

Macrophotography of Fern Gametophytes Using a Focus Stacking System



Jennifer I. Deegan



Tim Deegan

Department of Plant Sciences,
University of Cambridge, Downing Street, Cambridge, CB2 3EA.
e-mail: jic24@cam.ac.uk

This follows on from a previous article on focus stacking of fern gametophytes by the same author. It can be found at **Deegan, J., (2017)** 'Photographing the Fern Gametophyte' *Pteridologist*, **6(4)**: 263-265. The principles of this new system are the same as before, but several components have changed significantly. This article describes the modifications made, relative to the previous system.

The major components of the system are the rail, the rail driver, the optics and the lighting. The sections that follow will describe these new components.

The Rail

The previous rail was a modified flatbed scanner, and this has been replaced with a home-made system based on an old microscope. The bulk of the rail was made from the focusing block of a Nikon Optiphot or Labophot microscope. This was cut out and bolted horizontally onto a piece of scavenged worktop (Fig. 1). On top of the focus block, a flat sheet of metal was screwed down to act as a platform for the camera. The camera was placed on top, with the lens supported in a horizontal position using a piece of pencil eraser.



Fig. 1. The rail based on an old microscope.

The Rail Driver

The rail could be driven either by a Raspberry Pi B computer (Fig. 2) or an Arduino computer (Fig. 3), both of which were in turn controlled by a laptop computer. I am including

descriptions of both as they were quite complicated to set up and I don't know which system will continue to be available for longer, if someone should want to replicate this work. The purpose of the Raspberry Pi or Arduino was to drive a stepper motor, which then drove the fine focus knob of the microscope block via some plastic gears (Fig. 4).



Fig. 2. The Raspberry Pi B computer.

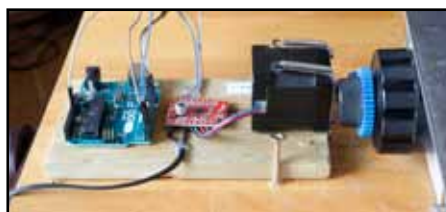


Fig. 3. The Arduino computer.



Fig. 4. Plastic gears driving the fine focus knob.

The stepper motor we used was an RS Pro Hybrid Stepper Motor 0.9°, 44 Ncm, 2.8 V, 1.68 A, 4-wire motor. The gear wheels were bought from <http://www.maplin.co.uk/> and then the centres were drilled out or cut out with a hacksaw blade. The plastic gears were very soft and easy to cut. They were then just pushed onto the focus knobs of the focus block.

We also used a "Big Easy Driver" Stepper motor control board (Fig. 5), which is fully documented at: <http://www.schmalzhaus.com/BigEasyDriver/>.

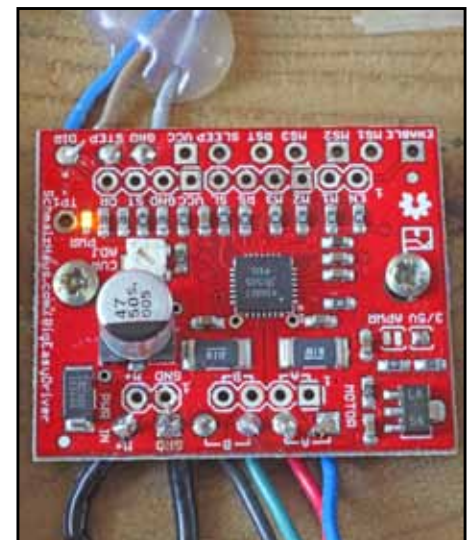


Fig. 5. The 'Big Easy Driver' motor control board

The board brought a number of specific advantages over just using an H-bridge and breadboard to control the stepper motor. The motor was able to make a single complete turn of its spindle in 400 steps, but with the driver board this could be changed to 6400 steps. The current reaching the motor could also be controlled so that the motor did not burn out. A potentiometer on the driver board had to be adjusted with a screwdriver in order to complete the setup. This was done by practical experiment until the movement was correct. The process is described on the website mentioned above.

Macrophotography of Fern Gametophytes Using a Focus Stacking System

The Big Easy Driver wiring was the same in both cases and the wires from this driver are seen on the left end of the set of GPIO pins of the Raspberry Pi in Fig. 6.



Fig. 6. Showing the wires on the Pi.

The wires on the right side of this image are those that drive the infra-red remote control for the camera, which is discussed later. The code for the Raspberry Pi is shown in the Fig. 7.

The wiring for the Arduino is shown in Figures 8 and 9, with the driver wires on the right and centre front and the infra red remote control wires shown on the left at the rear. The code for the Arduino is shown in Figure 10.



Fig. 8. The wiring for the Arduino showing the driver wires right and centre.



Fig. 9. The wiring for the Arduino showing the infra red wiring at the left.

The Arduino was a simpler system to setup and the Raspberry Pi was trickier but offered more sophisticated options for extension, as it is a more powerful computer.

```
#!/usr/bin/env python
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
#This assigns the pins of the raspberry pie
to variables called "pin".
pin0 = 18
pin1 = 23
pin2 = 24
pin3 = 17
#Set these pins to send messages out
rather than listen for messages coming in.
GPIO.setup(pin0, GPIO.OUT)
GPIO.setup(pin1, GPIO.OUT)
GPIO.setup(pin2, GPIO.OUT)
GPIO.setup(pin3, GPIO.OUT)
#Zero all of the pins between tests.
def zero_pins():
    time.sleep(0.1)
    GPIO.output(pin0, 0)
    GPIO.output(pin1, 0)
    GPIO.output(pin2, 0)
    GPIO.output(pin3, 0)
#This runs the test.
def test():
    count = 1
    while (count < 10000):
        print 'The count is:', count
        zero_pins()
        GPIO.output(pin0, 1)
        GPIO.output(pin1, 1)
        zero_pins()
        GPIO.output(pin1, 1)
        GPIO.output(pin2, 1)
        zero_pins()
        GPIO.output(pin2, 1)
        GPIO.output(pin3, 1)
        zero_pins()
        GPIO.output(pin3, 1)
        GPIO.output(pin0, 1)
        zero_pins()
        count = count + 1
    test()
```

Fig. 7. The code for the Raspberry Pi.

```
//Declare pin functions on Arduino
#define stp 2
#define dir 3
#define fire 13
void setup() {
    pinMode(stp, OUTPUT);
    pinMode(dir, OUTPUT);
    pinMode(fire, OUTPUT);
    digitalWrite(stp, LOW);
    digitalWrite(dir, HIGH);
    Serial.begin(9600); //Open Serial connection for
    debugging
}
//Main loop
void loop() {
    long int i, j;
    char input;
    Serial.println("Press Enter to start");
    do {
        input = Serial.read();
    } while (input != '\n');
    for (i = 0; i < shots; i++) {
        trigger();
        for (j = 0; j < distance; j++) {
            step();
        }
    }
    // Power on the IR trigger circuit to fire the camera
    void trigger()
    {
        delay(1000); // wait for vibration to settle
        digitalWrite(fire, HIGH); // take the picture
        delay(1000); // wait for that to happen
        digitalWrite(fire, LOW); // disable the IR circuit
        again
    }
    // Advance the motor by one step
    void step()
    {
        digitalWrite(stp, HIGH); //Trigger one step forward
        delayMicroseconds(25);
        digitalWrite(stp, LOW); //Pull step pin low so it can
        be triggered again
        delayMicroseconds(25);
    }
}
```

Fig. 10. The code for the Arduino.

Wiring chart for Big Easy Driver Wires

Big Easy Driver connection	Destination
GND (top left)	GND
STEP (top left)	2 Arduino, 24 Raspberry Pi
DIR (top left)	3 Arduino, 26 Raspberry Pi
M+ (bottom left)	Power cable for stepper motor
GND (bottom left)	Power cable for stepper motor
A	"A" wire of the stepper motor
A	"A" wire of the stepper motor
B	"B" wire of the stepper motor
B	"B" wire of the stepper motor

Infra Red remote control wiring

Infra Red remote control	Raspberry Pi	Arduino
Non-bulb end	GND	GND
Bulb-end	19	13

Stepper Motor Wiring

The stepper motor is a 4 wire stepper motor with two circuits. In order to wire it up correctly we checked the wires in pairs with an ohmmeter to see which pairs were connected together. We designated these, pair "A" and pair "B", which were then wired up as shown in the Big Easy Driver wiring table above.

Automatic Control of the Camera Shutter

The Raspberry Pi or Arduino computers were also used to control the camera shutter, so that it was easy to alternate the processes of the camera being winched toward the subject and the shutter firing. One additional piece of equipment was required: the infra-red remote control that would communicate directly with the camera (green circuit board in Fig. 1). The infra red remote control was available commercially and we stripped it down to the bare circuit board and wired it directly to the Raspberry Pi or Arduino computer. The wiring is shown in Fig. 11, and in this case the purple wire connects back to the black wire shown in the Raspberry Pi and Arduino wiring images.



Fig. 11. This stripped down circuit board is the infra red remote.

Macrophotography of Fern Gametophytes Using a Focus Stacking System

The Optics

The optics in this system were also completely different from the previous model in order to achieve higher resolution. In this case we used a Mitutoyo 10x M Plan APO Long Working Distance microscope objective, as recommended by forum members at the photomicrography.net website. This objective has a working distance of 33.5mm, which gives a lot of space for lighting the subject. It also avoids optical problems with image processing that might result from camera's movement relative to the subject.

The objective was mounted on an Olympus Zuiko 200mm manual focus prime lens. The adapter used was a M26X0.706 Mitutoyo microscope female thread to M55mm 55X0.75 adapter. The thread of the objective on its own was too long to screw fully into the adapter without hitting the glass of the prime lens. Without further adjustment, the objective was suspended at an angle, spoiling the focus entirely. We solved this problem by using two 55mm UV filters with the glass removed and inserting them as extenders between the prime lens and the objective adapter. This enabled the objective to be fully screwed in and the focus to be really sharp, as required. The last piece of optical equipment required was the Olympus OM Lens to Canon EF-Mount which was used to fix the prime lens onto the Canon 5d MkII body. The full setup is shown in Fig. 1.

The lighting

In the previous setup we had problems with specular highlights, which combined poorly when the slices (photos) were stacked to make a single fully focussed image. In order to resolve this problem we used a polystyrene cup as a light diffuser (Fig. 13), also recommended by forum members at the photomicrography.net website.



Fig. 13. The polystyrene cup used as a light diffuser.

As a light source, illuminating the outside of the cup, we used two Yongnuo YN-560 IV Flash Speedlite flashguns. These were controlled by a Yongnuo YN560-TX remote controller mounted on the hotshoe of the camera. The flashes allowed us to take short exposures of 1/60th second, freezing any vibration in the system that might have affected image quality (Fig. 14).



Fig. 14. The two flash guns.

We had great problems with the fern cells collapsing under the influence of the flash guns and substituting the flashguns with LED lamps did not help. We solved this problem by very careful adjustment of the number of photos taken to the absolute minimum and by speeding up the movement of the rail.

The first step of reducing the number of photos taken was made by carefully measuring the depth of field of the images. We made this measurement by photographing a stage micrometer, with 100 micron steps, mounted almost parallel relative to the optical axis of the lens (Fig. 15).

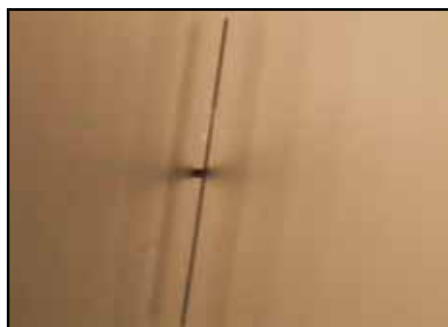


Fig. 15. Measuring the depth of field.

We found that the depth of field of this optical setup was about 20 microns and so we set the movement of the rail to 17 microns between photographs, which proved effective.

We speeded up the movement of the rail by making adjustments to the code run on the Arduino. (This modification has not yet been extended to the Raspberry Pi code, so any readers reusing that code will

need to adjust it accordingly).

The result of these two modifications was that we took very few photographs, very quickly, preventing harm to the fern. A common number of photographs for a young gametophyte would be about 40 photographs in just a few minutes.

The final setup is shown in Figure 16.



Fig. 16. The final setup.

Growing Ferns for Photography

Our photographic subjects were grown on sterilised peat-based Jack's Magic compost, in plastic pots, enclosed in a ziplock freezer bag. The sterilisation step was carried out by pouring boiling water through the compost, covering with clingfilm while the water drained, and then cooling the pots in the fridge. When cool, the pots were removed to room temperature, and sown thinly with fern spores. They were then enclosed in a freezer bag and put to grow in a suitable place. We found that temperature and light levels were critical. The spores grew best at room temperature (about 21 degrees C) and away from direct sunlight. We put them to one side of a window, rather than directly on the windowsill, and germination took a few weeks.

Operating the System

There were a number of steps to be followed when operating the system. The camera was connected to the laptop and controlled from the EOS Remote software, so that the image could be observed on the screen, and the fern carefully positioned on a shaped piece of blutac. The Arduino or Raspberry Pi was also connected to the laptop and controlled from there. The rail had to be winched back at the beginning of each run, so as to avoid running out of travel mid-shot. It was also important to pull the metal sheet back at the beginning of each run to fully engage the gears. The focus block is designed to be used vertically

Macrophotography of Fern Gametophytes Using a Focus Stacking System

under the force of gravity and has a large amount of play in the system before the gears fully engage.

The exposure was set by taking test shots to look at on the laptop screen and finally the Arduino or Raspberry Pi was triggered to drive the system. The set of images captured was then run through Helicon Focus on the standard settings as before.



Fig. 17. Gametophyte at an early stage of development.



Fig. 18. Gametophyte at a later stage of development.



Fig. 19. Gametophyte as archegonia begin to form.

Licensing

Full details of the system are provided at the following URL, in line with the conditions of the Biomaker grant. https://github.com/BioMakers/23_Focus-stacking-system-for-gametophyte-ferns.

If anyone would like to copy the system then they are welcome to do so, but full attribution is requested, to be made by citing this article.

Images

At the time of writing, the system has just been completed and we have taken some successful photographs of gametophyte *Asplenium scolopendrium* (hart's tongue) ferns. Figures 17, 18 and 19 show three different stages of gametophyte development.

Future Work

We hope to continue this work, with the aim of producing a more complete developmental series of photographs. Our hope would be to produce images showing fern development from the filamentous stage of growth to the emergence of the first sporophyte leaf. This may take some time as we will need to obtain microscope objectives of very different magnifications in order to cover such a large range of sizes of subject.

Acknowledgements:

Financial support was provided by the Biomaker Challenge 2017 which was jointly funded by OpenPlant, a BBSRC/EPSRC Synthetic Biology Research Centre Grant BB/L014130/1 and the Isaac Newton Trust. Biomaker Challenge 2017 was coordinated by University of Cambridge's Synthetic Biology Strategic Research Initiative and the CambridgeSens Network.

Jim Haseloff, Malcolm B. Wilkins and Adrian Dyer gave invaluable encouragement and advice. Richard Mortier and Matthew Couchman assisted with the grant proposal.

The members of photomacrography.net gave detailed and constant guidance in setting up the system. We especially thank ChrisR, Peter M. Macdonald and Doc.AI. Chris Slaybaugh built the "Bratcam" (Fig. 20) on which our system is based.



Fig. 20. The 'Bratcam' on which our system is based.

Lastly we would like to thank Ben Deegan (age 7) for his patience, and for always fitting the camera lenses so very carefully.