Program Report

Writer: Liu Yongfan 61516126
Team members: Shi Yunpeng 61516113
Deng Shuheng 61516116

1. Function Achieved

This program achieved the recognition about the color of the flower, the thickness of the stem and the split angle of the plant branches.

The color of the flower: By color recognition program, and then applied to the original mask manner, the program feedbacks to the user is a color corresponding to color plate.

The thickness of the stem: The program first identifies the size of each block in the graph paper and then identifies the stem of the tomato. The system measures the thickness of the stem at intervals and then compares the measurement to the size of the block. The final, program will return a picture containing the annotation of the measurement point of the stem and the measured data file.

The split angle of the plant branches: The program first identifies the skeleton of the plant, then linearizes the skeleton and removes some of the repeated lines. The program will return a line with a image to the plant skeleton and a data file containing the angle between any two lines.

2. Technical Introduction:

2.1.Color Recognition

Color recognition is a relatively simple task, and you only need to have a good understanding of the concept of color space. Flow chart such as fig. 1

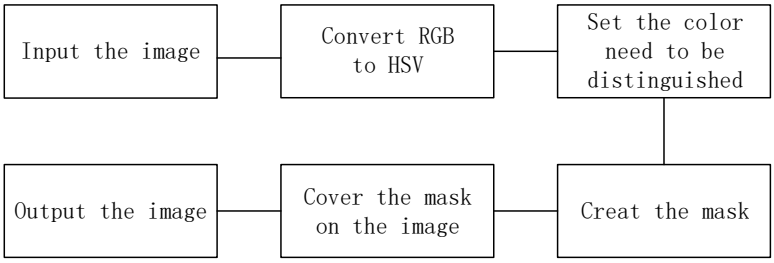


fig 1.

2.1.1. HSV Color Space

HSV (hue, saturation, value) are alternative representations of the RGB color model, designed in the 1970s by computer graphics researchers to more closely align with the way human vision perceives color-making attributes. In these models, colors of each hue are arranged in a radial slice, around a central axis of neutral colors which ranges from black at the bottom to white at the top. The HSV representation models the way paints of different colors mix together, with the saturation dimension resembling various shades of brightly colored paint, and the value dimension resembling the mixture of those paints with varying amounts of black or white paint. So the color extraction and recognition in the HSV color space will get better

results.

2.1.2. Mask

Based on the system's distinction of images in the HSV color space, masks can be made based on the color. In fact, there are two sets of color recognition functions defined. One of the function, yellowMask, is used to distinguish yellow, while other function, greenMask, is used to distinguish green. Since the flowers of tomato are mostly yellow, yellowMask is used for color recognition of this part. GreenMask will be used in later work.

After getting the mask, overlay it on the original image, so that the original image only has the part of the color we want. The work of color recognition is completed.

2.2.Size Detection

The work of size detection is more complicated than imagined. In fact, there are two different types of objects waiting for their size to be detected - the distance of the squares and the thickness of the stems. Detecting the distance of the square is a necessary job, and only then can we determine a unit distance. Next, compare this unit distance with the thickness of the stem to know the thickness of the stem in reality. It doesn't make sense to just detect that the stem occupies a few pixels in the picture. The flow chart is like fig2.

It should be pointed out that the image has been preprocessed by OSTU threshold segmentation before it is processed as follows.

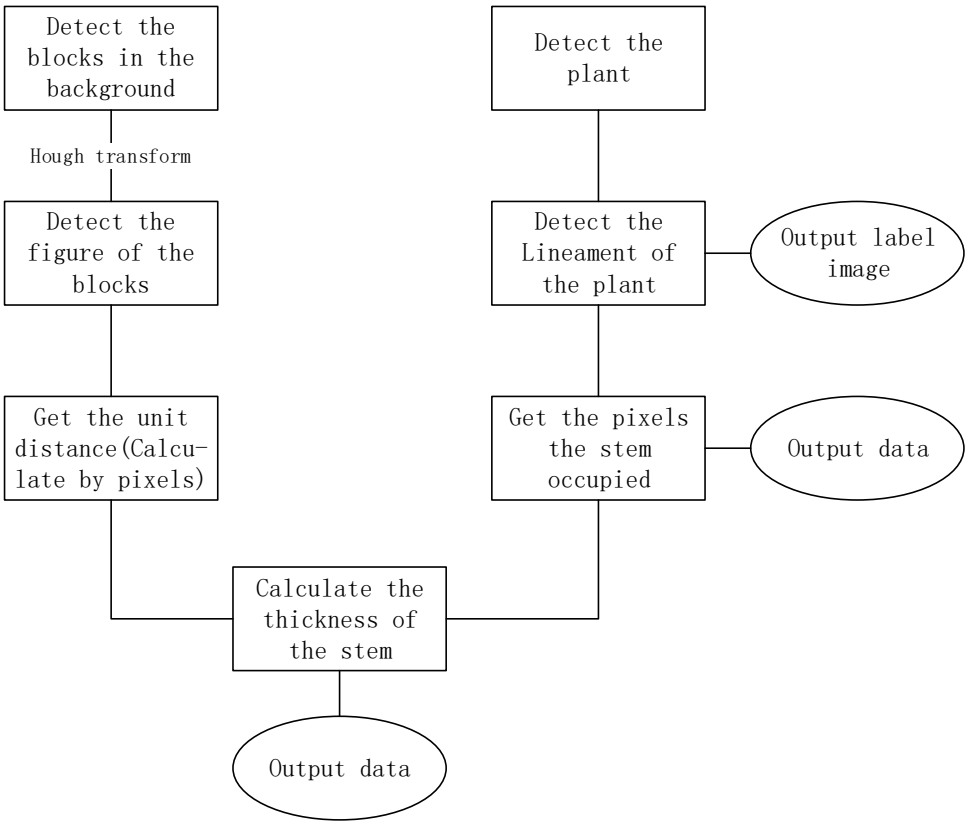


fig 2.

2.2.1. Detect the Blocks

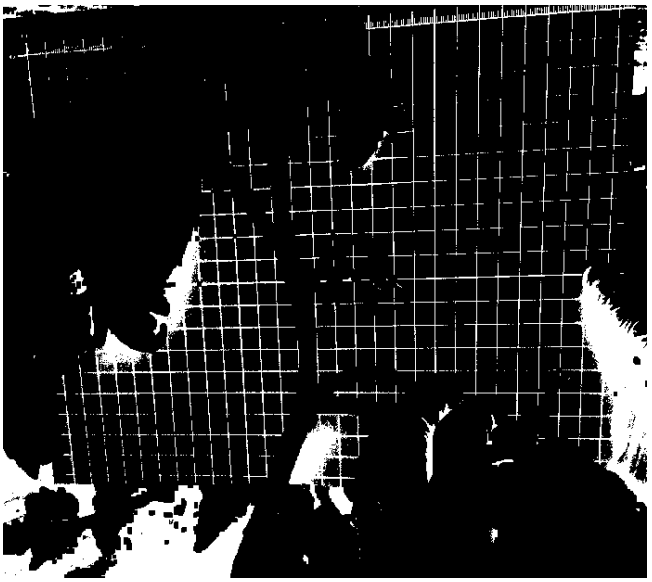
First, we need to separate the blocks in the background from the image. However, it is a

difficult job to put many black staggered thin lines from this. But we can consider this work from the opposite direction, how to remove the background block. This makes the problem simple, because these thin lines can be eliminated by open operations. Set the original image to $\mathbf{M}_{src}$. After the open operation, the image obtained is $\mathbf{M}_{open}$. Then the image $\mathbf{M}_{block}$ containing only the block is:

$$\mathbf{M}_{block} = \mathbf{M}_{src} - \mathbf{M}_{open} \quad (1)$$

The resulting rendering is like fig3.

(a) (b)



(c)

fig 3. (a) $\mathbf{M}_{src}$ (b)$\mathbf{M}_{open}$ (c) $\mathbf{M}_{block}$

As you can see, there are some unwanted images in the $\mathbf{M}_{block}$, but you don't need to worry about them. The impact they can have is limited. These limited impacts will be eliminated in later work.

2.2.2. Hough Transfer

The classical Hough transform was concerned with the identification of lines in the image, but later the Hough transform has been extended to identifying positions of arbitrary shapes, most commonly circles or ellipses. What we need is just to detect the lines in the $\mathbf{M}_{block}$. Since

the Hough transform is well known, its principle will not be repeated here. What is concerned is how to implement it.

We hope that the Hough transform will help us identify the lines in the $\mathbf{M}_{\mathbf{block}}$ so that we can accurately get the coordinates of the line in the block. In turn, the coordinates of the block can be calculated by these coordinates.

2.2.3. Get the unit distance

In fact, the Hough transform is not as friendly as we think. As mentioned above, there are some unnecessary images in the $\mathbf{M}_{\mathbf{block}}$ that have an effect. So when the Hough transform is done, we get bad results. Just like fig 4.

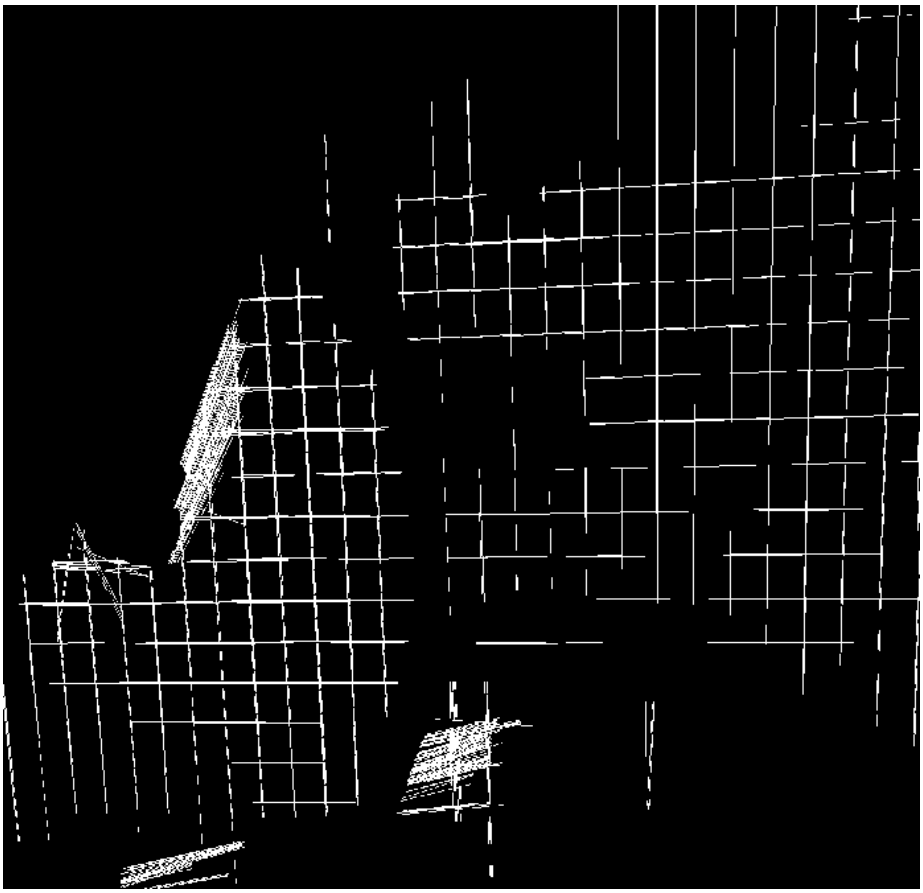


fig 4.

As you can see, there are many lines that coincide, and the program recognizes some images that are not part of the block as straight lines. The following two types of error information need to be processed:

- Lines not belonging to blocks
- Lines that overlap with each other

It is easy to see that the lines belonging to the block are vertical or horizontal, so you can screen out the lines that do not meet the requirements by calculating the slope of the line in the graph. Then, by calculating the distance between the straight lines, the lines that are coincident together can be eliminated, or the lines that are too close together to meet the requirements.

Finally, the pure $\mathbf{M}_{\mathbf{block}}$ shows, such as fig 5. We can calculate the unit distance of the block

based on this result. The end result is a block with approximately 57 pixels.

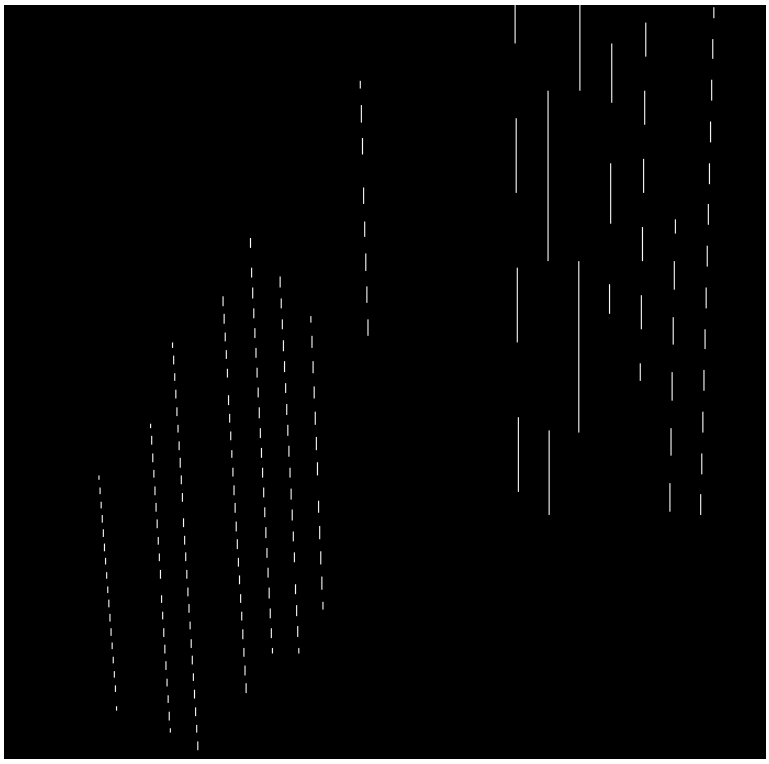


fig 5. The line looks broken, actually not. This is caused by image scaling.

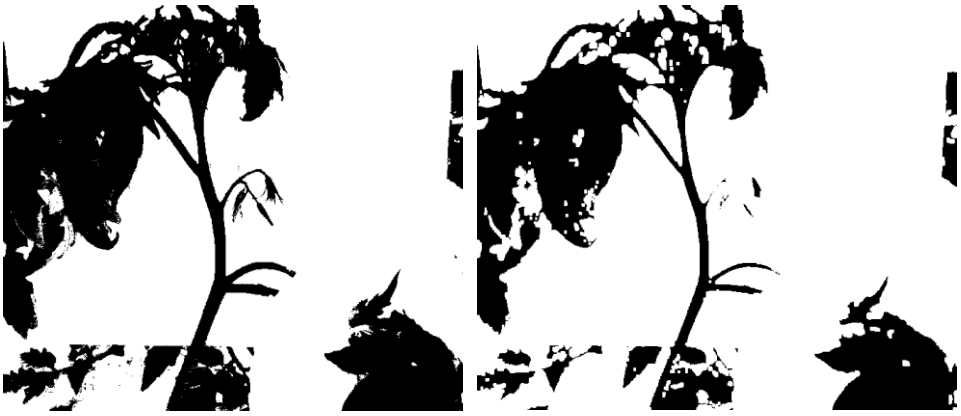
2.2.4. Detect the Plant and the Lineament

Detecting plants in an image is a relatively simple task. The mask method is proposed in 2.1.2, so the image area $\mathbf{M}_{\mathbf{plant}}$ containing only plants can be obtained by the greenMask function.

Getting the outline of the plant is a key task, and now $\mathbf{M}_{\mathbf{plant}}$ is ready, and if it is etched, it will get a $\mathbf{M}_{\mathbf{erode}}$ with a smaller stem than the original. The lineament map $\mathbf{M}_{\mathbf{lm}}$ is very simple:

$$\mathbf{M}_{\mathbf{lm}} = \mathbf{M}_{\mathbf{plant}} - \mathbf{M}_{\mathbf{erode}} \quad (2)$$

The resulting rendering is fig 6.



(a) (b)

fig 6. (a) M_{plant} (b) M_{erode} (c) M_{lm}

2.2.5. Measure the Stem

Now that we have the lineament map of the plant, the next task is to correctly identify the stem of the plant in the outline and measure the thickness of the stem.

We take a strategy of horizontal or vertical scanning at intervals. During horizontal scanning, stems have significant characteristics, such as fig 7:

fig 7.

The sides of the stem have a clear lineament (1 interval), the distance of 0 interval between the two contours will not be very wide, and the distance between the lineament (1 interval) will be narrower. So as long as the program recognizes this characteristic of the stem in each scan, the orientation of the stem can be determined. The state machine was designed:

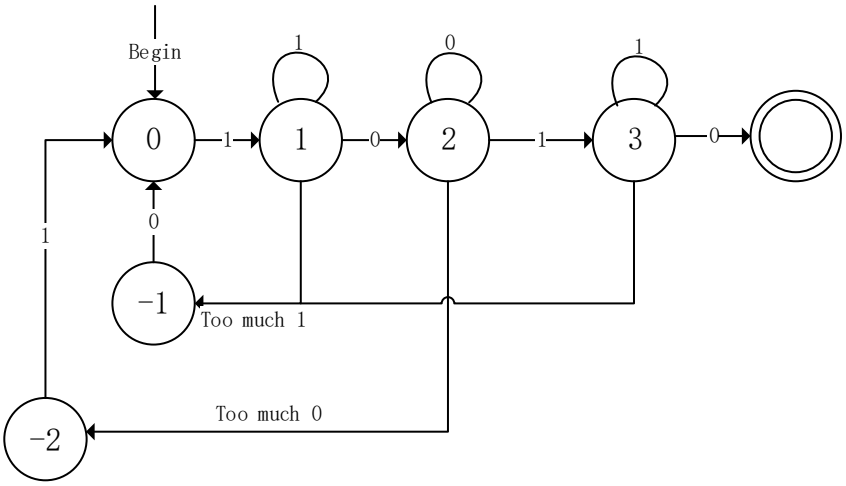


fig 8. The numbers in the circles are flags,
the numbers on the lines mean the next pixel's value.
'Too much 1' means the number of 1-pixels is out of range,

‘Too much 0’ means the same.

This state machine distinguishes stems from other objects well, and this state machine has a counter that, when it completes the identification of the stem, also completes the measurement of the pixels occupied by the stem.

2.3.Angle Detection

To detect the angle of a plant branch, first extract the skeleton of the plant. The drawing of the distance map is a preparation for extracting the skeleton. Since the skeleton extraction algorithm and the distance map drawing method are not well-known methods, it is necessary to introduce them in detail. After the skeleton is extracted, the Hough transform is used again. The Hough transform and our design of the filtering algorithm can complete the linearization of the skeleton to construct a rigorous measurement environment. Finally, the program calculates the angle between all the lines and returns them to the user in the form of a table and a label image. Flow chart such as fig 9.

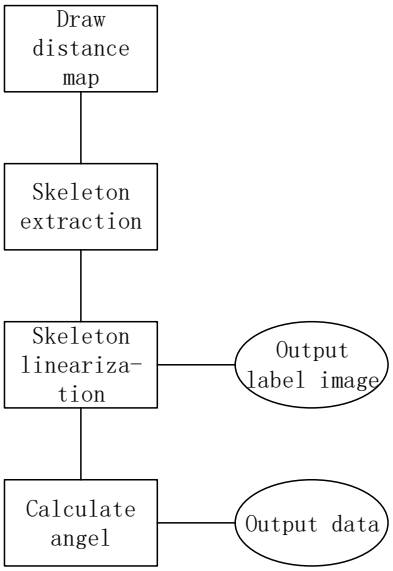


fig 9.

2.3.1. Distance map

The distance map is the shortest distance from each pixel in the foreground to the background in a binarized image. Suppose there is a pixel A with 4 neighbor pixels { N1, N2, N3, N4}. Distance(x) represents the shortest distance of pixel x. Then according to the idea of iteration, we can calculate:

$$distance(A) = \min\{ distance(N1), distance(N2), distance(N3), distance(N4)\} + 1 \quad (3)$$

The pseudo code for this algorithm is:

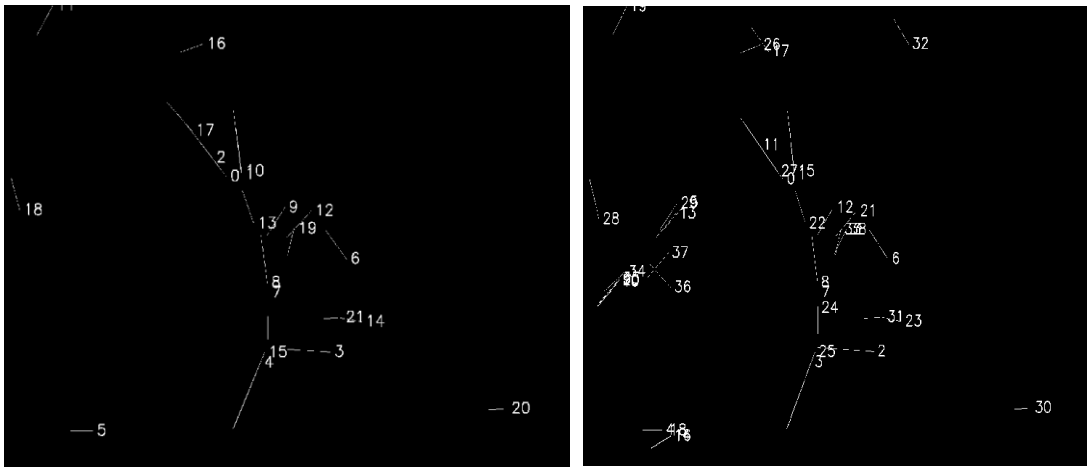
```
All the background pixel = 0
All the foreground pixel = 1
Scan the image form up to down, left to right
    distance(A) = min{ distance(N1), distance(N2)} + 1
Scan the image form down to up, right to left
    distance(A) = min{ distance(N3), distance(N4), distance(A) } + 1
```

2.3.2. Skeleton Extraction

When the distance map is obtained, we give the conclusion that the pixel at the local maximum in the distance map must be on the skeleton. According to this assertion, the skeleton of the plant was successfully extracted:

fig 10. The skeleton of the plant

The next task is to linearize the skeleton, and the Hough transform is used again, and like last time, the Hough transform gives us a lot of wrong information. Also after a certain screening algorithm, the excess line is removed. It should be noted that we have specially prepared a parameter LINE_CAPTURE in this section. Users can modify this parameter according to the actual situation and their own needs. When the value of this parameter is larger, the sensitivity of the system to the recognition of plant branches is higher, such as fig 11.



(a) (b)

fig 11. (a) LINE_CAPTURE = 12 (b) LINE_CAPTURE = 15

2.3.3. Angle Measurement

Completing the work of linearization means that we know exactly the exact orientation of each line. We calculate the angle between each line at a time and feed it back to the user. The user can check the angle between the two lines according to the actual needs, that is, the angle between any branches.

3. Program Instructions

Before using the program:

The user puts the image to be detected into the directory where the program is located, 2 images are required;

The picture used to measure stem thickness and branch angle is named "1.jpg"

The image used to detect the color is named "2.jpg"

While using the program:

It takes a while for the program to run, please be patient

The user can change the parameter LINE_CAPTURE in the program according to the actual situation and his own needs. When the value of this parameter is larger, the sensitivity of the system to the recognition of plant branches is higher.

After using the program:

The program will output 4 files, “color_config.jpg ”, “stem-measure.jpg”, ”stem-measure.csv”, “angle-measure.jpg” and ”angle-measure.csv”.

“color_config.jpg ” is the result of the color reorganization. The user can find the yellow in the image, so that he/she can know what the color they are of the flowers.

“stem-measure.jpg” is the label image for measuring the thickness of the stem.

”stem-measure.csv” is the result data for measuring the thickness of the stem. We suggest open this file by Excel. The first column is the serial numbers of the lines, the second column is the pixels the stem occupied, the third column is the unit distance of the stem thickness.

“angle-measure.jpg” is the label image for measuring the angle of the branch.

”stem-measure.csv” is the result data for measuring the angle of the branch. We suggest open this file by Excel. The first column and row are the serial numbers of the lines, the other data are the angle value between two lines in angle system. You can check any angles between any lines showed in the label image.

Wish you a pleasant experience in the process of using this program.