Follow-focus with Arduino

Christian Guerrero
Ludwig Maximilian Universität, München, Munich, Germany
christian.guerrero@campus.lmu.de

Arthur Moufounda
École Supérieure Des Technologies Industrielles Avancées, Bidart, France
a.moufounda@net.estia.fr

Alexander Schenker
Ludwig Maximilian Universität, München, Munich, Germany
alexander.schenker@campus.lmu.de

ABSTRACT
This paper covers the development of a novel device for focus pulling in movie production. It describes the field of application and gives an overview of established commercial and non-commercial solutions, before stating current research and deducing requirements for a new development. It further documents the challenges and solutions, that came up during the development process, before attempting a future prospect.

Author Keywords
Follow Focus; Movie Production; Remote Control; Film Set Automation; Arduino; Android;

MOTIVATION AND BACKGROUND

General Motivation

General Description of the Topic
In movie production, especially the part of *principal photography* is considered one of the most expensive ones. This is, among other reasons, due to the often high number of scene takes, that have to be shot. [10] Usually, manual control is required in order to zoom and to keep the focus on the desired object. In classical productions, the *focus puller* (or *first Assistant Camera*) is a dedicated position, whose job it is to keep the focus according to the cinematographers instructions. As there is no way to correct a falsely set focus in post-production, it is considered one of the most difficult jobs on a film set. [13]

State of the Art in Follow Focus
This section gives a brief overview of a selection of currently existing technologies in the field of follow focus.

Andra Motion Focus is a commercial hardware system developed by Andra Motion Technologies Inc. It combines hardware-supported motion capturing with a remote controlled follow focus system. It is currently controllable optionally with an iPad or a dedicated hardware interface, called *The Arc*. Andra Motion Focus is designed for portability and efficiency, in order to reduce the amount of time needed and hence the shots' cost. Due to the high price, it aims for professional and semi-professional film-makers. [8]

The FF-X1 by PROAIM Camera is exemplary for many similar products on the market. It is a purely manual gear-driven mechanism, that is supposed to "ensure smooth accurate emphasis over your subject" [5]. Usage is supported for a wide range of cameras.

Lenzhound is "A super-affordable wireless follow focus system for indie filmmakers." [4] It was originally featured on kickstarter by a company called Motion Dogs. The key idea of Lenzhound is, to have a simplistic, hardware- and remote-controlled Focus System. It is belt-driven and can be attached to almost any type of camera lens. Unlike Andra Motion Focus, Lenzhound addresses both amateur and independent film-makers.

"Follow Focus" by Adi Soffer is an open source and Do-it-yourself (DIY) follow focus project from 2013. It is based on Arduino hardware and standard electronical components, which makes it relatively easy to reproduce. The source code can be found on GitHub. [12]

Background

Vision
Considering the difficulties resulting from the precision required while shooting a film, and the often high number of takes, it is worth thinking about a holistic automated system. One that offers not only automated dolly shots, or remote controlled focus pulling, but one that includes everything from camera movement to image composition. As mentioned in the State of the Art section, there is a wide range of systems available. However, they are often not affordable for independent film-makers, due to their high costs. Also, for most solutions, the benefit is limited to one single use-case, such as focus pulling.

Consequently, our vision is to create a DIY system, which combines features from several existing solutions.

Current Research at Ludwig-Maximilian-Universität (LMU)
Previous and current research conducted by LMU in Munich, Germany, has come up with theoretical research on automated camera shots [9] and the development of a remote controlled camera slider [11].

This paper covers the prototypical development of a remote controlled focus pulling device with the long-term objective to get integrated into a system, which combines automated dolly shots, focus pulling and camera tilts.

SOLUTION

Our Follow Focus Approach

Our approach is based on the ideas of *DIY Follow Focus* [6] and *Soffer Follow Focus*, combined with an app-based remote control interface.

Having analyzed the existing solutions from the *State of the Art* section and current research, we came up with a list of requirements, that we impose on our prototype:

- The protypical device should be adaptable to a wide range of camera types.
- Focus pulling should be done in realtime.
- The speed of focus change should be adjustable.
- The lense's range should be calibratable.
- Focus sequences should be recordable and therefore easily repeatable.
- The system should be controllable through an Android-based app interface.
- The communication between hardware and software should happen via Bluetooth Low Energy (BLE).

The adaptation to different camera types happens through an adjustable gear ring.

BLE is a low-cost wireless technology, which allows data transmission with a comparatively low energy consumption. According to the official bluetooth developer portal, it has the "ability to run for years on standard coin-cell batteries". [2]

Methods

To realize our aims we decided to develop an Android App for input. It sends a perpetual signal via BLE to a BLE module connected to an Arduino Nano. The Arduino processes the byte information and sends commands to a stepper driver module. The driver then controls the movement of a stepper motor.

Electrical Engineering

Most importantly we wanted our follow focus to be portable. Therefore the elements had to be small, low energy powered and easily embeddable. These challenges had to be tackled in the planning phase of the process. So at first we had to decide between different technologies in order to meet our aims:

- For our requirements an Arduino Nano is more than sufficient. It is small and consumes only little.
- For the Wireless control we use BLE for the reasons mentioned above and for compatibility reasons with other LMU projects, which use this technology as well.
- A stepper motor works more precise and faster than a servo and has only small enertia [7]. This way we also always know the exact amount of rotation we have to apply.
- In order to link the motor to the system, we use a stepper driver. It will protect the Arduino Nano from electrical damages that may be caused by the sudden increase of intensity by the motor. It also allows to turn in both directions.
- The AccelStepper library for Arduino prevents the API from blocking and supports very slow speeds. Furthermore it helps to easily apply acceleration.
- None of the electronic parts is passive. To make them all work, we need electric energy. As we also want our global system to be portable, we will use different sets of battery, because the Arduino and the step motor need to be powered separately. At least 5V and 12V respectively.

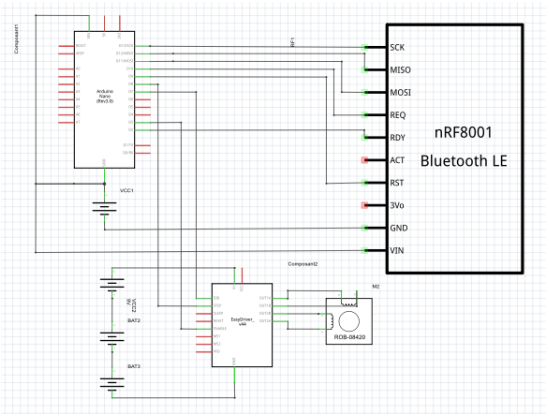

Figure 1. Circuit Diagram of the Electrical Engineering in our Follow Focus

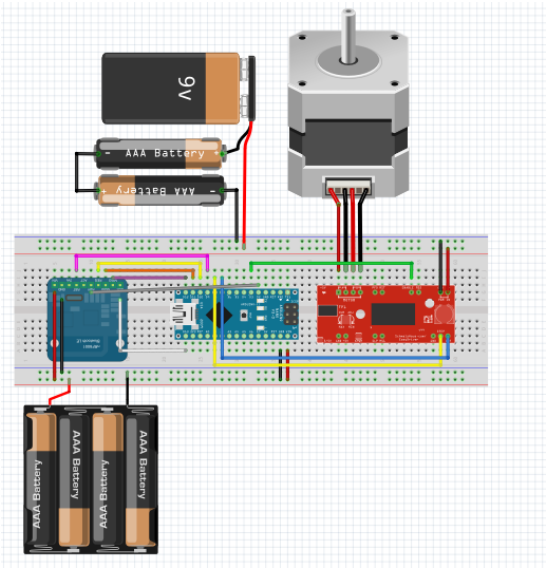

Figure 2. Assembly Diagram of the Follow Focus System

In Figures 1 and 2 you can see how all components are merged together into one system.

Listing 1 shows how the received BLE Serial data is being processed on Arduino side. The main loop always checks for connection and if so then at first we perform run() on the stepper. This will result in a step to the next position if there is one due.
There are basically two possibilities for our control. Either

we use a fixed maximum speed and only change the step size or we vary the speed and use always the same step size. Our experiments showed that the second method performs more fluid and exact. The react() function works as follows:
the Arduino listens to incoming values *0*, *43*, *45*, *49* to *57* which represent the ASCII values of *0*, +, -, respectively the digits *1-9*. A + adds one step to the stepper's position and - subtracts. *0* stops our stepper immediately. This is necessary as there might be some steps due, because of congestion in the output processing. With the digits *1* to *9* we control nine different speeds.
After react() we call run(). The due steps are getting performed and the new target position, which will be executed in the next loop, is set.

Listing 1. Processing of received BLE serial data

```
if (status == ACI_EVT_CONNECTED)
  {
    stepper.enableOutputs();
    stepper.run();
    if (BLEserial.available()>0)
    {
      dataReceive = BLEserial.read();
      react(); //function to react to input
      stepper.run();
      stepper.moveTo(encoderValue);
    }
        ...

void react()
{
  if (dataReceive > 48 && dataReceive < 58)
//[1-9]-Key
  {
    speedValue = map(dataReceive, 49, 57,
        minSpeedValue, maxSpeedValue);
    stepper.setMaxSpeed(speedValue);
  }
  else if (dataReceive == 43) //[+]-Key
  {
    encoderValue += stepSize;
  }
  else if (dataReceive == 45)  //[-]-Key
  {
    encoderValue -= stepSize;
  }
  else if (dataReceive == 0) {
    encoderValue = stepper.currentPosition();
  }
  ...
```

Case Modeling
Challenges & Solution

For building our case (see Figure 3) we faced two crucial questions: How can we achieve a small overall size to maintain portability? And which technology can we use to create a case, which is both stable and lightweight? We quickly realized, that premade plastic cases for electronics would be too inflexible in terms of spatial efficiency, while the quality and stability was excellent. Our first approach for a self-made case was using 3D-printing technology. First, we designed the parts with *CATIA*, a 3D modeling software by *Dassault Systemes* [3]. However, the printing took a very long time and the outcome was not satisfying. Even though we expected a maximum of freedom in terms of design possibilities, the

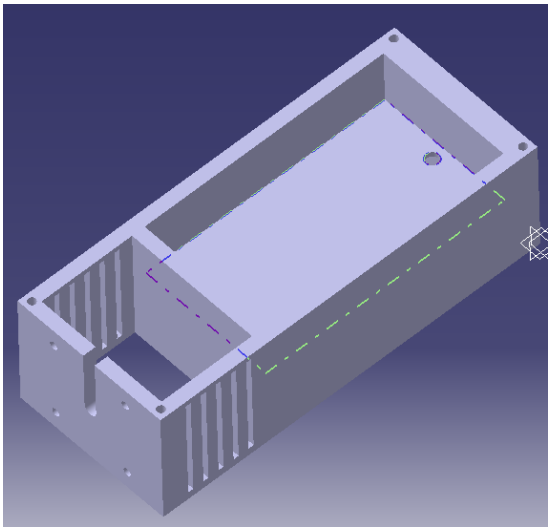

Figure 3. 3D Model of our Prototype Case

printed parts of the case turned out to be very uneven and unstable. Our third, and final option was the use of a laser cutter. We were able to re-utilize parts of the 3D model, which we created for the 3D-printing approach. The 3mm wooden material turned out to be ideal for our case, as it offers the needed stability, while being lightweight and easy to process.

The finished case is composed of three compartments. The first one accommodates the electronic parts. A second one lies underneath and is designed exclusively for the batteries and a power button. The back of the case has two notches, which allow to open the backside easily when a battery exchange is due. In the front part of the case, there is space reserved for the stepper motor, which can be fixated with screws. The slots on the sides around the stepper motor are supposed to evacuate the heat produced by the motor when it is at work. Two holes drilled into the partition walls allow wires to connect the Arduino, stepper and battery compartments.

App Development
Challenges

For the development of the remote controller app, we faced various challenges. As we defined the use of BLE as one of our requirements, only a part of the available Android devices is capable of executing our application, namely those with Android 4.3 or higher and with the required hardware module built in. Currently (as of August 3, 2015), 62.1 percent of all Android devices meet the requirements software-sided, while the share of devices, which provide BLE hardware is unknown. [1] However, we do not consider this being a problem, as the advantages of BLE technology outweigh the disadvantage (compare *Bluetooth Low Energy* in *Our Follow Focus Approach*).

The other challenge was, how to implement the required functionalities. Here, it was important to keep the extensibility in mind. As the main goal is, to eventually embed the app and the hardware into to a larger system, it must be developed in

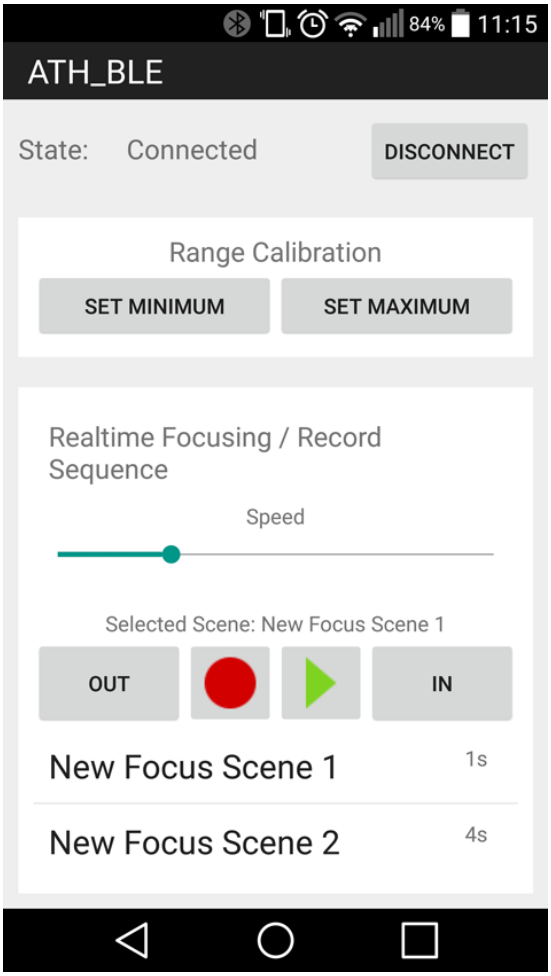

Figure 4. Prototypical User Interface

a way, that it can be easily extended on the one hand, and that the logic can be transferred into another app on the other hand (compare *Our Vision*). One of the key challenges, was the implementation of the focus sequence recording in a way, that sequences are exactly reproducible.

Last, but not least, the User Interface (UI) should be created in a way, that makes the app easy to use, especially when it offers multiple different functionalities. As the design of a UI is not directly part of our requirements, we neglect this topic in this paper (see Figure 4 for the prototypical UI).

Our Solution

As our solution for the controller, we developed an Android App for smartphones. Through the provided UI, the camera operator is able to connect to the Follow Focus device via BLE. Once connected, the lens range of the attached camera can be calibrated in order to prevent the stepper motor from damaging the lens focus thread. The underlying logic for focus movement is relatively simple. The app manages two *byte*-values for both speed and direction of the movement. The speed value can be regulated via a slider element (*SeekBar* in Android SDK) with a range from 0 to 100. The direction can be either right (IN), left (OUT) or neutral (no movement), and can be regulated via two buttons. The default direction is neutral, and direction changes only, while one of the two buttons is pressed.

To implement the focus sequence recording function, we had to think of a way, of how the user input could be saved including the temporal aspect. Therefore, the important information is not only *what* the user does, but also *when*. We basically came up with two approaches to this problem:

- Whenever a user input occurs, send the data and save the input type and value (e.g. speed 99), as well as the time delta from the beginning of the recording.
- Constantly check, save and send current speed and direction every x milliseconds.

As one of the key aspects of recording a sequence is reliable repeatability, we figured that the second option would offer the better results (in terms of result stability). Due to expected latency on both the master's side (i.e. the mobile android device) and the slave (the follow focus device), we expect the first option to lead to irregular playback result.

The sampling approach of the second option offers regular samples, which can be iterated through in the defined interval in order to exactly repeat the original input.

Listing 2. Focus Sequence Sampling

```
/*
* Timer Thread
* - executed every <EXECUTION_INTERVAL> ms
* - sends currentDirection and currentSpeed
*   via BLE to Arduino device
* */
public void startTimer (){
  final Handler handler = new Handler ();
  Timer timer = new Timer();
  timer.scheduleAtFixedRate(new TimerTask() {
    public void run() {
      handler.post(new Runnable() {
        public void run() {

// if not playing a recorded scene:
          if (!isPlayingScene) {

// send data via BLE

            if (isRecording) {
// if scene is currently recording,
// write values into FocusScene object
            }
          } else {
// if recorded scene is being
// played back, send recorded
// values as byte values.
// convert slide value (Integer)
// to byte values and send via BLE

// stop playing, when max frames reached:
            if (currentSceneFrame <
                selectedFocusScene
                .getSpeedValues().size()-1) {
              currentSceneFrame++;
            } else {
              currentSceneFrame = 0;
              isPlayingScene = false;
            }
```

Figure 5. Assembled Prototype

```
            }
          }
        });
      }
    }, 0, EXECUTION_INTERVAL);
}
```

As can be seen in Listing 2, a timer thread in the controller app executes a function every *EXECUTION_INTERVAL* miliseconds. We defined this as an integer constant with the value of 50, which turned out to deliver the most stable results in our tests.

The calibration works in the following way: The user can move the cameras' focus ring via the *IN* and *OUT* buttons to either the minimum or the maximum stop. By pressing the *set min*, respectively *set max* button, the app sends a signal to the Arduino. Upon arrival, it responds with the current position of the stepper motor. The minimum and maximum positions are saved in variables throughout the session. Knowing the current speed and direction, the app also knows the position of the stepper motor at all times. Hence, it can prohibit movement past the limits of the lens.

DISCUSSION & NEXT STEPS

Regarding the existence of other DIY projects, one might come up with the question *Why is this important?* Admittedly, the current state is not much more than a combination and extension of the named. Though, we believe that the real strength of this project lies in its technological and topical compatibility with current research projects at LMU. Having the vision of a system in mind, that remotely controls all matters of a camera operator via an Android app, this paper covers a necessary and logical step towards the big picture. However, there are things that have to be addressed up front:

- The UI of our controller app is rather provisorily. Even though we deliberately neglected this topic in the scope of this paper, it is a major point for a good usability within a complex control environment.
- Vernier adjustment of the mechanical parts. The size of the gear and the gear transmission ratio have to be tested under real circumstances, in order to make sure that the movement suits the needs of a film production scenario.
- The same applies to the parameters in the Arduino software. Step size and movement speed have to be adjusted to ensure smooth movements throughout a focus sequence.

The case is equipped with its own power supply, as well as braces, that allow attachment to a camera rail. That means, that the current system can be integrated into the existing camera slider (compare *Current Research at LMU*) independently, from a hardware perspective.

Software-sided, the next step would be to conceptualize and develop a mobile application, that controls and records not only focus sequences, but also many different aspects of the camera operator related tasks. This is why the outcome of our follow focus app should be seen more as a proof of concept, rather than a fully matured product.

Figure 5 shows the assembled prototype together with a DSLR Camera.

CONCLUSION

Conclusively, this work has shown, that a more advanced tool for focus pulling can be developed with relatively simple means. The availability of existing open source projects, such as Soffer Follow Focus [12] or DIY Follow Focus [6], as well as fairly low priced electronic components lay the foundation for many DIY opportunities. Advanced commercial projects like Andra Motion Focus [8] represent the current state of the art, and, with their multitude of novel features, serve as a source of inspiration for own projects.

REFERENCES

1. Android developers - platform versions. `https://developer.android.com/about/dashboards/index.html`.
2. Bluetooth low energy. `https://developer.bluetooth.org/TechnologyOverview/Pages/BLE.aspx`.
3. Dassault systemes. `http://www.3ds.com/de`.
4. Lenzhound - kickstarter. `https://www.kickstarter.com/projects/1608311032/lenzhound-wireless-lens-motor-control-system?lang=de`.
5. Proaim camera follow focus x1. `http://www.proaimshop.com/pas/PROAIM-Camera-Follow-Focus-X1.html`.

6. Diy follow focus, 2014. **https://vimeo.com/85608648**.
7. Stepper vs. servo, 2014. **http://www.amci.com/tutorials/tutorials-stepper-vs-servo.asp**.
8. Andra motion focus, 2015. **http://andra.com/**.
9. Burgdorf, P. Take-over strategies and their effect on control and recording of automated camera tracking shots. Bachelor's Thesis, 2015.
10. Crabb, K. *The Movie Business: The Definitive Guide to the Legal and Financial Secrets of Getting Your Movie Made*. Simon and Schuster, 2005.
11. Mörwald, P. Development of a remote controlled camera slider. Bachelor's Thesis, 2014.
12. Soffer, A. Github - follow focus. **https://github.com/soffer/Follow-Focus**.
13. Stapleton, O. So you wanna work in movies? - camera department. **http://www.cineman.co.uk/#camera**.