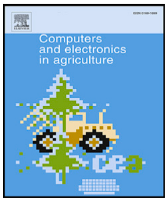
Computers and Electronics in Agriculture 220 (2024) 108863

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag



Original papers

Fast and stable pedicel detection for robust visual servoing to harvest shaking fruits

Yonghyun Park [a,b], Changjo Kim [a,b], Hyoung Il Son [a,b,c,*]

[a] *Department of Convergence Biosystems Engineering, Chonnam National University, Yongbong-ro 77, Gwangju 61186, Republic of Korea*
[b] *Interdisciplinary Program in IT-Bio Convergence System, Chonnam National University, Yongbong-ro 77, Gwangju 61186, Republic of Korea*
[c] *Research Center for Biological Cybernetics, Chonnam National University, Yongbong-ro 77, Gwangju 61186, Republic of Korea*

ARTICLE INFO

Keywords:
Fast point feature histogram
Harvesting robot
Stem detection
Robust visual servoing
Video stabilization

ABSTRACT

This study introduces a fast and stable pedicel detection method for robust visual servoing (RVS), supplemented with video stabilization (ViS) and fast pedicel detection, to realize automated harvesting of shaking fruits. By incorporating ViS, the system effectively mitigates the effects of motion blur, thereby ensuring consistent and precise object detection. In addition, the Fourier spectrum-based band-stop filter (FSBF) is used to improve clarity. The proposed approach also leverages the fast point feature histogram (FPFH) for fast pedicel detection, achieving real-time detection rates of 15–37 fps. Furthermore, it incorporates 6D pose estimation, culminating in the implementation of a 6D pose-based robust visual servoing (6DRVS) system. The performance of this system is evaluated using standard metrics such as perception accuracy, approach accuracy, precision, recall, accuracy, and F1-score in both preliminary tests and on-site experiments at two cucumber farms in Korea. The 6DRVS, supplemented with fast pedicel detection and ViS, exhibited improvements across all evaluation metrics. It recorded 90.00% perception accuracy, 82.22% approach accuracy, 0.957 precision, 0.938 recall, 0.900 accuracy, and 0.947 F1-score, highlighting its essential role in ensuring precise and efficient harvesting.

1. Introduction

Robotic technology has advanced rapidly and has profoundly influenced numerous sectors, notably agriculture (Xiao et al., 2022). The challenges posed by declining agricultural labor, an aging workforce, and unpredictable climate changes have accelerated the transition to smart agriculture (Kpadonou et al., 2017; Ju et al., 2022). With drones scanning fields from above (Ju and Son, 2019) and coordinated ground vehicles optimizing processes (Kim and Son, 2020), precise smart agriculture promises to provide consistent yields and economic growth (Kim et al., 2019; Seol et al., 2022; Park et al., 2023a). A burgeoning domain in this arena is the development of harvesting robots. These robots help address one of the most labor-intensive tasks in farming and ensure that crops are harvested at their peak. As global food demand surges, there is a growing need to improve the productivity and accuracy of harvesting processes (Mohamed et al., 2021). To this end, capabilities in terms of identifying crops, precisely determining their spatial coordinates, and harvesting them without causing damage are necessary (Campbell et al., 2022).

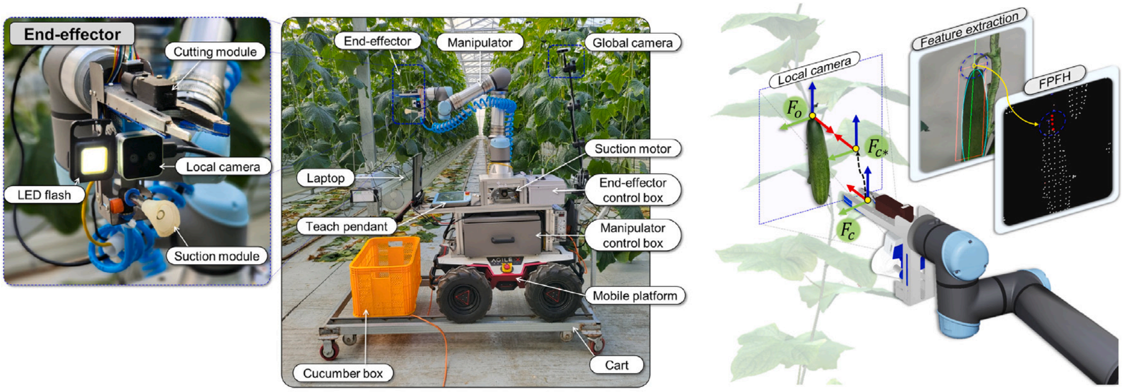
The unpredictable and unstructured environments that characterize the agricultural sector present unique challenges from the perspective of robot application (Bechar, 2021). Fruits exhibit diverse biological characteristics depending on their growth environment, spatial position, geometric shape, size, color, and hardness (Tang et al., 2020). These characteristics make it arduous for robots to function effectively (Li et al., 2020). For this reason, while fruit-harvesting robots have been advanced significantly, the path to the commercialization of such robots remains elusive (Gil et al., 2023). Multifaceted agricultural settings, combined with the delicate nature of crops, constitute the core of this challenge. Consequently, the current study focuses on devising a suitable approach for robot application by investigating crop detection and localization.

In harvesting robots, visual servoing (VS), which capitalizes on feedback from vision sensors to steer and control robotic systems, is being used increasingly (Zhao et al., 2016). VS has advanced rapidly in recent years, and extensive research efforts related to VS have been made across a broad range of applications, including the harvesting of tomatoes (Gao et al., 2022b), apples (Gao et al., 2022a), strawberries (Xiong et al., 2020), and sweet peppers (Arad et al., 2020). However, the dynamic environments of these crops introduce disruptions such as unexpected motions, which challenge the efficacy of traditional VS (Mehta

[*] Corresponding author at: Department of Convergence Biosystems Engineering, Chonnam National University, Yongbong-ro 77, Gwangju 61186, Republic of Korea.

E-mail addresses: dk03378@jnu.ac.kr (Y. Park), ckdwh12@jnu.ac.kr (C. Kim), hison@jnu.ac.kr (H.I. Son).
URL: http://hralab.jnu.ac.kr (H.I. Son).

https://doi.org/10.1016/j.compag.2024.108863
Received 27 September 2023; Received in revised form 13 March 2024; Accepted 19 March 2024



(a)

(b)

Fig. 1. Cucumber-harvesting robot developed in previous studies (Park et al., 2023b): (a) Structure of cucumber-harvesting robot and (b) visual servoing process.

and Burks, 2016). In view of this challenge, researchers are focusing on robust visual servoing (RVS), which is designed to offer unwavering performance even in the presence of such perturbations (Mehta and Burks, 2014; Mehta et al., 2016). However, environmental factors, such as uneven lighting, shadows, and partial covering of fruits with stems and leaves, and shaking fruits, can lead to partial fruit detection, where only a part of a fruit is identified. To realize the full potential of RVS, it is crucial to achieve impeccable object detection even under adverse conditions. This requirement accentuates the importance of embedding robust detection protocols within VS for realizing more robust and reliable robotic functionalities.

The integration of deep learning with harvesting robots is a burgeoning research frontier (Lawal, 2021). As this field progresses, various implementations are emerging. For instance, Mao et al. (2023) introduced a technique that leverages smartphones for detection at a speed of 19.00 frames per second (fps). Moreover, their MangoYOLO1 model, which drew from the features of the YOLOv3 and YOLOv2 models, achieved a detection speed of 14.30 fps on high-end computing clusters with a detection time of 70 ms/frame. In another study, Gao et al. (2022a) developed an automated image processing method tailored for counting apples in orchards with modern vertical fruit-wall structures. They achieved an accuracy of 99.35% at detection speeds of 2.00 to 5.00 fps. However, for real-time applications, detection speeds should ideally exceed 20.00 fps. Anything slower, especially when incorporated into robots, could hamper performance (Zhang et al., 2021). This challenge is exacerbated by the computational demands of deep learning algorithms, which often reduces the frame rate.

In addition, a critical objective in this domain is precise detection of the pedicel, that is, the flower-supporting stem, which is essential for efficient robotic harvesting. Kim et al. (2022) developed the Deep-ToMaToS model to estimate the six-dimensional (6D) pose of an object, and the average accuracy of this model was 96.83%. However, there was a trade-off: as the inference time increased, the frame rate decreased from 45.81 fps to 7.26 fps. To circumvent potential collisions between harvesting robots, Luo et al. (2022) explored the use of deep cameras for object detection and 6D pose estimation. Their methodology required approximately 1.79 s to detect and estimate the pose of one grape cluster. These findings suggest that the incorporation of 6D pose estimation for pedicel recognition might inherently increase computational demands. Therefore, further research on stable and fast detection is needed.

In previous studies, research on efficient cucumber-harvesting robotic systems was conducted (Park et al., 2023b). VS was realized using an easy and fast detection technology based on a simple and fast approach (Fig. 1). In this context, detection was achieved using computer vision-based pedicel detection, specifically by leveraging the differences between normal vectors in the fast point feature histogram (FPFH) to identify pedicels. Even if there exists some error in fruit detection and extraction of pedicel location, so long as the pedicel is within the ROI, 6D pose estimation can be performed using FPFH. The advantage of this proposed fruit detection approach is that it detects the shape of the fruit rather than its type, and therefore, the approach can be adapted easily to different fruits without the need for new datasets, unlike deep learning. A notable hurdle in this endeavor was the reduced perception accuracy during the approach phase, which was predominantly attributed to the shake generated as the robots engaged with plants (Mehta et al., 2014). Such interactions often induce image or motion blur, which significantly degrades the quality of the captured visuals. Moreover, because these blurs are caused by fast object dynamics or camera instabilities, they reduce image sharpness (Huihui et al., 2023). This reduction, in turn, increases the complexity of object detection from these images, which makes the task more error-prone. To address this concern, advanced techniques that can ensure reliable pedicel detection despite these complexities must be developed.

In this study, a fast and stable pedicel detection mechanism for the proposed 6D pose-based robust visual servoing (6DRVS) approach is introduced to efficiently harvest fruits that may be in motion. By adopting a straightforward computer-vision-based approach for pedicel detection in combination with video stabilization (ViS), the system effectively counters the challenges posed by fruit shake, which makes it suitable for use in real-time applications. The quick computation speeds offered by the computer-vision-based method make the system ideal for instantaneous image processing. Meanwhile, the ViS incorporated into the system minimizes image oscillations substantially. To improve video clarity, a Fourier spectrum-based band-stop filter (FSBF) that varies according to the measured blurriness is applied. Additionally, fast pedicel pose estimation is realized using FPFH-based 6D pose estimation. The primary goals of this work are to present techniques that enhance both the perception and approach precision of harvesting robots.

The contributions and novelty of this study are as follows:

- A computer-vision-based system tailored for swift and reliable detection of the 6D pose of pedicels in intricate agricultural settings is presented.
- The proposed system integrates ViS, FSBF, and FPFH to address issues related to video blurriness and improve the speed of 6D pose estimation for facilitating stable and fast pedicel detection.
- The efficacy of this system is tested rigorously in real-world conditions, specifically in two cucumber farms in Korea. The results underscore the robustness and competency of the system in actual farming scenarios.
- The problems and supplements encountered in the experiment are discussed in depth.

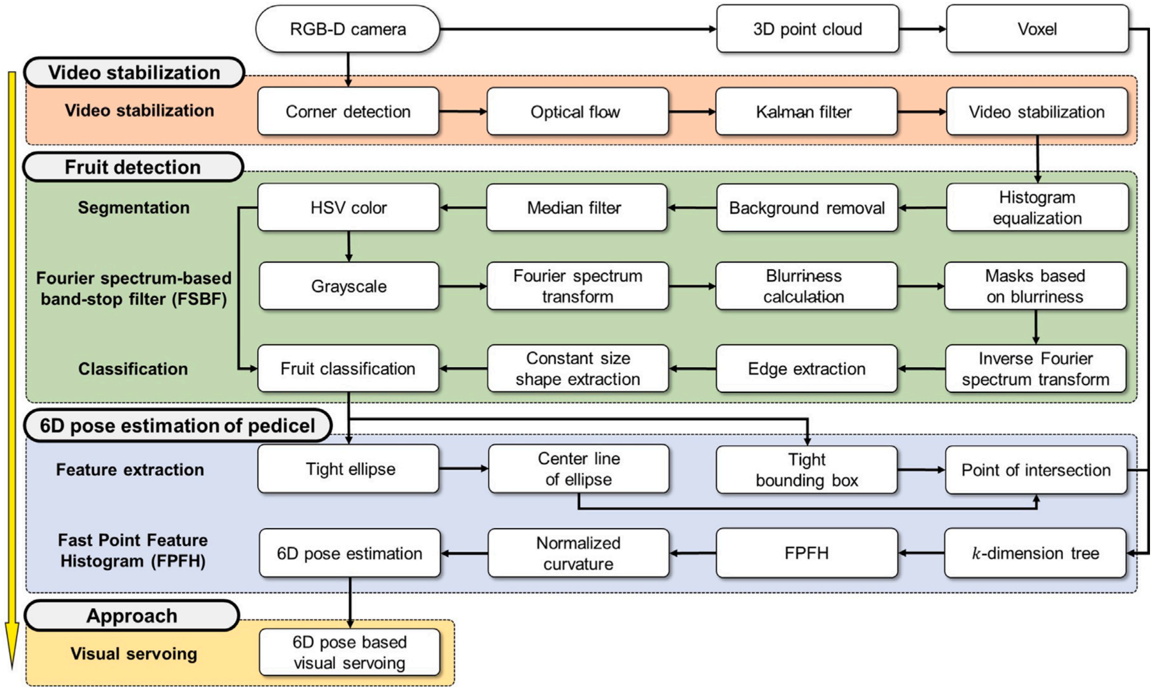


Fig. 2. Flowchart of 6D pose-based robust visual servoing (6DRVS) of fruit.

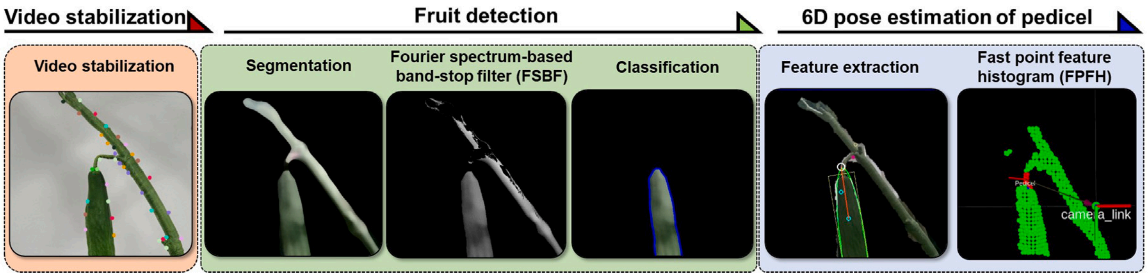


Fig. 3. 6D-pose-based robust visual servoing (6DRVS).

The remainder of this paper is organized as follows: Section 2 focuses on the estimation of 6D pose by using the 6DRVS. Section 3 describes the preliminary experiments conducted to evaluate the proposed system and analyzes the results. In Section 4, field experiments conducted in real-world environments are described, along with the methods used, and the results are analyzed. In Section 5, the associated problems and supplements are discussed comprehensively. The paper concludes with a summary of our findings and an outline for future research.

2. 6D Pose based Robust Visual Servoing (6DRVS)

In this section, the proposed method and approach for combining multiple technologies to realize efficient and precise harvesting are elucidated. The 6DRVS approach synergizes ViS, FSBF, and FPFH-based 6D pose estimation to yield optimal results in diverse agricultural environments. The process flow of the 6DRVS approach is depicted in Fig. 2, and this approach is designed to facilitate accurate object detection and pose estimation in real time. The results of executing the detection task in the 6DRVS approach according to this process flow are presented in Fig. 3.

2.1. Overall system

In a previous study, a cucumber-harvesting robot was developed (Park et al., 2023b). The hardware configuration of this robot is depicted in Fig. 1(a), and it is identical to that of the robot used in this study. The mobile platform (AgileX Robotics, Scout 2.0, China) is equipped with a manipulator (Universal Robots, UR5e, Denmark). An end-effector (EE) is attached to the tool center point of the manipulator. The custom-made EE is equipped with a hand-eye camera (local camera), cutting module, and grasping module. Additionally, to facilitate detection and maintain an adequate ambient lighting, an LED flash that is always switched on is mounted in EE. The aforementioned camera is a short-range stereo camera (Intel, D405, U.S.A) that provides high-resolution, color, and global shutter depth sensors for close-range computer vision applications. The parameters of the manipulator and camera of the harvesting robot capable of 6DRVS are summarized in Table 1.

Camera and hand-eye calibration are performed separately. The proposed system comprises a hand-eye camera and a UR5e robot. Camera calibration is performed using an ArUco marker to compute the camera coordinates. For hand-eye calibration, a calibration marker is positioned near the robot. Fig. 4 illustrates the relationships between

Table 1
Parameters of the manipulator and camera constituting the proposed system.

Camera		Robot manipulator	
Feature	Parameter	Feature	Parameter
Model	Intel RealSense D405	Model	UR5e
Type	Active stereo based RGB-D camera	Type	Universal robots
Resolution [Pixels]	640 × 480	DoF	6
FOV [deg]	87 (Horizontal) × 58 (Vertical)	End-effector	Custom
Depth accuracy [mm]	±2% at 500	Load [kg]	5
Ideal range [mm]	70 to 500	Accuracy [N·m]	0.3

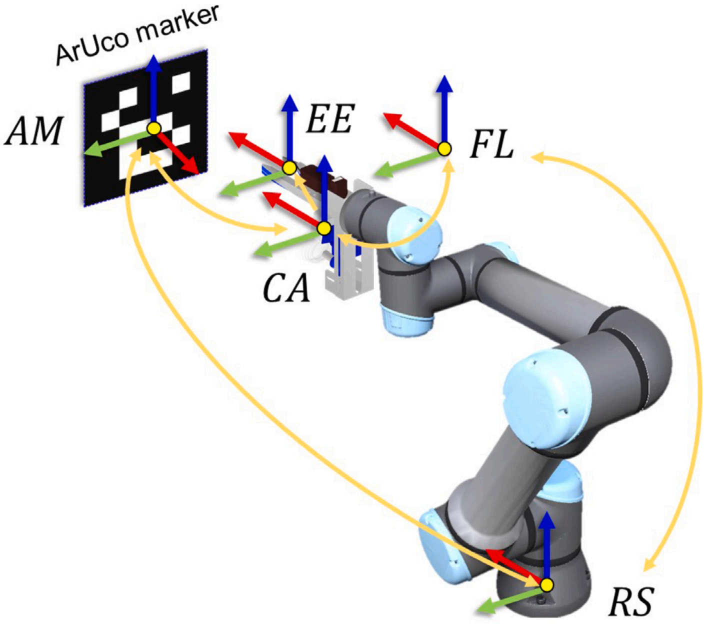


Fig. 4. Coordinates in the hand-eye system.

various coordinate systems: AM denotes the ArUco marker, EE denotes the EE, FL denotes the flange, RS denotes the robot's system, and CA denotes the camera. The coordinate transformation sequence is ted as follows:

$$ {}^{RS}_{FL}TR^{(j)} \cdot {}^{FL}_{CA}TR \cdot {}^{CA}_{AM}TR^{(j)} \cdot {}^{AM}_{RS}TR = I. \tag{1}$$

Here, I is the identity matrix. The hand-eye relationship is expressed as follows:

$$ {}^{FL}_{CA}TR = \left({}^{CA}_{AM}TR^{(j)}{}^{AM}_{RS}TR{}^{RS}_{FL}TR^{(j)}\right)^{-1}. \tag{2}$$

${}^{AM}_{RS}TR$ is a static transformation matrix from RS to AM, and it is calibrated by aligning the origin of E with the axes of AM. ${}^{CA}_{AM}TR^{(j)}$ is the transformation from AM to CA for the position of the jth robot, and it is determined using regression methods. ${}^{RS}_{FL}TR^{(j)}$ represents the transformation from FL to RS at the jth position, and it is obtained from forward kinematics. To ensure robustness, data from multiple positions are averaged:

$$ \widehat{{}^{FL}_{CA}TR} = \frac{1}{N_K}\sum_{i=1}^{N_K}\left({}^{CA}_{AM}TR^{(i)} \cdot {}^{AM}_{RS}TR \cdot {}^{RS}_{FL}TR^{(i)}\right)^{-1}. \tag{3}$$

N_K represents the count of different poses, and it is set to 10 in this study. The end effector's position, denoted as EE, is determined by the transformation from the flange to the robot's system at the jth position, represented as ${}^{RS}_{FL}TR^{(j)}$. This transformation is obtained from the robot's forward kinematics. However, to calculate the exact position of the end effector's cutting area, which is offset from the camera coordinates by 50 mm along the y-axis and 160 mm along the z-axis, additional conversion must be applied. This transformation can be represented as:

$$ {}^{CA}_{EE}TR = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 50 \\ 0 & 0 & 1 & 160 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4}$$

Then, the position of the EE in the robot's system can be calculated by:

$$ {}^{RS}_{EE}TR^{(j)} = {}^{RS}_{FL}TR^{(j)} \cdot {}^{FL}_{CA}TR \cdot {}^{CA}_{EE}TR. \tag{5}$$

This allows us to compute the exact position of the EE cutting area in the robot's coordinate system. ${}^{RS}_{EE}TR^{(j)}$, The EE cutting region coordinate F_c can be obtained as depicted in Fig. 1(b).

2.2. Video stabilization

In video stabilization, the primary goal is to mitigate the effects of camera shake and enhance video quality. This process encompasses three sequential steps: corner point extraction, optical flow computation, and motion smoothing via Kalman filtering.

2.2.1. Corner point extraction

Corner point extraction is the first process in video stabilization. Shi-Tomasi corner detection, which builds on the Harris corner detection method (Mstafa et al., 2020), identifies those points in video frames V that can be tracked reliably across successive frames. It focuses on the smallest eigenvalue of the second-moment matrix, which represents changes in the image gradient. The second-moment matrix M of a pixel is defined as follows:

$$M = \sum_{i} w_i \begin{bmatrix} V_{x_i}^2 & V_{x_i} V_{y_i} \\ V_{x_i} V_{y_i} & V_{y_i}^2 \end{bmatrix}, \tag{6}$$

where V_{x_i} and V_{y_i} denote the image gradients at the ith pixel, and w_i is the window function. This approach ensures that the points chosen for tracking are robust across successive frames, creating a foundation for accurate motion estimation.

By using the Shi-Tomasi method, n corner points $\mathbf{P} = \{p_1, p_2, \ldots, p_n\}$ are detected from a given video frame. Additionally, a user-defined point p_{user} is added to obtain the augmented set of points $\mathbf{P}' = \mathbf{P} \cup \{p_{\text{user}}\}$. The set $\mathbf{P}'$, containing a total of 100 points, is used in subsequent processes such as motion estimation using optical flow.

2.2.2. Optical flow

Following corner detection, optical flow computation is conducted using the Lucas-Kanade method to estimate motion between consecutive frames. For each detected corner point $\mathbf{p}_c$ in the set $\mathbf{P}'$, at location (x, y) in the frame at time t, the Lucas-Kanade method is used to calculate the flow vector (u_i, v_i):

$$\begin{bmatrix} V_x(\mathbf{p_c}) & V_y(\mathbf{p_c}) \end{bmatrix} \begin{bmatrix} u_i \\ v_i \end{bmatrix} = -V_t(\mathbf{p_c}), \tag{7}$$

where $V_x(\mathbf{p}_c)$, $V_y(\mathbf{p}_c)$, and $V_t(\mathbf{p}_c)$ represent the spatial and temporal intensity gradients at $\mathbf{p}_c$. The optical flow vectors $\mathbf{v}_i = [u_i, v_i]^T$ of each point $\mathbf{p}_c$ are then utilized in the Kalman filter for motion estimation.

2.2.3. Kalman filter for motion estimation

The Kalman filter plays a pivotal role in video stabilization by estimating and correcting camera motion to enhance video clarity. It operates in two main phases: prediction and update. Initially, the state vector $\mathbf{x}_k$, representing the camera's position and velocity, is predicted from the previous state $\hat{\mathbf{x}}_{k-1|k-1}$ and external control inputs u_k through:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}u_k, \tag{8}$$

where $\mathbf{A}$ is the state transition matrix, and $\mathbf{B}$ is the control input model. Updates leverage new observations $\mathbf{z}_k$:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}), \tag{9}$$

where $\mathbf{K}_k$ is the Kalman gain, informed by process noise covariance $\mathbf{Q}$ and measurement noise covariance $\mathbf{R}$:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R})^{-1}. \tag{10}$$

Dynamic adjustments are facilitated through:

$$\mathbf{P}_{k|k-1} = \mathbf{A}\mathbf{P}_{k-1|k-1}\mathbf{A}^T + \mathbf{Q}, \tag{11}$$

enhancing the filter's robustness. These configurations-specifically the calibration of $\mathbf{Q}$ and $\mathbf{R}$ are critical, determined through trial and error, to stabilize the video effectively. This methodical approach enhances video stabilization by compensating for camera motion, improving stabilization accuracy and reliability. Optical flow vectors $\mathbf{v}_i$ from all points in $\mathbf{P}'$ form the measurement vector $\mathbf{z}_k$, computed as $\mathbf{z}_k = \sum_{i \in \mathbf{P}'} \mathbf{v}_i$, crucial for the Kalman filter's update phase.

The final stage involves using the motion estimates from the Kalman filter to generate a stabilized video V_s. This is achieved by constructing the transformation matrix $\mathbf{T}_k$ from the estimated state vector $\hat{\mathbf{x}}_k$:

$$\mathbf{T}_k = \begin{bmatrix} \cos(\theta_k(\hat{\mathbf{x}}_k)) & -\sin(\theta_k(\hat{\mathbf{x}}_k)) & \Delta x_k(\hat{\mathbf{x}}_k) \\ \sin(\theta_k(\hat{\mathbf{x}}_k)) & \cos(\theta_k(\hat{\mathbf{x}}_k)) & \Delta y_k(\hat{\mathbf{x}}_k) \\ 0 & 0 & 1 \end{bmatrix}, \tag{12}$$

where $\theta_k(\hat{\mathbf{x}}_k)$, $\Delta x_k(\hat{\mathbf{x}}_k)$, and $\Delta y_k(\hat{\mathbf{x}}_k)$ represent the rotation angle and translations in the x and y directions, respectively, and they are derived from the state vector $\hat{\mathbf{x}}_k$ for the frame at time k. By applying this matrix $\mathbf{T}_k$ to each frame, stabilization is achieved as follows:

$$V_s(x, y, k) = \mathbf{T}_k \cdot V(x, y, k), \tag{13}$$

where $V_s(x, y, k)$ is the stabilized frame at time k, and $V(x, y, k)$ is the original frame. The stabilized video V_s can be obtained using this transformation.

2.3. Fruit detection

2.3.1. Segmentation

Given a V_s, the segmentation process starts with the enhancement of image contrast by means of histogram equalization. The objective of histogram equalization is to obtain a transformation function, such that the histogram of the transformed image is approximately uniform across all intensity levels. Let $p_r(r)$ be the probability density function (PDF) of the pixel intensities in V_s, as follows:

$$p_r(r) = \frac{h(r)}{V_t}, \tag{14}$$

where $h(r)$ is the number of pixels with intensity level r. The total number of pixels is V_t (image size of V_s is width V_w × height V_h). The cumulative distribution function (CDF) $C(r)$ is as follows:

$$C(r) = \sum_{i=0}^{r} p_r(i). \tag{15}$$

The transformation function $T(r)$ for each r is as follows:

$$V_{HE} = T(r) = (L-1) \times C(r), \tag{16}$$

where L is the total number of intensity levels, typically 256 for an 8-bit image. Each pixel in V_s with r is replaced with $T(r)$ to obtain V_{HE}, that is, the histogram equalized image. After histogram equalization, background removal is performed. In many applications, parts of an image with depths greater than a depth threshold d_d are considered the background. By using the depth information of every pixel in V_{HE}, one obtains

$$V_{BR}(p) = \begin{cases} V_{HE}(p) & \text{if } d_p \leq d_d \\ 0 & \text{otherwise.} \end{cases} \tag{17}$$

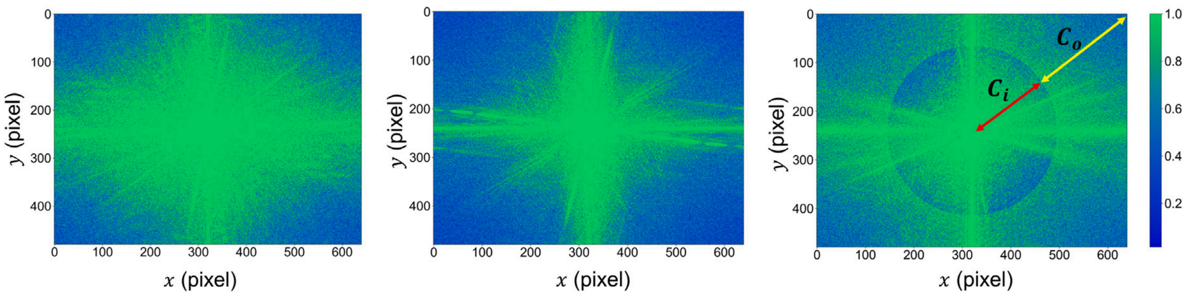
Here, V_{BR} is the background removal image. To enhance the image quality of V_{BR} and mitigate noise, a median filter is utilized. V_m is the resulting median filter image. Subsequently, the improved image V_m is transformed into the hue, saturation, value (HSV) color space to obtain the HSV image V_{HSV}. By building upon insights from the extant research on cucumber-harvesting robots, a specific color range in the HSV space, characterized by $H \in [30, 255], S \in [35, 200], V \in [5, 140]$, is used for segmentation. This leads to creation of the segmentation mask, wherein each pixel p is assigned a value of 1 if it falls within the aforementioned HSV range and 0 otherwise.

2.3.2. Fourier spectrum-based band-stop filter (FSBF)

To enhance image clarity by improving the accuracy of contour extraction and sharpening, the FSBF is employed (Fig. 5). As shown in Fig. 5(a), (b), a comparison of the scenarios with and without shaking reveals that the high-frequency area is widely distributed in the scenario without shaking. By using this distribution, blurriness can be calculated. In this work, by leveraging the fact that amplification of the high-frequency area in the presence of shaking increases image sharpness, a blurriness-based variable filter is applied to design the FSBF.

First, convert V_{HSV} to grayscale transformation image V_g to make it a single channel. V_g is subjected to a two-dimensional Fourier transform, which leads to the derivation of the magnitude spectrum F_s:

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} V_g(x, y) e^{-j(ux+vy)} dx dy. \tag{18}$$



(a) (b) (c)

Fig. 5. Fourier spectrum-based band-stop filter (FSBF) design: (a) Fourier spectrum corresponding to video without shaking, (b) Fourier spectrum corresponding to video with shaking, and (c) variable filter design using Fourier spectrum.

Here, e is the base of the natural logarithm, and j is the imaginary unit (equivalent to the square root of -1). From this expression, the magnitude spectrum, F_s is derived as follows:

$$F_s(u,v) = \log_{10}(|F(u,v)| + 1). \tag{19}$$

Blurriness F_b is computed as the average magnitude of the spectrum:

$$F_b = \frac{\sum F_s(u,v)}{V_t}, \tag{20}$$

A smaller F_b value indicates higher levels of blurriness in the video.

Second, to prevent such blurriness, a band-stop filter is designed (Fig. 5(c)). The inner and outer radii C_i and C_o, respectively, of this filter are influenced by F_b:

$$C_i = C_{ii} + C_{oo}F_b \tag{21}$$

$$C_o = C_i + C_{oo}. \tag{22}$$

Here, the constants C_{ii} and C_{oo} are determined empirically to define the design of the band-stop filter. The mask $M_f(u,v)$ of the band-stop filter is constructed using the values of C_i and C_o, and it is defined as follows:

$$M_f(u,v) = \begin{cases} 1 & \text{if } (u-u_c)^2 + (v-v_c)^2 \le C_i^2 \textit{ and } (u-u_c)^2 + (v-v_c)^2 \ge C_o^2 \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

The spectrum is then filtered using this mask, as follows:

$$F_f(u,v) = F_s(u,v) \times M_f(u,v). \tag{24}$$

A predetermined $M_f(u,v)$ is used to amplify the frequencies within this mask for enhancing video clarity. After filtering, the inverse Fourier transform is applied to obtain the sharpened video frame V_f:

$$V_f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F_f(u,v) \cdot e^{j(ux+vy)}\, du\, dv. \tag{25}$$

2.3.3. Classification

In fruit classification, it is essential to differentiate a fruit from its stem and any surrounding foliage. In cases where the color of the stem or background is different from that of the fruit (e.g., tomatoes, non-green bell peppers, strawberries, apples, and grapes), classification can be accomplished straightforwardly by using the V_{HSV}. However, the color of cucumbers, the primary object fruit in this study, is similar to that of the stem and background, that is, predominantly green.

In the case of cucumbers, characteristics such as fruit length and diameter significantly influence consumer preferences. Mature cucumbers typically have various shapes and sizes, and consumers from different regions have specific preferences for various fruit shapes (Zhang et al., 2019). In Korea, cucumbers are deemed ready for harvesting when their length and diameter are approximately 250 mm and 40 mm, respectively. Therefore, harvest-ready cucumbers can be identified based on their measured length and width (aspect ratio of 1:6).

However, during the robot's approach (i.e., pedicel entering the cutting area), the region of interest (ROI) of the hand-eye camera installed on the EE is restricted. Consequently, only half of the cucumber is detected in the ROI of the hand-eye camera. Given that the aspect ratio of cucumbers is approximately 1:6, shapes with aspect ratios of 1:3–1:6 are classified as cucumbers. By leveraging this attribute, bounding boxes b_b are generated around each detected outline in V_f. Any b_b that does not conform to the aforementioned aspect ratio range is eliminated. Objects that remain encapsulated within these b_b are classified as fruits O_{fruit}.

2.4. 6D pose estimation of pedicel

2.4.1. Calculation pertaining to feature point extraction

After classifying the cucumber as O_{fruit}, the subsequent step is to discern the features corresponding to the approximate position of the pedicel P_{pe}. The pedicel is significant because it provides information about the orientation and attachment of the cucumber.

The initial step is to delineate the contour of O_{fruit}. fitting an ellipse around this contour is a standard approach in computer vision for shape analysis. Subsequently, an ellipse is fitted around this detected contour. A bounding box b_{be} encapsulating the ellipse is computed, and it is designated by its vertices, (x_1, y_1) and (x_2, y_2). Considering that the pedicel's position is roughly 3 mm higher than that of the cucumber, the bounding box is translated upward by this distance, resulting in $(x_1, y_1 - \delta)$, where $\delta = 3$ mm denotes the equivalent translation distance in the image. The point of intersection of the elongated major axis of the ellipse and the relocated b_{be} is P_{pe}. P_{pe} is p_{user}, and it is included in one of the 100 corner points detected during ViS.

2.4.2. Fast point feature histogram (FPFH)

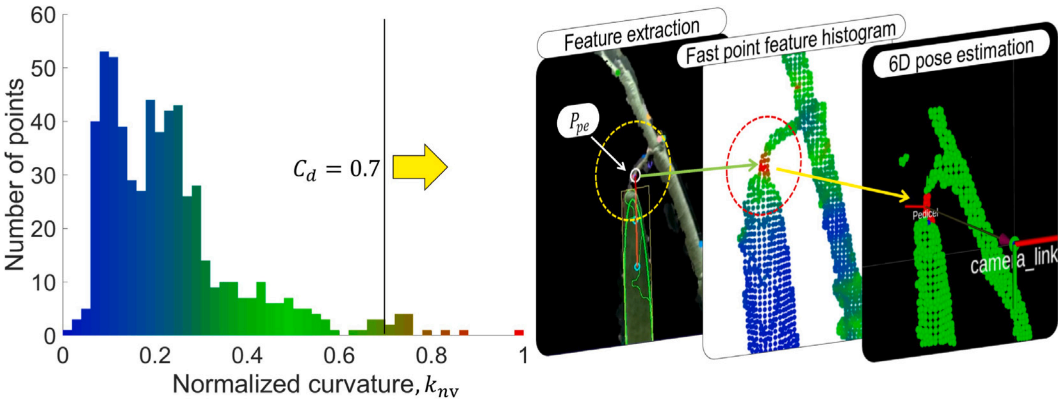
By using a specific pixel of the cucumber represented by $P_{pe}(u_f, v_f)$, adjacent point clouds are grouped to form a region of interest (ROI) (Fig. 6(b)). To convert the coordinate of this pixel into 3D point cloud attributes, the depth information recorded by the depth camera and the intrinsic parameters of the camera, (f_x, f_y) are used. Using (V_w, V_h) and d_p, a 3D position vector of point $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ within the camera's coordinate system is derived as follows:

$$\mathbf{X} = \frac{(V_w - c_x)\cdot d_p}{f_x}, \tag{26}$$

$$\mathbf{Y} = \frac{(V_h - c_y)\cdot d_p}{f_y}, \tag{27}$$

$$\mathbf{Z} = d_p. \tag{28}$$

Here, (c_x, c_y) symbolizes the principal point coordinates of the image, typically at the center, and (f_x, f_y) denotes the camera's focal lengths along the x and y axes. The ROI point cloud is aligned with $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$,



(a)

(b)

Fig. 6. Histogram distribution of curvature: (a) curvature distribution and (b) curvature histogram.

and the set of point clouds inside the ROI is $\mathbf{p}_r$. Considering the given context, the for $\mathbf{p}_r$ could be defined as follows:

$$\mathbf{p}_r = \mathbf{p}_{po}(\mathbf{X}_i, \mathbf{Y}_i, \mathbf{Z}_i) \mid \sqrt{(\mathbf{X}_i - \mathbf{X})^2 + (\mathbf{Y}_i - \mathbf{Y})^2 + (\mathbf{Z}_i - \mathbf{Z})^2} \leq R_o. \tag{29}$$

$\mathbf{p}_{po}(\mathbf{X}_i, \mathbf{Y}_i, \mathbf{Z}_i)$ represents the 3D points in the point cloud. R_o is a predefined radius that determines the size of the ROI around the point of interest. The set $\mathbf{p}_r$ consists of all points within the ROI that are within a distance R_o from the point of interest. For every point p_i within $\mathbf{p}_r$, the k-nearest neighbors are identified. By using these neighbors, the surface's normal and curvature at each point are computed. The FPFH of a point p_i is expressed as the following histogram (Rusu et al., 2009):

$$\mathbf{FPFH}(p_i) = \mathbf{SPFH}(p_i) + \frac{1}{k}\sum_{j=1}^{k} \frac{1}{\omega_k}\mathbf{SPFH}(p_j), \tag{30}$$

where k denotes the number of k-nearest neighbors of p_i, and ω_k is a weight function that is typically defined as the inverse of distance. $\mathbf{SPFH}(p_i) = [\alpha, \phi, \theta]$ denotes the SPFH. As depicted in Fig. 7, the angles α, ϕ, and θ are the angles between the normal vectors $\mathbf{n}_i$ and $\mathbf{n}_j$ of points p_i and p_j, respectively, and they are computed as follows:

$$\alpha = \mathbf{v} \cdot \mathbf{n}_i, \tag{31}$$

$$\phi = \mathbf{u} \cdot (\mathbf{n}_i - (\mathbf{v} \cdot \mathbf{n}_i)\mathbf{v}), \tag{32}$$

$$\theta = \arctan\left(\frac{\mathbf{w} \cdot \mathbf{n}_i}{\mathbf{u} \cdot \mathbf{n}_i}\right). \tag{33}$$

The unit vectors $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$ are defined as follows:

$$\mathbf{u} = \frac{\mathbf{n}_i \cdot \mathbf{n}_j}{\|\mathbf{n}_i \cdot \mathbf{n}_j\|}, \tag{34}$$

$$\mathbf{v} = \mathbf{n}_j - (\mathbf{u} \cdot \mathbf{n}_j)\mathbf{u}, \tag{35}$$

$$\mathbf{w} = \mathbf{u} \cdot \mathbf{v}. \tag{36}$$

FPFH employs 33 bins per point in the point cloud to succinctly capture the extensive geometric and spatial attributes of the local neighborhood around a point. These bins are pivotal for precisely depicting the complex features and variations of the local surface. They incorporate unique angular features and relationships between the point of interest and its neighbors to provide a robust representation of the local surface morphology. The sum of squared differences k_k between the FPFH histograms of a specific point and those of its neighboring points is calculated accordingly:

$$k_k(p_i, p_j) = \sum_{k_k=1}^{33} \left(\mathbf{FPFH}(p_i)_k - \mathbf{FPFH}(p_j)_k\right)^2. \tag{37}$$

The normalized variance k_{nv} of k_k for each point is calculated using the following equation:

$$k_{nv}(p_i) = \frac{k_k(p_i) - k_{k\,\min}}{k_{k\,\max} - k_{k\,\min}}. \tag{38}$$

This approach enables visualization of the local geometric properties and differences in properties between each point and its neighbors in the point cloud. In this context, a calculated k_{nv} serves as a measure of the curvature at each point within a point cloud. Given a set of points P with associated curvatures denoted by k_{nv}, the subset P_b is defined as $P_b = \{p_i \in P \mid k_{nv}(p_i) \geq C_d\}$, where each point p_i in P_b has a curvature k_{nv} greater than or equal to the predetermined curvature constant C_d (Fig. 6(a)). This subset P_b is called $p_{pedicel}$ (Fig. 6(b)).

2.4.3. 6D pose estimation

The iterative closest point (ICP) algorithm is employed to accurately estimate the 6D pose of the pedicel within the coordinate system of the EE. The algorithm facilitates the alignment of two point clouds: the target point cloud $p_{pedicel}$, which represents the pedicel's current position, and a source point cloud. The source point cloud, denoted as p_{co}, is initially a copy of $p_{pedicel}$ but is transformed iteratively to best match $p_{pedicel}$.

The transformation matrix T, which aligns p_{co} with $p_{pedicel}$, is computed through the ICP algorithm. The matrix Q_c, representing the EE's orientation and position, serves as a reference for aligning p_{co} to the EE's coordinate system. The ICP algorithm iterates, minimizing the distance between p_{co} and $p_{pedicel}$, until convergence criteria based on either a minimum distance threshold or a maximum number of iterations are met. The mathematical objective of this alignment is expressed as follows:

$$T^* = \arg\min_T \sum_{i=1}^{n} \left\| T \cdot p_i - Q_c \cdot p_{co} \right\|^2. \tag{39}$$

The derived transformation matrix T^* from this process encapsulates the 6D pose (position and orientation) of the pedicel relative to the EE. It is important to note that p_{co} is utilized as a mutable representation of $p_{pedicel}$ during the ICP process, allowing for the calculation of the transformation matrix without altering the original pedicel point cloud data.

2.5. 6D pose based visual servoing

6D pose based VS is employed to compute the control inputs that minimize the error between the current pose of the robot's EE and the desired pose located within the EE's truncation region. $\mathbf{e}_t$ and $\mathbf{e}_r$ denote the translational (position) and rotational (orientation) errors, respectively. The translational error is the difference between the current and desired positions:

$$\mathbf{e}_t = \mathbf{p}_d - \mathbf{p}_{cp}. \tag{40}$$

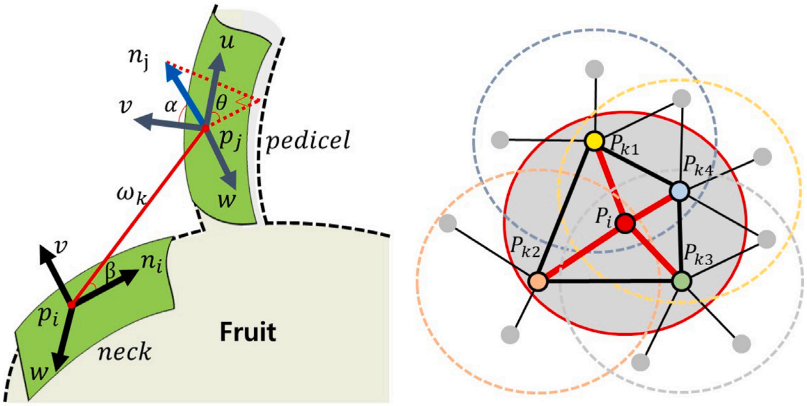


Fig. 7. For the query point p_i, the algorithm determines its simplified point feature histogram (SPFH) values by pairing it with its neighbors (shown in red). The SPFH values are then re-weighted using its k-neighbors to form the p_i fast point feature histogram (FPFH), where the added connections are depicted in black. A few pairs, highlighted by thicker lines, are counted twice.

Here, $\mathbf{p}_d$ denotes the desired position, and $\mathbf{p}_{cp}$ is the current position. In Fig. 1(b), F_{c*} is $\mathbf{p}_d$, and F_c is $\mathbf{p}_{cp}$. The rotational error is computed based on the angle difference between the current and desired orientations:

$$\mathbf{e}_r = \frac{1}{2}(\omega_x \mathbf{R}_r + \mathbf{R}_r^T \omega_x^T). \tag{41}$$

In this equation, $\mathbf{R}_r$ denotes the current orientation matrix, and ω_x is the skew-symmetric matrix derived from the axis-angle representation of the desired orientation. After establishment of the error vector, the following control laws are applied:

$$\dot{\mathbf{p}}_{cp} = -\lambda_p \mathbf{e}_t \tag{42}$$

$$\boldsymbol{\omega} = -\lambda_\omega \mathbf{e}_r. \tag{43}$$

Here, $\dot{\mathbf{p}}_{cp}$ is the linear velocity of the EE, $\boldsymbol{\omega}$ is its angular velocity. In our control scheme, the gain for linear velocity, denoted as λ_p, is carefully selected to ensure a balance between responsiveness and stability during the robot's approach to the target. The value of λ_p is chosen based on empirical testing and system dynamics analysis, considering the need for precise yet smooth linear motion towards the pedicel. Similarly, the gain for angular velocity, denoted as λ_ω, is set to optimize the system's ability to correct orientation errors without inducing oscillations. This control law modulates the EE's velocity to ensure that the error vector converges to zero. In this specific context, the desired pose corresponds to the 6D pose of the pedicel denoted by T^*. Thus, the desired position and orientation can be expressed as follows:

$$T^* = \begin{bmatrix} \mathbf{R}_d & \mathbf{p}_d \\ 0 & 1 \end{bmatrix}. \tag{44}$$

The current pose is characterized by the 6D pose of the EE, represented as T_e. Therefore, the current position and orientation can be expressed as follows:

$$T_e = \begin{bmatrix} \mathbf{R}_r & \mathbf{p}_{cp} \\ 0 & 1 \end{bmatrix}. \tag{45}$$

To move the EE from its current pose T_e to the desired pose T^*, e is used within a control loop to determine the robot's movements.

3. Evaluation of proposed system

To assess the efficacy of the 6DRVS and validate it, preliminary experiments were conducted in laboratory settings that closely emulated a smart farm greenhouse by replicating the conditions employed in prior studies (Park et al., 2023b). The parameters for 6DRVS are shown in Table 2. These evaluations covered two critical dimensions: perception accuracy and approach accuracy. During these experiments, eight motion-capture cameras were deployed to measure the poses of the fruit (in this case, cucumber) and the EE.

Table 2
6DRVS of the parameters.

Parameters	Specification
Process noise covariance, Q	0.001
Measurement noise covariance, R	0.01
Pixels with intensity, r	3
Image size, $V_w \times V_h$	640 × 480 [pixels]
Depth threshold, d_d	300 [mm]
Inner radii, C_i	30 [pixels]
Outer radii, C_o	250 [pixels]
Parameters of the camera, (f_x, f_y)	545, 448 [pixels]
Predefined radius, R_o	30 [mm]
Predetermined curvature constant, C_d	0.7
Control gain, λ_p, λ_ω	0.2, 0.2

3.1. Perception accuracy

The perception accuracy evaluations covered three critical dimensions: Qualitative analysis of captured images: As depicted in Fig. 8, in the absence of the 6DRVS, artifacts such as image and motion blurs were often generated, which hampered accurate detection. However, after 6DRVS integration, the numbers of such disturbances decreased noticeably, leading to clearer and more consistent detection. Additionally, a Fourier spectrum-based quantitative method (expressed in Eqs. (18)–(20)) was used to calculate blurriness. The findings indicated that when the 6DRVS was applied, the blurriness was 0.548, whereas without the 6DRVS, it was 0.414 (without shaking fruits: 0.792). These values indicated a clear decrease in blurriness when using the 6DRVS, which underscored the potential of the proposed system to achieve high perception accuracy even when relying solely on computer vision techniques.

Pixel representations were charted to visualize the stabilization afforded by the 6DRVS more clearly (Fig. 9). The results indicated that the 6DRVS enhanced the smoothness and stability of pedicel detection. Frame rate performance in detection: In the 6DRVS-based pedicel detection process, frame rates of 15–37 fps (640 × 480 images) were achieved, which represented a remarkable improvement over the frame rates of 3–18 fps achieved using traditional deep-learning-based recognition techniques. Such continuous and rapid detection is paramount for VS, given the inherent real-time manipulation requirements of the method. The results of this experiment affirmed that increasing the detection robustness significantly augmented the performance and reliability of the 6DRVS system.

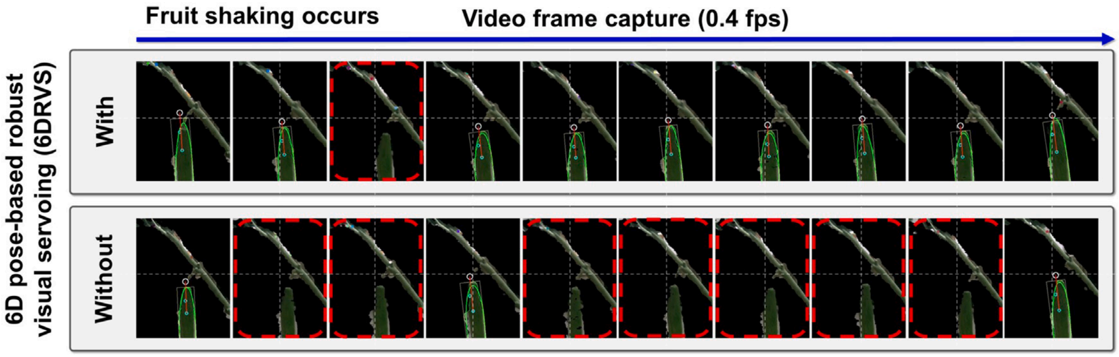

Fruit shaking occurs
Video frame capture (0.4 fps)
6D pose-based robust visual servoing (6DRVS)
With
Without

Fig. 8. Detection test with/without 6D pose-based robust visual servoing (6DRVS) with occurrence of fruit shake.

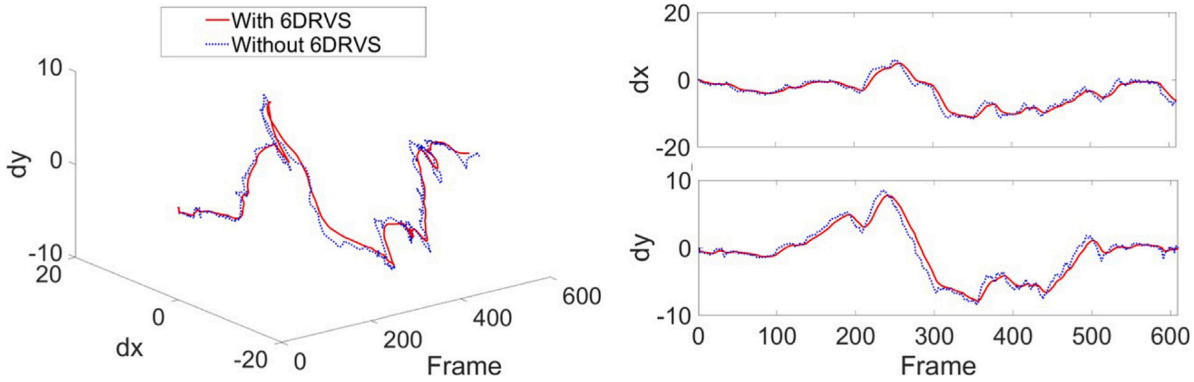
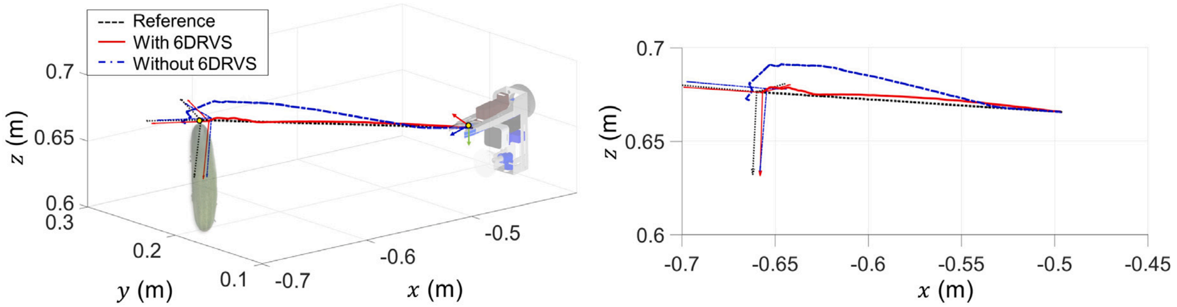


Fig. 9. Stabilization of pedicel detection with/without 6DRVS in situations with shaking fruits.



(a)

(b)

Fig. 10. Approach trajectory with/without 6D pose-based robust visual servoing (6DRVS): (a) 3D plot and (b) 2D plot (xz).

3.2. Approach accuracy

First, the trajectory with/without the implementation of 6DRVS was examined. The reference trajectory was determined by the shortest distance between the pedicel and the EE, which represents the trajectory under optimal conditions. The resulting trajectories are depicted in Fig. 10. The trajectory obtained by applying the 6DRVS method adeptly traces the reference trajectory. This implies that the incorporation of 6DRVS allowed for more accurate and efficient alignment of the trajectory with the reference value, thus demonstrating the system's proficiency in terms of maintaining optimal proximity.

Second, the EE was set randomly within the camera's ROI to ensure that the fruit was visible, and it was subsequently moved to the target pose. This process was repeated $n = 50$ times. The objective, depicted in Fig. 1(b), was to move F_c on the image plane to F_{c*}. By using the motion-capture system, F_c was designated as T_e according to Eq. (45). The position of F_{c*} was aligned with the fruit's coordinate position F_o to guarantee that the pedicel was placed within the truncation region of the EE. Here, F_{c*} corresponds to T^* from Eq. (44). The pose error is $PE = T^* - T_e$. The root mean square error (RMSE) was used in the validity calculations:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} PE^2}{n}}. \tag{46}$$

Table 3 and Fig. 11(b) consolidate the outcomes of the preliminary experiments conducted to assess the efficacy of the 6DRVS system. These experiments were crucial for evaluating how well the system could align the pedicel within the truncation region of the proposed EE. The findings indicated that the pedicel was positioned accurately within this specific region, which is a prerequisite for subsequent operations. With the assistance of the 6DRVS system, the EE could be moved proficiently in real-time and aligned with the pedicel, as depicted in Fig. 11(a). This implied that the proposed visual servoing system

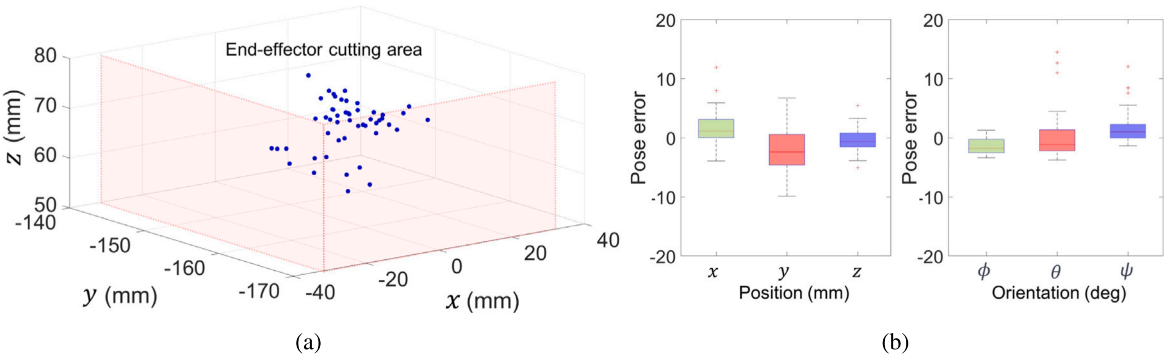


Fig. 11. Pose error of 6D pose-based robust visual servoing (6DRVS): (a) 3D scatter plot and (b) pose error.

Fig. 12. Experimental environment.

Table 3
Results of preliminary experiments for 6DRVS.

	Position [mm]			Orientation [deg]		
	x	y	z	ϕ	θ	ψ
PE average	1.60	−2.00	−0.37	1.50	−0.13	1.61
RMSE	3.27	4.23	1.77	2.00	3.76	3.14

was able to maintain the pedicel's alignment within the cutting region, thereby ensuring the precision and efficiency of the maneuvering process.

4. Field experiments

4.1. Experimental setup

To validate the effectiveness and reliability of the 6DRVS, field experiments were conducted on cucumbers, which were the representative target fruit in these trials. The experiments proceeded in two distinct cucumber farms located in Korea, as depicted in Fig. 12, to ensure the availability of a diverse and comprehensive set of environmental conditions and variables for assessing the versatility and adaptability of the 6DRVS. In total, 100 cucumbers were harvested during these experimental trials, and the resulting data were gathered and analyzed. The experimental setup, execution, and results were documented to assess the perception and approach accuracy of the 6DRVS under real-world, practical conditions. The 6DRVS methodology was evaluated using two distinct criteria: perception accuracy and approach accuracy. First, the perception accuracy was evaluated using standard metrics such as precision, recall, accuracy, and F1-score. These metrics are commonly used to assess the performance of classification models. Precision measures the proportion of true-positive identifications among all positive identifications made; recall evaluates the proportion of actual positives that are correctly identified; accuracy assesses the overall correctness of the model; and F1-score provides a balance between precision and recall. These metrics are crucial for understanding the effectiveness and reliability of the classification models used in this study. Second, approach accuracy was assessed by counting the number of fruits that the system approached successfully. Owing to the practical challenges associated with deploying a motion-capture system in field environments, the length of the cucumber pedicels after they were cut by the EE was used as an indicative metric of position accuracy. In addition, system performance was evaluated against the approach time measured in a previous study (Park et al., 2023b) (second approach time in the previous study).

Table 4
Field experiment results of proposed 6DRVS.

6DRVS	Perception accuracy		Approach accuracy	
	Success rate [%]	Success [pcs]	Success rate [%]	Success [pcs]
Without	70.00	35	77.14	29
With	90.00	45	82.22	37

Table 5
Pedicel detection using 6DRVS: precision, recall, accuracy, and F1-score.

6DRVS	Precision	Recall	Accuracy	F1-score
Without	0.854	0.795	0.700	0.824
With	0.957	0.938	0.900	0.947

4.2. Experimental result

4.2.1. Perception accuracy

The results of the experiment are presented in Table 4. Implementation of the 6DRVS system yielded improvements across all evaluated metrics. These outcomes highlighted the crucial role of the 6DRVS system in the fruit detection task within the harvesting process. As

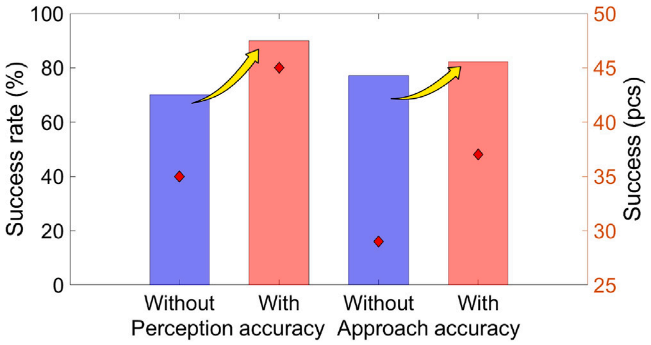


Fig. 13. Results of field experiment.

summarized in Table 5, performance metrics such as precision, recall, accuracy, and F1-score were computed. A comparative analysis with prior experiments revealed that the perception accuracy improved. The highest indicators in the previous study were precision, recall, accrual, and F1-score, and their values were 0.957, 0.927, 0.890, and 0.942, respectively (Park et al., 2023b). When using the 6DRVS proposed herein, the corresponding values were 0.957, 0.938, 0.900, and 0.947.

4.2.2. *Approach accuracy*

A comparison of the success rates achieved in the experiments with and without the use of the 6DRVS, a significant improvement was observed upon the integration of the 6DRVS. Specifically, in the absence of the 6DRVS, the success rate was 77.14%. By contrast, when the 6DRVS was used, the success rate increased noticeably to 82.22%. This enhancement, illustrated in Fig. 13, clearly demonstrated the beneficial impact of incorporating the 6DRVS into the harvesting process.

The approach accuracy in the cucumbers cutting process was assessed by measuring the post-cut stem length, that is, the length of the cut pedicel. The range of pedicel lengths corresponding to an accurate approach was defined as 0 to 30 mm. If the length of the cut pedicel fell within this range, the approach was considered accurate. The observed lengths of the cut pedicels in this study ranged from 1 to 20 mm, encompassing both the minimum and maximum extents of the cut lengths. These findings indicated that the EE executed the approach with a high level of precision, aligning accurately with the intended target pose. This precision in terms of aligning and cutting within this specified range of pedicel lengths reflected the effectiveness and accuracy of the approach method used herein.

Compared to the previous approach time of 30.7 s, in this study, the approach time decreased slightly to 29.3 s (Park et al., 2023b). This improvement was attributed to the application of the 6DRVS, which was designed to enhance the precision and stability of robots' movement trajectory. Owing to the stabilization of its approach trajectory, the time required by the robot to approach its target is reduced. Thus, the robot can complete its tasks more efficiently. This suggests that the 6DRVS improved the control and accuracy of robotic movements, thereby enhancing the overall performance of the specific task.

5. Discussion

5.1. *Perception issue*

The fruit harvesting domain introduces unique challenges in terms of perception and manipulation, especially considering the diverse nature of fruits. These challenges include predicting the direction of fruits, prioritizing cluster fruits, and occlusion-related issues.

5.1.1. *Predicting the direction of fruits*

The morphologies and growth patterns of fruits vary widely, and they directly influence the complexity of robotic harvesting tasks. The inherent linear shapes of elongated fruits, such as cucumbers, provide a clear indication of the stem's direction. This attribute ensures relatively straightforward mapping of robotic EEs to ensure their alignment with the fruit's axis and effective execution of the harvesting task. By contrast, cluster fruits, such as tomatoes, present a more intricate scenario. These fruits grow in bunches, and they are often tangled and overlap each other. Consequently, the directions of their pedicels are neither linear nor consistent. This variation demands a more sophisticated perception mechanism to accurately predict the direction of each fruit within the cluster. Mistakes in predicting this direction can lead to improper cuts, potential damage to the fruit, or incomplete harvesting. Hence, for cluster fruits, direction prediction is not only more complex but also critical to ensure efficient and damage-free harvesting.

5.1.2. *Prioritizing cluster fruits*

Cluster fruits introduce a layered challenge in robotic harvesting. A wrong choice could damage neighboring fruits, generate inefficient harvesting sequences, or cause the system to miss ripe fruits while picking less mature ones. For this reason, a harvesting priority should be established. The factors influencing this priority include ripeness of the fruit, ease of access to adjacent fruits, and predicted time to harvest. Approaches to issue, such as harvest ordering, provide the basis for solving problems with situations to determine the order of harvest (Park et al., 2023b). However, given the dynamic nature of cluster growth and the variabilities across different crops and growth conditions, there is an emerging need for more adaptive and context-aware priority-determination systems. These systems are expected to leverage real-time data, possibly coupled with predictive analytics, to make real-time decisions about which fruit to harvest next. Such approaches would not only ensure the desired produce quality but also increase the efficiency and effectiveness of robotic harvesters.

5.1.3. *Occlusion issues*

Detecting fruits that are occluded by obstacles and subsequently accessing their pedicels is significantly challenging. In this study, the authors focused on occluded fruits and conducted experiments in open spaces after defoliation. To address occlusion, Kim et al. (2023) proposed a methodology centered on cucumber segmentation and occlusion recovery. They employed amodal segmentation coupled with a U-net reconstruction network. This segmentation technique diligently recovered the parts hidden from view by considering both the visible and obscured portions of a cucumber. Following detection, a skeleton was extracted based on the cucumber's identified region to pinpoint the pedicel meant for cutting.

However, challenges persist. Even with successful pedicel detection in an occluded cucumber, the dense foliage surrounding the cucumber can obstruct the EE's trajectory. To solve this problem, SepúLveda et al. (2020) are investigating the use of dual-arm robots. Their approach aims to recognize occluded fruits and, subsequently, clear the obstructing foliage by leveraging the direction vector of the leaves, thus creating an accessible path to the target fruit. This strategy hinges on cooperative control between the two robot arms. Such innovations hint at future research in this domain, which seems to be poised to explore the potential of dual-arm harvesting robots.

5.2. *Control issue*

In this study, a specific controller for manipulator control was not designed. Instead, a conventional approach was adopted, and 6DRVS was implemented by employing the widely used position-based VS method. During harvesting, the manipulator or EE may come into contact with the fruit, causing the fruit to shake. Such shaking can abruptly shift the fruit's coordinates, which may cause the manipulator to move abruptly. Such abrupt movement can, in turn, induce secondary contact-induced fruit shaking. From the perception standpoint, such shaking can induce motion blur, thereby increasing the complexity of the harvesting process and increasing the detection time, which extends the overall harvesting duration.

To address these challenges, Xu et al. (2022) introduced adaptive VS. This method adaptively adjusts the parameters of the transformation matrix to minimize the distance between the EE's trajectory and the reference trajectory. Their experimental results demonstrated that the EE was able to track the reference trajectory with high precision by using adaptive learning. Furthermore, Mehta et al. (2016) developed a robust image-based visual servo controller to adjust a robot manipulator with respect to a target fruit in the presence of unknown fruit movements. A Lyapunov-based stability analysis ensured uniform ultimate bounded control of the robot EE relative to the target fruit.

While research on RVS through controller design is ongoing, there is a pressing need for additional research in the agricultural sector, particularly on harvesting robots. The inherent complexities and dynamic nature of agricultural environments, combined with the importance of precise and gentle handling, underscore the significance of these efforts.

5.3. Delayed harvesting time issue

The principal factors contributing to the delayed harvesting time are twofold: firstly, the imperative to minimize potential damage to the fruits and surrounding vegetation; and secondly, the necessity for precise and cautious navigation to access the pedicel. To elaborate, the robotic system is designed with an inherent emphasis on the delicate handling of agricultural produce to ensure that the quality of the fruits remains uncompromised throughout the harvesting process. This entails meticulous control and slower movements, which, while enhancing safety and precision, inadvertently extend the overall time required for harvesting.

Moreover, the pedicel, which is crucial for a successful harvest, necessitates accurate identification and approach. The complexity of this task is magnified by the diverse orientations and positions of pedicels in a natural setting, coupled with the dynamic environmental conditions of outdoor farms. These factors necessitate a careful and measured approach to accurately position the harvesting mechanism, further contributing to the lengthier harvesting time.

To mitigate these challenges and potentially expedite the harvesting process without compromising on safety or accuracy. These include: (1) leveraging more sophisticated machine learning algorithms that can more accurately and rapidly identify the pedicel and assess the optimal approach path. This could reduce the time spent on these tasks while maintaining high precision. (2) the deployment of advanced sensing technologies could provide richer environmental data, enabling quicker adaptation to varying conditions and more efficient path planning.

6. Conclusions

This study introduced a fast and stable pedicel detection approach for RVS, supplemented with ViS and fast pedicel detection, to realize automated harvesting of shaking fruits. This study successfully implemented the 6DRVS system, a composite of ViS, FSBF, and FPFH-based 6D estimation. The inclusion of ViS effectively addressed motion blur, ensuring stable segmentation and classification during image preprocessing. Additionally, video clarity was improved by using an FSBF that adapts to blurriness. Following this stabilization, the system estimated the approximate pedicel location through coordinate estimation. Subsequently, point clouds within a defined range around this estimated location were extracted. By utilizing the FPFH, the system was able to differentiate between fruits and pedicels based on curvature differences in the normal vectors of the point cloud. Clusters of point clouds exceeding a certain curvature threshold were identified as pedicels. Then, 6DRVS was implemented to locate the EE can until extraction of the 6D pose.

The 6DRVS, with its capability to compute a fast and stable 6D pose at 19–37 fps, guarantees real-time visual servoing. This ensures effective fruit harvesting, even when fruits shake. Preliminary experiments were conducted in a lab environment mirroring real smart farms to evaluate the 6DRVS. The findings confirmed the system's ability to robustly servo to the target position, and location errors were analyzed to identify improvements over previous studies. For further validation, field tests involving 100 repetitions of the experiment were conducted at two farms in Korea. The system was evaluated in terms of its precision, recall, accuracy, F1-score, perception accuracy, and approach accuracy. The results obtained using the 6DRVS (0.957, 0.938, 0.900, 0.947, 90.00%, 82.22%) indicated a performance enhancement compared the results obtained when not using the 6DRVS (0.854, 0.795, 0.700, 0.824, 70.00%, 77.14%). In addition, the access rate improved from 77.14% to 82.22%. These experimental results convincingly attest to the efficacy of the 6DRVS system. The enhancements provided by the 6DRVS system can significantly optimize the fruit harvesting process, leading to improved productivity. In conclusion, application of the 6DRVS system is promising from the perspective of enhancing automated fruit harvesting.

CRediT authorship contribution statement

Yonghyun Park: Writing – original draft, Writing – review & editing, Visualization, Validation, Software, Methodology, Investigation, Conceptualization. **Changjo Kim:** Software, Methodology, Investigation, Conceptualization. **Hyoung Il Son:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This research was supported, in part, by the Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry (IPET) through the Smart Farm Innovation Technology Development Program funded by the Ministry of Agriculture, Food and Rural Affairs (MAFRA) (421031-04), and, in part, by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF–2023R1A2C1003701).

References

Arad, B., Balendonck, J., Barth, R., Ben-Shahar, O., Edan, Y., Hellström, T., Hemming, J., Kurtser, P., Ringdahl, O., Tielen, T., et al., 2020. Development of a sweet pepper harvesting robot. J. Field Robotics 37 (6), 1027–1039. http://dx.doi.org/10.1002/rob.21937.

Bechar, A., 2021. Agricultural robotics for precision agriculture tasks: concepts and principles. Innov. Agric. Robot. Precis. Agric.: Roadmap Integr. Robot. Precis. Agric. 17–30. http://dx.doi.org/10.1007/978-3-030-77036-5_2.

Campbell, M., Dechemi, A., Karydis, K., 2022. An integrated actuation-perception framework for robotic leaf retrieval: Detection, localization, and cutting. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, pp. 9210–9216. http://dx.doi.org/10.1109/IROS47612.2022.9981118.

Gao, F., Fang, W., Sun, X., Wu, Z., Zhao, G., Li, G., Li, R., Fu, L., Zhang, Q., 2022a. A novel apple fruit detection and counting methodology based on deep learning and trunk tracking in modern orchard. Comput. Electron. Agric. 197, 107000. http://dx.doi.org/10.1016/j.compag.2022.107000.

Gao, J., Zhang, F., Zhang, J., Yuan, T., Yin, J., Guo, H., Yang, C., 2022b. Development and evaluation of a pneumatic finger-like end-effector for cherry tomato harvesting robot in greenhouse. Comput. Electron. Agric. 197, 106879. http://dx.doi.org/10.1016/j.compag.2022.106879.

Gil, G., Casagrande, D.E., Cortés, L.P., Verschae, R., 2023. Why the low adoption of robotics in the farms? Challenges for the establishment of commercial agricultural robots. Smart Agric. Technol. 3, 100069. http://dx.doi.org/10.1016/j.atech.2022.100069.

Huihui, Y., Daoliang, L., Yingyi, C., 2023. A state-of-the-art review of image motion deblurring techniques in precision agriculture. Heliyon http://dx.doi.org/10.1016/j.heliyon.2023.e17332.

Ju, C., Kim, J., Seol, J., Son, H.I., 2022. A review on multirobot systems in agriculture. Comput. Electron. Agric. 202, 107336. http://dx.doi.org/10.1016/j.compag.2022.107336.

Ju, C., Son, H.I., 2019. Modeling and control of heterogeneous agricultural field robots based on Ramadge–Wonham theory. IEEE Robot. Autom. Lett. 5 (1), 48–55. http://dx.doi.org/10.1109/LRA.2019.2941178.

Kim, S., Hong, S.-J., Ryu, J., Kim, E., Lee, C.-H., Kim, G., 2023. Application of amodal segmentation on cucumber segmentation and occlusion recovery. Comput. Electron. Agric. 210, 107847. http://dx.doi.org/10.1016/j.compag.2023.107847.

Kim, J., Kim, S., Ju, C., Son, H.I., 2019. Unmanned aerial vehicles in agriculture: A review of perspective of platform, control, and applications. Ieee Access 7, 105100–105115. http://dx.doi.org/10.1109/ACCESS.2019.2932119.

Kim, J., Pyo, H., Jang, I., Kang, J., Ju, B., Ko, K., 2022. Tomato harvesting robotic system based on Deep-ToMaToS: Deep learning network using transformation loss for 6D pose estimation of maturity classified tomatoes with side-stem. Comput. Electron. Agric. 201, 107300. http://dx.doi.org/10.1016/j.compag.2022.107300.

Kim, J., Son, H.I., 2020. A voronoi diagram-based workspace partition for weak cooperation of multi-robot system in orchard. IEEE Access 8, 20676–20686. http://dx.doi.org/10.1109/ACCESS.2020.2969449.

Kpadonou, R.A.B., Owiyo, T., Barbier, B., Denton, F., Rutabingwa, F., Kiema, A., 2017. Advancing climate-smart-agriculture in developing drylands: Joint analysis of the adoption of multiple on-farm soil and water conservation technologies in West African Sahel. Land Use Policy 61, 196–207. http://dx.doi.org/10.1016/j.landusepol.2016.10.050.

Lawal, O.M., 2021. YOLOMuskmelon: quest for fruit detection speed and accuracy using deep learning. IEEE Access 9, 15221–15227. http://dx.doi.org/10.1109/ACCESS.2021.3053167.

Li, J., Tang, Y., Zou, X., Lin, G., Wang, H., 2020. Detection of fruit-bearing branches and localization of litchi clusters for vision-based harvesting robots. IEEE Access 8, 117746–117758. http://dx.doi.org/10.1109/ACCESS.2020.3005386.

Luo, L., Yin, W., Ning, Z., Wang, J., Wei, H., Chen, W., Lu, Q., 2022. In-field pose estimation of grape clusters with combined point cloud segmentation and geometric analysis. Comput. Electron. Agric. 200, 107197. http://dx.doi.org/10.1016/j.compag.2022.107197.

Mao, D., Sun, H., Li, X., Yu, X., Wu, J., Zhang, Q., 2023. Real-time fruit detection using deep neural networks on CPU (RTFD): An edge AI application. Comput. Electron. Agric. 204, 107517. http://dx.doi.org/10.1016/j.compag.2022.107517.

Mehta, S., Burks, T., 2014. Vision-based control of robotic manipulator for citrus harvesting. Comput. Electron. Agric. 102, 146–158. http://dx.doi.org/10.1016/j.compag.2014.01.003.

Mehta, S., Burks, T., 2016. Adaptive visual servo control of robotic harvesting systems. IFAC-PapersOnLine 49 (16), 287–292. http://dx.doi.org/10.1016/j.ifacol.2016.10.053.

Mehta, S., MacKunis, W., Burks, T., 2014. Nonlinear robust visual servo control for robotic citrus harvesting. IFAC Proc. Vol. 47 (3), 8110–8115. http://dx.doi.org/10.3182/20140824-6-ZA-1003.02729.

Mehta, S.S., MacKunis, W., Burks, T.F., 2016. Robust visual servo control in the presence of fruit motion for robotic citrus harvesting. Comput. Electron. Agric. 123, 362–375. http://dx.doi.org/10.1016/j.compag.2016.03.007.

Mohamed, E.S., Belal, A., Abd-Elmabod, S.K., El-Shirbeny, M.A., Gad, A., Zahran, M.B., 2021. Smart farming for improving agricultural management. Egypt. J. Remote Sens. Space Sci. 24 (3), 971–981. http://dx.doi.org/10.1016/j.ejrs.2021.08.007.

Mstafa, R.J., Younis, Y.M., Hussein, H.I., Atto, M., 2020. A new video steganography scheme based on Shi-Tomasi corner detector. IEEE Access 8, 161825–161837. http://dx.doi.org/10.1109/ACCESS.2020.3021356.

Park, Y., Kim, H.-J., Son, H.I., 2023a. Novel attitude control of Korean cabbage harvester using backstepping control. Precis. Agric. 24 (2), 744–763. http://dx.doi.org/10.1007/s11119-022-09973-5.

Park, Y., Seol, J., Pak, J., Jo, Y., Kim, C., Son, H.I., 2023b. Human-centered approach for an efficient cucumber harvesting robot system: Harvest ordering, visual servoing, and end-effector. Comput. Electron. Agric. 212, 108116. http://dx.doi.org/10.1016/j.compag.2023.108116.

Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast point feature histograms (FPFH) for 3D registration. In: 2009 IEEE International Conference on Robotics and Automation. IEEE, pp. 3212–3217. http://dx.doi.org/10.1109/ROBOT.2009.5152473.

Seol, J., Kim, J., Son, H.I., 2022. Field evaluations of a deep learning-based intelligent spraying robot with flow control for pear orchards. Precis. Agric. 23 (2), 712–732. http://dx.doi.org/10.1007/s11119-021-09856-1.

SepúLveda, D., Fernández, R., Navas, E., Armada, M., Gonzalez-De-Santos, P., 2020. Robotic aubergine harvesting using dual-arm manipulation. IEEE Access 8, 121889–121904. http://dx.doi.org/10.1109/ACCESS.2020.3006919.

Tang, Y., Chen, M., Wang, C., Luo, L., Li, J., Lian, G., Zou, X., 2020. Recognition and localization methods for vision-based fruit picking robots: A review. Front. Plant Sci. 11, 510. http://dx.doi.org/10.3389/fpls.2020.00510.

Xiao, B., Chen, C., Yin, X., 2022. Recent advancements of robotics in construction. Autom. Constr. 144, 104591. http://dx.doi.org/10.1016/j.autcon.2022.104591.

Xiong, Y., Ge, Y., Grimstad, L., From, P.J., 2020. An autonomous strawberry-harvesting robot: Design, development, integration, and field evaluation. J. Field Robotics 37 (2), 202–224. http://dx.doi.org/10.1002/rob.21889.

Xu, F., Zhang, Y., Sun, J., Wang, H., 2022. Adaptive visual servoing shape control of a soft robot manipulator using bezier curve features. IEEE/ASME Trans. Mechatronics 28 (2), 945–955. http://dx.doi.org/10.1109/TMECH.2022.3210762.

Zhang, T., Li, X., Yang, Y., Guo, X., Feng, Q., Dong, X., Chen, S., 2019. Genetic analysis and QTL mapping of fruit length and diameter in a cucumber (*Cucumber sativus* L.) recombinant inbred line (RIL) population. Sci. Hortic. 250, 214–222. http://dx.doi.org/10.1016/j.scienta.2019.01.062.

Zhang, W., Liu, Y., Chen, K., Li, H., Duan, Y., Wu, W., Shi, Y., Guo, W., 2021. Lightweight fruit-detection algorithm for edge computing applications. Front. Plant Sci. 12, 740936. http://dx.doi.org/10.3389/fpls.2021.740936.

Zhao, Y., Gong, L., Huang, Y., Liu, C., 2016. A review of key techniques of vision-based control for harvesting robot. Comput. Electron. Agric. 127, 311–323. http://dx.doi.org/10.1016/j.compag.2016.06.022.