RootRobot: Camera Calibration Guide

Version 1.0

Peter Pietrzyk, Suxing Liu - University of Georgia

September 2021

Contents

Contents 2
Introduction 3
Tools 4
VNC Viewer 4
TCam Capture 4
Pre-calibration: Setting up baseline adjustments 5
Camera calibration 5
1. Camera position 5
2. Camera orientation 6
3. Focus 7
4. Depth of Field (aperture) 8
5. Illumination 9
6. Exposure time 10
7. Gain 10
TCam Capture settings and Python Code 11
Example Settings 11
Pre- and Post-calibration comparison 12
TIS Camera Properties 13
Imaging Chamber Operation Procedure 14
Introduction 14
Operation 14

Introduction

This document gives an overview to calibrate the camera settings of the RootRobot in order to increase point density and improve quality of 3D root models. This will be achieved by adjusting all cameras such that they provide higher quality images of roots. Improvements can be summarized by improved image composition, image sharpness, image brightness and reduced noise. An overview of all goals to achieve this is given in Table 1. Generally, there exist tradeoffs between some goals and settings. The user should determine how to balance these to get the best images.

Table 1: Overview of goals to achieve a high quality in root images

Goals	**Action**	**Tradeoffs/dependencies/other**
Capture image of whole root.	Reposition/reorient camera.	Depends on root position and size
Increase overlap between images.	Reposition/reorient camera.	Depends on root position and size
Focus on center of root.	Center the focal plane on vertical axis of root.	Depends on root position and size. Front or back parts of root might be out of focus.
Keep all root parts (front and back) in focus by increasing depth of field.	Close aperture by using a larger F-number (e.g. F/11 or F/16).	Depends on root position and size. Results in darker images (compensate by adding more light).
Homogenous lightning of root.	Add more diffuse light from front.	If adding more light is not possible, then increase exposure time and gain.
Avoid camera shake and motion blur.	Keep exposure time as short as possible (ideally shorter than 1/30 seconds = 33,333µs).	Results in darker images (compensate by adding more light)
Limit noise in image.	Set gain close to minimum value (higher gain results in speckle). Ideally set to zero.	Results in darker images (compensate by adding more light)
Avoid overexposure of root.	Set gain to lower value. Reduce exposure time. Use diffuse light from multiple sources.	Always avoid overexposed parts, since they cannot be fixed. Underexposed parts are more acceptable.

The items below show the overview of steps needed to calibrate the scanner cameras**. The calibration should be performed in this order** since they might affect other adjustments. Otherwise, subsequent steps might need to be repeated (e.g. the focus has to be re-adjusted after a camera has been moved). These steps are further explained in subsequent sections.

- Pre-calibration of cameras
- Calibration of cameras
  1) Repositioning of cameras
  2) Reorientation of cameras
  3) Adaptation of focus
  4) Adaptation of depth of field (aperture)
  5) Setting illumination
  6) Setting exposure time (with zero gain) -> Go back to 5) if still too dark
  7) Setting gain -> Go back to 5) or 6) if still too dark

Tools

This section gives an overview of software that can be used to access individual cameras.

VNC Viewer

Install the VNC Viewer on your computer to remotely access each RaspberryPi individually using local network IP address: 192.168.1.101~192.168.1.110, corresponding to PI01~Pi09 and PiController. Note that PiController's IP address is 192.168.1.110. (File -> New Connection -> Enter IP address under VNC Server -> Press Ok)

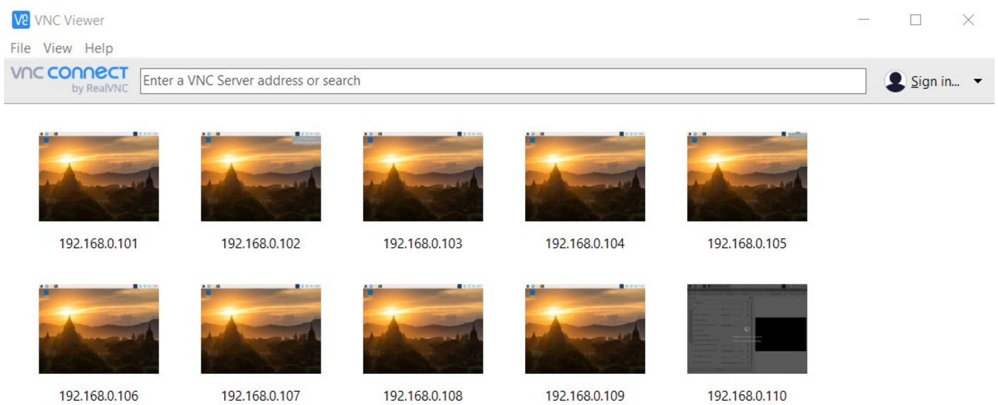

Figure 1: VNC Viewer

TCam Capture

Log into each individual RaspberryPi using the VNC Viewer and open TCam Capture (SDK of camera module), which is installed on each RaspberryPi and can be accessed , to view images of individual cameras in real time, to test different camera settings and to save sample images. By default, TCam Capture saves all images in the "/tmp" folder.

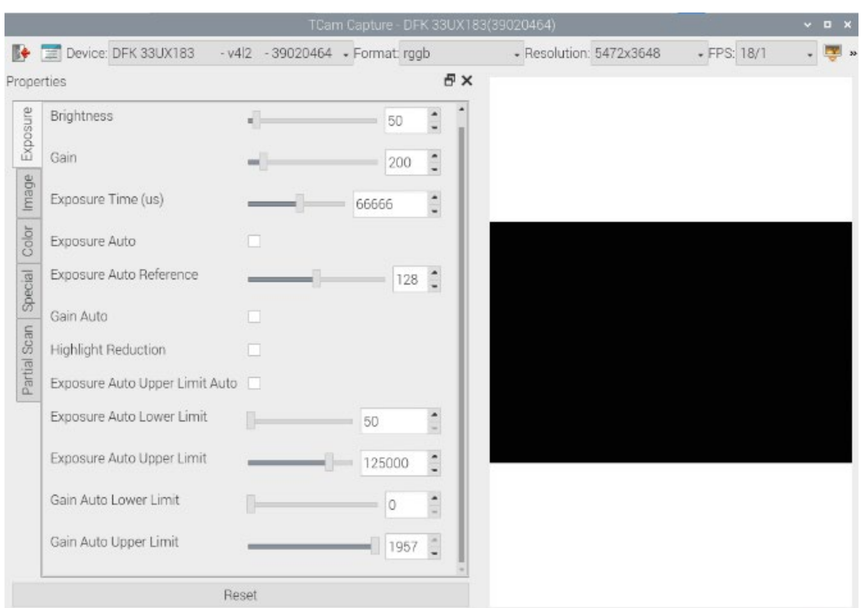

Figure 2: TCam Capture Software can be used to view camera image and to change settings in real time

Pre-calibration: Setting up baseline adjustments

As a first step, make rough adjustment, such that images can be used for an initial reconstruction. If the images are already sufficiently good then skip this step. If desired, scan a set of roots with these settings and construct 3D root models. These can be compared to reconstructed roots after the calibration. As a starting point, you can reset the settings in TCam Capture and select to use 'Exposure Auto' and 'Gain Auto' by checking the corresponding boxes.

Example: Before our calibration, many images were overexposed (see Figure 3). We therefore turned off most lights and used only the small light strip next to the cameras. Additionally, we set all camera apertures to F/8 to increase the depth of field and to make the images darker (see Figure 4). The goal of increasing the depth of field was to capture more layers of root inner structures. All other settings remained the same (e.g. focal distance, auto exposure, auto gain).

Figure 3: Before baseline adjustments; Image is overexposed

Figure 4: After baseline adjustments; Better, but still overexposed image

Camera calibration

1. Camera position

Next, adjust the camera position. Start by adjusting the position of the top camera (connected to PiController). During our calibration, we moved the top camera slightly up to increase the distance to the root and to allow for better focusing of the closest roots (Figure 5). This was performed because the camera manufacturer specified a minimum focusing distance of 0.3 meters (note: lower distances seemed to be in focus as well during our calibration). The best position for the top camera will be greatly influenced by how high the root is placed in the chamber.

Next, adjust the position of the bottom camera (connected to Pi09). We moved this camera to the right, thus closer to the root and slightly down to obtain a better view of brace roots (Figure 5). Note that it should be still >0.3 meters from the root. All other camera positions have been adjusted to obtain an equal distribution of cameras in form of an arc (see Figure 5).

2. Camera orientation

To adjust the orientation of cameras it is helpful to place markers inside the scanner box where the root will be located. For example, stretch a white thread from one bottom corner to the opposite bottom corner, and another thread between the other two corners (Figure 5, right image) to determine the horizontal center. In addition, add a vertical marker in the center to determine height and add a white thread from top to bottom in the center of the black back panel (Figure 5, right image). Then, adjust each camera orientation individually using TCam Capture to view the image in real time. Be patient because the viewer lags behind the movement by several seconds. The image orientation should be straight and should not show the white side panels. Each camera should cover as much of the root as possible. See Figure 6 for a comparison of images taken before and after adjustments. Note that camera 9 can capture the whole imaging chamber and the root itself does not always have to be centered in the image. Test multiple roots on each camera to guarantee correct orientation for various sizes of roots.

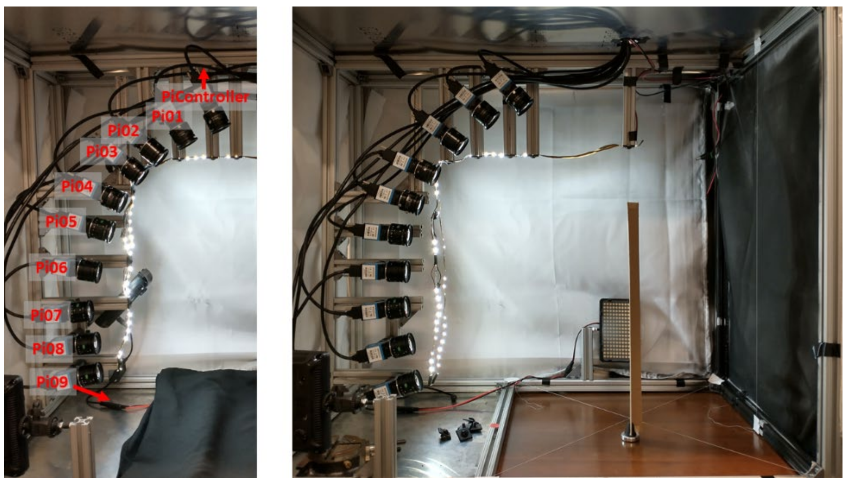

Figure 5: New positions and orientations of cameras

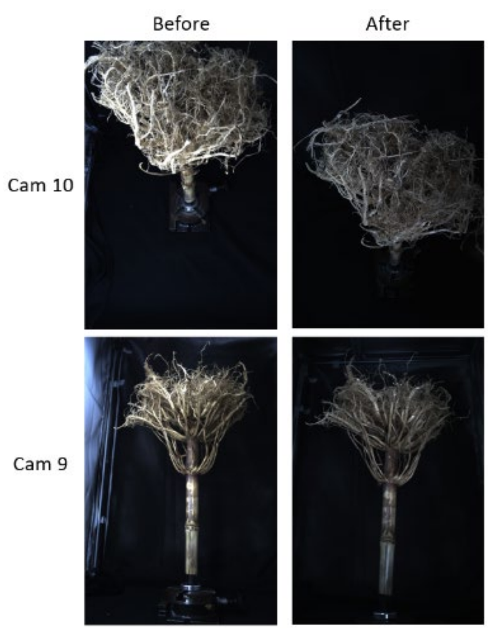

Figure 6: Before and after adjustment of position and orientation

3. Focus

Next, adjust the focus for each camera individually. The focus can be adjusted by turning the corresponding ring on the camera's lenses (loosen the screw before turning the focus ring). This step is the most difficult step and will require the most time.

Print a calibration board (e.g. https://packshotstudio.fr/wp-content/uploads/2018/10/focus21-18.pdf), glue it to a flat surface (like cardboard) and place it inside the imaging chamber such that:

- Its center aligns with the horizontal center of the chamber (Figure 7. Left image step 1))
- The orientation of the board is at a 45 degree angle to the camera (Figure 7. Left image step 2))
- And the height of the board is such that its center aligns with the vertical center in the image (Figure 7. Left image step 3; and right image).

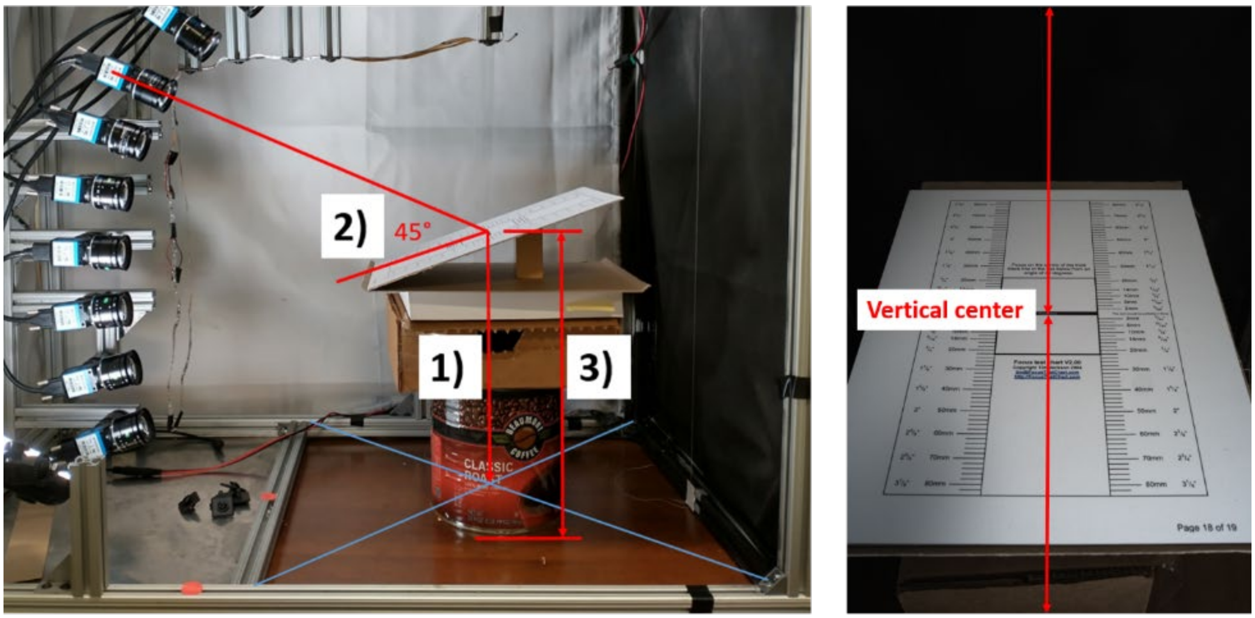

Figure 7: Instructions to place the calibration board correctly into the imaging chamber. Left image shows the three constraints. The blue crossing lines indicate the horizontal center. The right image shows an image taken by the camera and the red lines indicate that the center of the board is in the vertical center of the image.

Set the aperture of the camera lens to the lowest F-number (F/2.8) to create a narrow depth of field by turning the aperture ring of the lens (Note: This is not the same as the focus ring). Take an image using TCam Capture and on the image determine where the camera focuses. Adjust the focus by turning the focus ring accordingly. If the focus is too close to the camera, then increase the focus distance, and decrease it if the focus is too far from the camera. Take another image using the TCam Capture tool to check if the focus is correct and adjust until the focus is on the centerline of the calibration board. This step might take several iterations. Figure 8 shows that the focus is in the center of the board (up to around 20mm on both sides) and that the markers further away from the center are out of focus (e.g. at 80mm).

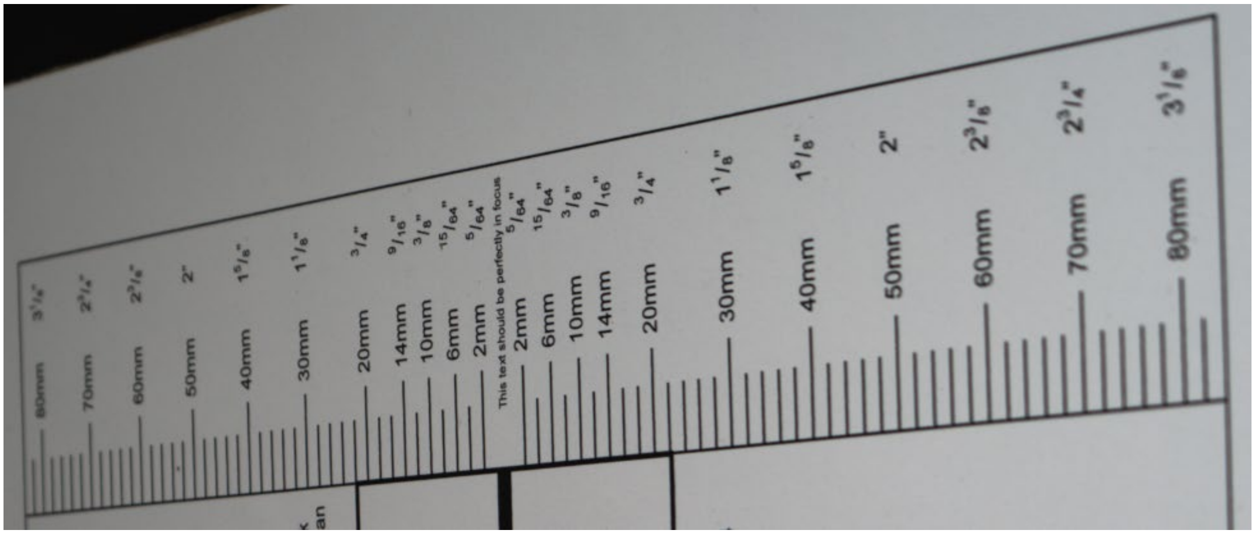

Figure 8: The focal point was found using a calibration board at an aperture of F/2.8. Only markers up to approximately 20mm are sharp.

4. Depth of Field (aperture)

Finally close the aperture by rotating the aperture ring on the camera lens to an F-number that has sufficient depth of field (Figure 9) to keep the whole root in focus. Ideally, this value should be as large as possible (e.g. F/11 or F/16). Note though that a larger F-number darkens the image though. Repeat this for each camera.

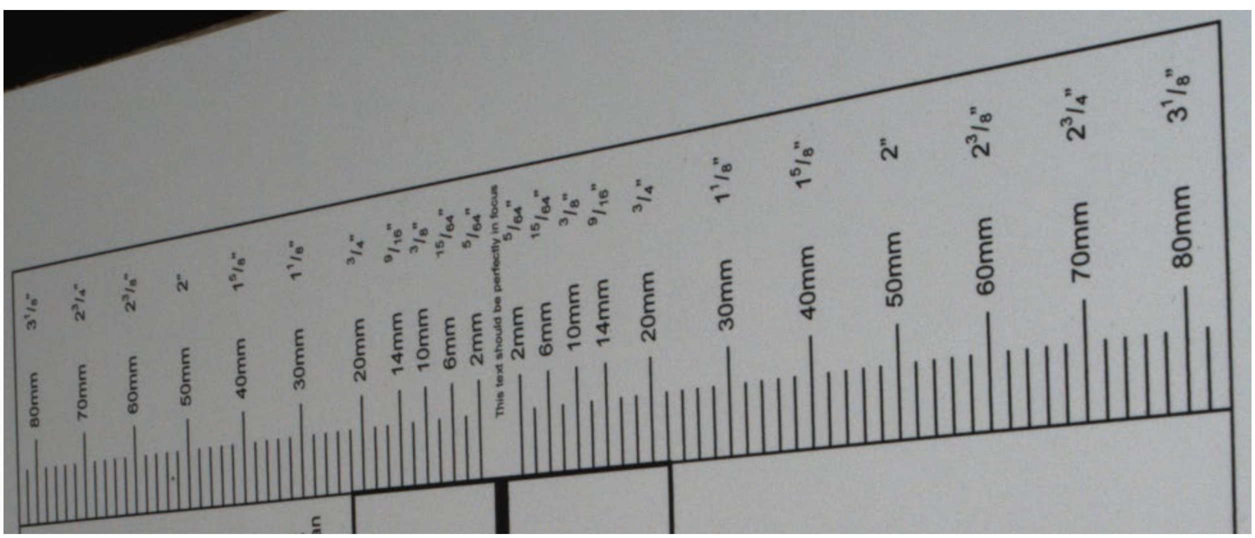

Figure 9: The depth of field was increased by changing the aperture to F/11. All markers on the board are sharp.

5. Illumination

The overall exposure of the image is influenced by the light conditions inside the chamber, the aperture, exposure time and gain. The aperture was already set in the previous step and should not be changed to improve brightness, unless all other options have been exhausted. While opening the aperture would increase image brightness, it will also decrease depth of field, which can result in blurry foreground and background roots.

Unchecking the box for “Exposure Auto” and “Gain Auto” in TCam Capture to turn off the automatic exposure time and gain setting. Set the “Gain” to zero and adjust the exposure time manually to a low value. A short exposure time will prevent camera shake and motion blur. This value will heavily depend on the actual movement of the robot and the scanner itself. If the scanner is static, then 33,333µs (=1/30 seconds) should work fine as a starting point, but a shorter exposure time is preferred.

Now place diffuse lights inside the imaging chamber such that it points mainly from the cameras towards the roots (Figure 10). Diffuse light on the sides might also improve image quality and reduce shadows inside the root. Finally, increase the light until the images are well exposed and add more or stronger lamps if necessary.

Use TCam Capture to save images and inspect them. **Avoid overexposure of any root parts**, as these parts cannot be fixed during 3D model reconstruction**. The ideal image is just bright enough so that no roots are overexposed.** And keep in mind that images with well exposed roots might appear overly dark due to the dark background. Consider only the brightness of the root to determine how much light is needed.

Figure 10: Lamps set up such that the point from the cameras towards the root.

6. Exposure time

As mentioned before, the exposure time should be kept as short as possible to avoid camera shake and motion blur. A useful rule of thumb from photography with handheld cameras to avoid camera shake could be applied here to determine what the maximum exposure time should be. The rule states that the exposure time should not exceed the lens' focal length. In our case the focal length is 12mm (Computar V1228-MPY2 lens) and the maximum exposure should be 1/12 seconds (83,333µs). In practice, this value might have to be much lower (e.g. 1/100 seconds or 10,000µs) to the movement of the root itself, which could result in motion blur.

If a sufficiently short exposure time cannot be achieved, then go back to step 6 and increase the light. Avoid changing the aperture to achieve better image brightness.

7. Gain

Finally, the gain manually. Gain can be used to get a brighter image, but it also increases noise (speckle) in the image (see Figure 11). Therefore, try to keep the gain as low as possible. Small amounts of gain might not add much speckle though. If sufficient brightness cannot be achieved with low gain, then add more light source or increase exposure time.

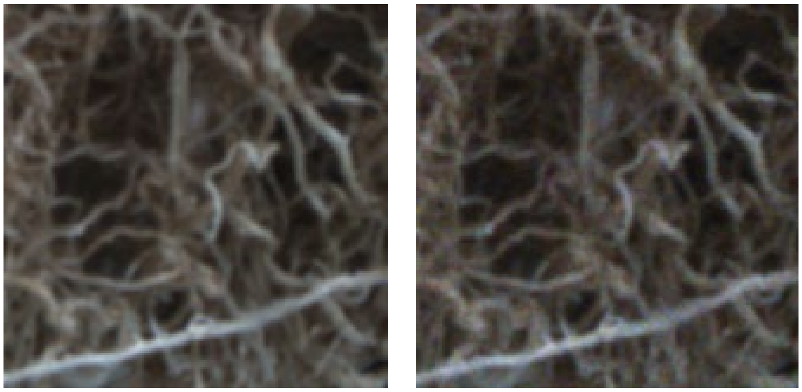

Figure 11: Low gain (left) versus high gain (right). The high gain image has more speckle.

There is an obvious tradeoff between illumination, exposure time and gain. It is ideal to add more light and to keep exposure time and gain low. Iterate through steps 5) to 7) until you reached a good solution.

TCam Capture settings and Python Code

It might be useful to reset the settings in TCam Capture if any other changes have been made. This will set all parameters to the default values. Then make sure that “Exposure Auto” and “Gain Auto” are off and set “Exposure Time (us)” and “Gain” to the desired values (see Figure 2 for an example).

It is unclear however whether these settings will be used if the scanning process starts. We had the impression that it automatically switches back to the default values when the scanning process starts from the python script. Therefore, it is necessary to specify these in the file *snap_image_buffer.py* (see Figure 12 for an example). A list of all parameters is given in section TIS Camera Properties at the end of this manual.

```
    # setup camera paramters
    Tis.Set_Property("Exposure Auto", False)
    Tis.Set_Property("Exposure Time (us)", 66666) # 1/15 sec
    Tis.Set_Property("Gain Auto", False)
    Tis.Set_Property("Gain", 200)
```

Figure 12: Python code to set parameter in snap_image_buffer.py

Example Settings

During our calibration we came up with the settings shown in Table 2. Because we could not add more lights to the chamber nor increase their power, we opened the aperture to F/11, increased exposure time to 66666µs and gain to 200. It would be ideal to add more or stronger lamps inside the chamber, such that an aperture of F/16, a shorter exposure time and lower gain can be used.

Table 2: Camera settings after calibration

Camera	Focal Distance	Aperture	Exposure time (us)	Gain
10 (ControllerPi)	0.28	F/11	66666	200
1 (Pi01)	0.3	F/11	66666	200
2 (Pi02)	0.3	F/11	66666	200
3 (Pi03)	0.35	F/11	66666	200
4 (Pi04)	0.43	F/11	66666	200
5 (Pi05)	0.4	F/11	66666	200
6 (Pi06)	0.35	F/11	66666	200
7 (Pi07)	0.38	F/11	66666	200
8 (Pi08)	0.38	F/11	66666	200
9 (Pi09)	0.4	F/11	66666	200

Pre- and Post-calibration comparison

Table 3 shows the increase in number of points from 3D models before calibration to 3D models after calibration for 12 different genotypes. In nearly all models the number of points increased by a factor of two.

Table 3: Comparison of point number before and after calibration

Comparison	B101	B112	DKIB014	DKPB80 X3IIH6	H96 X3IIH6	LH59 XPHG29	LH123HT	PA762	PA762 X3IIH6	PHG50 XPHG47	PHZ51	PHZ51 XLH59
Before calibration	1,928,853	1,986,030	1,592,580	2,068,432	1,681,955	1,192,388	687,004	261,392	759,195	490,841	501,798	652,584
After calibration	4,533,962	5,600,208	4,155,521	3,961,875	4,841,340	3,283,718	2,238,260	760,259	1,936,404	1,745,432	1,171,799	2,173,042
Increase of number of points	**2.3506**	**2.8198**	**2.6093**	**1.9154**	**2.8784**	**2.7539**	**3.258**	**2.9085**	**2.5506**	**3.556**	**2.3352**	**3.3299**

TIS Camera Properties

List of all camera properties that can be specified with *Tis.Set_Properties(PropertyName, value)* in Pythons scrip *snap_image_buffer.py*.

Brightness
Saturation
Hue
Whitebalance Auto
Whitebalance Red
Whitebalance Blue
Gamma
Gain
Exposure Time (us)
Exposure Auto
Exposure Auto Reference
Gain Auto
Trigger Mode
Software Trigger
Trigger Delay (us)
Strobe Enable
Strobe Polarity
Strobe Exposure
Strobe Duration
Strobe Delay
GPOut
GPIn
Offset X
Offset Y
Offset Auto Center
Trigger Polarity
Trigger Operation
Trigger Burst Count
Trigger Debounce Time (us)
Trigger Mask Time (us)
Trigger Noise Surpression Time (us)
Auto Functions ROI Control
Auto Functions ROI Left
Auto Functions ROI Top
Auto Functions ROI Width
Auto Functions ROI Height
Whitebalance
Whitebalance Auto Preset
Whitebalance Green
White Balance Temperature Preset
Whitebalance Temperature
Reverse Y
Highlight Reduction
Exposure Auto Upper Limit Auto
Exposure Auto Lower Limit
Exposure Auto Upper Limit
Override Scanning Mode
Auto Functions ROI Preset
Gain Auto Lower Limit
Gain Auto Upper Limit
Trigger Global Reset Release

Imaging Chamber Operation Procedure

Introduction

Pi cluster was built using raspberry pi 4 model b+ with USB 3.0 interface, matching with camera connection. All pi units are labelled as the name of this pi unit, such as "Pi01, Pi02…Pi09", and "Controller pi".

The detailed steps of building the pi cluster can be referenced https://github.com/Computational-Plant-Science/raspberry_pi_cluster

Operation

1. Turn on the power of surge protector, which connects to all the power of scanner.
2. Check the display directly connects to "pi controller", Make sure wireless mouse/keyboard was turned on.
3. Open command window from "pi controller" terminal, type command lines:
    a. cd code/cam   #enter working path
    b. python cluster_connection.py   # check all the raspberry pi unit's online status
    c. python cluster_capture_syn.py -s 1 -n 360   # scan root, image data will be stored in the folder named as current year-date format such as /2021-06-16/
    d. The main script was snap_image_buffer.py, it is located at the same path in each pi. cluster_capture_syn.py was to run each script in a parallel way.
    e. If error or some camera was not synchronized due to camera warm up delay. Go to step 6 delete files.
    f. Rotate the camera arm to its original position and restart the scan process.

   g. If the camera arm reaches 360 degrees, python stepper.py will rotate the arm back to original position.
4. Transfer files from raspberry pi units to “pi controller”.
   a. python camera_img_transfer.py -p /home/pi/code/cam/2021-06-16/   -a 1
5. Transfer files to your local computer.
   a. Connect to your local computer to the same network with scanner via WIFI.
      Name: RootRobot5G   PW: SmartAg214
   b. Router PW: Simon123321
   c. Install “FileZilla Client” from https://filezilla-project.org/,
   d. Login in to the “pi controller”, Host: 192.168.1.110, Username: pi, Password: raspberry, Port:22
   e. and navigate to the folder containing the image files, in the format as “Year-Date”, such as “/2021-06-16/”
   f. Download the folder /2021-06-16/ from “pi controller” to your local computer destination folder.
6. Delete files from “pi controller” and raspberry pi units.
   a. python camera_img_transfer.py -p /home/pi/code/cam/2021-06-16/ -a 2   # delete captured images in all raspberry pi units named as /2021-06-16/
7. Shutdown system
   a. python cluster_shutdown.py
8. Turn off the power of surge protector

Notice:

Please check the cable connection between cameras to the raspberry pi, part of the cables is integrated inside the box, unstable connections happened. Please make sure the cables are in the format of USB3.0 and functions stable. In our test, we skipped the cables is integrated inside the box, we use USB3.0 cables directly connecting from cameras to each pi units.