Recognition of Orchard Path Based on Machine Vision

Abstract:

In traditional orchards, lots of labors and material resources are required to carry fruits, spray pesticides and remove weeds. It has become increasingly prominent that autonomous labor burdens should be reduced. In this paper, a general orchard platform based on Field Programmable Gate Array (FPGA) is designed, which can be equipped with mowing and spraying systems to complete the whole orchard path. However, due to the limitation of orchard circumstances and other factors such as extreme illumination or light interference, there will be some position deviations in navigation by machine vision. To solve this problem, a method of orchard path recognition and location based on machine vision is proposed. The results show that this method can effectively adjust the deviation of path recognition through cameras and the identification and positioning accuracy are improved of the general orchard platform.

***Key Words*：** *Orchard; Machine Vision; Path Recognition; FPGA*

Introduction

With the advent of the intelligent era, there is also an urgent need to enhance the intelligence in agriculture. This paper mainly focuses on the navigation problem of the intelligence-based general orchard platform, of which the key technologies mainly include inter-ridge path recognition of the general orchard platform, and fitting of actual path and path curve by machine identification and etc.

In view of the complicated circumstances in the orchard field, some errors in identifying orchard environment only by means of computer vision will occur if there is no other auxiliary information. The image reflected from the camera is not the actual situation, it is a mapping of objects in the world coordinate system in the image coordinate system. In this research, image processing technology has been mainly utilized to optimize the visual part of the tracked trolley and make it obtain the driving navigation line in the orchard and realize the automatic operation.

Then comes to the recognition of ridges. Cameras are equipped to collect the images of the orchard according to the actual circumstance of orchard ridges, and the images are processed to identify fruit trees and backgrounds. In this way, the planting lines of fruit trees can be determined, thus obtaining the navigation reference line. Then, on the basis of the navigation reference line and the auxiliary sensors, the robot's posture at that moment and the angle and distance deviated from the reference line can be located. Finally, the motor is controlled by FPGA to make the general orchard platform move along the planned route.

In this experiment, OV5640 is used as the image sensor, and FPGA as the core controller. By means of such technologies as image acquisition, path recognition and lookup table, the control system of mowing robot is realized.

AN RIDGE LINE FITTING

It is noted that there are relatively few objects characterized by red, white or other colors in the orchard[1-3]. In view of the favorable condition, the method of “brushing red” on the trunks of fruit trees, that is, adding markers can be utilized. The process mainly consists of the following steps:

(1) Tie the red cloth on the bottom of the trunk to extract the red area: Based on converting the image from RGB to Lab color space, red extraction is performed based on "a-value" to quickly get the image of red trunk background. For the white trunk, the image is grayed out, and on this basis the white trunk background is extracted using the threshold segmentation method. According to the comparative study and analysis, red color can be selected as the feature.

(2) Noise removal: morphological closure operation is used to smooth the boundary and eliminate small noise.

(3) Trunk area extraction: The lateral scanning method is used to filter and distinguish, and identify the one-sided trunks on both sides of the ridge from left to right.

(4) Planting line fitting: Each connection domain (the intersection of the inner trunk with the ground) is scanned from top to bottom to fit a unilateral planting line.

(5) Planting line comparison: The fitted planting lines of the collected images are compared with the fitted planting lines of the standard images to determine the angles[4-5].

IMAGE ANALYSIS AND PRE-TREATMENT

Figure 1a is a natural scene image of experimental field of citrus orchard in Central China Agricultural University taken at 4: 00 p.m in the early March. Figure 1 shows that the trees are regularly arranged as a whole with dense leaves and gray branches, but the branches are blocked by leaves seriously. The field is covered by weeds so that trunks are not easy to be seen, without sharp contrast with other parts. In irregular shapes, trunks grow in different directions, and there are too many weeds. As a result, the image of the bottom of trunks is not clear, let alone there are some fallen leaves and uneven illumination.

In view of factors such as complex circumstances and obscure trunks, two methods - adding white markers or red markers (as shown in Figure 1(b) and Figure 1(c)) on trunks - are adopted. The image of the "white-painted" trunks has the following characteristics: ①The gray scale value of the white part of trunks is always higher than other parts, showing that white is easier to be noticed. ②The white part is greatly affected by light, and the gray scale value of different trunk parts varies. ③ There are many factors affecting the image, especially the color white. For example, a tendency for the image to show the white sky. The image of "red-painted" trunks has the following advantages: ①The R component of the red part of trunks is higher than other parts, which is always the most obvious color. ②Although there are great differences in the growth patterns of trunks in the forest, they are seldom influenced by light due to the inherent characteristics of red.

(a)Orchard environment map (b) Tree trunk painting white figure (c) Tree trunk brush red figure

Figure 1 pictures of orchard

In the actual treatment process for the "red-painted" trunks, firstly, based on the conversion between RGB color space and Lab color space, the original color image is converted into the Lab color space. The "a component" of Lab color space is able to make the target farmland bright and the background dark, which is helpful for distinguishing the target from the background. In order to make the treatment effect clearer, a trunk is painted red for demonstration, just as shown in Figure 2 below.

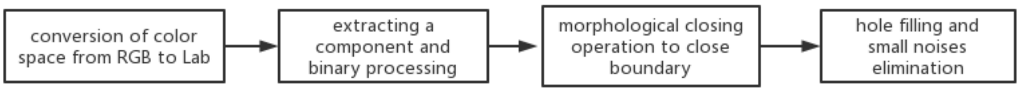


Figure 2 the flowchart of “brushing red trunks”

Color expression is separated from brightness information by the color space, which can overcome the influence of ambient light in the actual growing process to some extent. In the actual treatment process, firstly, based on the conversion between RGB color space and Lab color space, the obtained original color image is converted into the Lab color space. The conversion process is as follows:

$$\begin{cases} L=116*f(y)-16 \\ a^*=500*(f(x)-f(y)) \\ b^*=200*((f(y)-f(z)) \end{cases} \tag{1}$$

$$\begin{pmatrix} x & y & z \end{pmatrix} = \begin{pmatrix} 0.4125 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{pmatrix} * \left(\frac{r}{255} \quad \frac{g}{255} \quad \frac{b}{255}\right) \tag{2}$$

$$f(t) = \begin{cases} t^{1/3} & t \leq 0.008856 \\ 7.787*t+0.1379 & t > 0.008856 \end{cases} \tag{3}$$

The converted image can be turned into Figure 4b through grayscale processing. If there is a threshold segmentation of the a* component, the result is shown as figure 4b as the threshold is set to 30.

$$a^* = \begin{cases} a^* & a^* > T \\ T & a^* \leq T \end{cases} \tag{4}$$

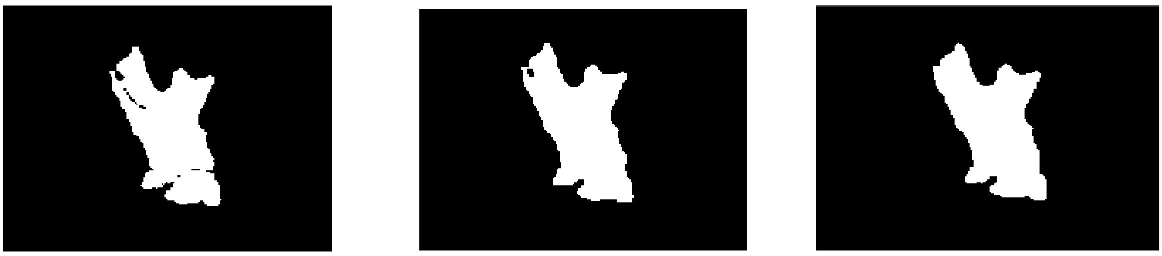
In order to better observe the processed trunk area, a trunk area in the image of Figure 4b is selected for demonstration, as shown in Figure 4c. There are local small holes in the segmented image. In view of this situation, morphological operation is used to remove the tiny holes, and the processing effect is shown in Figure 4d. It can be seen that small holes have been filled, but there are still large holes, so the process of hole filling is as follows:

$$X_k = (X_{k-1} \oplus B) \cap A^c \qquad k=1,2,3\ldots \tag{5}$$

A^c is the complementary set of image A, and B is a four-connected structural element.

The processed image is shown in figure 4e. It can be seen that all holes have been filled, and the boundary of image is obvious.

(a)Graying of a* component **(b) binary image**



(c) binary image **(d) morphological closing operation** **(e) hole filling.**

Figure 3 Effect drawings of image processing

FEATURE POINTS EXTRACTION.

There are many methods to obtain feature points, which can be based on specific shapes or some corners. According to the characteristics of different growth forms of fruit trees, and the trunk is not upright, the intersection point between trunk and ground is extracted as a feature point to represent the fruit tree, thus reducing the interfering factors. The algorithm used for extraction in this study is as follows:

(1) marking the processed trunk image to obtain the number of connected

domains(M) , and establishing an empty matrix a of M*2.

(2) Scanning each connected domain num line by line and column by column. The upper left corner of the image is the starting point of scanning, and the value of i increases from top to bottom and the value of j increases from left to right. If the pixel value at the current point (i, j) is 1, and the values at points (i, j+1) and (i+1, j) are 0, it can be considered that the point (i, j) is the feature point of the intersection point between trunk and ground, In which i=1, 2, …, a-1; j=1, 2,…, b-1; num=2,…, M。

(3) all the feature points are stored in the corresponding matrix A.

Using the above method to extract the feature points in the seven regions in the figure, the effect of which is shown in the figure 4. Star-shaped points represent the extracted results, and each point represents the junction of the trunk and the ground. From the figures 4, we can see that the points selected by this method are distinguishable, a requirement for a good feature, and can be used to extract feature lines.

Figure 4 Extraction of feature points

FEATURE LINE FITTING

The planting of fruit trees is generally arranged in a straight line, and the image taken also presents straight, so the straight line model is appropriate for fitting. The common methods to generate straight lines are least square method, Hough transform and so on. Because the number of extracted feature points is not large, and the miscellaneous points have been basically eliminated in the previous processing, the least square method with the fastest generation speed is adopted to fit the planting line. The algorithm for the least squares method can be described as: for n pixel data Pi(xi,yi) to be fitted, the straight line equation to be fitted is y=kx+b. Through calculation, the parameters of fitting straight line can be got, and the formulas are as follows:

$$\min \sum_{i=1}^{n} (y - kx_i - b)^2$$

$$k = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2} \qquad (6)$$

$$b = \frac{\sum y_i \sum x_i^2 - \sum x_i \sum x_i y_i}{n\sum x_i^2 - (\sum x_i)^2}$$

The effect is shown in Figure 5 below. The Blue line and red line respectively represent the left and right planting line generated by fitting feature points. The orchard navigation directrix is generated by calculating the perpendicular bisector of the left and right planting lines. The effect is shown by the white line in Figure 5.

Figure 5 navigation feature lines generation

POSE OBTAINMENT AND ADJUSTMENT IN IMAGES

The principle of image shooting is mainly perspective photography, that is, what the image reflects is not the actual situation, but the mapping of objects in the world coordinate according to the image coordinate system. Figure. 6 explicates, from a visual point of view, the differences between two generic parallel tree crop rows represented by a geographic coordinate system (Fig. 6a) and in an image-based coordinate system. What’s known is that complex 3D reconstruction technology is required to obtain 3D information from 2D images. Therefore, the fitting ridge lines of images obtained at various angles are measured from a fixed camera position in this study. Then by comparing the angles of images taken by cameras at various angles with those taken at standard positions, the transformation relationship between ridge line angles in world coordinate system and fruit tree line angles in image coordinate system under various camera angles is established, that is, the relationship between angles α and θ in Figure 6(c).

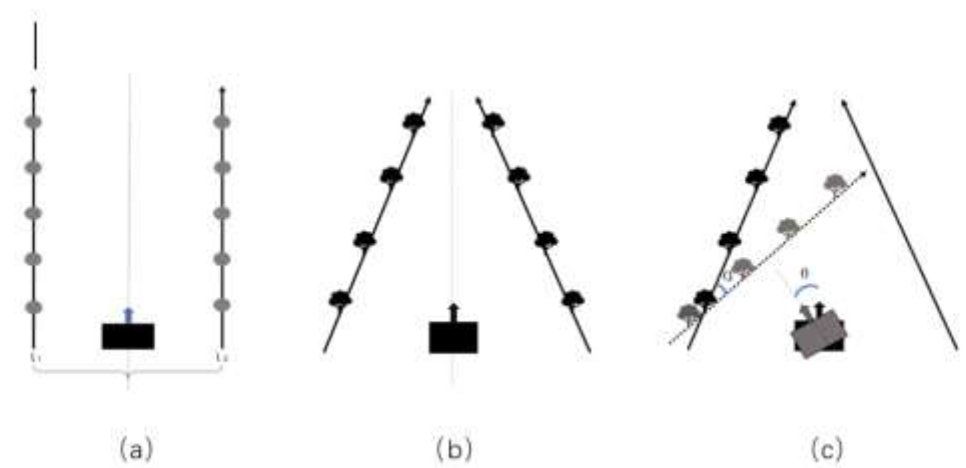


Figure 6 Fruit tree rows in different coordinate systems.

(a) fruit tree rows in world coordinate system (xw,yw,zw); (b) fruit tree rows in image coordinate system (u,v)

In a small angle, the left and right ridge planting lines can be extracted respectively, and so do the perpendicular bisectors of the two ridge lines. In this way, the deviation can be obtained. However, if there is a deviation of large angle, only a single ridge line can be identified in the image, which is the only basis for obtaining the deviation. Therefore, in this study, multiple experiments are carried out to shoot images of one-side ridge lines at various angles in a camera of fixed position, and establish the relationship between the image difference angle and the actual pose angle. Subsequently, by analyzing the included angle between the fitted feature line of the image taken from the real-time pose angle and that of the image from the standard pose, the real-time pose angle can be obtained through fitting their functional relationship.

The process consists of the processing the date of angles and the angle of the standard image, the obtainment of deviation angle of each camera angle and the deletion of some erroneous data. Then the fitting curve is obtained by Fourier fitting in Matlab, as shown in the following figure, and the relationship between the two angles is established as follows:

$$a = f(\theta) = a_0 + a_1 * \cos(\theta * w) + b_1 * \sin(\theta * w) \tag{7}$$

$$a_0 = 4.479, \quad b_1 = -0.8418, \quad w = 0.04138$$

The sum variance is SSE=0.6465, and the coefficient of determination is R-square=0.9979. The model has a good fitting effect on the data. As long as fitting the feature line from images collected in real time, and calculating the included angle α' of fitted feature lines in real time and standard pose, the actual pose angle θ' can be obtained according to formula (7).

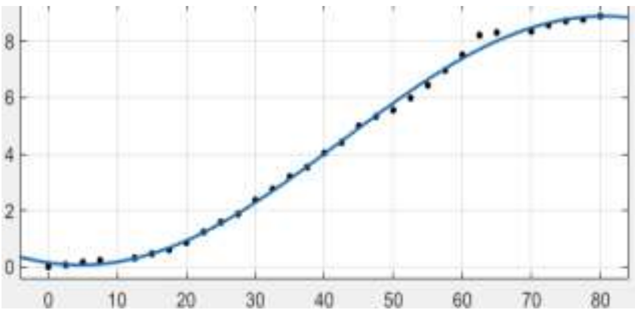


Figure 7 Fitting line of attitude angle and image difference angle

PATH RECOGNITION

Path recognition begins with binary processing of the image. Although several researches on image binarization algorithm have been proposed by several authors[8-9], the simplest fixed threshold method is generally used because of the high operation speed of visual navigation. However, the fixed threshold binarization method is sensitive to environmental lighting conditions and background noise, thus the corresponding path recognition effect is unstable and many conditions of environment should be met. Therefore, an adaptive path recognition method is proposed, which combines the two separate steps—image binarization and the determination of path position and width. In this method, starting from a set of preset path widths and binarization thresholds, then processing the image line by line, and the preset values of width and threshold are adjusted according to the width and threshold calculation results of the previous line. When the preset value and the measured value tend to be balanced, the recognition results tend to be accurate[6].

1) perform mean filtering on a row of images.

2) binarize the image according to the preset threshold.

3) search for path edges on both sides of FOV (field of view) from the preset path center point.

4) calculate the path width and the path center point, and compare the calculated path width value with the preset value. When the calculated width value is larger than the preset width value, increase the preset width and binarization threshold. On the contrary, adjust the path center point directly as the new preset path center point.

5) If the last line of the image frame is reached , skip to step 6), otherwise return to step 1) to process the next row of image data.

6) at the beginning of the next frame image, the path width and threshold obtained from the last line of the above image are returned as preset values to 1) for

processing.

In the adaptive algorithm above, image filtering and binarization are carried out at the same time, the system only needs to store a row of binarized images, which not only improves the processing speed, but also effectively reduces the storage demand. In step 3), the convolution method is used to detect the edge of the path and calculate the position and width of the path, just as shown in figure 8. Scanning from the preset center point to both sides at the same time to determine the edge of the path. Taking advantage of the continuity of the path itself, and the detection results of the previous row are applied in the detection of the next row, which effectively improves the stability and adaptability of the algorithm.

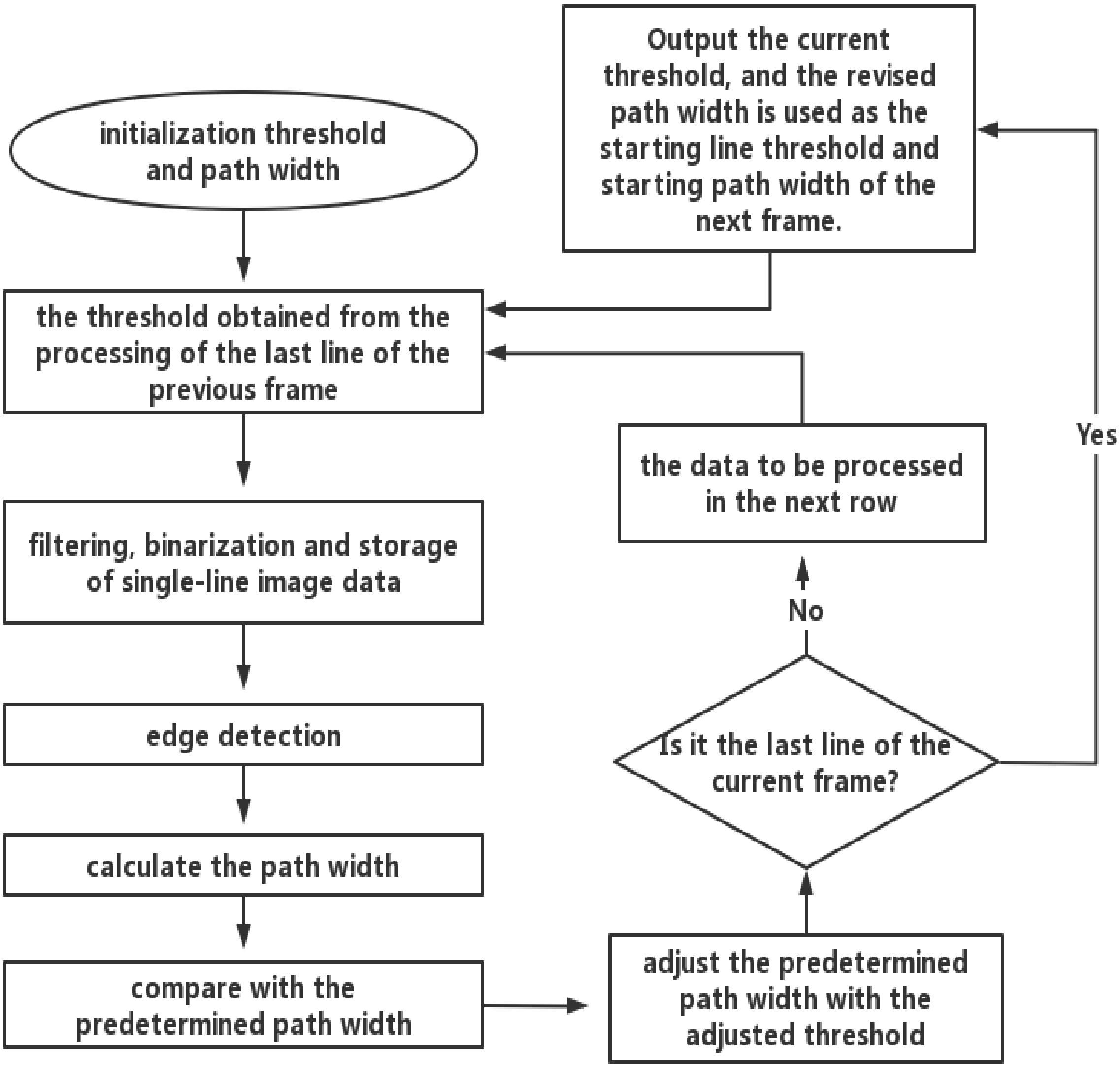


Figure 8: Flow chart of adaptive path recognition algorithm

Formula 8 the hardware implementation of the 2 convolution functions is as follows:

$$V_1 = \sum_{k=0}^{7} p_k \wedge 0 + \sum_{k=8}^{15} p_k \wedge 1; \qquad (8)$$
$$V_2 = \sum_{k=0}^{7} p_k \wedge 1 + \sum_{k=8}^{15} p_k \wedge 0.$$

As can be seen from the formula above, the value range of V1 and V2 is 8-16, so it is defined as the path edge point criterion when the convolution value is larger than or equal to 12. When the left and right edge search is over, the path center point and path width of this line are determined according to the following rules. Note that the path center point of the previous line is M0, the position of the left edge scanned previously is L, the position of the right edge is R, and the path center position M and the path width W of the current line are calculated as follows:

1) If the left and right edges are found at the same time, then $M=(L+R)/2$, $W=R-L$.

2) Only the left edge is found, then $M=(L+M0)/2$, $W=M-L$.

3) Only the right edge is found, then $M=(M0+R)/2$, $W=R-M0$

4) If the left and right edges are not detected, then $M=M0$, and the measured path width remains unchanged.

Among the above 4 points, except for 4), the other 3 points all adjust the threshold and the predetermined path width. we use the center position and width of the first line path of the previous image as the preset values of the center position and width of the next frame image.

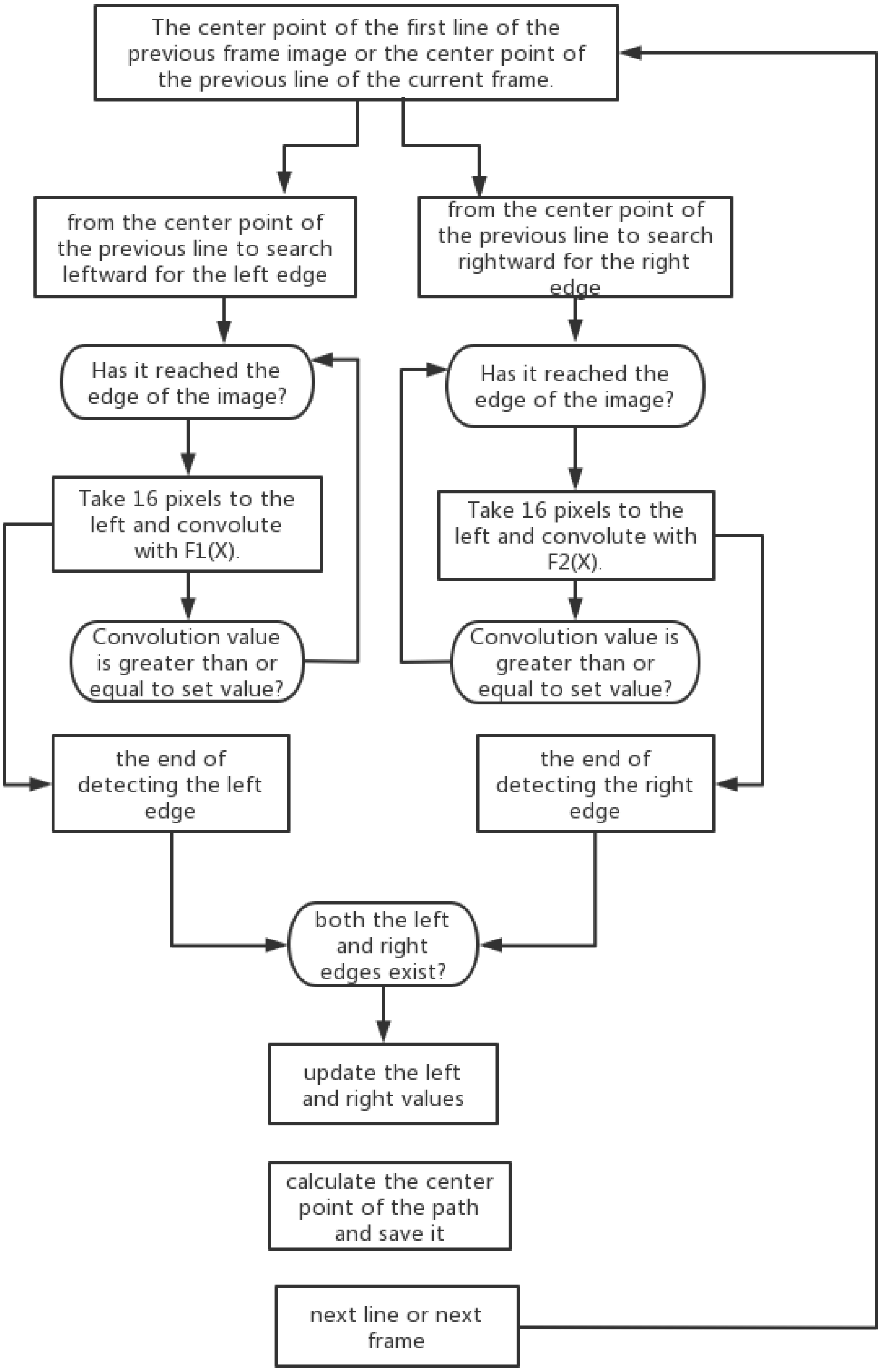


Figure 9: Flowchart of edge detection

Experimental Verification

Hardware and Software Platform

During the data acquisition, the crawler chassis is driven by manual remote control to simulate the normal working state of the equipment. The roof FPGA is equipped with OV5640 camera for data acquisition, and the video data is stored in the main control chip of FPGA. The camera is installed on the chassis cover through the PTZ (Pan-tilt Zoom), which is longitudinally m away from the front of the chassis, transversely located on the central axis of the chassis box, m away from the ground, and the shooting direction is directly in front of the engine body. The traveling speed of the machine is about 0.5m/s, the image resolution of the camera is 5 million pixels.

Figure 10: General orchard platform prototype and it hardware structure

Experimental Results

The images captured by cameras at various angles are processed to analyze the included angle between the characteristic lines at various camera angles and those in standard posture, and the data with large errors has been deleted. Fit the relationship between the two angles as follows:

$$\alpha = a_0 + \sum_{i=1}^{3}(a_i cosi * w\theta + b_i sini * w\theta) \tag{9}$$

For that：

[ai,bi]=[-1.623e+10,0;0,2.443e+12;2.921e+10,-1.954e+12;1.298e+10,4.885e+11]，

ω=0.002223。When the coefficient of determination R-square for measuring the goodness of fit is 0.9973, it means that the established model matches with the statistical fitting.

Table 1 Image information of each posture

Camera angle	Fitting the characteristic line	Image Angle	Camera angle	Fitting the characteristic line	Image Angle
0°	Y=-0.15715X+1583.1709	0°	42.5°	Y=-0.079076X+1608.1359	4.4098
2.5°	Y=-0.15600X+1591.2746	0.0642°	45°	Y=-0.068784X+1597.7458	4.9962
5°	Y=-0.15445X+1601.5084	0.1509°	47.5°	Y=-0.06287X+1418.6947	5.3335
7.5°	Y=-0.15322X+1594.9086	0.2199°	50°	Y=-0.058935X+1588.2498	5.5583
10°	Y=-0.16564X+1632.7864	0.474°	52.5°	Y=-0.051375X+1579.78	5.9901
12.5°	Y=-0.15169X+1634.4968	0.3055	55°	Y=-0.043541X+1571.8048	6.4380
15°	Y=-0.14898X+1621.8908	0.4574	57.5°	Y=-0.034177X+1559.6587	6.9736
17.5°	Y=-0.14637X+1636.5501	0.6037	60°	Y=-0.02467X+1545.7294	7.5179
20°	Y=-0.14185X+1635.2228	0.8574	62.5°	Y=-0.012415X+1527.179	8.2198
22.5°	Y=-0.13528X+1635.0452	1.2270	65°	Y=-0.010794X+1524.826	8.3127
25°	Y=-0.12908X+1634.0633	1.5762	67.5°	Y=-0.008127X+1507.0108	8.4654
27.5°	Y=-0.12389X+1634.5116	1.8686	70°	Y=-0.010038X+1534.2886	8.3560
30°	Y=-0.11519X+1630.2367	2.3604	72.5°	Y=-0.006124X+1514.2138	8.5801
32.5°	Y=-0.10817X+1627.5717	2.7575	75°	Y=-0.003572X+1543.4836	8.7263
35°	Y=-0.10048X+1621.4510	3.1930	77.5°	Y=-0.0026473X+1525.6206	8.7793
37.5°	Y=-0.094535X+1619.4611	3.5307	80°	Y=-0.00052882X+1523.6126	8.9008
40°	Y=-0.08564X+1447.2473	4.0361			

The pictures below show the sobel edge detection between orchard ridges in Central China Agricultural University through FPGA platform. Figure 11 is the overall road map and ridge line labeling in the image coordinate system. Figure 12 is the image of the trunk navigation line extracted under sobel edge detection at 0~180°, every 15 degrees of the trolley.

Figure 11 the ridge lines in the image coordinate system before image fitting

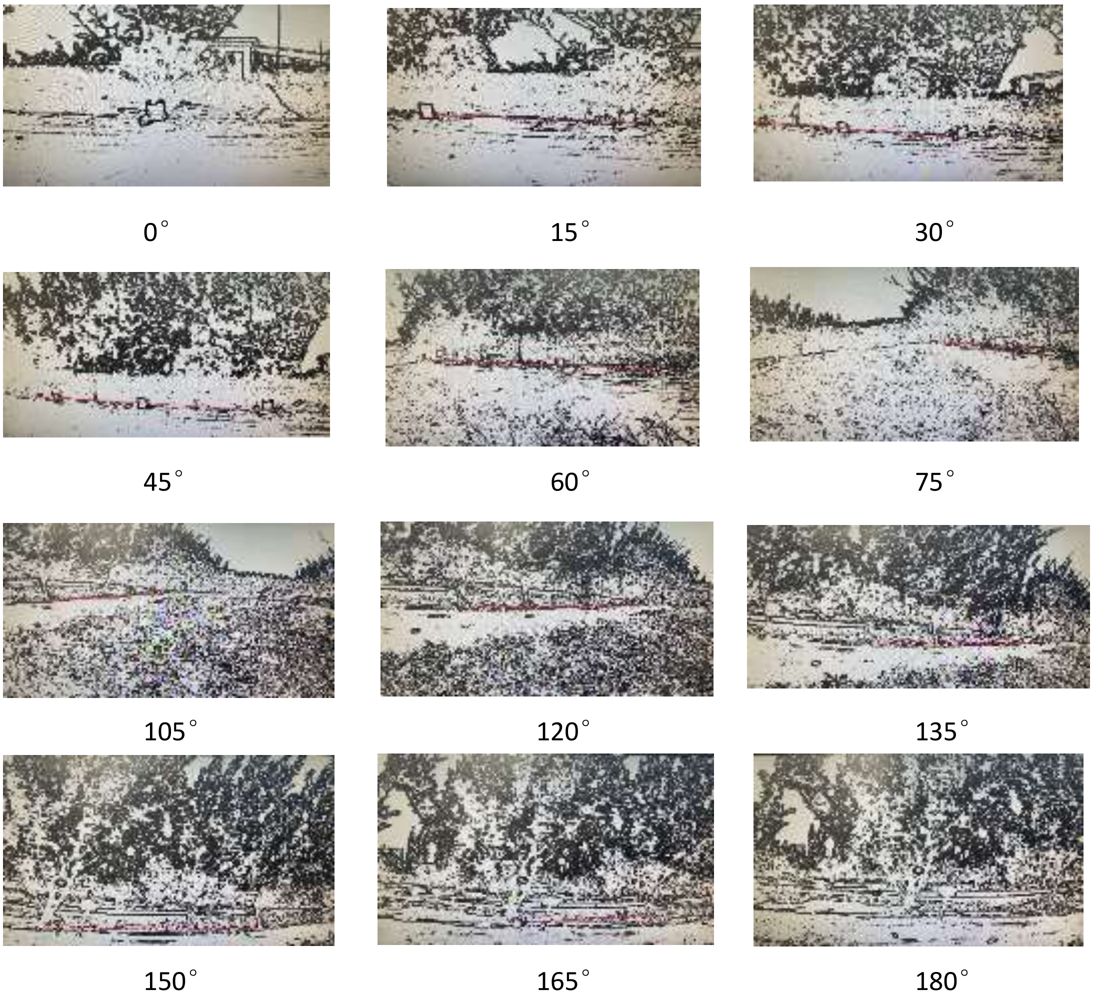


Figure 12 Image of trunk navigation line extracted under sobel edge detection

The experimental results prove that, by means of the algorithm, the ridge lines in the image coordinate system can be effectively fitted, and the actual deviation relationship in the image coordinate system can be reflected as the general platform rotates its position in the world coordinate system.

CONCLUSION

Aiming to solve problems of path identification in the visual navigation of general orchard platforms, this paper puts forward an overall design scheme based on FPGA to realize path image identification and processing with the help of the image processing technology. By comparing the angles of images taken by cameras at various angles with those taken at standard positions, the transformation relation between ridge line angles in the world coordinate system and corresponding fruit tree row angles in the image coordinate system under various camera angles is established. Then the curve between them can be fitted. For the fast moving average algorithm used in the paper of directrix, an adaptive path identification method is proposed in the paper. The method combines the two discrete steps of image binarization and the determination of path position and width, and processes the image line by line from a set of preset path width and binarization threshold. With simple hardware structure, this system mentioned in the paper features high integration and impressive real-time performance, providing strong support for the follow-up work of path identification.

REFERENCES

[1]Chen Hao & Weng Xianyao. (2010). FPGA-based preprocessing design of smart car path image recognition. Automation Technology and Application (02), 107-110+116. doi:CNKI:SUN:ZDHJ.0.2010-02-031 .

[2]Gong Jinliang, Wang Xiangxiang, Zhang Yanfei & Lan Yubin. (2020). Corn rhizome navigation line extraction method based on edge detection and region positioning. Journal of Agricultural Machinery (10), 26-33. doi:

[3]Li Muzi. (2019). Design and implementation of target detection and tracking system based on image recognition (Master's thesis, Beijing University of Posts and Telecommunications). dbname=CMFD201902&filename=1019047169.nh

[4]Lei Haijun, Li Dehua, Xue Lei, Guan Jinghuo. (2002). FPGA+DSP Aerial Photo Parallel Processing System. Journal of Huazhong University of Science and Technology (Natural Science Edition) (11), 28-30. doi: 10.13245/j.hust .2002.11.010.

[5]Yang Fan,Zhang Hao,Ma Xinwen,Jiang Yong.Image processing system based on FPGA[J].Journal of Huazhong University of Science and Technology:Natural Science Edition,2015,43(2):119-123.

[6]Zheng Xinqian, Wang Fuming, Ma Hua, Lin Yun & Zhang Wei. (2012). FPGA-based design and implementation of visual navigation car. Journal of Xiamen University (Natural Science Edition) (03), 331-335. doi:

[7]Wang Zhen. Wheeled robot visual navigation and control system based on ARM/DSP [D]. North China University of Technology, 2015.

[8]Liu Jing, Liu Yu & Lu Yuhua. (2010). Visual navigation wheeled robot based on FPGA+ARM. Computer Engineering (21), 194-195+198. doi:CNKI:SUN:JSJC.0.2010-21- 070.

[9]Wu Yan. Research and implementation of basic algorithms for digital image processing based on FPGA[D]. Harbin Institute of Technology, 2008.

[10]Jiao Jianxiong, Zhao He, Luo Yinglong, Yin Pengcheng & Liu Ziyan. (2016). Design of

smart car based on FPGA. Microcomputer and Application (11), 22-24. doi:10.19358/j.issn.1674-7720.2016. 11.007.

[11]Jing Gu, Feng Xiaoxue & Zhang Xuesong. (2016). Integrated control of direction and speed of blind guidance system and FPGA implementation. Journal of Harbin University of Science and Technology (06), 55-60. doi:10.15938/j.jhust.2016.06.011.

[12]HanDapeng，WeiQing，I．iZexiang．Pathfollowingofmo—bilerobotsusingavirtualvehicleapproach[C]／／Control Conference．Harbin，China：IEEE，2006：1533-1537．

[13]WangYue，TeohEK，ShenDinggang．Lanedetectionand trackingusingBsnaker'J]．ImageandVisionComputing， 2004(22)：269-280．

[14]Yang Bin, Xie Liang & Jin Xiangliang. (2021). FPGA-based real-time image acquisition and display system design. China Integrated Circuit (11), 61-64+72. doi:CNKI:SUN:JCDI.0.2021-11 -013.

[15]Liu Tengfei & Liu Wei. (2020). FPGA image acquisition and binary image display system design. Single chip microcomputer and embedded system application (10), 39-42. doi:CNKI:SUN:DPJY.0.2020-10 -014.

[16]Pei Zhengxiong & Peng Anjin. (2019). Image acquisition and processing system design based on FPGA and OV5640. Electromechanical Information (32), 159-160. doi:10.19514/j.cnki.cn32-1628/tm.2019.32. 089.

[17]Lei Haodong. Design of image acquisition and preprocessing system based on FPGA[D]. University of Electronic Science and Technology of China, 2019.

[18]Li Yuchuan. Design of smart car based on FPGA control[J]. Computer Measurement and

Control, 2018,26(07):52-56.DOI:10.16526/j.cnki.11-4762/tp.2018.07.012 .

[19]Li Bingqi. Target detection and tracking based on FPGA [D]. Beijing University of Posts and Telecommunications, 2019.

[20]Wu Jinlan. Intelligent car control technology based on FPGA[D]. Hebei University of Technology, 2015.

[21]Rui Qingbo. Intelligent car system based on FPGA[D]. Jiangsu University, 2007.

[22]Cui Chenpeng. Research and hardware realization of AGV visual guidance algorithm based on FPGA[D]. Nanjing University of Aeronautics and Astronautics, 2011.

[23]Shan Dan, Cong Guotao & Zhang Xiaoxu. (2018). FPGA-based multifunctional smart car design. China Integrated Circuit (09), 67-71. doi:CNKI:SUN:JCDI.0.2018-09-019.