World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:9, No:4, 2015

Object Motion Tracking Based On Color Detection for Android Devices

Zacharenia I. Garofalaki, John T. Amorginos, John N. Ellinas

***Abstract*—**This paper presents the development of a robot car that can track the motion of an object by detecting its color through an Android device. The employed computer vision algorithm uses the OpenCV library, which is embedded into an Android application of a smartphone, for manipulating the captured image of the object. The captured image of the object is subjected to color conversion and is transformed to a binary image for further processing after color filtering. The desired object is clearly determined after removing pixel noise by applying image morphology operations and contour definition. Finally, the area and the center of the object are determined so that object's motion to be tracked. The smartphone application has been placed on a robot car and transmits by Bluetooth to an Arduino assembly the motion directives so that to follow objects of a specified color. The experimental evaluation of the proposed algorithm shows reliable color detection and smooth tracking characteristics.

***Keywords*—**Android, Arduino Uno, Image processing, Object motion detection, OpenCV library.

I. Introduction

ONE of the most important and challenging tasks of computer vision is object detection and motion tracking. Numerous online or offline applications have been developed for objects surveillance, traffic control, security environments, medical image processing, augmented reality, etc. [1]-[3]. A review of tracking algorithms and their classification to different categories is presented in Alper Yilmaz's paper [4]. Among them, the low computational complexity *Kernel-based tracking* is a localization method based on the maximization of a similarity measure like the color of an object [5], [6]. The implemented kernel-based tracker, which is a typical scheme for object detection from an image, is based on color tracking, range thresholding and contour detection techniques. However, the challenge for the various approaches is to discriminate objects from their background and the success of most of the presented methods depends on the color difference between them.

Mobile devices such as smartphones and tablets with Android operating systems can be integrated into robotic applications and used to detect and track objects. Today's devices have enough processing strength and can sufficiently respond to medium complexity computer vision algorithms. One of the most used libraries for image processing and computer vision algorithms implementation is the Open source Computer Vision library (OpenCV), which has been selected to be part of the presented Android application [7].

The proposed work has been developed on a robot car which is equipped with hardware for the motion and an Android device that runs an application for object's color detection and a motion tracking algorithm. The object's captured image is subjected to a number of different image processing steps, with OpenCV functions, that finally result in a blob of the object with an estimate of its area and its center with respect to mobile screen coordinates.

The tracking algorithm estimates the relative blob position with respect to screen center taking into account the Android device orientation. However, the car movement disorders, towards left or right, that are created by small deviations around the center of the screen may be alleviated by separating screen into three vertical zones and checking if object's blob crosses the borders. The width of the middle zone may be trimmed so that the forward movement to be consistent with the motion of the object.

The following sections describe the used hardware and the image processing steps for object detection and motion tracking.

II. Proposed Work

A. Overview

The proposed work concerns the development of a robot car that can track and follow colored moving objects with an Android mobile device as Fig. 1 illustrates. The chassis consists of two 20x15 cm plastic panels on top and bottom spaced by separator screws. The four wheels are 66 mm diameter rubber tires, which are turned by four dc motors, with a reduction rate 48:1 and average speed 190 rpm. The four wheel robot car is driven by an Arduino Uno board, a motor shield that can drive four dc motors and a Bluetooth device that have been placed on the top of the chassis. Next to them, an Android device has been plugged in the front side of the car and communicates with Bluetooth to the rest hardware.

B. Arduino Application

The motion of the robot car is controlled by an Arduino Uno board with a motor shield that can drive four independent dc motors. These boards can communicate with an Android device by a Bluetooth module with a baud rate of 57.600 bps. The Arduino's software provides motion in four directions after receiving the first character of the words Forward,

Z. I. Garofalaki and J. T. Amorginos are with the Department of Electronic Computer Systems, Piraeus University of Applied Sciences, 250 P. Ralli & Thivon, 12244 Egaleo, Greece (e-mail: rania@ teipir.gr, amorgi@windtools.gr).

J. N. Ellinas is with the Department of Electronic Computer Systems, Piraeus University of Applied Sciences, 250 P. Ralli & Thivon, 12244 Egaleo, Greece (phone: +302105381208; fax: +302105381436; e-mail: jellin@teipir.gr).

Backward, Left, Right and can stop motion with character "S". In the proposed work optical encoders or Hall effect sensors for wheel synchronization are not used, as this was beyond the purpose of the project. Future work would incorporate encoders and a PID algorithm for precise movement of the robot car.

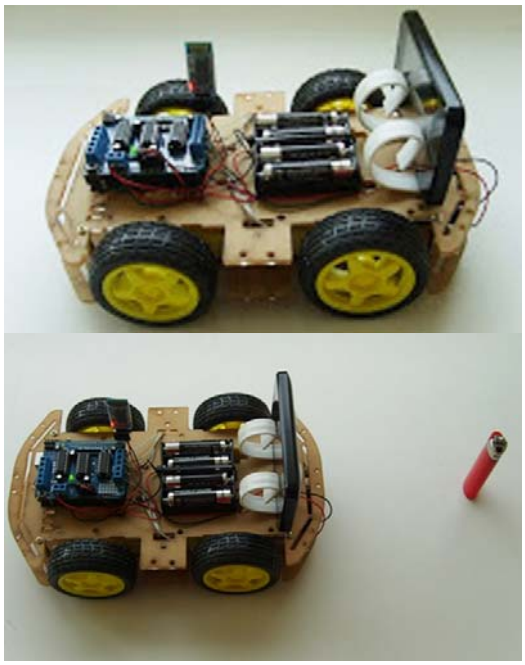


Fig. 1 Robot car for object and motion tracking

C. Android Application

The Android device runs an application that has been developed in the Android development suite and incorporates the OpenCV 2.4.9 API. The selection of this Java API to handle vision algorithms, instead of developing native code, is for simplicity reasons although is far more computational speed expensive. The basic application uses the abstract class *SampleCvViewBase.java* and the image processing steps are embedded in a subclass of it. All the following processing manipulations are functions of OpenCV and they have been selected for efficient results with low computational complexity.

D. Object Detection Procedure

Object detection is a kernel-based method that tracks an object from its color that has been pre-specified by the user. The Android device captures a scene by its back camera and the resulting image frame is subjected to various image processing steps in order to isolate the object with the pre-determined color. The image processing steps are illustrated in the following block diagram, Fig. 2. The Android device continuously captures an image, as Fig. 3 (a) illustrates, and converts the default RGBA color space into HSV because colors are limited to one variable instead of three and this facilitates image processing. The conversion of the colored image to a binary image is performed by color filtering using thresholds for the three components of HSV color space. After filtering, the object that has the pre-defined color resembles like a group of white pixels, Fig. 3 (b). However, there are a lot of white pixels dispersed across the frame, which are pixel noise.

Usually, the removal of pixel noise is performed by erosion and dilation morphological operations with a structuring element that may be of suitable size in order to eliminate the noise but to leave the object intact, Fig. 3 (c). The resulting object blob, which is a group of white pixels that form a coherent space, is analyzed for determining the largest contour, which is assumed to be the desired object. The largest contour is shown with a white rectangle that bounds the desired object whereas other objects of the same color are ignored. This is a simplistic and rough approach but with low computational complexity. The alternative and more precise way is the implementation of a feature extraction algorithm that is far more complicated than the proposed method.

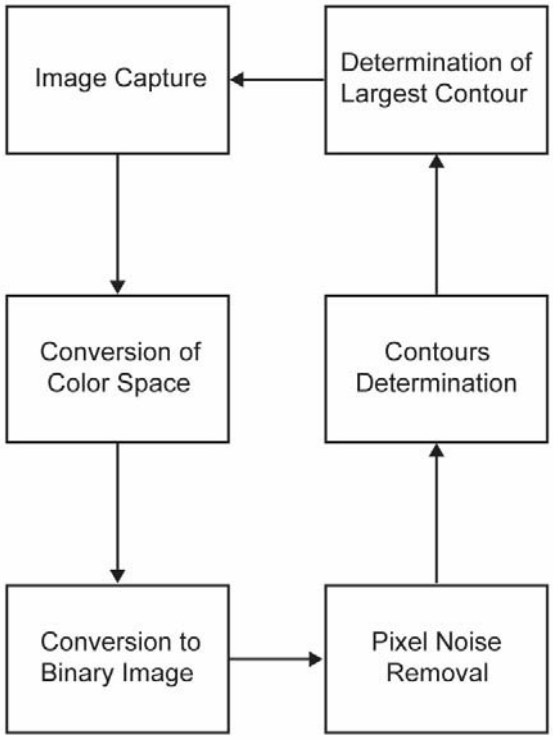
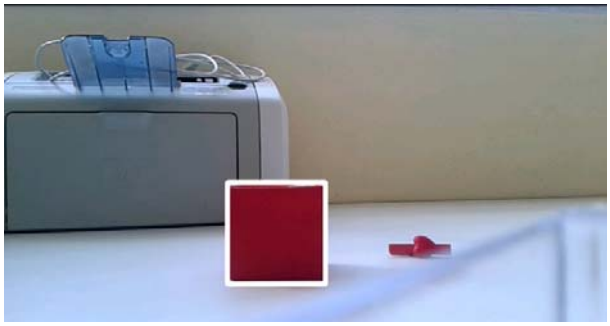


Fig. 2 Block diagram for object tracking by color detection

(a)

(b)

(c)



(d)

Fig. 3 Image processing steps: (a) Frame of the captured image; (b) Binary image after color filtering; (c) Binary image after pixel noise removal; (d) Object tracking after largest contour evaluation

Fig. 4 Object is bounded by a white rectangle for motion tracking

E. Object Motion Tracking Procedure

After the above procedure of the largest contour determination, the resulting image shows the largest object bounded by a white rectangle. OpenCV provides functions for rectangle dimensions or area calculation, which facilitates the implementation of the remaining procedure for motion tracking. Fig. 4 illustrates the bounded object with the coordinates of its upper-left corner and the size of the bounding rectangle.

The easiest way for motion tracking is to determine if the object's center is left or right with respect to the middle of the screen when Android device is in landscape mode. However, for small object movements about the middle of the screen, the application sends messages for left or right motion and the robot car moves continuously. The mitigation of this abnormal behavior can be obtained by dividing the mobile device's screen into three zones where the middle zone is used as a buffer zone. If the center of the object is within the middle zone, the car is moving forward whereas if it crosses the zone to the left or right, the device sends a message to the robot car in order to turn left or right respectively. The area and the dimensions of the bounding rectangle may be evaluated by the Open CV's functions and therefore the relative position of the object with respect to smartphone's view may be determined. Fig. 5 shows smartphone's view with the bounding rectangle of the tracked object and the estimated quantities. The width and height of the view are represented by *W* and *H*, the width and height of the object are *w* and *h* and the coordinates of the top-left corner of the object are *x* and *y* respectively.

The view is divided into three equal zones, as shown in Fig. 5 with the two imaginary lines P1 and P2, and the direction of the robot car's motion is defined by the determination of the object's relative position. This is estimated by the following logical statement, where $\left(x+\frac{w}{2}\right)$ is the middle of the object's boundaries. The width of the middle zone may be trimmed so that the motion of the robot car to be more consistent to the object's movement. It should not be too small because the car will turn left or right very easily and it should not be very large because the car will mainly move forward losing the object.

The approaching speed of the car to the object may be stable or vary with inverse proportion to the area of the white rectangle and the car may stop when the area overcomes a threshold value.

$$\left(x+\frac{w}{2}\right) \le \frac{W}{3} \qquad Turn\ Left \qquad (1)$$

$$\left(x+\frac{w}{2}\right) \ge \frac{2\cdot W}{3} \qquad Turn\ Right \qquad (2)$$

$$\frac{W}{3} < \left(x+\frac{w}{2}\right) < \frac{2\cdot W}{3} \qquad Move\ Forward \qquad (3)$$

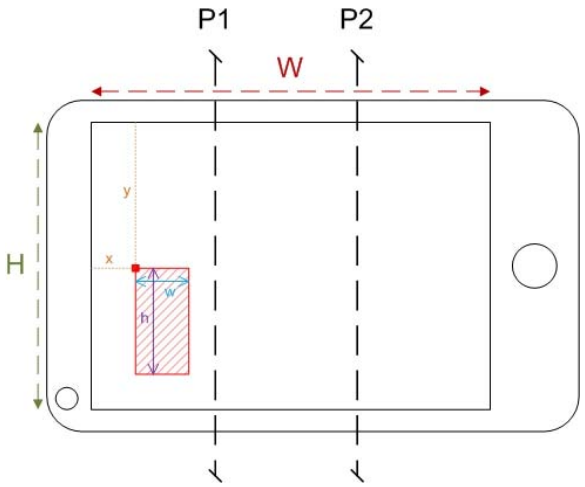


Fig. 5 Determination of the relative object's position

III. Conclusion

This paper describes the development of a four wheeled robot car moving by an Arduino Uno microcontroller board with a motor shield and a Bluetooth module, controlled by an Android device that is running an application that can track moving objects of a pre-specified color. As a first step, the Android device captures an image that is subjected to various image processing manipulations, such as color space conversion, color filtering, noise removal and largest contour estimation, resulting in a blob for an object with a predefined color. The object of the largest contour is bounded by a white rectangle and its center is determined in order to track its motion and inform the robot car about the direction of movement. The motion of the robot car to the correct direction, in order to follow object's motion, can be more accurate if the landscape screen of the smartphone device is divided into three vertical zones and the decision is taken when the center of the object's bounding rectangle crosses a zone.

Acknowledgment

This paper is part of a Master's degree thesis project for the Department of Electronic Computer Systems Engineering, which is funded by the Research Committee of Piraeus University of Applied Sciences.

References

[1] Tang Sze Ling et al, "Colour-based Object Tracking in Surveillance Application", *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. I, Hong Kong, March 2009.
[2] J.F. Engelberger, "Health-care Robotics Goes Commercial: The Helpmate Experience", *Robotics*, vol. 11, 1993, pp. 517-523.
[3] L. Davis, V. Philomin and R. Duraiswami, "Tracking Humans from a Moving Platform", *IEEE Computer Society Proceedings of the International Conference on Pattern Recognition*, vol. 4, 2000, pp. 4171.
[4] O. Javed and M.S. Yilmaz, "Object Tracking: A survey", *ACM Journal of Computing Surveys*, vol. 38, no. 4, article 13, 2006.
[5] D. Comanciu, P. Meer, "Mean shift: A robust approach toward feature space analysis", *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 24, no. 5, 2002, pp. 603–619.
[6] D. Comanciu, V. Ramesh, P. Meer, "Kernel-based object tracking", *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 25, 2003, pp. 564–575.
[7] http://opencv.org/platforms/android.html

Zacharenia I. Garofalaki received the B.Sc. degree in Electronic Computer Systems Engineering, School of Engineering, from Piraeus University of Applied Sciences, Greece, in 2000. She is currently pursuing her M.Sc. degree in "Applied Information Systems" at the Piraeus University of Applied Sciences, Department of Electronic Computer Systems Engineering. Her research interests are in object oriented programming, digital image processing and embedded systems.

John T. Amorginos received the B.Sc. degree in Electrical Engineering, from Piraeus University of Applied Sciences, Greece, in 2009, where he is currently a research associate. He is currently pursuing his M.Sc. degree in "Applied Information Systems" at the Piraeus University of Applied Sciences, Department of Electronic Computer Systems Engineering. His research interests are in computer architecture and mechatronic systems.

John N. Ellinas graduated in Electrical and Electronic Engineering from the University of Sheffield, Sheffield, England, in 1977 and received the M.Sc. degree in Telecommunications from the University of Sheffield, in 1978. He also received the Ph.D. degree in informatics and telecommunications from the University of Athens, Department of Informatics and Telecommunications, in 2005. Since 1983, he has been with the Department of Electronic Computer Systems, Piraeus University of Applied Sciences in Greece, where he is a Professor.

His main interests are in the field of embedded computer systems and digital image processing. His research activity is focused on microcontroller systems design, image and video coding, image restoration and watermarking.