The Implementation of a Red Object Tracking Algorithm with a Single, Static Camera [†]

Ali Arshad *, Saman Cheema * and Umair Ahsan

Department of Electrical Engineering, GIFT University, Gujranwala 52250, Pakistan; nashariam401@gmail.com

* Correspondence: ali.arshad@gift.edu.pk (A.A.); saman.chaeema@gift.edu.pk (S.C.)

[†] Presented at the 1st International Conference on Energy, Power and Environment, Gujrat, Pakistan, 11–12 November 2021.

Abstract: In recent years, activity recognition and object tracking are receiving extensive attention due to the increasing demand for adaptable surveillance systems. Activity recognition is guided by the parameters such as the shape, size, and color of the object. This article purposes an examination of the performance of existing color-based object detection and tracking algorithms using thermal/visual camera-based video streaming in MATLAB. A framework is developed to detect and track red moving objects in real time. Detection is carried out based on the location information acquired from an adaptive image processing algorithm. Coordinate extraction is followed by tracking and locking the object with the help of a laser barrel. The movement of the laser barrel is controlled with the help of an 8051 microcontroller. Location information is communicated from the image-processing algorithm to the microcontroller serially. During implementation, a single static camera is used that provides 30 frames per second. For each frame, 88 ms are required to complete all three steps from detection to tracking, to locking, so a processing speed of 12 frames per second is implemented. This repetition makes the setup adaptive to the environment despite the presence of a single static camera. This setup can handle multiple objects with shades of red and has demonstrated equally good results in varying outdoor conditions. Currently, the setup can lock only single targets, but the capacity of the system can be increased with the installation of multiple cameras and laser barrels.

Keywords: image processing; frames per second; Matlab; object detection

Citation: Arshad, A.; Cheema, S.; Ahsan, U. The Implementation of a Red Object Tracking Algorithm with a Single, Static Camera. *Eng. Proc.* **2021**, *12*, 40. <https://doi.org/10.3390/engproc2021012040>

Academic Editor: Qasim Awais

Published: 28 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Computer vision is gaining considerable attention in the field of detection, tracking, and identification systems. Detection and tracking systems are designed to work with a vast range of indoor and outdoor environments. The purpose of such systems is to identify the objects of interest and their trajectories [1]. The outdoor environment is more challenging, as compared with the indoor environment.

Computer vision is used in outdoor environments for surveillance purposes, as well as for situational awareness. To be more aware of the surrounding environment, these systems are deployed to observe the traffic flow, label the busiest road toward a destination, and provide an alternative. In the outdoor environment, autonomous vehicles are among the major applications of such systems. Autonomous vehicles are driven by very sophisticated algorithms that decide the path in a predictable and rational manner. With the help of cameras and sensors on these systems, they can identify the nature of obstacles [2,3].

The proposed algorithm is an outdoor object detection system with a static camera. Among the number of object tracking and detection algorithms, the proposed algorithm stands out as a color-based detection and tracking system.

2. Related Work

Object detection and tracking involve many challenges. In every proposed system of object detection and tracking, only a few aspects or challenges are dealt with.

Swantje and Tews [1] focus on the classification of the objects that appear in real-time videos. The traditional way to deal with this is to divide the task into three different steps—namely, motion segmentation, object tracking, and object classification [2,3]. In the first step, the motion of the object is detected by comparing the background of each frame with the updated one. The difference between the frames conveys information about the direction of motion of the object. Stauffer and Grimson [4] use the same concept of classification, along with a technique named a two-pass grouping algorithm. This algorithm significantly improves the result of the system.

The concept of classification is not a difficult one, but the problem arises when the objects cross each other during motion, and the system becomes totally blind about the tags assigned to each object. In [4], a linear prediction algorithm is used to solve this problem. The same problem is addressed in [5,6] but despite all these efforts, they fail to present a complete solution.

3. System Overview

A red object surveillance system was implemented and tested in an outdoor environment by a single static camera. This tracking system is capable to identify red objects in the view of the camera and lock their positions with the help of a laser gun. An adaptive object detection algorithm was incorporated that can track the motion of existing red objects and can also identify the presence of a new red object in the area. The flow diagram of the system is shown in Figure 1.

Figure 1. Overview of the implemented system.

A surveillance camera provides 30 frames per second (FPs). These frames were required by the object detection algorithm to update the location. The red object extraction step employed a digital-image-processing technique to choose the region of interest from the current frame. This information was used by the lesser barrel to lock the target. Information hand-off between surveillance camera to laser barrel was achieved with the help of an 8051 microcontroller.

This research was conducted in two parts—hardware and software. For hardware, canon EOS R3, a laser, and two stepper motors in the full-step mode were used. For the software part, the C++ language was used to code the image-processing algorithm, and coding of the microcontroller and subsequent compilation were performed with Keil u Vision 2.0.

3.1. Red Object Extraction

An RGB image was collected with the help of the image-processing software from a live video that was available on the surveillance camera. Two different versions of the images were extracted from the original RGB image. One was named purely red-layer image (RLI), and the other was named gray-scale image (GSI). Then, color-based background subtraction was performed by subtracting GSI from RLI to identify red objects' positions in the current frame. After the subtraction process, still there were some objects that had values near the red color. These objects were removed by the 30% threshold criteria. The result of this stage was an image with red objects only (ROI). ROI might have some areas where traces of red color were present. The process of area opening was used to remove all those traces and to obtain a refined binary image (RBI). RBI was then forwarded to the image-processing software. Image-processing software provided us with the result in the form of coordinates and centroid location for each red object in the frame. At this stage of

the algorithm, make boundaries around all red objects in an image individual. We could lock as many objects as we want.

As one single, laser barrel was used during the implementation of the algorithm, only the object that had the largest size would be locked by the laser gun. The largest object in the image was filtered with the help of boundary coordinates.

$$Size = \sqrt{dx^2 + dy^2}$$

where

dx = Width of object;

dy = Height of object.

Then, we applied the algorithm on first frame by taking screenshot of coming video from the camera. The algorithm extracted the information of centroid and coordinates from the image and sent it to the controller. This process repeated for every incoming frame.

After completion of the one cycle, another frame was collected from the video by taking a screenshot of it. Proposed algorithm consists multiple image processing functional sets to perform red object extraction and tracking. In case of skipping any steps, the system would deliver inappropriate results if one of the previous steps did not achieve satisfactory performance.

3.2. Locking the Target

3.2.1. Angle Calculation

The laser barrel used angular distance to follow the object. Angular distances were calculated from the coordinates of the red object in the image. These calculations were performed by the controller, which received the coordinate's information from the image-processing software at regular intervals.

Calculations were performed using the following equations:

$$\theta_x = \tan^{-1}(x)\theta_y = \tan^{-1}(y)$$

where

$$x = dx \times 0.001805y = dy \times 0.001805$$

where dx is the horizontal distance of the object from the center in pixels, and dy is the vertical distance of the object from the center in pixels, and 0.001805 is a unit to convert pixels in real distance.

3.2.2. Step Calculation

The laser barrel was supported by the two stepper motors, one was responsible for movement in the horizontal axis, and the other was responsible for movement in the vertical axis. To move in the appropriate direction, each stepper motor was provided with the number of steps calculated by the controller. The controller continued to update these steps with the arrival of new coordinates from the image-processing software.

For the calculation of how many steps to move, one must know the step angle of the motor. In our case, the step angle of the motor was 0.2°.

Our controller received angles from serial ports, horizontal angle for the horizontal motor, and vertical angle for the vertical motor. After obtaining the angles, we had to calculate the steps, which were calculated by the following equations:

$$Horizontal_steps = \frac{\theta_x}{0.2} Vertical_steps = \frac{\theta_y}{0.2}$$

After calculating the number of steps needed, the 8051 microcontroller began to drive both motors according to the number of steps, and this process repeated continuously.

4. Conclusions

We demonstrated a color-based system for tracking and extracting red objects in an outdoor environment. The system was built on an adaptive object detection algorithm written in C++ and simulated with the help of MATLAB. The system can handle multiple objects with shades of red and demonstrated equally good results in varying outdoor conditions. This system works in real time, and with the help of careful calculations, the system response time is merely 88 ms. This response time leads to achieving a frame rate of 12 frames per second. Our approach differs from existing approaches in that the objects are reliably tracked on the basis of their color rather than on their shape and movement patterns.

5. Future Work

Within this work, the basis of object extracting and tracking was developed, which is quite effective in terms of response time. To make the system reliable, the next step is to make the frame rate adaptive to the environment, which is now fixed at 12 frames per second.

Furthermore, the static camera of the system can be replaced with multiple cameras to increase the angle of observation to 360 degrees. Viewing the object in different cameras would improve the object tracking of the system.

Funding: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Swantje, J.; Tews, A. Real-time object tracking and classification using a static camera. In Proceedings of the IEEE International Conference on Robotics and Automation, Workshop on People Detection and Tracking, Kobe, Japan, 12–17 May 2009.
2. Javed, O.; Shah, M. Tracking and object classification for automated surveillance. In Proceedings of the European Conference on Computer Vision, Copenhagen, Denmark, 28–31 May 2002; Springer: Berlin/Heidelberg, Germany, 2002; pp. 343–357.
3. Zhang, L.; Li, S.Z.; Yuan, X.; Xiang, S. Real-time object classification in video surveillance based on Appearance Learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17–22 June 2007; pp. 1–8.
4. Stauffer, C.; Grimson, W.E.L. Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 747–757. [[CrossRef](#)]
5. Israd, M.; Blake, A. CONDENSATION—Conditional Density Propagation for Visual Tracking. *Int. J. Comput. Vis.* **1998**, *29*, 5–28. [[CrossRef](#)]
6. Chen, H.-T.; Lin, H.-H.; Liu, T.-L. Multi-Object Tracking Using Dynamical Graph Matching. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December 2001; Volume 2, pp. 210–217.