Novel method for the visual navigation path detection of jujube harvester autopilot based on image processing

Xiongchu Zhang[1], Bingqi Chen[1*], Jingbin Li[2], Xin Fang[1], Congli Zhang[1], Shubo Peng[1], Yongzheng Li[1]

(1. *College of Engineering, China Agricultural University, Beijing 100083, China*;
2. *College of Mechanical and Electrical Engineering, Shihezi University, Shihezi 832003, China*)

Abstract: To realize automatic harvesting of the jujube, the jujube harvester was designed and manufactured. For achieving the jujube harvester autopilot, a novel algorithm for visual navigation path detection was proposed. The centerline of tree row lines was taken as the navigation path. The method included four main parts: image preprocessing, image segmentation, tree row lines access, and navigation path access. The methods of threshold segmentation, noise removal, and border smoothing were utilized on the image in Lab color space for the image segmentation. The least square method was employed to fit the tree row lines, and the centerline was obtained as the navigation path. Experimental results indicated that the average false detection rate was 3.98%, and the average detection speed was 41 fps. The algorithm meets the requirements of the jujube harvester autopilot in terms of accuracy and speed. It also can lay the foundation for accomplishing the jujube harvester vision-based autopilot.

Keywords: visual navigation path, jujube orchards, image processing, Lab color space, seed region growing
DOI: 10.25165/j.ijabe.20231605.7638

Citation: Zhang X C, Chen B Q, Li J B, Fang X, Zhang C L, Peng S B, et al. Novel method for the visual navigation path detection of jujube harvester autopilot based on image processing. Int J Agric & Biol Eng, 2023; 16(5): 189–197.

1 Introduction

Autonomous mobile robots have widely been applied in agricultural tasks[1]. Autopilot is the key technology for developing an autonomous mobile robot[2]. To solve the problem of the high labor intensity, low efficiency, and high labor cost caused by manual harvesting, the jujube harvester has been designed and developed by the Shihezi University[3]. Meanwhile, the navigation path detection is the prerequisite of the agricultural robot autopilot[4,5]. However, it is also a challenging task to figure out the navigation path due to the complex agricultural environment[6,7]. Hence, a novel method was developed in this study to find the navigation path for the jujube harvester autopilot.

In the past decade, Global Navigation Satellite System (GNSS) has gained more attention in the research of the agriculture mobile robot autopilot according to the increased availability of differential corrections[8]. The state-of-the-art Real Time Kinematic (RTK) GNSS can achieve centimeter-level accuracy[9]. However, the performance in agricultural fields with foliage and surrounding obstacles decreased due to propagation effects[10]. Meanwhile, the satellite of the GNSS system cannot be in a stable state for a long time due to the impacts of multi-path reflections and signal a blockage in agricultural fields[11]. Furthermore, GNSS-based solutions are expensive and require a long preparation phase where the field has to be surveyed with a GNSS rover to collect waypoints for the navigation path[12].

Light Detection and Ranging (LiDAR) scanners, a local perception sensor, can be provided as an alternative navigation approach according to the ability to track geometric features in the perceived scene[12]. Nevertheless, since LiDAR cannot directly distinguish un-traversable obstacles (actual crop plant stalk) and traversable obstacles (hanging leaves, weeds, uneven terrain), LiDAR-based methods cannot estimate distance and angle from the row, leading to low robustness of autonomy[13]. Meanwhile, the high cost of LiDAR scanners prevents the public from utilizing this technology on autopilot[14].

With the development of the digital image processing technology, Machine Vision (MV) has successfully applied in agricultural robot navigation according to irreplaceable visual information and low-cost hardware costs[15]. Detection algorithms of the navigation path have generated considerable recent research interest these years[16],which research has covered almost all agricultural scenes, such as orchards[17,18], farmland[19,20] and green house[21,22].

Nevertheless, there is little research about visual navigation path detection in jujube orchards. The algorithm to work out the navigation path has been proposed for circumstance of short stalked and close-planting jujube orchard based on image processing technology[23]. However, in this study, since the row spacing of jujube orchards is more extensive, and the camera installation is higher, the segmentation method in the above research cannot work. Hence, it is an urgent task to design a method to find the navigation path for the jujube harvester.

As concerned above, this paper presented a novel method to find the visual navigation path based on the image processing technology for the jujube harvester autopilot. For our method, the centerline of tree row lines was taken as the navigation path. The algorithm consists of four parts: image preprocessing, image segmentation, tree row lines access, and navigation path access. The image captured was converted to the Lab color space for image pre-processing. For image segmentation, first, the image is transferred

Received date: 2023-01-22 **Accepted date:** 2023-06-12
Biographies: Xiongchu Zhang, PhD, research interest: image processing, Email: 15201238665@163.com; **Jingbin Li,** PhD, Professor, research interest: visual navigation, Email: ljb8095@163.com; **Xin Fang,** MSc, research interest: image processing, Email: gxyyjs@163.com; **Congli Zhang,** MSc, research interest: image processing, Email: conglizh@163.com; **Shubo Peng,** MSc, research interest: image processing, Email: peng_xy034@163.com; **Yongzheng Li,** MSc, research interest: image processing, Email: 473020906@qq.com.
***Corresponding author: Bingqi Chen,** PhD, Professor, research interest: image processing and autopilot. No.17 Qinghua East Road, College of Engineering, China Agricultural University, Beijing 100083, China. Tel: +86-10-62966687, Email: fbcbq@163.com.

into the binary image by the binarization method without grayscale processing. Then, the methods of area-based noise removal and border smoothing were applied for further segmentation and improving the accuracy of the navigation path. Seed Region Growing (SRG)[24] method is utilized on the in-row area for extraction of navigation points. Last, the least square method is employed for the border lines fitting, and the middle line is achieved as the navigation path. This algorithm can provide the theoretical foundation to realize the jujube harvester autopilot. We believe, it can also enrich the detection algorithm of the vision-based navigation path for agricultural scenes. The main contribution is as follows:

1) A novel method based on image processing technology was proposed to work out the navigation path for the jujube harvester autopilot.

2) A new binarization method without grayscale processing for the color image was designed.

3) An area-based noise removal method was proposed to remove the noise after binarization.

4) A border smoothing method was developed to remove the sharp edges.

2 Materials and methods

We aim to design an algorithm to work out the visual navigation path for the jujube harvester autopilot. Figure 1 is the schematic diagram of the navigation path. The solid blue line, the centerline of the tree row lines (solid red lines), is the navigation path[25]. For this study, the border of the in-row area marked by the black cross is the tree row line.

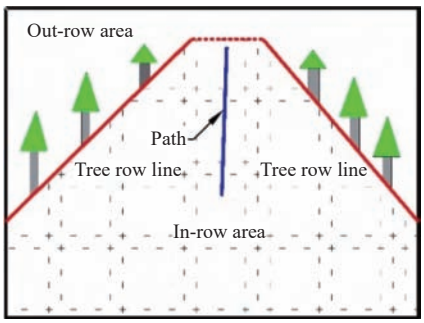


Figure 1 Schematic diagram of the navigation path in jujube orchards

2.1 Platform and equipment

The platform and equipment, such as the jujube harvester and camera, are introduced in the following chapters.

2.1.1 Jujube harvester

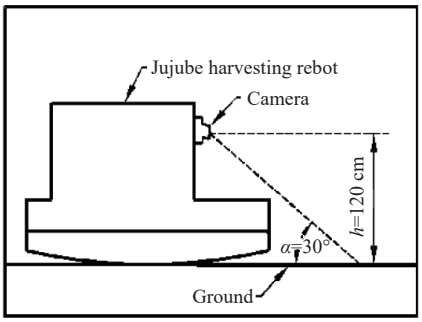
To improve efficiency and decrease labor cost, the jujube harvester based on the full-hydraulic self-propelled has been designed and manufactured by the Shihezi University[3] shown in Figure 2a. The harvester mainly works in the dwarf and close planting jujube orchards.

2.1.2 Camera

A camera with the parameters listed in Table 1 was applied in this study. Figure 2a is the physical picture of the camera installation, and the installation diagram was manifested in Figure 2b. The angle is 30° between the optical axis of the camera and the ground, and the vertical height is 120 cm from the camera to the ground.

Furthermore, the algorithm is developed with the help of the computer with an AMD Ryzen 5 5600X6 Core Processor 3.70 GHz CPU and a 32G RAM, and the Microsoft Visual Studio 2010 software.

a. Physical picture of the camerain stallation



b. Schematic diagram of the camera installation

Figure 2 Physical picture and schematic diagram of the camera installation

Table 1 Main parameters of the camera

Product model	Output image format	Resolution@ frame rate	Interface type	Operating voltage	Operating temperature
RER-USBFHD01M-LS36	MJPEG / YUV2 (YUYV)	640X480 VGA MJPEG@60 fps	USB2.0 High Speed	DC5V	0°C-60°C

2.2 Video acquisition

The video sampling operation was conducted in Alar, Xinjiang, between November 28 and December 2, 2021.The operator drove the jujube harvester at an operating speed of 0.5 m/h to collect videos. The videos are saved in AVI format with a resolution of 640×480 pixels (horizontal×vertical) of every frame.

2.3 Image preprocessing

Image preprocessing plays a crucial role in the digital image processing technology which is a data preparation step for contrast enhancement, noise reduction, or filtering[26,27]. Image Enhancement, one common method of Image preprocessing, has been utilized in image processing which can improve the visual quality of images[28], such as removing noise, de-blurring edges of objects, and highlighting some specified features[29]. Color space conversion is a popular method of Image enhancement that has an important effect on the result of image processing[30-32]. Meanwhile, it has been successfully applied in navigation path detection methods for agricultural mobile robots[33], and the most commonly used color spaces include HSI[22,34], Lab[35], and HSV[33,36].

The Lab color space has been widely applied in agricultural image segmentation[37-39]. More importantly, experimental results indicate that the Lab color space has better performance than the color space of the HSV and HSI on the image segmentation in this project. Hence, Lab color space was used in this study.

RGB color space cannot be transformed directly into Lab, and it needs *XYZ* as an intermediate variable. Equation (1) describes the method to converse RGB to *XYZ*.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = 100M \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

$$M = \begin{bmatrix} 0.436\,052\,025 & 0.385\,081\,593 & 0.143\,087\,414 \\ 0.222\,491\,598 & 0.716\,886\,060 & 0.060\,621\,486 \\ 0.013\,929\,122 & 0.097\,097\,002 & 0.714\,185\,470 \end{bmatrix} \tag{2}$$

The method to achieve the components R, G, and B of RGB is displayed in Equation (3).

$$\begin{cases} R = gamma\left(\dfrac{r}{255.0}\right) \\ G = gamma\left(\dfrac{g}{255.0}\right) \\ B = gamma\left(\dfrac{b}{255.0}\right) \end{cases} \tag{3}$$

where, r, g, and b are the color component value of the current pixel.

$$gamma(x) = \begin{cases} \left(\dfrac{x+0.055}{1.055}\right)^{2.4}, & (x > 0.040\,45) \\ \dfrac{x}{12.92}, & (\text{others}) \end{cases} \tag{4}$$

Equation (5) is utilized to transfer *XYZ* to Lab.

$$\begin{cases} L = 166 \times f\left(\dfrac{Y}{Y_n}\right) - 16 \\ a = 500 \times \left[f\left(\dfrac{X}{X_n}\right) - f\left(\dfrac{Y}{Y_n}\right)\right] \\ b = 200 \times \left[f\left(\dfrac{Y}{Y_n}\right) - f\left(\dfrac{Z}{Z_n}\right)\right] \end{cases} \tag{5}$$

where,

$$f(t) = \begin{cases} t^{\frac{1}{3}}, & \left(t > \left(\dfrac{6}{29}\right)^3\right) \\ \dfrac{1}{3} \times \left(\dfrac{6}{29}\right)^2 \times t + \dfrac{4}{29}, & (\text{others}) \end{cases} \tag{6}$$

and

$$\begin{cases} X_n = 95.047 \\ Y_n = 100 \\ Z_n = 108.883 \end{cases} \tag{7}$$

2.4 Image segmentation

Image segmentation plays a significant role in image recognition[40]. In this study, the thresholding method was utilized to carry out segmentation, and methods of noise removal and border smoothing are applied to improve the segmentation effect.

2.4.1 Threshold segmentation

The thresholding method was utilized in this study, which is the most common and simplest approach to segment an image[41]. In common, the conventional thresholding methods consist of two steps: image grayscale and image binarization. The image grayscale algorithm is to unify the component values of each pixel by the formulas which mainly include Chromatic aberration method[42-46], Weighted average grayscale method[47], and Average gray scale value method[48]. Especially, the threshold of each scan line method was proposed for short stalked and close-planting jujube orchards[23]. However, the above methods are not fitting to this project according to the low chromatic aberration and contrast of the image in jujube orchards. Due to this, conventional binarization methods cannot be utilized in this study, either, such as Otsu[49] which has a wide application in navigation path detection for the agricultural mobile robot autopilot[50,51].

To address this issue, a new binarization method without grayscale processing was proposed, which has better performance on the segmentation for this project. Furthermore, the method proposed can improve the speed and reduce the complexity of the algorithm according to that it requires only one step. The method is described in Equation (8).

$$g(x,y) = \begin{cases} 255, & (p_a > p_L \text{ and } p_a > p_b) \\ 0, & (\text{others}) \end{cases} \tag{8}$$

where, $g(x,y)$ is the gray value of pixel(x,y). p_a, p_b, and p_L are component values of the Lab color space of the pixel(x,y), respectively.

2.4.2 Noise removal

After the threshold segmentation process, there is lots of noise within the in-row area. To remove the noise, the area-based denoising algorithm was proposed. After image segmentation, the grayscale value of the pixels in the in-row area is 255, and 0 is the grayscale value for the pixels in the out-row area.

The first step of the proposed algorithm is to find all the connected areas by using the SRG algorithm. Then, for the in-row area, the grayscale value of pixels that are not in the maximum connected area is set to 0, and 255 for the inter-row area. The specific operation is described in Algorithm 1.

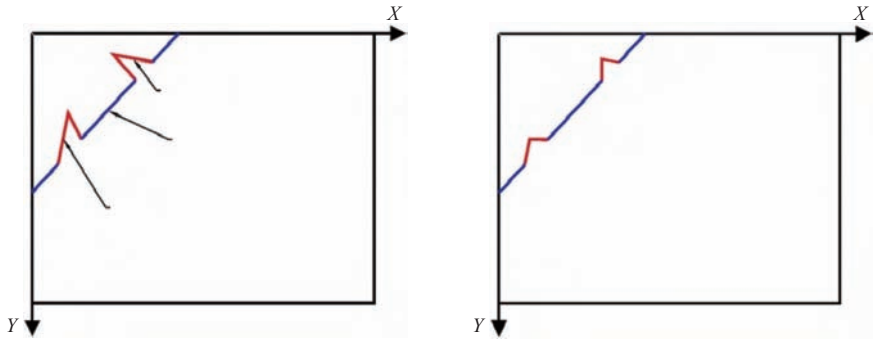
Algorithm 1: Noise removal method

Input : A binary bitmap I_m of size $w \times h$
output: A bitmap I'_m of size $w \times h$ after noise removal, $I'_m \leftarrow 0$
Initialization: L, matrix of size $w \times h$ for pixel labels in I_m, $L \leftarrow 0$; p_c, to save coordinates of the pixel in the current connected area, $p_c \leftarrow 0$; p_n, to save coordinates of the pixel in the connected area which is noise, $p_n \leftarrow 0$; s_c, size of p_c, $s_c \leftarrow 0$; s_n, size of p_n, $s_n \leftarrow 0$; d_c, to save 4 directions of the current seed, $d_c \leftarrow [(-1, 0), (1, 0), (0, -1), (0, 1)]$; PixelValue() is the function to get the value of the pixel

```
for l ← 0 to 2 do
  if l == 0 then p = 0;
  else p = 255;
  s_m = 0;
  for j ← 1 to h − 1 do
    for i ← 1 to w − 1 do
      if L_(i,j) == 1 then continue;
      this ← PixelValue(Im[i, j]);
      if this == p then
        p_c = []
        s_c = 0
        add (i, j) into p_c
        seed = []
        add (i, j) into seed
        while seed ≠ [] do
          scur ← Remove a seed from seed;
          for h ← 0 to 4 do
            cor = scur + d_c[h];
            p_c ← PixelValue(Im[cor]);
            if p_c == p and L[cor] == 0 then
              s_c = s_c + 1
              L[cor] = 1
              add cor into seed
              add cor into c
          end
        end
        if s_c < s_m then
          add p_c into p_n
          s_n = s_n + s_c
        else s_m = s_c;
      end
    end
    for k ← 0 to s_n do
      Im'[p_n[k]] = 255 − p
    end
  end
```

2.4.3 Border smoothing

After the denoising process, a better segmentation result was obtained. However, the noise of sharp edges still exists at the border, which can decrease the accuracy of the navigation path. There are two types of sharp edges, the horizontal and the vertical sharp edges, as shown in Figure 3a.



a. Before border smoothing b. After border smoothing

Figure 3 The border of the in-row area after and before edge noise removal

For vertical sharp edges, scanning from bottom to top column-by-column, the pixel value of the point after the first point that does not belong to the in-row area is set to 0. For horizontal sharp edges, the first step is to divide the image into two parts. Then, the scanning direction is that, for the left part, line-by-line scanning from the right to the left, and the opposite for the right part. The next process is the same as the processing of vertical sharp edges. The border after border smoothing is depicted in Figure 3b. The specific operation of border smoothing is described in Algorithm 2.

Algorithm 2: Border smoothing method

Input : A binary bitmap I_m of size $w \times h$
output: A bitmap I_m after border smoothing

Initialization: i_p, the pixel column index for the image is divided into left and right parts, $ip \leftarrow 0$; PixelValue() is the function to get the value of the pixel

for $t \leftarrow 0$ **to** 2 **do**
  if $t == 0$ **then**
    for $i \leftarrow 0$ **to** w **do**
      for $j \leftarrow h - 1$ **to** 0 **do**
        $this \leftarrow$ PixelValue($Im[i, j]$);
        if $this == 0$ **then** $j_k = j$
        for $j_s \leftarrow j_k$ **to** *end* **do**
          $Im[i, j_s] = 0$
        end
      end
    end
  end
  else
    $p_m = 0$
    for $i \leftarrow 0$ **to** w **do**
      $P_n = 0$
      for $j \leftarrow 0$ **to** $(h - 1)$ **do**
        $this \leftarrow$ PixelValue($Im[i, j]$);
        if $this == 255$ **then** $p_n = p_n + 1$;
      end
      if $p_n > p_m$ **then**
        $p_m = p_n$
        $i_p == i$
      end
    end
    for $l \leftarrow 0$ **to** 2 **do**
      if $l == 0$ **then**
        $start = i_p$
        $end = 0$
      end
      else
        $start = i_p$
        $end = w$
      end
      for $j \leftarrow start$ **to** end **do**
        for $i \leftarrow i_p$ **to** 0 **do**
          $this \leftarrow$ PixelValue($Im[i, j]$);
          if $this == 0$ **then** $i_k = i$;
        end
        for k $\leftarrow 0$ **to** s_n **do**
          $Im[i_s, j] = 0$
        end
      end
    end
  end
end

2.5 Tree row lines access

2.5.1 Reference point generation of tree row lines

In this study, the reference points are the border points of the in-row are according to that the border of the in-row is the tree row line. For this method, the first step is to find the pixel point within the in-row area, and this pixel point is used as the seed for the SRG algorithm. Then the pixel point that can no longer grow is the reference point. Algorithm 3 describes details of this method.

Algorithm 3: Method for navigation points extraction based on the SRG algorithm

Input : A binary bitmap I_m of size $w \times h$
output: Points p for fitting navigation path, p←[]

Initialization: d_c, to save 4 directions of the current seed, $d_c \leftarrow [(-1, 0),(1, 0),(0, -1),(0, 1)]$; PixelValue() is the function to get the value of the pixel

for $j \leftarrow 1$ **to** h − 1 **do**
  for $i \leftarrow 1$ **to** $w - 1$ **do**
    $this \leftarrow$ PixelValue($Im[i, j]$);
    if $this == p$ **then** continue;
    $seed = []$;
    add (i, j) into *seed;*
  end
  break
end
while $seed \neq []$ **do**
  $scur \leftarrow$ Remove a seed from *seed*;
  $time = 0$
  for $h \leftarrow 0$ **to** 4 **do**
    $cor = scur + d_c[h]$;
    $p_c \leftarrow$ PixelValue($Im[cor]$);
    if $p_c == 255$ **then**
      $time = time + 1$
      add *cor* into *seed*
    end
  end
  if $time \neq 4$ **then** add *scur* into *p*;
end

2.5.2 Line fitting of the tree rows

Fitting methods are mainly composed of the Hough Transform[52-55] and the Least Square[56]. Since Hough Transform is computationally expensive[57], the Least Square method was applied in this study. The tree row line is detected by using least square algorithm shown in Equations (9) and (10) based on reference points. x_i and y_i is the coordinates of the i_{th} reference point. k and b denoted the slope and intercept of the navigation path.

$$\begin{cases} x_{sum} = \dfrac{\sum_{i=1}^{n} x_i}{n} \\ y_{sum} = \dfrac{\sum_{i=1}^{n} y_i}{n} \\ xx_{sum} = \dfrac{\sum_{i=1}^{n} x_i x_i}{n} \\ yy_{sum} = \dfrac{\sum_{i=1}^{n} y_i y_i}{n} \end{cases} \tag{9}$$

$$\begin{cases} k = \dfrac{nxy_{sum} - x_{sum}y_{sum}}{nxx_{sum} - x_{sum}x_{sum}} \\ b = \dfrac{-x_{sum}xy_{sum} + xx_{sum}y_{sum}}{nxx_{sum} - x_{sum}x_{sum}} \end{cases} \tag{10}$$

2.6 Navigation path access

Figure 4 is the schematic diagram of the navigation path access. The two solid red lines are the tree row lines after fitting, and the solid blue line is the navigation path. According to the principle that two points determine a straight line, we only need to find two points on the centerline. The method to work out the two points $p_{n1}(p_{n1\_x}, p_{n1\_y})$ and $p_{n2}(p_{n2\_x}, p_{n2\_y})$ is described in Equation (11).

$$\begin{cases} p_{n1\_x} = \dfrac{p_{l1\_x} + p_{r1\_x}}{2} \\ p_{n1\_y} = \dfrac{p_{l1\_y} + p_{r1\_y}}{2} \\ p_{n2\_x} = \dfrac{p_{l2\_x} + p_{r2\_x}}{2} \\ p_{n2\_y} = \dfrac{p_{l2\_y} + p_{r2\_y}}{2} \end{cases} \tag{11}$$

The top left corner of the image is the origin of the coordinate system, and the orientations of arrows are the positive directions of the *X-axis* and the *Y-axis*. $p_{l1}(p_{l1\_x}, p_{l1\_y})$ and $p_{l2}(p_{l2\_x}, p_{l2\_y})$, $p_{r1}(p_{r1\_x}, p_{r1\_y})$ and $p_{r2}(p_{r2\_x}, p_{r2\_y})$ are the two end points of the left and right border lines respectively.

The flowchart of the algorithm is shown in Figure 5. Figure 6 describes the result of each processing step in the proposed algorithm.

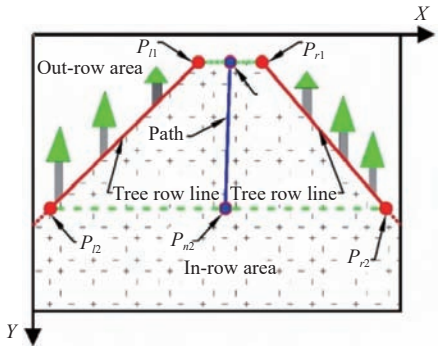


Figure 4 Schematic diagram of the navigation path access

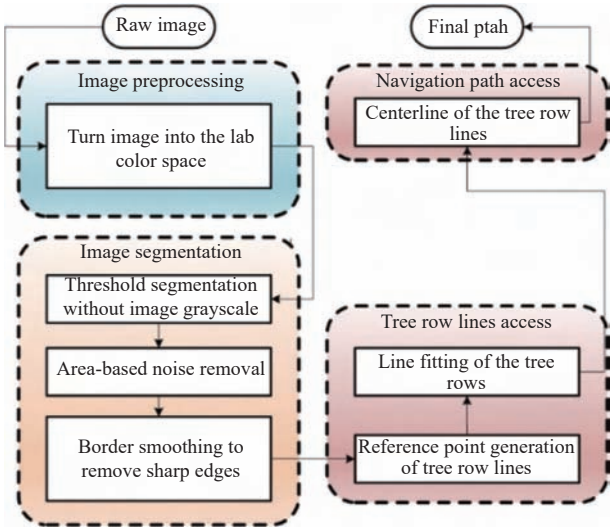
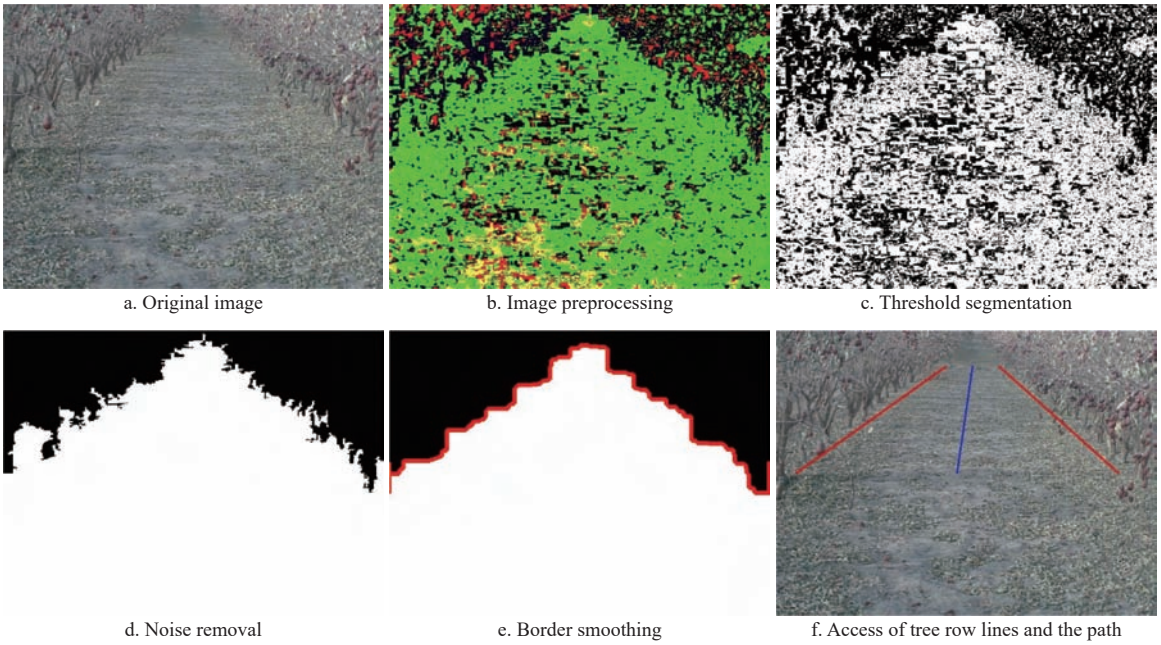


Figure 5 Flow chart of the proposed algorithm



a. Original image b. Image preprocessing c. Threshold segmentation

d. Noise removal e. Border smoothing f. Access of tree row lines and the path

Figure 6 Processing result of each step in the proposed algorithm

3 Results and analyses

3.1 Results

To verify the algorithm performance in terms of the detection speed and false detection rate, the test experiment has been carried out through 8 sets of videos randomly selected from videos collected. In this study, it is a false detection frame when the detection result has a large error from the visual inspection. Furthermore, the false detection rate r_w is calculated by Equation (12). w_f and t_f are the number of false detection frames and the total frames, respectively. Experimental results are listed in Table 2.

$$r_w = \frac{w_f}{t_f} \tag{12}$$

It can be seen from Table 2 that the average detection speed and false detection rate are 41 fps and 3.98% respectively, which

Table 2 Experimental results of the false detection rate and detection speed

No	t_f	w_f	r_w /%	Average detection speed/fps
1	2910	131	4.50	40
2	3021	88	2.91	42
3	3241	165	5.09	43
4	3623	134	3.70	38
5	2891	127	4.40	42
6	2641	103	3.90	40
7	2951	121	4.10	42
8	3970	135	3.40	37
average			3.98	41

can meet the requirement of the jujube harvester autopilot.

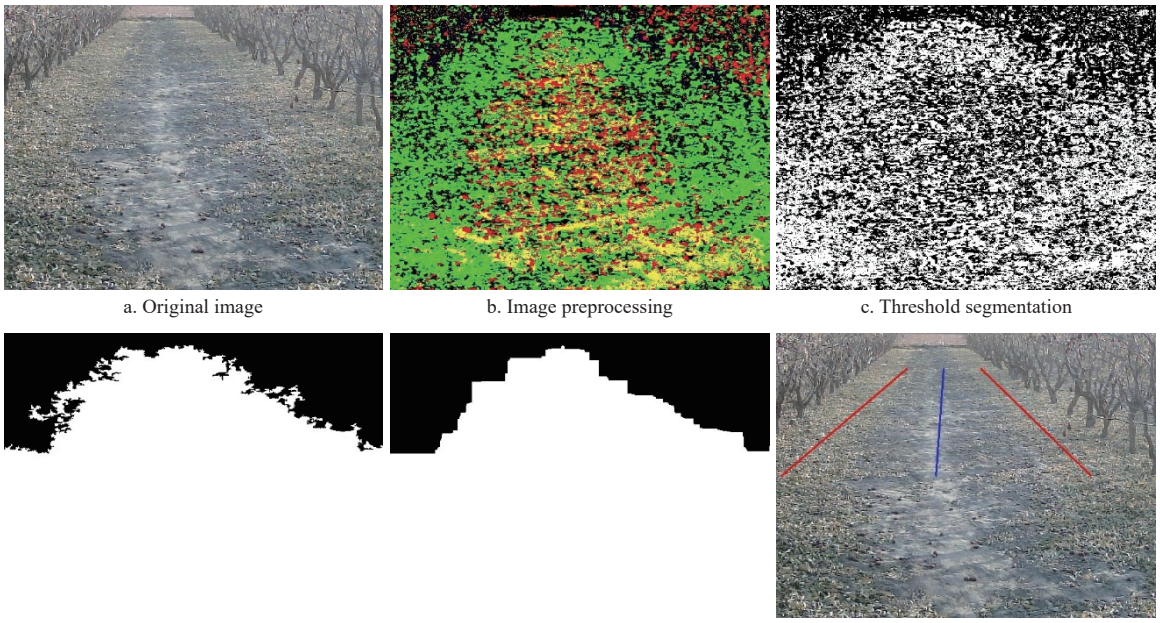
3.2 Analysis of the false detection

Severe image exposure is the main cause of the false detection in this study. The processing results of all steps of the severe exposure image are displayed in Figure 7. Due to the exposure impact, there is lots of noise in the in-row area and out-row area displayed in Figure 7b, and the border is not obvious. After noise removal and border smoothing, the accuracy of border points is not up to the requirement which leads to false detection. Hence, image exposure should be avoided in actual operation.

3.3 Image preprocessing

To select a suitable color space, the comparison experiments of RGB, HSI, HSV, and Lab color space were carried out. The images on the first row in Figure 8 are the images in different color spaces, and the ones on the second row are the results after the threshold segmentation. For the image in RGB, HSI, and HSV color spaces, the methods of the Average grayscale value method[48], and the Otsu method[49] were applied for threshold segmentation, which can obtain the best segmentation result for the image in this study. The result of the image in the Lab color space was obtained by the threshold segmentation method proposed in this paper.

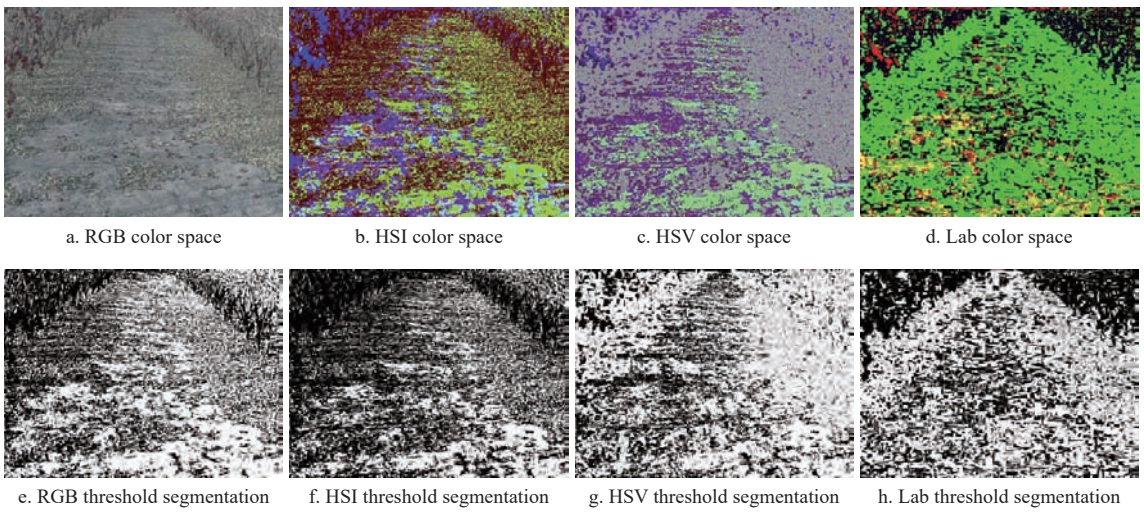
It can be seen in Figure 8d that the image in Lab color space can be better divided into two parts, the in-row area and the out-row area, and the borders of the in-row area are obviously. However, these could not be observed well from RGB, HSI and HSV color space displayed in Figure 8a, Figure 8b, Figure 8c. Meanwhile, the Lab color space has better performance from the threshold segmentation results by comparing Figure 8b, Figure 8d, Figure 8f, and Figure 8h. Hence, the Lab color space is more suitable than other color spaces.



a. Original image b. Image preprocessing c. Threshold segmentation

d. Noise removal e. Border smoothing f. Access of tree row lines and the path

Figure 7 Processing results of each step in the proposed algorithm under false detection



a. RGB color space b. HSI color space c. HSV color space d. Lab color space

e. RGB threshold segmentation f. HSI threshold segmentation g. HSV threshold segmentation h. Lab threshold segmentation

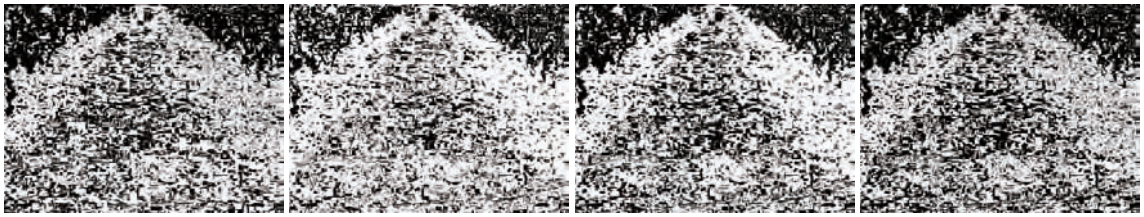
Figure 8 Threshold segmentation result of the image in different color spaces

3.4 Image segmentation

3.4.1 Threshold segmentation

The comparative experiment was carried out among different threshold methods for finding a suitable thresholding method for image segmentation. Figure 9 describes threshold segmentation result of different methods. The methods of the Chromatic Aberration method[42], Weighted Average Grayscale method[47], Average Grayscale Value method[48] are utilized to grayscale the image, and the Otsu[49] is utilized to work out the threshold value.

Comparing with Figure 9a, it can be seen that there is more noise in the out-row area in Figure 9b, Figure 9c, and Figure 9d, and the size of the in-row area also is larger which results in the wrong tree row line. Thus, the threshold method proposed in this paper can be applied.



a. Method proposed b. Average grayscale value c. Weighted average grayscale d. Chromatic aberration

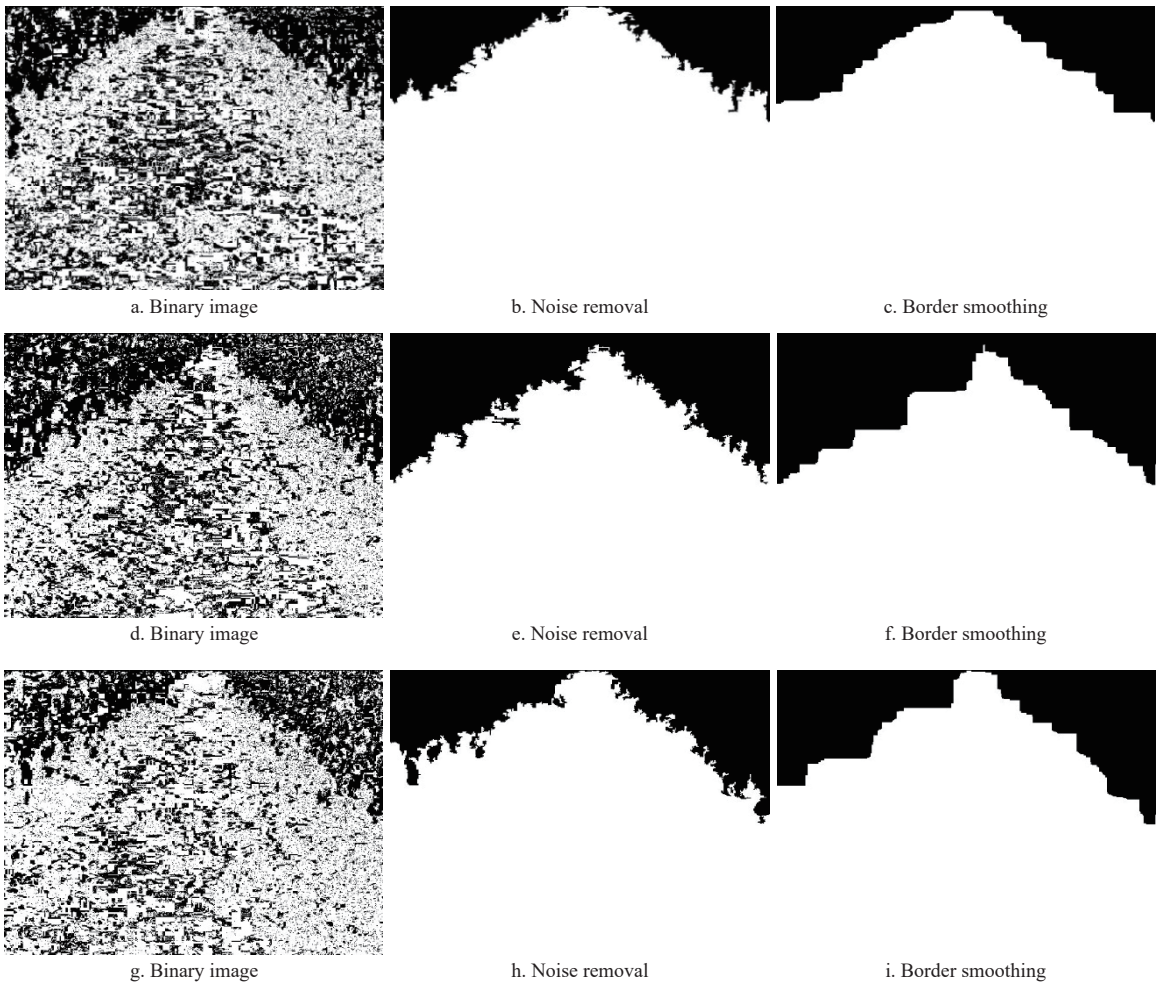
Figure 9 Threshold segmentation result of different methods

3.4.2 Noise removal

The noise removal method is employed for further segmentation due to the fact that, after threshold segmentation, there is lots of noise in the in-row area and out-row area displayed in Figure 10a, Figure 10d, and Figure 10g. The images after noise removal are shown in Figure 10b, Figure 10e, and Figure 10h. As can be seen that the image after noise removal is divided clearly into two parts, and there is no noise in the interior of the in-row area and out-row area. Hence, the area-based threshold method can work well.

3.4.3 Border smoothing

Figure 11 describes the tree row line points, tree row lines, and navigation path under different conditions. The accuracy of the left tree row line in Figure 11b is more accurate than the one in Figure 11d. Thus, the border smoothing method is necessary.



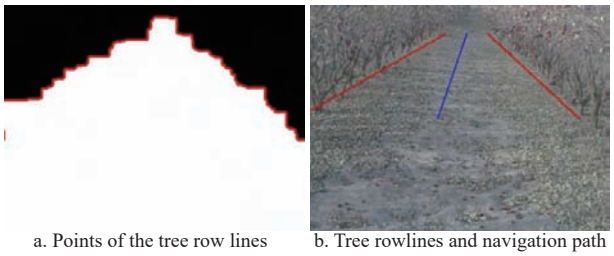
a. Binary image b. Noise removal c. Border smoothing

d. Binary image e. Noise removal f. Border smoothing

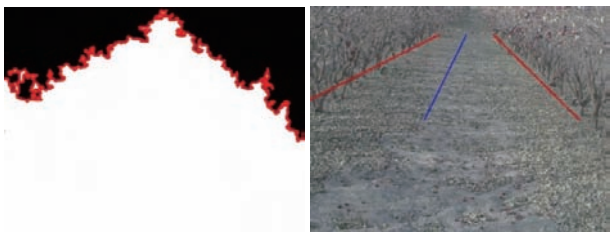
g. Binary image h. Noise removal i. Border smoothing

Figure 10 The processing result of noise removal and border smoothing on different videos

Meanwhile, we can see that the borders of the in-row area after border smoothing are clearer, and more coherent and smooth by comparing images Figure 10b with Figure 10c, Figure10e with Figure 10f, and Figure 10h with Figure 10i.



a. Points of the tree row lines after border smoothing b. Tree rowlines and navigation path



c. Points of the tree row lines without border smoothing d. Tree row lines and navigation path

Figure 11 Reference points of tree row lines, tree row lines, and navigation path under different conditions

4 Conclusions

This study set out to develop an algorithm of vision-based navigation path detection for the jujube harvester autopilot. The study contributes to realizing the jujube harvester autopilot and promotes the automation of the jujube production.

The proposed algorithm consists of four steps: image preprocessing, image segmentation, tree row lines access, and navigation path access. For image preprocessing, the Lab color space has better performance than RGB, HSV, and HSI color space in this project. The methods of area-based noise removal and border smoothing can further improve the image segmentation effect, and improve the accuracy of the navigation path. Experimental results confirmed that the algorithm performance of the average detection speed and false detection, 41 fps and 3.98% respectively, which can meet the requirement of the jujube harvester autopilot.

The limitations of the present studies naturally include the test experiment in jujube orchards further verified the performance of the proposed algorithm. Meanwhile, future research should consider the potential effects of image quality more carefully, for example, image exposure.

Acknowledgements

This work was financially supported by the National Key R&D Program of China (No. 2016YFD0701504). The authors would like to thank the editors and anonymous reviewers for their constructive comments and suggestions that significantly improved this paper.

[References]

[1] Sharifi M, Chen X. A novel vision based row guidance approach for navigation of agricultural mobile robots in orchards. In: 2015 6th International Conference on Automation, Robotics and Applications (ICARA), 2015; pp.251–255.

[2] Raikwar S, Fehrmann J, Herlitzius T. Navigation and control development for a four-wheel-steered mobile orchard robot using model-based design. Computers and Electronics in Agriculture, 2022; 202: 107410.

[3] Zhou L, Li J B, Ding L P, Ding H Z, Shi G K, Li D L. Design and evaluation of a mechanical floor-standing jujube picker. Agriculture, 2022; 12(8): 1–20.

[4] Jiang H B, Xiao Y L, Zhang Y W, Wang X J, Tai H J. Curve path detection of unstructured roads for the outdoor robot navigation. Mathematical and computer modelling, 2013; 58(3-4): 536–544.

[5] Torres-Sospedra J, Nebot P. A new approach to visual-based sensory system for navigation into orange groves, Sensors. 2011; 11(4): 4086-4103.

[6] English A, Ross P, Ball D, Corke P. Vision based guidance for robot navigation in agriculture. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, 2014; pp.1693–1698.

[7] Shalal N, Low T, McCarthy C, Hancock N. A review of autonomous navigation systems in agricultural environments. SEAg 2013: Innovative Agricultural Technologies for a Sustainable Future, Engineers Australia, 2013.

[8] Kim K, Deb A, Cappelleri D J. P-AgBot: In-row and under-canopy agricultural robot for monitoring and physical sampling. IEEE Robotics and Automation Letters, 2022; 7(3): 7942–7949.

[9] Bakker T, van Asselt K, Bontsema J, Müller J, van Straten G. Autonomous navigation using a robot platform in a sugar beet field. Biosystems Engineering, 2011; 109(4): 357–368.

[10] Pini M, Marucco G, Falco G, Nicola M, De Wilde W. Experimental testbed and methodology for the assessment of RTK GNSS receivers used in precision agriculture. IEEE Access, 2020; 8: 14690–14703.

[11] Rovira-Más F, Chatterjee I, Sáiz-Rubio V. The role of GNSS in the navigation strategies of cost-effective agricultural robots. Computers and electronics in Agriculture, 2015; 112: 172–183.

[12] Nehme H, Aubry C, Solatges T, Savatier X, Boutteau R, Rossi R. Lidar-based structure tracking for agricultural robots: Application to autonomous navigation in vineyards. Journal of Intelligent & Robotic Systems, 2021; 103: 1–16.

[13] Higuti V A H, Velasquez A E B, Magalhaes D V, Becker M, Chowdhary G. Under canopy light detection and ranging‐based autonomous navigation. Journal of Field Robotics, 2019; 36(3): 547–567.

[14] Raj T, Hanim H F, Huddin A B, Ibrahim M F, Hussain A. A survey on LiDAR scanning mechanisms. Electronics, 2020; 9(5): 741.

[15] Wang T, Chen B, Zhang Z, Li H, Zhang M. Applications of machine vision in agricultural robot navigation: A review. Computers and Electronics in Agriculture, 2022; 198: 107085.

[16] Bonin-Font F, Ortiz A, Oliver G. Visual navigation for mobile robots: A survey. Journal of Intelligent and Robotic Systems, 2008; 53: 263–296.

[17] Zhou M K, Xia J F, Yang F, Zheng K, Hu M J, Li D, et al. Design and experiment of visual navigated UGV for orchard based on Hough matrix and RANSAC. Int J Agric & Biol Eng, 2021; 14(6): 176–184.

[18] Lyu H K, Park C-H, Han D H, Kwak S, Choi B. Orchard free space and center line estimation using Naive Bayesian classifier for unmanned ground self-driving vehicle. Symmetry, 2018; 10(9): 355.

[19] Choi K H, Han S K, Han S H, Park K-H, Kim K-S, Kim S. Morphology-based guidance line extraction for an autonomous weeding robot in paddy fields. Computers and Electronics in Agriculture, 2015; 113: 266–274.

[20] Jiang G, Wang Z, Liu H. Automatic detection of crop rows based on multi-ROIs. Expert systems with applications, 2015; 42(5): 2429–2441.

[21] Chen J, Qiang H, Wu J, Xu G, Liu X. Extracting the navigation path of a tomato-cucumber greenhouse robot based on a median point Hough transform. Computers and Electronics in Agriculture, 2020; 174: 105472.

[22] Gao G, Li M. Navigating path recognition for greenhouse mobile robot based on K-means algorithm. Transactions of the CSAE, 2014; 30(7): 25–33. (in Chinese)

[23] Peng S, Kan Z, Li J. Extraction of visual navigation directrix for harvesting operation in short-stalked and close-planting jujube orchard. Transactions of the CSAE, 2017; 33(9): 45–52. (in Chinese)

[24] Adams R, Bischof L. Seeded region growing. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1994; 16(6): 641–647.

[25] Zhou J, Geng S, Qiu Q, et al. A deep-learning extraction method for orchard visual navigation lines. Agriculture, 2022; 12(10): 1650.

[26] Chen T. The past, present, and future of image and multidimensional signal processing. IEEE Signal Processing Magazine, 1998; 15(2): 21–58.

[27] Bhattacharyya S. A brief survey of color image preprocessing and segmentation techniques. Journal of Pattern Recognition Research, 2011; 1(1): 120–129.

[28] Vasuki P, Kanimozhi J, Devi M B. A survey on image preprocessing techniques for diverse fields of medical imagery. In: 2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE), IEEE, 2017; doi:10.1109/ICEICE.2017.8192443.

[29] Wang D C C, Vagnucci A H, Li C C. Digital image enhancement: A survey. Computer Vision, Graphics, and Image Processing, 1983; 24(3): 363–381.

[30] Ahirwal B, Khadtare M, Mehta R. FPGA based system for color space transformation RGB to YIQ and YCbCr. In: 2007 International Conference on Intelligent and Advanced Systems, IEEE, 2007; pp.1345–1349.
[31] Liu Z, Chen W, Zou Y, Hu C. Regions of interest extraction based on HSV color space. In: IEEE 10th International Conference on Industrial Informatics, IEEE, 2012; pp.481–485. doi:10.1109/indin.2012.6301214.
[32] Asmare M H, Asirvadam V S, Iznita L. Color space selection for color image enhancement applications. In: 2009 International Conference on Signal Acquisition and Processing, IEEE, 2009; pp.208–212.
[33] Jiang W J, Yang Z T, Wang P F, Cao Q X. Navigation path points extraction method based on color space and depth information for combine harvester. In: 2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM), IEEE, 2020; pp.622–627. doi:10.1109/ICARM 49381.2020.9195358.
[34] Du Q, Wang D, Sha L. Recognition of mobile robot navigation path based on K-means algorithm. International Journal of Pattern Recognition and Artificial Intelligence, 2020; 34(8): 2059028.
[35] Mao J, Cao Z, Wang H, Zhang B, Niu W. Agricultural robot navigation path recognition based on k-means algorithm for large-scale image segmentation. In: 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA), IEEE, 2019; pp.1233–1237. doi:10.1109/ICIEA. 2019.8834296.
[36] Jin Y, Wen S, Shi Z, et al. Target recognition and navigation path optimization based on NAO robot. Applied Sciences, 2022; 12(17): 8466.
[37] Loresco P J M, Dadios E P, Valenzuela I C. Color space analysis using KNN for lettuce crop stages identification in smart farm setup. In: TENCON 2018: IEEE Region 10 Conference, IEEE, 2018; pp.2040–2044. doi:10.1109/TENCON.2018.8650209.
[38] Larijani M R, Asli‐Ardeh E A, Kozegar E, Loni R. Evaluation of image processing technique in identifying rice blast disease in field conditions based on KNN algorithm improvement by K-means. Food Science & Nutrition, 2019; 7(12): 3922–3930.
[39] Hossain E, Hossain M F, Rahaman M A. A color and texture based approach for the detection and classification of plant leaf disease using KNN classifier. In: 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), IEEE, 2019;doi:10.1109/ ECACE.2019.8679247.
[40] Weeks A R, Hague G E. Color segmentation in the HSI color space using the k-means algorithm. In: Nonlinear image processing VIII. SPIE, 1997; 3026: 143–154.
[41] Senthilkumaran N, Vaithegi S. Image segmentation by using thresholding techniques for medical images. Computer Science & Engineering:An International Journal, 2016; 6(1): 1–13.
[42] Woebbecke D M, Meyer G E, Von Bargen K, et al. Shape features for identifying young weeds using image analysis. Transactions of the ASAE, 1995; 38(1): 271–281.
[43] Ribeiro A, Fernández-Quintanilla C, Barroso J, Garcíaalegre M C. Development of an image analysis system for estimation of weed pressure. Precision Agriculture, 2005; 5: 169–174.
[44] Kataoka T, Kaneko T, Okamoto H, Hata S. Crop growth estimation system using machine vision. In: Proceedings 2003 IEEE/ASME international conference on advanced intelligent mechatronics (AIM 2003), IEEE, 2003; 2: b1079–b1083. doi:10.1109/AIM.2003.1225492.
[45] Hague T, Tillett N D, Wheeler H. Automated crop and weed monitoring in widely spaced cereals. Precision Agriculture, 2006; 7: 21–32.
[46] Meyer G E, Neto J C. Verification of color vegetation indices for automated crop imaging applications. Computers and electronics in agriculture, 2008; 63(2): 282–293.
[47] Pratt W K. Digital image processing. Wiley-Interscience. 2007.
[48] Jack K. Video demystified: a handbook for the digital engineer. Elsevier, 2011.
[49] Otsu N. A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics, 1979; 9(1): 62–66.
[50] Ramos A R, Gaspar P D. Algorithm for path recognition in-between tree rows for agricultural wheeled-mobile robots. International Journal of Mechanical and Mechatronics Engineering, 2019; 13(1): 34–37.
[51] García-Santillán I, Guerrero J M, Montalvo M, Pajares G. Curved and straight crop row detection by accumulation of green pixels from images in maize fields. Precision Agriculture, 2018; 19: 18–41.
[52] Zhang X Y, Li X N, Zhang B H, Zhou J, Tian G Z, Xiong Y J, et al. Automated robust crop-row detection in maize fields based on position clustering algorithm and shortest path method. Computers and Electronics in Agriculture, 2018; 154: 165–175.
[53] Chen J Q, Qiang H, Wu J H, Xu G W. Navigation path extraction for greenhouse cucumber-picking robots using the prediction-point Hough transform. Computers and Electronics in Agriculture, 2021; 180: 105911.
[54] Hough P V C. Method and means for recognizing complex patterns: U. S. Patent 3, 069, 654. 1962-12-18.
[55] Winterhalter W, Fleckenstein F V, Dornhege C, Burgard W. Crop row detection on tiny plants with the pattern hough transform. IEEE Robotics and Automation Letters, 2018; 3(4): 3394–3401.
[56] He B, Liu G, Ji Y, Si Y S, Gao R. Auto recognition of navigation path for harvest robot based on machine vision. In: Computer and Computing Technologies in Agriculture IV: 4th IFIP TC 12 Conference, CCTA 2010, Nanchang, China, October 22-25, 2010, Springer Berlin Heidelberg, 2011; 344: 138–148. doi:10.1007/978-3-642-18333-1_19.
[57] Amin A, Fischer S. A document skew detection method using the Hough transform. Pattern Analysis & Applications, 2000; 3: 243–253.