

**Open Access** 

# Object Detection and Tracking for Wireless Motion Control using MATLAB and Irobot Create

## Ejofodomi OA<sup>1\*</sup>, Ofualagba O<sup>1</sup>, Ovoradua VO<sup>1</sup>, Oviri JO<sup>1</sup>, Ettoh GO<sup>1</sup>, Gonu S<sup>1</sup>, Bambe AE<sup>2</sup>, Onah Ll<sup>3</sup>, Ovokerie JU<sup>3</sup> and Idongesit PJ<sup>3</sup>

<sup>1</sup>Department of Electrical and Electronics Engineering. Federal University of Petroleum Resources. Nigeria

<sup>2</sup>Department of Marine Engineering, Federal University of Petroleum Resources, Nigeria

<sup>3</sup>Department of Mechanical Engineering, Federal University of Petroleum Resources, Nigeria

## Abstract

This paper presents a robust algorithm for object detection and tracking using MATLAB. This program captures a single image using a webcam attached to the laptop. The captured image is processed to find a predefined object. The centroid of the object is calculated. Subsequent images are then acquired. Each time a new image is acquired, the predefined object and its centroid is extracted and compared to the centroid of the previous image. The differences in the x and y coordinates of the centroids are used to determine the direction of movement of the lrobot Create. Once the algorithm determines the direction of movement, the program issues the move command wirelessly to the robot with the aid of a Bluetooth Adapter Module.

**Keywords:** Object detection; Object tracking; Robot; MATLAB; Motion control

# Introduction

Object tracking and wireless motion control are important topics in robotics and automation that are currently being explored [1]. This paper explores the use of the motion of an object to wirelessly control the motion of a robot. The algorithm for wireless motion control was written in MATLABTM (The MathworksTM, Natick, MA, USA) [2]. A webcam or camera attached to a PC takes continuous snapshots of a particular object (a red object in this case), detects the presence and position of that object using color detection and image segmentation, and tracks the movement of the object in subsequent snapshots. The object movement is quantified and used to determine the direction and distance moved by a robot, the Irobot Create [3]. Motion commands are sent to the Irobot wirelessly by means of a Bluetooth Adapter Module (BAM).

The algorithm for the object detection, tracking and wireless motion control is presented in detail in the materials and methods section. Possible improvements to the existing algorithm are briefly explained in the discussion and conclusion section.

# Materials and Methods

#### Materials

**Irobot create:** The Irobot Create, shown in Figure 1, is a low cost preassembled mobile robotic platform, a complete robotic development kit for programming new robot behaviours.

The Create has a cargo bay area that houses a 25 pin port for digital and analog input and output. One of the major advantages of using the Irobot Create is the ease of attaching external hardware and electronic devices [3].

The Irobot possesses a serial port through which sensor data can be read and motor commands can be generated using the "Irobot Roomba Open Interface protocol" (a MATLAB based function dedicated to control Irobot Create). In order to control the Irobot using MATLAB, a toolbox named 'Irobot Create toolbox' has to be used that is not present in MATLAB. The toolbox replaces the native low-level numerical commands, with a set of high level, intuitive, MATLAB functions that:

• Create a link between your PC and Create using your PC's serial port or Bluetooth connections.

• Provide a variety of user-friendly drive commands, calibrated in SI units.

• Read the bump and cliff sensors, determine distance driven, and battery life in SI units.

• Use the MATLAB command line or script files to control the robot. All code is developed, stored and executed on the PC base station-not the Create [4].

**Element direct bluetooth adapter module:** It is possible to control the Irobot Create using a serial cable. However, this option requires a physical connection to a PC, and would require the PC to be placed in the cargo bay area of the robot during operation.

To overcome this deficiency, an Element Direct Bluetooth Adapter module (BAM) was utilized in this project (Figure 2). The BAM,



\*Corresponding author: Ejofodomi OA, Department of Electrical and Electronics Engineering, Federal University of Petroleum Resources, Nigeria, Tel: +234 703 919 0064; E-mail: tegae@yahoo.com

Received September 02, 2015; Accepted September 22, 2015; Published September 28, 2015

**Citation:** Ejofodomi OA, Ofualagba O, Ovoradua VO, Oviri JO, Ettoh GO, et al. (2015) Object Detection and Tracking for Wireless Motion Control using MATLAB and Irobot Create. Adv Robot Autom 4: 136. doi: 10.4172/2168-9695.1000136

**Copyright:** © 2015 Ejofodomi OA, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.



Figure 2: Bluetooth Adapter Module (BAM) (a) BAM (b) BAM connected in Irobot cargo bay.



Figure 3: Flow chart of MATLAB algorithm for Object Detection, Tracking and Wireless Motion Control.



Figure 4: Wireless motion control using object detection and tracking. (a) Positioning red object in front of camera. (b) Snapshot of predefined red object. (c) Predefined object detection using color detection and image segmentation in MATLAB (d) second snapshot of predefined object with object displaced to the left. (e) Predefined object detected. X and Y displacements are calculated from object centroids in (c) and (e). (f) Irobot Create turns left and moves in that direction since displacement in X is greater than displacement in Y, and the X displacement is negative.

connected to the cargo bay port of the Irobot without any extra wires or cables, enables wireless control of the robot from a Windows, Macintosh, or Linux PC. It accomplishes this by providing a virtual serial port connection between the Irobot Create and the PC, so the PC can communicate with the robot in the same way it would as if it were attached with a serial cable. Unlike WiFi and other wireless systems, Bluetooth is both a low-power and low-cost option. **Host PC:** A computer with a webcam or camera attached is required to capture images. The MATLAB software and the Irobot Create toolbox should be installed on the host computer. The MATLAB software is used to capture images from the webcam, process the images and send wireless instructions to the Irobot via the BAM.

## Methodology

The algorithm of MATLAB program presented in Figure 3. The steps of the algorithm are explained in further detail below.

**Camera initialization:** A video object was created in MATLAB and linked to the camera or webcam. The video object was set to acquire every 10th image in the video stream and the size and color space of the captured images was specified. Images were acquired using the getsnapshot() command.

**Object detection:** Once an image was acquired, the color components of each pixel was separated into blue, red and green, resulting in the creation of three matrices containing the blue, red and green color components for each pixel in the image. The blue and green color values were subtracted from the red color values to yield an image containing only red-hued pixels. This image was then converted to a black and white image. Segmentation of the black and white image was performed to extract the object from the image.

**Object property acquisition:** Object properties were acquired using the regionprops function in MATLAB. This function was used to obtain the area, centroid and bounding box of segmented regions in the image. The region with the largest area indicated the presence of the predefined object. Once the region containing the predefined object was identified, the x and y coordinates of its centroid  $(x_1,y_1)$  was extracted.

**Object tracking:** When another image is acquired, the object detection and property acquisition steps are repeated. A second set of x and y coordinates  $(x_2,y_2)$  is obtained, indicating the centroid of the predefined object. To determine object displacement in the x-axis, the x coordinate of the first centroid is subtracted from the x coordinate of the second centroid (i.e.  $X_d = x_2 - x_1$ ). To determine object displacement in the y-axis, the y coordinate of the first centroid is subtracted from the y coordinate of the second centroid (i.e.  $Y_d = y_2 - y_1$ ).

Wireless motion control: If displacement in the x direction is greater than the displacement in the y direction (i.e.  $X_d > Y_d$ ), it is assumed the object has been displaced in the x-direction only. The value of  $X_d$  is rounded to the nearest hundred). If  $X_d > 0$ , instructions are sent wirelessly to the Irobot via BAM, causing it to turn right, travel a distance equal to the rounded value of  $X_d$ , and then turn left to face its original direction If  $X_d < 0$ , the Irobot is instructed to turn left, travel a distance equal to the rounded value of  $X_d$ , and then turn right to face its original direction (Figure 4).

Similarly, if displacement in the y direction is greater than the displacement in the x direction (i.e.Y<sub>d</sub> > X<sub>d</sub>), it is assumed the object has been displaced in the y-direction only. The value of Y<sub>d</sub> is rounded to the nearest hundred). If Y<sub>d</sub> >0, instructions are sent wirelessly to the Irobot via BAM, causing it to move forward a distance equal to the rounded value of Y<sub>d</sub>. If Y<sub>d</sub> < 0, the Irobot is instructed to move backwards a distance equal to the rounded value of Y<sub>d</sub>. It was assumed that a single pixel in either the x or y direction was equivalent to 1mm of motion by the Irobot in the physical world.

After Irobot has completed its movement, the values of  $x_2$  and  $y_2$  are transferred to  $x_1$  and  $y_1$  respectively, so that the x and y centroid coordinates of the next acquired image can then be stored in  $x_2$  and  $y_2$ .

Citation: Ejofodomi OA, Ofualagba O, Ovoradua VO, Oviri JO, Ettoh GO, et al. (2015) Object Detection and Tracking for Wireless Motion Control using MATLAB and Irobot Create. Adv Robot Autom 4: 136. doi: 10.4172/2168-9695.1000136

# **Discussion and Conclusion**

The MATLAB algorithm was able to successfully detect the predefined object and track its movement in subsequently acquired frames. It should be noted that a red object was selected as the predefined object for this experiment (Figure 4), and so the object detection algorithm uses this quality in accurately identifying the position of the object in the image. The algorithm will have to be altered if another object with a different color is selected as the predefined object. The algorithm successfully tracked the object in subsequent images, displaying the displacements in the x and y direction and determining the direction to move the Irobot based on the values of the x and y displacements, as outlined in the methodology section above. Moving the red object upwards caused the Irobot to move forward, while moving the object downwards caused the Irobot to move backwards. Moving the object left caused the Irobot to turn left and move in that direction, while moving the object right caused the Irobot to turn right and move in that direction.

It is important to note that the movement of the Irobot was calibrated with the values of the x and y displacement, with a single pixel representing 1 mm. However, this scale can be readily adjusted to elicit greater motion from the Irobot (e.g. a single pixel could represent 1 cm or 10 cm). Moving the red object closer to the PC causes it to have a larger size and area in captured images, while moving it further away from the PC makes it appear smaller. This third dimension can be

used to refine the Irobot's movements. For instance, the computed area of the object in the image could be used to set the speed of the Irobot movement. Larger areas could set the speed to higher values, while a lower area could be used to reduce the speed of movement of the Irobot.

The algorithm can be further modified so that both x and y displacements are recognized simultaneously by the Irobot, instead of selecting only the x or y displacement to determine direction of motion. This would enable the robot to more accurately follow the movement of the object.

In conclusion, this paper demonstrates how object detection and tracking can be used to wirelessly control a robot's movement. Object detection was achieved by means of color detection and image segmentation. The color detection section of the algorithm needs to be improved to accommodate objects with different colors.

## References

- Buksh R, Routh S, Mitra P, Banik S, Mallik A, et al. (2014) Implementation of MATLAB based object detection technique on Arduino board and Irobot Create. International Journal of Scientific and Research Publications 4: 1-5.
- 2. http://in.mathworks.com/matlabcentral/fileexchange/http://www.mathworks.com
- 3. http://www.irobot.com/http://www.Irobot.com
- 4. Esposito JM, Barton O, Koehler J, Lim D (2011) MATLAB Toolbox for the Create Robot.