Available online at www.sciencedirect.com

ScienceDirect

Procedia Technology 22 (2016) 556 – 561

Procedia Technology

9th International Conference Interdisciplinarity in Engineering, INTER-ENG 2015, 8-9 October 2015, Tirgu-Mures, Romania

Implementation of Tracking of a Moving Object Based on Camshift Approach with a UAV

Musab Coşkun[a,]*, Sencer Ünal[a]

[a]*Electrical Electronic Engineering, Fırat University, Elazığ 23200, Turkey*

Abstract

In this paper, a report is presented on the findings of a study conducted on evaluating AR.Drone performance for computer vision applications. Object tracking is a challenging problem because of tracked object motion and its size changes in the scene, illumination changes and egomotion. An approach that is able to cope with those problems using an unmanned air vehicle was applied. Unmanned air vehicle has been an increasing field of research in both civilian and military applications. Among its different models, quadrotors have advantages such as high maneuverability and moving in three directions. The unmanned air vehicle used in this paper is AR.Drone. The application is implemented using version 2.4.9 of OpenCV library with C programming language in Visual Studio 2010 environment and is capable of tracking moving objects by using the front camera of AR.Drone with resolution of 640x360 and at a frame rate of 30 fps. Object tracking process is carried on independently from the distance between AR.Drone and the tracked object. Experimental results testify that object tracking is carried out successfully with AR.Drone which is a mobile platform despite of change in object's size in the scene and of illumination.

Peer-review under responsibility of the "Petru Maior" University of Tirgu Mures, Faculty of Engineering

Keywords: moving object tracking; camshift algorithm; unmanned air vehicle.

1. Introduction

Object tracking can be described as the process of estimating the trajectory of an object over time as the object moves around a scene. It is bearing significance in the realm of computer vision due to the proliferation of high

* Corresponding author. Tel.: 00905467831770;
E-mail address: musabcoskun@yandex.com

Peer-review under responsibility of the "Petru Maior" University of Tirgu Mures, Faculty of Engineering
doi:10.1016/j.protcy.2016.01.116

powered computers and the increasing need for automated surveillance systems and is widely used for applications such as human-machine interface, vehicle navigation, automated surveillance, motion-based recognition, video indexing, robotics and traffic monitoring. A large number of those applications need to have reliable tracking methods that meet real-time constraints and are complex and challenging regarding changes of movement of the object, scene illumination, scale and appearance and occlusion. Tracking results can be influenced by variation of one of these parameters. In order to overcome those mentioned challenges and many others in object tracking many approaches have been proposed [1]. In a tracking application, target object can be described as anything that is interesting for analysis. To give examples, person walking on a street, flying vehicle in the air, cars on a road, face in motion, etc. Object tracking approaches can mainly be classified with point, kernel and silhouette tracking techniques. Selecting the right feature plays a crucial role in the object tracking system. One can simplify the tracking problem by imposing on the motion and appearance of object. Mostly all tracking algorithms suppose the object motion to be smooth. Object motion can be further constrained to be of constant velocity or constant acceleration based on priori information. Prior knowledge about the number and the size of the objects can also be utilized to reduce the complexity of the problem. Some algorithms assume prior knowledge about the object appearance or shape, texture, color, etc. Many tracking algorithms generally use a combination of these features and have been developed and proposed. For example, KLT, the kalman filter [2], meanshift [3,4] and Camshift [5].

The rest of this paper is organized as follows: in section two, we briefly describe the principle methods of commanding and retrieving information from AR.Drone. In section three object tracking system is presented. Section four summarizes our work and draw outlines of prospective further improvements.

2. Hardware Platform

AR.Drone is a commercially available quadrotor which has been increasingly used in education and research area due to its low cost, robustness to crashes, safety and reliability for both indoor and outdoor applications. It is constructed of plastics, carbon fiber, battery, four BLCD motors, equipped with 6 degree of freedom inertial measurement unit, 3 axis gyroscope and accelerometer, control board with ultrasonic sensors and two cameras. Users can directly set the yaw, pitch, roll, vertical speed and control board can adjust return motor speed to match state requirements. It can achieve speed of more than 5 m/s for a continuous flight of 15 minutes.

Fig. 1. AR. Drone

AR.Drone is controlled by ARM Cortex A8 processor whose CPU clock speed has 1GHz with 1GB DDR2 RAM at 200MHz. It provides a console via ad-hoc wireless network to control the AR.Drone using Ipad/Iphone or Android devices. Thanks to the supplied open-source Software Development Kid, several control parameters of flight can be set via its Application Programming Interface which also provides access to the data of sensors and images from the cameras[6,7].

3. Object Tracking System

The goal of an object tracking system is simply to know where the object constantly is in the image. Object tracking is composed of two steps: firstly tracked object is selected and afterwards Camshift algorithm is processed.

3.1. Camshift Tracking Algorithm

The principle of the Camshift algorithm is based on the principle of the algorithm Meanshift. While Meanshift algorithm is employed for static distributions, Camshift algorithm is employed for dynamic distributions. This difference allows the Camshift algorithm to compute distributions again for each frame that makes it beneficial in video as this allows the algorithm to anticipate object movement to constantly track an object between frames. Thus, Camshift algorithm is considered effective in tracking moving objects. It is based on the color characteristics. For the reason that RGB color space is more sensitive to changes in illumination brightness, Camshift algorithm transforms images from the RGB color space to HSV color space to reduce the effect of changes in light intensity on the tracking performance.

Camshift Algorithm works as follow:

1. Firstly, select the location in the search window that is to be tracked. This is meanshift search window. It sets the hue component from HSV color space of the object that is going to be tracked. This hue can also set using a color wheel or a similar color selection method.
2. Calculate the probability distribution of the selected area centered at the meanshift window. This is represented as a histogram of colors that represents the object. The amount of pixels that have hue component in selected region can be seen by means of the histogram.
3. Iterate the meanshift to find the centroid of the probability image. This point is used as the zeroth moment. That will be the new center point of the search window.
4. For the next video frame, center the search window at the new centroid and then go to step two to repeat the process [8].

I(x,y) is the intensity of the discrete probability image at the point (x,y) within search window and x and range over the search window. The zeroth moment is found by equation (1) as follows:

$$M_{00} = \sum_x \sum_y I(x, y) \tag{1}$$

The first order moments are calculated by equation (2) and the centre of mass is given by equation (3).

$$M_{10} = \sum_x \sum_y x \times I(x, y),\ M_{01} = \sum_x \sum_y y \times I(x, y) \tag{2}$$

$$x_c = \frac{M_{10}}{M_{00}},\ y_c = \frac{M_{01}}{M_{00}} \tag{3}$$

Equations (4), (5) and (6) describe the second and first moments.

$$M_{20} = \sum_x \sum_y x^2 \times I(x, y) \tag{4}$$

$$M_{02} = \sum_x \sum_y y^2 \times I(x, y) \tag{5}$$

$$M_{11} = \sum_x \sum_y x \times y \times I(x, y) \tag{6}$$

The object orientation is defined by the equation (7).

$$\theta = \frac{\tan^{-1}\left(\dfrac{2\times(\dfrac{M_{11}}{M_{00}} - x_c \times y_c)}{(\dfrac{M_{20}}{M_{00}} - x_c^2) - (\dfrac{M_{02}}{M_{00}} - y_c^2)}\right)}{2} \tag{7}$$

The parameters to calculate length and width of the probability distribution are calculated in closed form in the equations (8), (9) and (10). The length and width of the distribution around the centroid of the object are then given by equations (11) and (12) respectively [9].

$$a = \frac{M_{20}}{M_{00}} - x_c^2 \tag{8}$$

$$b = 2\times(\frac{M_{11}}{M_{00}} - x_c \times y_c) \tag{9}$$

$$c = \frac{M_{02}}{M_{00}} - y_c^2 \tag{10}$$

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}} \tag{11}$$

$$w = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}} \tag{12}$$

4. Experimental Results

Tracking application was organized as shown in Figure 2.

Fig. 2. Tracking System

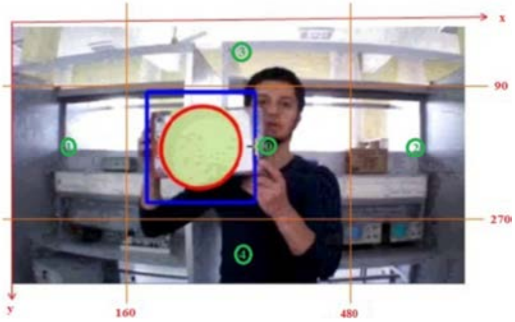

Fig. 3.The method for finding the region of the object in every frame

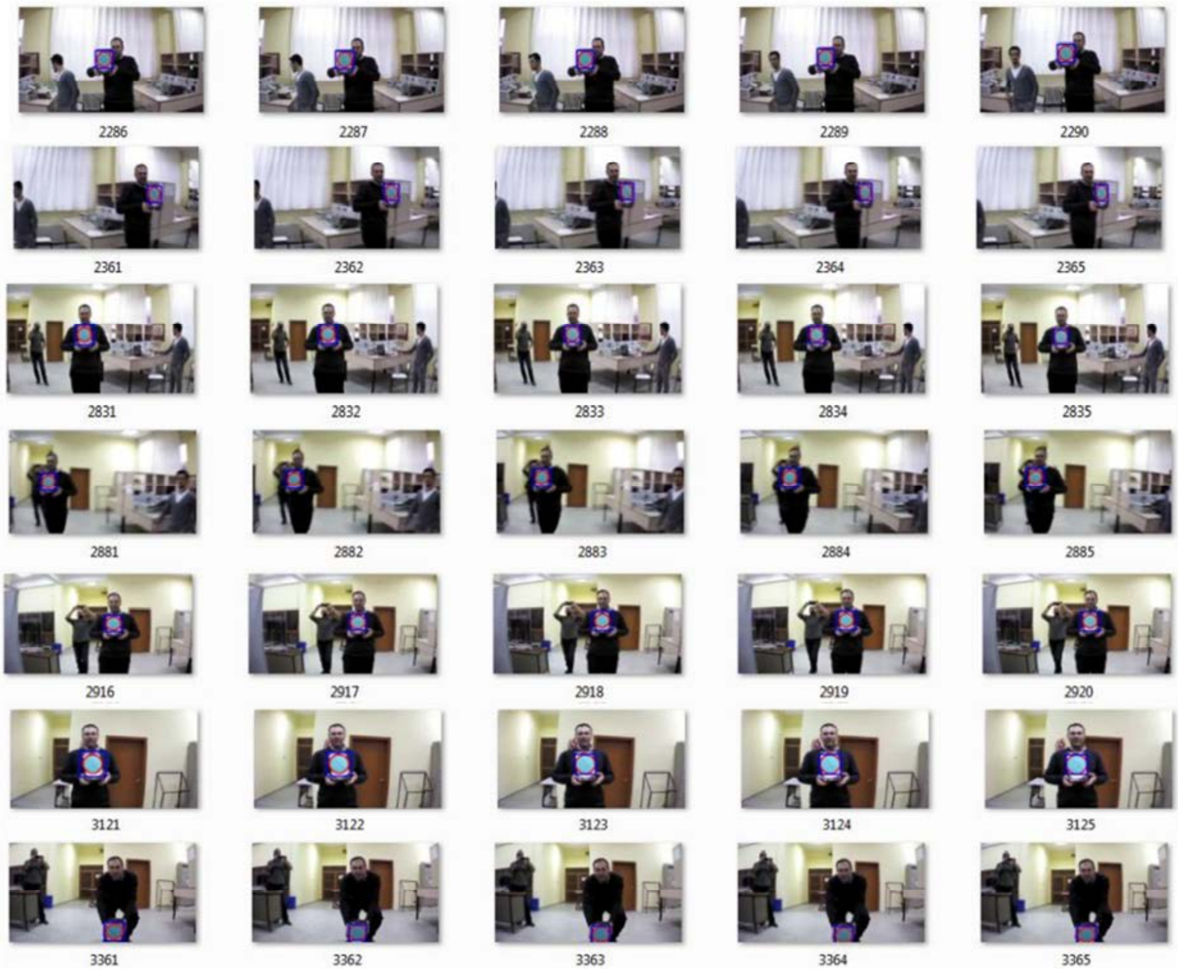

Fig. 4.The result of our tracking system from given frames

It begins with selecting the tracked object in the scene. To detect exact position of the tracked object, a limitation for search area was considered as shown in Figure 3. Camshift algorithm then continues to track the object around the centre coordinates of selected area. If the object's centre is out of the search area, then AR.Drone commences to change its rotor's speed according to the object's position. Let us assume that the object moves to 1 region in the Figure.3. The AR.Drone has to turn right in yaw movement to centre the object in 0 region. This process goes on until the process is stopped by base station or in case of low battery. After the tracking application gets started all the frames are saved and numbered in a folder meantime. It can be seen in Figure 4 that some frames from 2286 to 3365 are proof of that the AR.Drone tracks the object in six directions which are forward, back, right, left, up and down.

5. Conclusion and Future Work

In this paper, just a test of tracking a moving object was successfully fulfilled in the laboratory. A traditional Camshift approach was employed to track the object. Since the AR.Drone is a mobile platform, its speed can be different than the tracked objects. Thus there may occur some challenges. The AR.Drone can track the object in partial occlusion in this implementation. However it can fail if it comes up with a full occlusion. To solve this problem and improve the tracking system, a robust approach that can overcome those challenges mentioned will be applied in our further studies.

References

[1] Yilmaz A, Javed O, Shah M. Object Tracking: A Survey, ACM Comput. Surv., 2006; 38(4):13.
[2] Sun X, Yao H, Zhang S. A Refined Particle Filter Method for Contour Tracking, Visual Communications and Image Processing , 2010; 8:1-8.
[3] Chen D, Bai F, Li B, Wang T. BEST: A Real-time Tracking Method for Scout Robot, IEEE/RSJ International Conference on Intelligent Robots and Systems 2009; 6:1001-1006.
[4] Ido L, Micheal L, Ehud R. Mean Shift tracking with multiple reference color histograms, Computer Vision and Image Understanding 2010; 9:400-408.
[5] Chen X, Li X, Wu H, Qiu T. Real- time Object Tracking via Camshift-Based Robust Framework, IEEE Int. Conference on Information Science and Technology, Wuhan, Hubei, China; March 23-24, 2012.
[6] Dan C, Pham H, Pham T, Truong N. Vision Based Ground Object Tracking Using AR.Drone Quadrotor, ICCAIS, 2013.
[7] AR.Drone Developer Guide SDK 2.0.
[8] Gary RBm Computer Vision Face Tracking For Use in a Perceptual User Interface, Intel Technology Journal, http://developer.intel.com/technology/itj/q21998/articles/art_2.htm, Q2, 1998.
[9] Salhi A. Yengui Jammoussi A., Object tracking system using Camshift, Meanshift and Kalman Filter, World Academy of Science, Engineering and Technology, Vol:6, 2012.