SelfieBot: A Robot to Detect, Photograph, and Tweet Smiles

Kuo-Pao Yang, Patrick McDowell, Patrick Dolan, Cameron Otts, Grace Chenevert, Cara Tunstall
Department of Computer Science
Southeastern Louisiana University
Hammond, LA 70402 USA

Abstract— **SelfieBot is a robot that searches for a face to take a picture and post to a social media site. It is successfully outfitted with a Raspberry Pi, Camera, and Arduino. Using OpenCV library, SelfieBot recognizes faces. The SelfieBot detects smiling faces, takes photographs, and then uploads these pictures to Twitter. There are myriad things that can be done with an autonomous image-processing robot. We plan to use SelfieBot to recognize the faces of animals and take animal selfies in the future.**

Keywords- OpenCV; Raspberry Pi; Arduino

I. INTRODUCTION

In the modern world, selfie-taking is a predominant part of social media. Whether on Instagram, Twitter, or Facebook, users are constantly uploading photos of themselves. In fact, the number of selfies increased 90,000% between 2012 and 2014[1]. A 'selfie' is a photograph that the subject has taken by themselves usually via a front-facing mobile phone camera. Selfies are also approximately 1/3 more likely to receive likes and comments on social media than pictures without faces [2]. Taking a photo of one's self with one's own hands can be unwieldy and poses difficulty for capturing the proper angle and scale. Before SelfieBot, the solution to this problem is the Selfie Stick. A Selfie Stick is an extendable handle with a mobile device holster. Selfie Sticks allow the selfie photographer to take selfies with better angles and of larger width and breadth. While the Selfie Stick does alleviate some of the difficulties involved in proper selfie-taking, it still requires the user to take the photo.

SelfieBot offers a solution to this problem. Using facial detection technology, SelfieBot finds faces, then automatically takes photos of these faces. SelfieBot requires no user input from the person being photographed. SelfieBot provides a new way for the world to take selfies.

In this paper, related research to SelfieBot is outlined first. Then, an in-depth look at the implementation of SelfieBot is explored. Final sections include an evaluation of the SelfieBot implementation, concluding statements, and an outline of plans for future work.

II. RELATED WORK

A selfie is a self-portrait photograph, typically taken with the front-facing camera on a smartphone. Selfies are often shared on social networking services such as Facebook, Instagram and Twitter. Most selfies are taken with a camera held at arm's length or pointed at a mirror, rather than using a timer.

A selfie stick is a monopod used to take "selfie" photographs by positioning a smartphone or camera beyond the normal range of the arm. The metal sticks are typically extendable, with a handle on one end and an adjustable clamp on the other end to hold a phone in place. Some have remote or Bluetooth controls, letting the user decide when to take the picture, and models designed for cameras that have a mirror behind the view screen so the shot can be lined up.

There have been several robots for photographers made to roam around and photograph people. One such robot is the autonomous robot photographer named Lewis [3]. Lewis consists of a video camera and a digital camera mounted on a large wheeled robot that can roam around a room and take pictures to be sent via email [4]. Lewis uses its video camera to observe human behavior to choose appropriate times to take photographs [5]. Furthermore, Lewis uses geometric constraints to accurately detect faces and other objects [6].

Another mobile photographing robot was a four wheeled robot with a digital camera mounted on a tripod and a laptop computer as the processor [7]. This robot finds a person's face by finding dark pixels that are not skin colored as eyes [8]. Other photographing robots use body detection to find an identify humans [9]. Lastly, some photographing robots use laser-based methods of tracking people and crowds [10].

All of these robots vary from the SelfieBot, mostly through the design of the robot. It is unique in that it uses a Raspberry Pi along with an Arduino. The Raspberry Pi controls the higher level functions, i.e. path planning and camera functions such as image processing and face detection through the use of an image processing API, while the Arduino controls the robotic platforms actuators and limb movement.

III. IMPLEMENTATION

A. *PhantomX AX Quadruped Mark II*

This project is based on the robotic kit, PhantomX AX Quadruped Mark II shown in Fig. 1. SelfieBot has a total of twelve servos. These servos are Dynamixel AX-12A servo actuators. These actuators have the ability to track speed, temperature, shaft position, voltage, and load, thus providing feedback to the servo controller. Standard hobby servos do not have this ability. This added feature allows, the control algorithms to more precisely control the shaft positions of the servos. Also, since each servo has a built in microcontroller the servo's feedback loop parameters can be adjusted individually, allowing for more control of the speed and strength of the servo motor's response. This distributed approach leaves the main controller free to perform other functions.

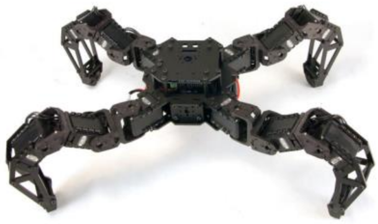


Fig. 1. PhantomX AX Quadruped Mark II

B. *Raspberry Pi*

The Raspberry Pi is chosen as the central processor and controller for all operations for the SelfieBot project. The SelfieBot uses a Raspberry Pi 1 Model B due to its availability and the flexibility and its ease of use. The Raspberry Pi Model B has 512 MB of RAM, two USB ports and a 100mb Ethernet port. In order to best use the built in features of the Raspberry Pi, Python is chosen as the language to handle all on board processing.

Originally, all computations and image processing were to be performed on board of the Raspberry Pi. After initial testing and usage, it became apparent that the Raspberry Pi was not capable of the quick computations and memory intensive processes that are needed for responsive image processing. In order to circumnavigate this problem, the SelfieBot project developed and used an API that uses an external processor to improve computation time by performing resource intensive operations like image processing on a machine with more resources than the Raspberry Pi. The API designed and developed for the SelfieBot project is hosted on an online server. The Raspberry Pi is able to access it using a wired connection to the internet.

The SelfieBot's Raspberry Pi acts as the control hub for interaction between the various hardware and software systems. The Raspberry Pi acts as an intermediary for the Raspberry Pi camera to the API, the API to the robot. Fig. 2 shows the flow of data through the Raspberry Pi in the SelfieBot Project.

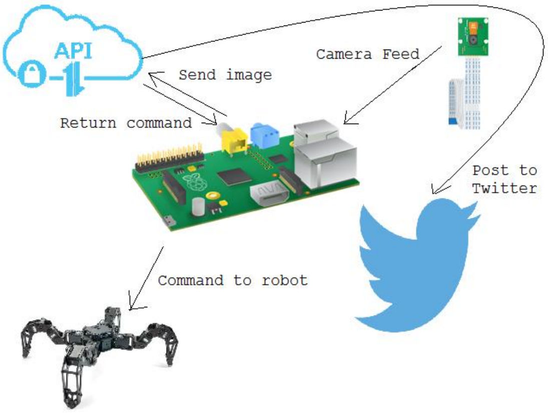


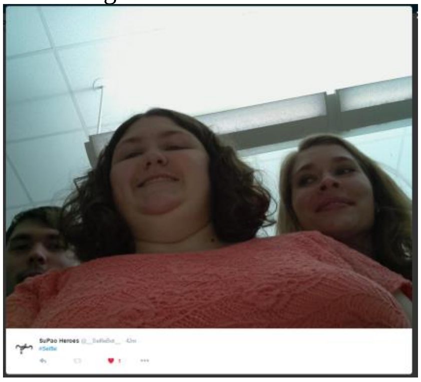
Fig. 2. Flow of Data in the SelfieBot Project

In its communication from the Raspberry Pi camera and the API, the Raspberry Pi handles the passing of images. The Raspberry Pi uses a combination of built in functions of the Raspberry Pi camera as well as functions from the Open Source Computer Vision Library (OpenCV) to capture video frames as images in python. The images are then converted from images to base-64 strings on the Raspberry Pi using Python.

After the API receives and processes the incoming base-64 strings, it transmits a string containing a command back to the Raspberry Pi. There are two types of commands that can be returned from the API. The first is a motion command for the PhantomX Quadruped. The other is a capture command for the Raspberry Pi Camera.

When the API returns a motion command string, this triggers the Raspberry Pi's communication with the PhantomX Quadruped. The Raspberry Pi then uses the string from the API to select the appropriate command for the Quadruped robot. Once the proper command is found, the Raspberry Pi sends a command in the form of a single character to the Quadruped for it to execute.

To take a picture, the API returns a capture command object, which triggers the Raspberry Pi to signal the Raspberry Pi camera. The Raspberry Pi will run a Python script that captures a video frame and saves it as an image. This image is then transmitted using the Twitter API to be posted on social media as shown in Fig. 3.



SuPao Heroes @__SelfieBot__ 42m
#Selfie

Fig. 3. Selfie Posted to Twitter by SelfieBot

C. *Raspberry Pi Camera*

The SelfieBot uses a Raspberry Pi Camera to take in a video feed in order to find a face. The Raspberry Pi Camera is placed on top of the robot then the video feed is processed on the Raspberry Pi itself.

The Raspberry Pi Camera is 25mm x 20mm x 9mm in size. The overall weight of the Raspberry Pi Camera is only around 3 grams, which makes it easy to carry around on the Raspberry Pi. To connect to the Raspberry Pi and the Camera using a ribbon cable, which is connected to the Raspberry Pi's BCM2835 processor via the Camera Serial Interface (CSI) bus. The CSI bus travels down the ribbon cable to the Pi and the camera.

The sensor on the Raspberry Pi Camera is a 5 megapixel camera. The Raspberry Pi Camera is capable of taking 1080p30, 720p60 and VGA90 video modes and still captures that can be saved as PNGs, JPGs, and other picture files. The Raspberry Pi Camera can be used to take high resolution video and still picture captures at up to 30 frames per second for a 1080p resolution.

For the SelfieBot project, the Raspberry Pi Camera is plugged into a Raspberry Pi Model B. Due to the limited ability of the 512 MB RAM and processers of the Raspberry Pi Model B, the Raspberry Pi Camera is not able to be used to its full potential.

D. OpenCV

OpenCV is an open source computer vision library. It is a tool to aid programmers by providing multiple libraries and functions to do both basic and advanced computer vision. This library provides many effective image processing routines, thus mitigating spin-up time that developers had to previously take.

The SelfieBot is receiving visual input from the Raspberry Pi Camera through the OpenCV interface. The image will then be processed through various methods and functions to clean and evaluate the image.

The SelfieBot uses Haar Cascades, which are reference files of what things should look like. A Haar Cascade is basically a classifier which is used to detect the object for which it has been trained for, from the source. Using these cascades, SelfieBot can detect faces, which in turn will trigger a callback. Once a callback is detected, the Raspberry Pi will send a serial command to the Arbotix-M Robocontroller which will execute the command.

E. Arduino

The Arduino integrated development environment is development environment in which users can write code called sketches. The sketch written for this project uses arrays of servo positions shown in Fig. 4.

```
PROGMEM prog_uint16_t lean3[] = {12, 512, 510, 503, 325,
507, 516, 510, 511, 506, 325, 509, 517};
PROGMEM prog_uint16_t lean2[] = {12, 512, 510, 503, 350,
507, 516, 510, 511, 506, 350, 509, 517};
PROGMEM prog_uint16_t lean1[] = {12, 512, 510, 503, 375,
507, 516, 510, 511, 506, 375, 509, 517};

PROGMEM transition_t leanSeq[]  = {{0,1}, {lean1, 500}};
PROGMEM transition_t leanSeq2[] = {{0,1}, {lean2, 500}};
PROGMEM transition_t leanSeq3[] = {{0,1}, {lean3, 500}};
```

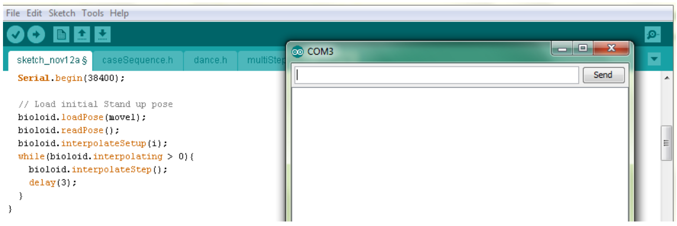
Fig. 4. Position Arrays and Sequences

This project plays the sequences via void functions in the main loop shown in Fig. 5.

```
void tilt(){
  bioloid.playSeq(leanSeq);
  while(bioloid.playing){
     bioloid.play();
  }
}
```

Fig. 5. Play Sequence Function

It is based on commands being sent for the Serial Monitor in the Arduino integrated development environment shown in Fig. 6.



```
  Serial.begin(38400);

  // Load initial Stand up pose
  bioloid.loadPose(movel);
  bioloid.readPose();
  bioloid.interpolateSetup(i);
  while(bioloid.interpolating > 0){
    bioloid.interpolateStep();
    delay(3);
  }
}
```

Fig. 6. Serial Monitor

F. ArbotiX-M Robocontroller

The Arbotix-M Robocontroller is an Arduino microcontroller-comparable device shown in Fig. 7. The robocontroller incorporates a powerful AVR microcontroller, XBee wireless radio, dual motor drivers, and 3-pin servo-style headers for digital and analog Input/Output.

Fig. 7. ArbotiX-M Robocontroller

G. XBee 1mW Communication Module

The 2.4GHz XBee 1mW Communication Module from Digi allows a very reliable and simple communication between microcontrollers, computers, systems, and virtually anything with a serial port. In Fig. 8, XBee 1mW Communication Modules work well with Arduino-based systems, and work with the built in serial port.

Fig. 8. XBee 1mW Communication Module

IV. EVALUATION

The functionality provided by the SelfieBot project is superior to traditional selfie taking because it eliminates the need for any user input in the photography. Users need only smile and face the SelfieBot camera. SelfieBot is superior to other image-processing robots in that it takes photographs automatically, as well as uploading photographs to social media automatically.

The SelfieBot project is an innovative project that integrates many different computing and hardware mediums: Arduino, Python, Raspberry Pi, and robotics. The SelfieBot project takes all these elements and uses them together to solve one problem: automatic selfie-taking using facial recognition technology and image processing to find faces with a Raspeberry Pi camera, using Arduino code to position the robot for ideal photograph angles, and then uploads the photo to social media.

V. CONCLUSION AND FUTURE WORK

Selfies are now a popular method of communication via social networking. The problem with selfie-taking is that it is quite awkward and difficult for users. SelfieBot offers an innovative solution to this problem. SelfieBot finds smiling faces using facial recognition technology and takes photos of these faces. After taking these photos, SelfieBot automatically uploads them to a pre-selected social media like Twitter.

The future of SelfieBot is exciting. There are a myriad things that can be done with an autonomous image-processing robot. We plan to use SelfieBot to recognize the faces of animals and take animal selfies. This will make SelfieBot a marketable solution to the problem of humans in the dangerous field of wildlife photography. SelfieBot's photographs will then be applicable not only to social media sites like Instagram and Twitter, but also publications like National Geographic and Time magazine.

REFERENCES

[1] F. Souza, D. Casas, V. Flores, S. Youn, M. Cha, D. Quercia, and V. Almeida, "Dawn of the Selfie Era: The Whos, Wheres, and Hows of Selfies on Instagram," Proceedings of the 2015 ACM on Conference on Online Social Networks (COSN '15), pp. 221 – 231, 2015.

[2] M. Kalayeh, M. Seifu, W. LaLanne, and M. Shah, "How to Take a Good Selfie?" Proceedings of the 23rd ACM International Conference on Multimedia (MM '15), pp. 923 – 926, 2015.

[3] Z. Byers, M. Dixon, W. Smart, and C. Grimm, "Say Cheese! Experiences with a Robot Photographer," American Association for Artificial Intelligence (AAAI), AI Magazine, 25(3):37 – 46, 2004.

[4] G. Hsu, J. Huang, "A Photographer Robot with Multiview Face Detector," 2016 IEEE International Conference on Industrial Technology (ICIT), pp. 2152 – 2156, 2016.

[5] W. Smart, C. Grimm, M. Dixon, and Z. Byers, "Interacting with a Robot Photographer," Proceedings of the AAAI Spring Symposium on Human Interaction with Autonomous Systems in Complex Environments, pp. 181 – 186, 2003.

[6] M. Dixon, F. Heckel, R. Pless, and W. Smart, "Faster and More Accurate Face Detection on Mobile Robots Using Geometric Constraints," 2007 IEEE/RSJ International Conference on Robots and Systems (IROS '07), pp. 1041 – 1046, 2007.

[7] S. Suzuki, Y. Mitsukura, H. Takimoto, T. Tanabata, N. Kimura, and T. Moriya, "A Human Tracking Mobile-Robot with Face Detection", 2009 35th Annual Conference of IEEE Industrial Electronics, pp. 4217 – 4222, 2009.

[8] M. Putro, K. Jo, "Real-time Face Tracking for Human-Robot Interaction," 2018 International Conference on Information and Communication Technology Robotics (ICT-ROBOT), DOI: 10.1109/ICT-ROBOT.2018.8549902, 2018.

[9] T. Yoshimi, M. Nishiyama, T. Sonoura, H. Nakamoto, S. Tokura, H. Sato, F. Ozaki, N. Matsuhira, and H. Mizoguchi, "Development of a Person Following Robot with Vision Based Target Detection," 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5286 – 5291, 2006.

[10] J. Cui, H. Zha, H. Zhao, and R. Shibasaki, "Robust Tracking of Multiple People in Crowds Using Laser Range Scanners," 18th International Conference on Pattern Recognition (ICPR'06), DOI: 10.1109/ICPR.2006.1017, 2006.