McCool, C. (2016). Deepfruits: A fruit detection system using deep neural networks. *Sensors*, *16*(8), 1222.
- Shiigi, T., Kurita, M., Kondo, N., Ninomiya, K., Rajendra, P., Kamata, J., & Kohno, Y. (2008). Strawberry harvesting robot for fruits grown on table top culture. An ASABE Meeting Presentation, American Society of Agricultural and Biological Engineers, Providence, Rhode Island, 1–9.
- Silwal, A., Davidson, J. R., Karkee, M., Mo, C., Zhang, Q., & Lewis, K. (2017). Design, integration, and field evaluation of a robotic apple harvester. *Journal of Field Robotics*, *34*(6), 1140–1159.

- Sridharan, M., & Stone, P. (2007). Color learning on a mobile robot: Towards full autonomy under changing illumination. *International Joint Conference on Artificial Intelligence*, Hyderabad, India, 2212–2217.
- Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics*. Cambridge, USA: MIT Press.
- Vakilian, K. A., Jafari, M., & Zarafshan, P. (2015). Dynamics modelling and control of a strawberry harvesting robot. 2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM), IEEE, Tehran, Iran, 600–605.
- VanHenten, E. J., Hemming, J., VanTuijl, B., Kornet, J., Meuleman, J., Bontsema, J., & VanOs, E. (2002). An autonomous robot for harvesting cucumbers in greenhouses. *Autonomous Robots*, 13(3), 241–258.
- Vijayarangan, S., Sodhi, P., Kini, P., Bourne, J., Du, S., Sun, H., & Wettergreen, D. (2017). High-throughput robotic phenotyping of energy sorghum crops. *Field and Service Robotics* (5, pp. 99–113). Cham, Switzerland: Springer.
- Vitzrabin, E., & Edan, Y. (2016). Changing task objectives for improved sweet pepper detection for robotic harvesting. *IEEE Robotics and Automation Letters*, 1(1), 578–584.
- Xiong, Y., From, P. J., & Isler, V. (2018). Design and evaluation of a novel cable-driven gripper with perception capabilities for strawberry picking robots. 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, Brisbane, Australia, 7384–7391.
- Xiong, Y., Ge, Y., Liang, Y., & Blackmore, S. (2017). Development of a prototype robot and fast path-planning algorithm for static laser weeding. *Computers and Electronics in Agriculture*, 142, 494–503.
- Xiong, Y., Peng, C., Grimstad, L., From, P. J., & Isler, V. (2019). Development and field evaluation of a strawberry harvesting robot with a cable-driven gripper. *Computers and Electronics in Agriculture*, 157, 392–402.
- Yaguchi, H., Nagahama, K., Hasegawa, T., & Inaba, M. (2016). Development of an autonomous tomato harvesting robot with rotational plucking gripper. 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Daejeon, South Korea, 652–657.
- Yamamoto, S., Hayashi, S., Saito, S., Ochiai, Y., Yamashita, T., & Sugano, S. (2010). Development of robotic strawberry harvester to approach target fruit from hanging bench side. *IFAC Proceedings Volumes*, 43(26), 95–100.
- Yamamoto, S., Hayashi, S., Yoshida, H., & Kobayashi, K. (2014). Development of a stationary robotic strawberry harvester with a picking mechanism that approaches the target fruit from below. *Japan Agricultural Research Quarterly*, 48(3), 261–269.
- Ye, Y., Wang, Z., Jones, D., He, L., Taylor, M. E., Hollinger, G. A., & Zhang, Q. (2017). Bin-dog: A robotic platform for bin management in orchards. *Robotics*, 6(2), 12.
- Zhang, L., Gui, G., Khattak, A. M., Wang, M., Gao, W., & Jia, J. (2019). Multi-task cascaded convolutional networks based intelligent fruit detection for designing automated robot. *IEEE Access*, 7, 56028–56038.
- Zion, B., Mann, M., Levin, D., Shilo, A., Rubinstein, D., & Shmulevich, I. (2014). Harvest-order planning for a multiarm robotic harvester. *Computers and Electronics in Agriculture*, 103, 75–81.

## SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section.

**How to cite this article:** Xiong Y, Ge Y, Grimstad L, From PJ. An autonomous strawberry-harvesting robot: Design, development, integration, and field evaluation. *J Field Robotics*. 2019;1–23. <https://doi.org/10.1002/rob.21889>

## APPENDIX: INDEX TO MULTIMEDIA EXTENSIONS

The table shows some videos of the field experiments presented in this paper.

Extension	Media	
	type	Description
1	Video	Gripper actions and field test of the robot
2	Video	Obstacle separation actions in the field
3	Video	Failure cases in the field
4	Video	Lab demo of the robot (including the newest version of the gripper)
5	Video	Navigation in the farm

