Intelligent Computer Vision Tracking and Embedded Microcontroller in the Sporting Domain

Bassam Shaer and Timothy Stewart
bshaer@uwf.edu and tljstewart@gmail.com
Electrical and Computer Engineering Department, University of West Florida
Fort Walton Beach, Florida, United States

Abstract – *This paper presents the design and implementation of an intelligent embedded microcontroller utilizing computer vision within the basketball sporting domain. The design objectives of the overall system are to identify and track a basketball shooter utilizing feature detection capabilities of computer vision in real time, and return the basketball to the shooter regardless of their location or movement pattern. The embedded system is used to provide a communication link between the camera interface and system hardware, to provide image data processing for computer vision feature detection, to provide signal processing for precise motor control in response to the processed image to allow real time tracking. This paper will describe the design, implementation, and results of each of these objectives*

Keywords: Intelligent, Tracking, Computer Vision, Basketball, Electronic Sporting Device

1. Introduction

The system known as the “Computer Vision Basketball Rebound and Return System” is an autonomous tracking system that attaches to the backboard of a basketball hoop to provide the serious basketball athlete with continuous rebounding and returning of the basketball for productive and efficient shooting repetition while reducing distractible work which is defined to be any work other than shooting.

Basketball has over 24 million active participants in America, which is the most compared to other major sports such as football, baseball, soccer, and hockey [10]. The motivation for such a system stems from the needs of the basketball athlete. Current commercially available rebounding and return systems on the market cost upwards of $6000. The computer vision basketball rebound and return system can be built for under $400. These expensive systems are employed by professional teams and are rigidly preprogrammed return systems, none utilize the expansive field of computer vision to solve this problem for the basketball athlete. The current systems are agnostics to who is shooting and unintelligent when it comes to returning the ball to the athlete, those systems simply return the ball to the preselected location on the court, regardless if the athlete is there or not.

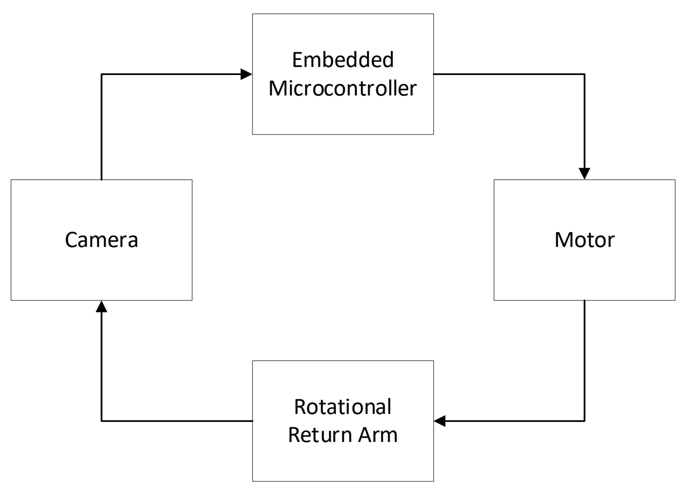

Figure 1: Control Flow Feedback Diagram

2. Embedded System Overview

The embedded system is composed of the Raspberry Pi 3 Model B+, using the Broadcom BCM2835 SoC [8], core architecture of ARM11 and 700 MHz clock. The CPU is a low power ARM1176JZFS applications processor with 512MB SDRAM soldered on top of the Broadcom chip.

The GPU is dual core with a VideoCore IV® Multimedia Co-processor which provides Open GL ES 2.0, hardware-accelerated OpenVG, and 1080p30 H.264 high-profile decoding capable of 1Gpixel/s, 1.5Gtexel/s or 24GFLOPs with texture filtering and DMA infrastructure.

The Operating system boots from Micro SD card, and runs Raspbian Jessie. The physical dimensions of the Pi are 85 x 56 x 17mm.

The Pi is powered via 5V supply which has polarity protection, 2A fuse and hot-swap protection so you can plug/unplug USB without resetting the board.

The Pi Camera V2 is a high quality 8 megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi, featuring a fixed focus lens. It's capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p60/90 video.

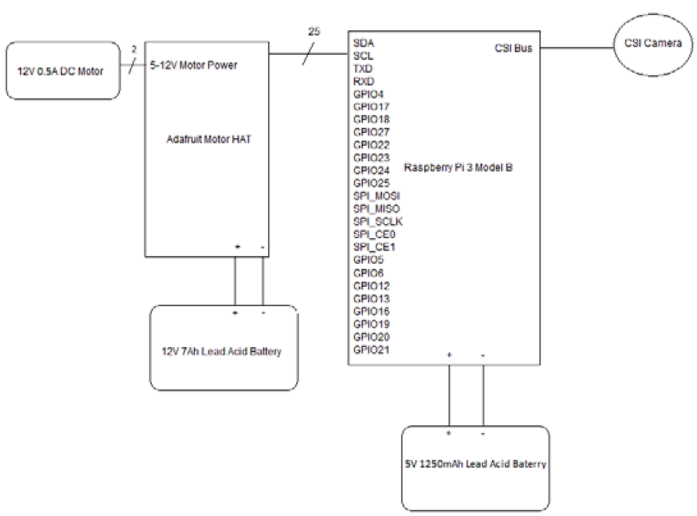

Figure 2: System Block Diagram

It attaches to the Pi by way of one of the small sockets on the board upper surface and uses the dedicated CSI interface, designed especially for interfacing with cameras. The board itself is tiny, at around 25mm x 23mm x 9mm. It also weighs just over 3g, making it perfect for mobile or other applications where size and weight are important [6].

Driving the 12V 0.5A 10RPM high torque motor is the Adafruit Motor 'HAT' where 'HAT' stands for hardware attached on top. The HAT drives motors by using TB6612 MOSFET drivers with 1.2A per channel continuous current supply rating with a 3A peak for approximately 20ms at a time [1]. This chip handles all the motor and speed controls over I^2C. Only two pins SDA and SCL are required to drive the multiple motors, and since it's I^2C you can also connect any other I^2C devices or HATs to the same pins.

3. Computer Vision Processing

The software which controls the motor and computer vision aspects of the system was implemented on the Raspberry Pi utilizing an open source computer vision library called OpenCV, and programmed in the Python computer language [7].

Considering Figure 4. First an image is captured using the Pi Camera, then the image processing begins. The first step is to resize the image to a known pixel width, in this case 600 pixels. Then, the image is converted from RGB color space to HSV color space for the best color detection results. Once in the HSV color space a range of acceptable values for the color we are looking to track is applied to the image this results in a black and white image with the feature being tracked becoming white. Further image processing is applied to the image to reduce noise and false positives, namely, erode and dilate functions from the OpenCV library [3].

The largest enclosed area is tracked in the resulting image and is identified by a yellow circle and red trail indicating the path it has traveled on the final image.

Finally, a signal is sent to the motor indicating the direction to turn the basketball return arm to get the ball back to the shooter. To achieve consistently accurate return of the ball the shooter is kept in the center of the image at the 300 pixel mark utilizing the 600 pixel width frame we resized the image to.

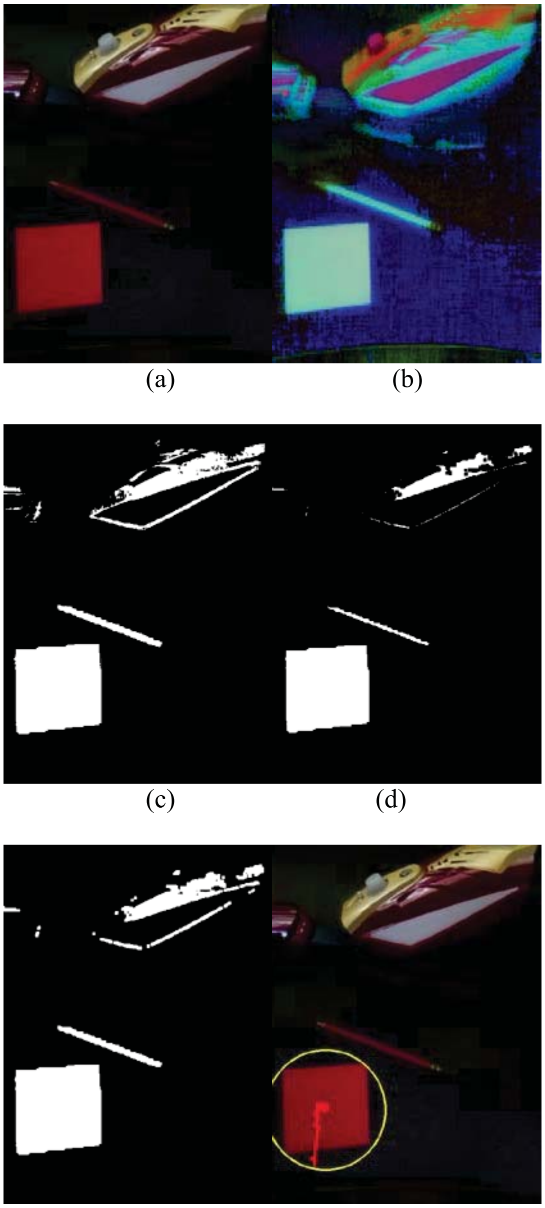

Figure 3. (a) Image to be processed (b) Image converted to HSV (c) Image processed with inRange function (d) Image processed with the erode function (e) Image processed with the dilate function (f) Final result being tracked.

An improvement upon color feature tracking would be facial recognition, which can be implemented using the OpenCV library. The camera lens would require auto-zoom, auto-focus functionality as well as a 60 FPS frame rate and a 1/120th shutter speed to maintain high resolution of the athlete's facial features during motion.

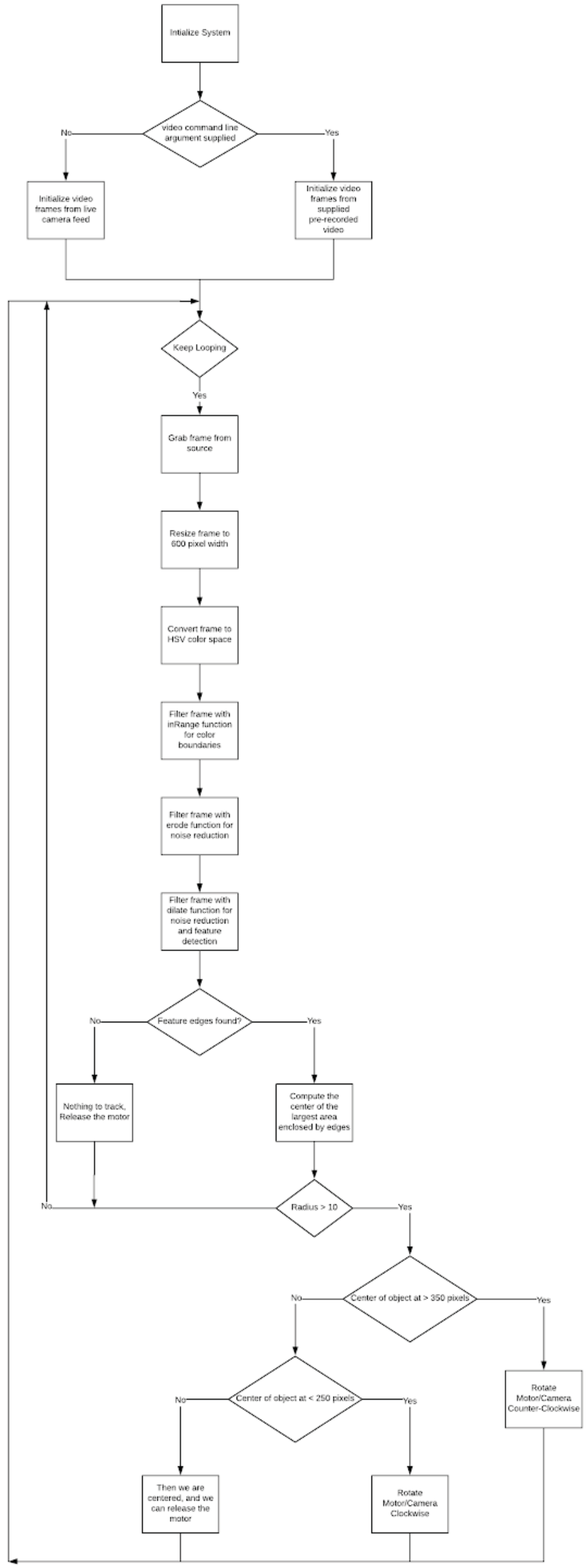

Figure 4: Computer Vision Filtering and Tracking Algorithm Flow Chart

To note a few packages, Numpy is a scientific computing python package. CV2 is the OpenCV library and the Adafruit MotorHAT package is the motor controller library which attaches to the GPIO pins on the Raspberry Pi.

The upper and lower boundaries are defined as shown in Figure 5 to detect and track a wide range of red colors in the spectrum and accomplish this task well under varied lighting conditions and reflective surfaces.

```
# define the lower and upper boundaries of "red"
# in the HSV color space
redLower = ( 160 , 140 , 50 ) #red
redUpper = ( 179 , 255 , 255 ) #red
```

Figure 5: HSV Color Space for Red Upper and Lower Boundaries

The source code for the entire implementation can be downloaded from the GitHub repository at https://github.com/timothyljstewart/bcm [11].

4. Structural Frame

The frame of the system is built from Polyvinyl chloride (PVC) 2½ inch and ¾ inch rods using a slotting system such that a hole is drilled into the 2 ½ inch rod and the ¾ inch is passed through. This design gives the structure strength and durability while also remaining light enough such that the basketball hoop can support the system. Also, the structure was built intentionally as to not shift the center of gravity of the original unloaded basketball hoop.

In Figure 6. One can observe in the must upper left of the image a horizontal bar parallel with the ground. This is a ¾ inch PVC rod with a threaded solid steel rod passed through. This is the pivotal weight distribution point, it bares the load of the system and translates the load to the backboard and basketball hoop itself. While maintain center of gravity.

A nylon mesh netting is used to enclose the retrieval structure. The enclosed retrieval structure measures 70in in length, 70in in width and is 40in above the rim. This design provides ball returns to the tracked athlete on 99.9% of shots taken, made or missed within the nylon mesh netting.

Also, the height of the structural frame above the rim reinforces optimal shooting arc for the practicing athlete approximately 55 degrees, this angle optimizes shooting distance and diameter of the rim that the ball observes on its path [2].

There are four mount points on the four corners of the backboard which support the 85 pounds of the entire system. These mount points are steel hinge plates that bolt to the backboard with a ¾ inch nylon locking nuts. The head of one of these 2½ inch bolts can be seen in Figure 6. Another 4½ inch bolt and ¾ inch locking nut are then passed through the steel hinge plate parallel with the backboard. This bolt also passes directly through the polyvinyl chloride rods behind the backboard at the four corners which finalizes the implemented mounting system.

5. Return Arm and Rotation Operations

To translate the software logic in to real world actions a 12 Volt 0.5 Amp 10RPM motor was used to rotate a belt-driven pulley system. Attached to the motor shaft is a 3D fabricated gear designed in SolidWorks.

Figure 6: Structural Frame

The rotational return arm consists of an inner ring which provides structure and support to an outer ring. The outer ring of the rotational arm is supported only by contact with the two mount points which support the inner ring. This enables the outer ring to rotate freely. The toothed belt is wrapped around the outer ring to provide rotation from the gear and motor upon signal from the microcontroller. The camera is mounted on the outer ring, so that when the rotational arm is rotated by the feedback loop seen in Figure 1, the camera will also be moved to continue data accusation of the athlete and complete the feedback loop.

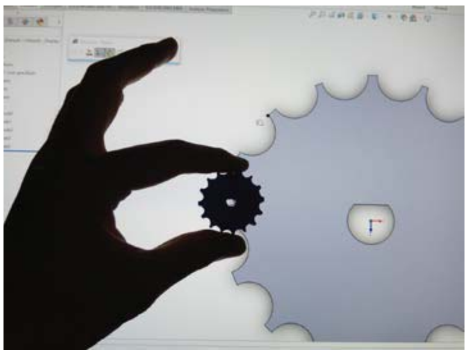

Figure 7. 3D Printed Gear used for Belt-Driven Rotation of the Return Arm.

Figure 7 illustrates the 3D printed gear used for belt-driven rotation of the return arm. This gear attaches to the shaft of the motor, the toothed belt is then fitted and tensioned between the gear and the return arm, this tension results in the rotation of the return arm.

6. Operational Results & Testing

Primary testing was focused on: camera computer vision feature detection, structural integrity, rebound rate and return rate.

To accomplish the computer vision feature detection, it was important to assert the exact HSV color space min and max values. Obtaining this min and max was a bit of an art as with lighting color detection confidence changes drastically. The camera was mounted in outdoor lighting and these HSV values were fine tuned for outdoor environment, however, these same values also work exceptionally well in low lighting. The feature detection is consistent and maintains tracking with a moving athlete regardless of the pattern of movement at a reasonable translational speed. Faster translational speeds can be achieved by implementing a faster motor.

The structural integrity was shock and vibration tested. Vibration testing was conducted using a reciprocating tool to provide sufficient vibrational disturbance to various parts of the system. This proved the system could handle violent vibrations safely and without adverse effect.

Rebound rate of the enclosed nylon mesh netting was tested and of 100 shots taken only one rebound was lost. That is a 99% rebound rate.

The rate at which a ball returns to the athlete was tested and at 6.02 meters the ball returns to the athlete under 3 seconds.

7. Conclusions

Basketball is the most actively participated sport in America and basketball athletes require productive and efficient shooting repetition to improve. Current commercially available rebounding and return systems on the market cost upwards of $6000 and are rigidly preprogrammed return systems, and unintelligent when it comes to returning the ball to the athlete. Those systems will return the ball to the preselected location on the court, regardless if the athlete is there or not.

The computer vision basketball rebound and return system can be built for under $400 and implements computer vision to identify and track the athlete for hours of continuous productive shooting practice no matter the location on the court or complex movement patterns performed.

Further recommendations to consider for future improvements and implementations are for personalized facial detection combined with multi-user intelligent queuing functionality. To implement this one would require a wide-angle camera to capture the entire basketball court with long range zoom and focus capabilities, as the current camera has a fixed lens, to capture facial features at distance. As well as a gimbal system to offset vibration when returning the ball. The OpenCV library has support for facial recognition.

The multi-user intelligent queuing system would recognize and track each shooter with facial recognition. Each shooter would be placed in an intelligent FIFO queue, and if a shooter does not have his or her hands ready to receive a pass then they are moved to the back of the queue, mimicking how a human would not pass the ball to someone unprepared or unaware.

Also, based on the large number of basketball goal designs, an on-the-ground system would be more suitable for a general purpose, easily deployable and removable system, such that one can setup for shooting practice and then remove for a fully-completive basketball game. Additionally, a grounded system would allow the implementation of pneumatics to return the ball at a greater rate, more comparable to a human passing a basketball.

8. References

[1] Adafruit DC & Stepper Motor Hat For Raspberry Pi. Retrieved April 20, 2018 from https://www.adafruit.com/product/2348

[2] ESPN Sports Science, Sport Science: Stephen Curry. Retrieved April 19, 2018 from https://www.youtube.com/watch?v=HOiH1eVCggw

[3] Eroding and Dilating. Retrieved April 20, 2018 from https://docs.opencv.org/2.4/doc/tutorials/imgproc/erosion_dilatation/erosion_dilatation.html

[4] inRange. Retrieved April 20, 2018 from https://docs.opencv.org/2.4/modules/core/doc/operations_on_arrays.html

[5] Numpy. Retrieved April 20, 2018 from https://docs.scipy.org/doc/numpy-1.8.1/index.html

[6] Pi Camera. Retrieved April 20, 2018 from https://www.raspberrypi.org/documentation/hardware/camera/README.md

[7] Python. Retrieved April 20, 2018 from https://docs.python.org/2/

[8] Raspberry Pi 3 Model B+. Retrieved April 20, 2018 from https://www.raspberrypi.org/documentation/

[9] Reading and Writing Video. Retrieved April 20, 2018 from https://docs.opencv.org/3.0-beta/modules/videoio/doc/reading_and_writing_video.html

[10] "Wayback Machine" (PDF). Web.archive.org. October 21, 2011. Archived from the original on October 21, 2011. Retrieved April 19, 2018.

[11] Source Code. Retrieved May 24, 2018 from https://github.com/timothyljstewart/bcm