Tracking Objects using Artificial Neural Networks and Wireless Connection for Robotics

Johnny Mauricio Barreno Oñate[1], Darío José Mendoza Chipantasi[1], Nancy del Rocío Velasco Erazo[2]
[1]*Departamento de Energía y Mecánica,*
[2]*Departamento de Ciencias Exactas,*
Universidad de las Fuerzas Armadas ESPE, Sangolqui, Ecuador.
jmbarreno@espe.edu.ec

***Abstract*—Tracking objects are used in many areas, and one of them is robotics. The goal in this work focuses on a robot that can follow an object that is in front of it. This application has two links: wireless and Bluetooth. The first one connects a mobile phone mounted on a robot for image acquisition and a personal computer (PC), and the second links a PC and a mobile robot to control the motors by open source, Arduino Board. The algorithm uses several patterns for training the Artificial Neural Network (ANN) and for object identification. Then, it is complemented by the extraction feature in Hue Saturation Value (HSV) color space. This algorithm uses C ++ language with OpenCV libraries for computer vision.**

***Index Terms*—Feature Extraction; ANN; HSV; Wireless; Object Tracking.**

I. INTRODUCTION

Tracking objects is used in several fields for human computer applications, like traffic monitoring systems or counting people in a specific space.

However, detecting and tracking objects in real time is a hard task because of the dynamic changes in the environment causing a cluttered background and object dimensions that is affected by its rotation or the movement of the camera, which can also generate an occlusion [1]. These characteristic problems should be seen in the design method for tracking. Therefore, it is generally difficult to detect a point in a dynamic background [2].

The different techniques to track objects include the motion vector estimation and optical flow [3], improvement of mean shift in algorithm [7], and robust and adaptive key point [4]. In [5], the tracking is a compact and low cost system that combines some algorithms. Bay in [13] develops SURF (Speeded up Robust Features) and Bradsky in [14] develops a computer vision face tracking with Continuously Adaptive Mean Shift (CAMShift). In [6], the moving object tracking is achieved by using CAMShift and the Kalman filter.

In [8], ANN is used for robust mobile tracking. The tracking is done with the whole body area or with a part, as in [9], which uses the face. In [10], the author uses OpenCV library to give a short implementations of the object detection.

One of the ways in which robots can know the correct path in which to go is posing the route previously or following an object. In [11], an efficient model of tracking algorithm is developed for the vision of the robot. Other authors in [12] use vision information (tracking) for Robot Control.

In most cases, researchers are looking for a visual control based on the extraction of 3D information, and a threshold that distinguishes the object. Then, it generates an interactive system with two main requirements: The identification of the object and its location as well as the filtering of unwanted information.

This paper develops an application for a mobile robot that has an Arduino Target and a camera phone to track an object by taking the image that contains the object. First, we get a base of image samples from a particular object and train an ANN. Then, we segment the object by color to make it useful for Mobil robot navigation. We use two parts: wireless network (image acquisition) and Bluetooth network (for servo motor control).

A description of the phases is presented: Section II explains the process on how an image is acquired by the phone camera. It is processed by the computer and finally, the robot moves according to the direction of the object detected. In Section III, the experimental results are presented, and Section IV contains the conclusion of the observations.

II. EXPERIMENT

The algorithm follows a sequential process divided in some phases. Figure 1 indicates the phases of the whole process starting from beginning to end.

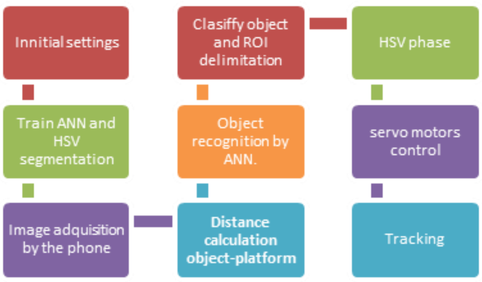

Figure 1: Phases of the process

A. Image Acquisition

The Software starts with determining a database (images) containing object patterns to be followed. The object patterns should be extensive (about 100 images per object). The pattern images (samples) are read during the process of filtering and thresholding. Figure 2 shows how the images are standardized, where parts that are not of interest are cut out. Then, scaling is conducted to reduce the size of the image to a conventional standard (16x16), hence reducing the computational cost. Consecutively, the matrix containing the pixel value of the object in a range of 0 to 255 becomes a vector of zeros and ones according to a threshold for storing a file.txt.

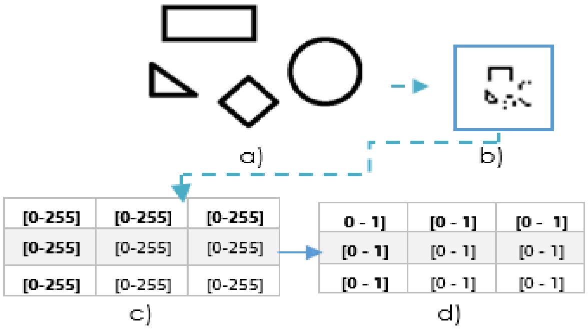

Figure 2: a) Original image, b) Staggered image (16x16) c) Gray scale image and d) Binary image

B. *Artificial Neural Network Setting*

The next step, creating the ANN, focuses on establishing three essential layers (input, hidden, output): The first layer contains the number of pixels per sample of each object (256), the second layer is set to a value of 16, and the third layer contains the number of objects assigned to search. Initially, the file is used to train ANN as well as identify and recognize the object in question, where various parameters are set as the number of iterations (1000), the synaptic weights (0,000001), the type of training (Back-Propagation) and the value of the latter.

When configuring the ANN has been completed, the veracity of the training for which a third of the total samples of the database (30%) is taken, the output layer presenting the allocation of weights is checked (0-1), and the largest value corresponds to the object is recognized by the ANN.

C. *Model HSV*

This part illustrates the improvement of the object recognition. In this context, the algorithm is based on the use of a minimum and maximum values of threshold for detecting the object by color and using HSV color space in three layers to verify if the object sought is located.

D. *Distance Calculation Object-Platform*

The object centroid is calculated [coordinates x, y], once the location of the object sought is satisfied using the method of "moments" and the area. This value is used for the mobile platform for it has an autonomous control as shown in Figure 3. The size is used to triangulate the distance object-camera (similar triangles): A larger area of the object reduces the distance between them and vice versa:

$$focal\_distance = AnchoObj\ [Pixel] \times Real\_distance\ [cm] \ /\ Weight\_Obj\ [cm] \quad (1)$$

$$distance_{object} = weight\_Obj\ [cm] \times Real\_distance\ [Pixel] \ /\ Height\_Obj\ [Pixel] \quad (2)$$

E. *Mobile Platform Configuration*

The hardware built for the platform for tracking the object is as shown in Figure 4. It consists of a smartphone for the usage of the video camera available on it, and it configured as IP camera using a commercial Android application that

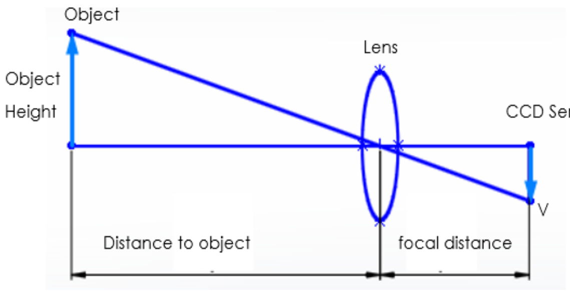

Figure 3: Representation of the distance between Object and camera position

allows to be assigned to a specific IP direction. Further, for the movement of the platform to have an idler wheel and two servo motors connected to the same bridge H, it is integrated to an Arduino UNO. For the phase control of the actuators, there is also an integrated Bluetooth device, which allows bi-directional communication between the board and the computer. In addition, a liquid crystal display (LCD) screen is required to display relevant user information like the received data to perform remotely autonomous control sent from a computer that did a control using the image received since the smartphone also has a display that shows the direction of the platform and the object distance. The power supply for the electronic board and the H bridge makes it through two 9V batteries, and a source Arduino 5V power is used to turn on to the LCD and Bluetooth card.

Figure 4: System scheme

III. Results And Discussion

The comparison with other existing algorithms mentioned at the beginning of this document was to compare the validity of the algorithm developed. There has been a basic test to look for the pattern in Figure 5.

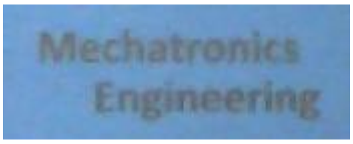

Figure 5: Pattern for tracking

The SURF algorithm is applied in Figure 6.a. Although it follows the object quite well, its computational cost is high. CAMSHIFT algorithm is used (Figure 6.b): This works well, but it is affected by some brightness changes. In Figure 6.c, the Kalman filter algorithm is applied, which partitioned the searched pattern and the prediction is not adequate.

Finally, in Figure 6.d, ANN algorithm is tested and it successfully identifies the whole pattern.

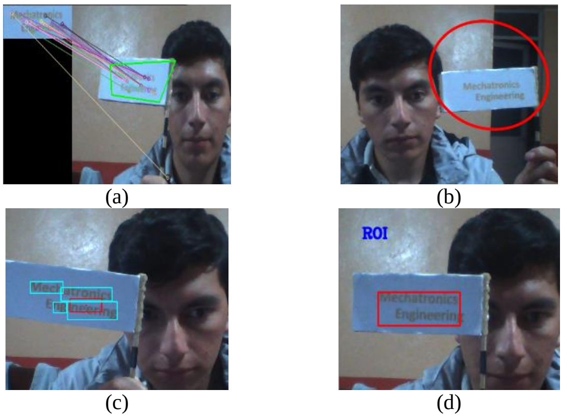

Figure 6: a) SURF, b) CAMShift, c) Kalman-Filter and d) ANN algorithm

When the ANN recognizes the object of interest, as shown in Figure 7(a), the algorithm delimits a region of interest (ROI). It should be noted that initially one webcam connected serially to the computer is used. Figure 7(b) shows the object applied the HSV model with a color mask and a uniform background. Figure 7(d) shows that when the object is in a complex background, unwanted information is generated, but it is small when using the HSV color space.

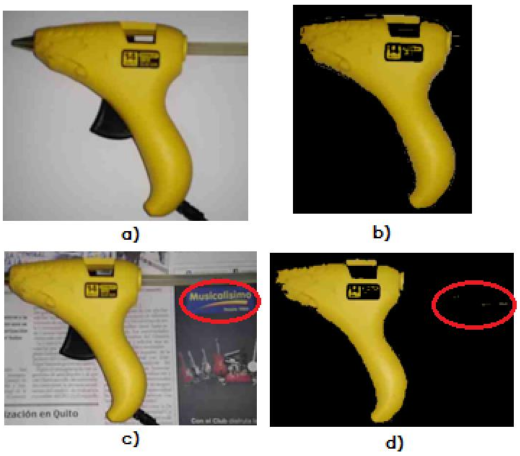

Figure 7: a) Original image, b) HSV image with color mask, c) Image with dynamic background, d) HSV image with color mask

Table 1
Control Logic for the Platform

Section	Direction	Left Servo motor Counter-clockwise	PWM [0-255]	Right Servo motor Clockwise	PWM [0-255]
A	↑	↻	60%	↻	60%
B	↑	↻	100%	↻	100%
C	↑ + →	↻	100%	↻	75%
D	→ + ↑	↻	75%	↻	50%
E	→ + ↑	↻	50%	↺	25%
F	↑	↻	30%	↻	30%
G	← +↑	↺	25%	↻	50%
H	← +↑	↻	50%	↻	75%
I	↑ + ←	↻	75%	↻	100%
No object	→	↻	30%	↺	30%
	←	↺	30%	↻	30%

The data platform-object distance is critical to position the platform at a safe distance from the object. Moreover, with the coordinates [x, y], its intended targets is the platform towards the object, as shown in Figure 8.

Figure 8: a) Original image b) Image with reference of coordinates [x, y] (mass center)

The control of the power units is based on a pulse-width modulation (PWM) that will increase in proportion to the distance generated between the platform and the object (Figure 9).

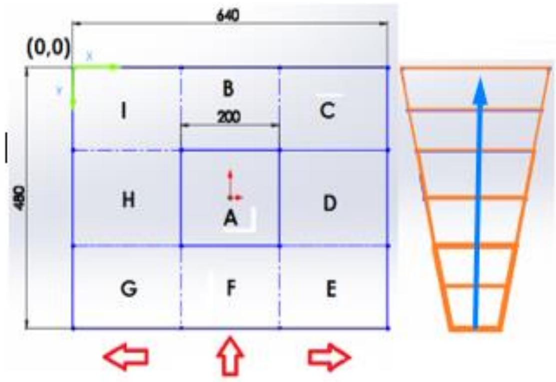

Figure 9: Correlation between position and distance

Table 1 indicates the priority possibilities as the platform moves according to the distance and the position of the object is presented.

The so-called Internet Protocol (IP) Webcam application can transmit the captured images (640x480) through a wireless network without internet.

In carrying out the final test and with the mobile platform (Figure 10), there is an object that can be anything used for ANN training.

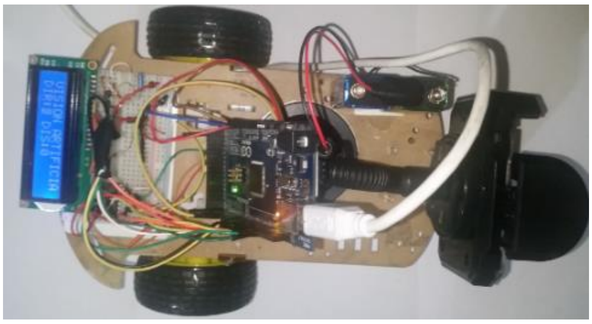

Figure 10: Mobile robot

First, the algorithm recognizes the object within the dynamic environments (Figure 11.a), immediately afterwards its center of the mass is sought (to know your location; thus sending this data to the personal computer (PC) to perform the control with feedback of information received and in turn, the PC sends the response to control the actuators of the platform, thus closing the control loop. In the remote regions (B, C) (Figure 11.c), the platform will move at a maximum capacity, but in Figure 11d to the nearest regions (E, F, G) the opposite will happen. Therefore, the process follows the claims that the object has always been in the region (Figure 11.e), which is the central part and the spotlight for proper detection and subsequent monitoring.

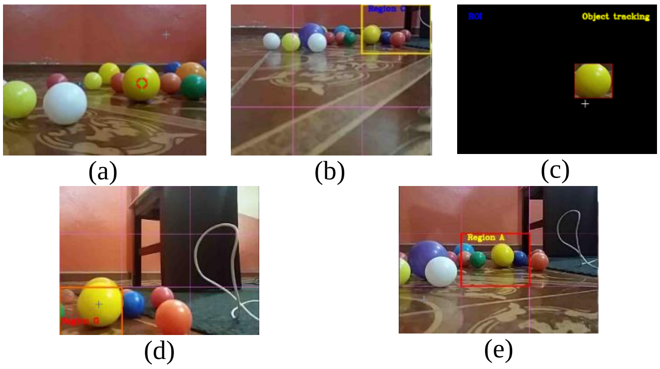

Figure 11: a) object mass center b) the ROI object, c) Object isolated, d) near object e) Object center

IV. CONCLUSION

This paper presents the implementation of ANN for tracking objects by a robot, where it requires a considerable number of samples of the object image to collate the information.

As shown by the tests, ANN recognizes the object of interest quite well. In addition, it also has the use of the HSV model that helps to get a robust system, as long as the object does not have a very diverse range of colors.

However, the evidence confirms that a delay in the execution time occurs because of the wireless connection, which is a very common problem in tele-robotics. However, when the webcam PC is used for recognizing the process, it runs faster. Future research could find an optimization in the sending and receiving of frames.

ACKNOWLEDGEMENT

Our sincere thanks for all the support received from our professors, friends and department managers: Energy and Mechanics (Mr. Marco Singaña), Exact Sciences (Ms. Jacqueline Pozo) and especially the team that makes up the Mechatronics Engineering directed by Mr. Vicente Hallo, friend and career director; who belong to the "Universidad de las Fuerzas Armadas - ESPE Extension Latacunga".

REFERENCES

[1] Wang J. and Yang J. 2008. Relative discriminant coefficient base multi-cue fusion for robust object tracking. Springer Link.

[2] Wang L. Y. Y. 2008. Real-Time Tracking of the Shoot Point from Light Pen Based on Camshift. IEEE. 560-564.

[3] Kale K., Pawar S. and Dhulekar P. 2015. Moving object tracking using optical flow and motion vector estimation. IEEE. 1 – 6.

[4] Pieropan, A., et al. 2016. Robust and adaptive keypoint-based object tracking. Advanced Robotics, 30(4): 258-269.

[5] Karam A., Qadi W., Melkon K., Fakes, B., Sababha B., & Al-Mousa A. 2014. A compact portable object tracking system. In 2014 5th International Conference on IEEE Information and Communication Systems (ICICS). 1-5.

[6] Huang S. and Hong J.. Moving object tracking system based on camshaft and Kalman filter. IEEE. 1423 – 1426.

[7] Li N., Zhang D., Gu X., Huang L, Liu W. and Xu T. 2015. An improved mean shift algorithm for moving object tracking. IEEE.1425 - 1429

[8] Islam A., Hasan Md. R., Rahaman R., Raiyan K. S. M. and Ahmme S. 2007. Designing ANN using sensitivity & hypothesis correlation testing. IEEE. 1 – 6.

[9] Ching, Y. K. 2009. Visitor face tracking system using OpenCV library. IEEE. 196 – 199.

[10] Druzhkov, P. N. et al. 2011. New object detection features in the OpenCV library. Pattern Recognition and Image Analysis, 21(3):384-386.

[11] Comport, Andrew I., Marchand, Éric Chaumette, François. 2005. Efficient model-based tracking for robot vision. Advanced Robotics, 19(10):1097-1113.

[12] Lang, H., Wang, Y., & de Silva C. W. 2010. Vision based object identification and tracking for mobile robot visual servo control. In Control and Automation (ICCA), 2010 8th IEEE International Conference on IEEE. 92-96.

[13] Bay H., Tuytelaars T., & Van Gool L. 2006. Surf: Speeded up robust features. In Computer vision–ECCV 2006, Springer Berlin Heidelberg, 404-417.

[14] Bradski G. R. 1998. Computer vision face tracking for use in a perceptual user interface.