PROJECT REPORT

TOPIC: AUTONOMOUS PEOPLE TRACKING USING A PARROT AR-DRONE 2.0

SUBMITTED BY:

PRAVEER SINGH

AAKANKSHA RANA

SUPERVISORS:

Prof. Jean Luc Dugelay

Prof. Ludovic Apvrille

ABSTRACT

We build here a robust and versatile technique for people detection and tracking that can be used for Parrot AR-Drone 2.0. Detection is carried out by an adaptive Histogram of Gradients (HOG) feature. For tracking we use an adaptive Rao-Blackwellised particle filter. All the commands to the drone are sent through the basic ROS hydro library.

INTRODUCTION

People detection and Tracking has long been a key topic of research in the field of computer vision and video surveillance. With the advent of high camera resolutions and recent vision techniques, research related to object detection and tracking, especially of humans has gained high momentum in the past decade. There have been couple of works on real time tracking but most of them employ algorithms that run on recorded video sets. Here in our case, we try to implement first a detection and then a tracking algorithm in real time for people detection and tracking with a parrot AR-Drone. Since we need to detect and track people and at the same instance send commands to the AR-Drone to control its movement, there is a requirement to set a trade off between robustness of the algorithm and its computational complexity. Robustness so that it can adhere to any climatic conditions (including winds when in motion outside) and in dull as well as bright light. Computation complexity is desired so that the drone reacts as quickly as possible while tracking and following a person. Taking all the above factors into account, we have developed a technique for efficient movement of the drone while following a person.

TECHNIQUE

The entire process mainly comprises of two parts. First step involves detection of people which is quite well addressed by the work of Histogram of Oriented Gradients (HOG) for human detection, a method well known in the Computer Vision community, proposed by Navneet Dalal and Bill Triggs. Classification is carried out with the help of a linear SVM classifier. For the second step of tracking, there are broadly two most commonly used techniques namely the kalman filter and the particle filter. In our method we use the latter version. The advantage of particle filter over kalman is in terms of there incredible versatality and robustness. Particle filters can model any probability distribution - continuous or discrete. Against this, Kalman filters - while orders of magnitude faster - only work with approximate gaussians under approximate linearity. Our tracking algorithm is an implementation of the paper “Object tracking with adaptive HOG detector and adaptive Rao-Blackwellised particle filter” by Stefano Rosa and all which we have modified in accordance to a moving platform like Drone. The ROS hydro library was used for sending and receiving commands from the AR-Drone. Tracking was successfully implemented in both indoor and outdoor conditions as well as with and without occlusion.

Histogram Of Oriented Gradients (HOG)

HOG is a feature descriptor that has mainly been used for object detection in the field of image processing. It basically involves counting the number of occurrences of different orientations of gradients inside a bounding box fixed by us and then rounding it to the correct bin of the histogram. The image is divided into blocks and then further into small cells for which the histogram of gradients is computed. Finally the concatenation of all resulting histograms leads to the descriptor for the entire image.

Linear Support Vector Machines (SVM) Classifier

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outlier detection. The advantages of support vector machines are basically that it is effective in high dimensional spaces, uses a subset of training points in the decision function (called support vectors), so it is also memory efficient and lastly it is versatile with different Kernel functions, though in our case we use a linear function.

Particle Filtering:

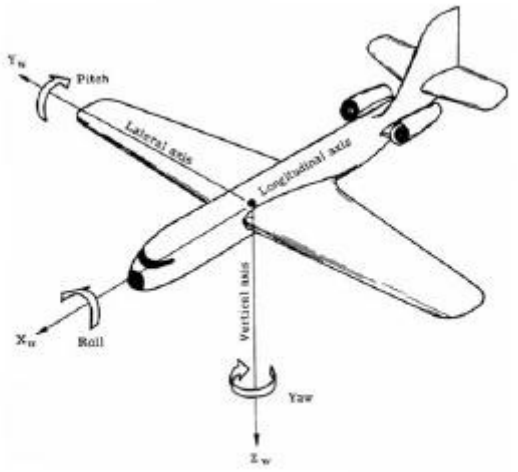
One way to reduce the computational cost of the problem is to reduce the number of samples that we analyze. Particle filtering essentially combines the particles at a particular position into a single particle, giving that particle a weight to reflect the number of particles that were combined to form it. This eliminates the need to perform redundant computations without skewing the probability distribution. Particle filtering accomplishes this by sampling the system to create N particles, then comparing the samples with each other to generate an importance weight. After normalizing the weights, it resamples N particles from the system using these weights. This process greatly reduces the number of particles that must be sampled, making the system much less computationally intensive.

Rao-Blackwellised Particle Filtering:

An additional way to improve our computational efficiency is to reduce the complexity of each sample. If we sample 2 variables instead of 3, then we will reduce the number of dimensions in the system. RBPF reduces the number of variables that must be sampled by identifying variables that do not need to be sampled to be computed. In the case of the model proposed above, RBPF would marginalize out P and V because they can be determined by sampling A. This optimization enables RBPF to perform significantly better than simple PF for a particular computational cost. The more that a network can be broken down into subnetworks, the greater performance increase will be.

AR-DRONE CONTROL

Before moving onwards, we would first like to highlight the various degree of freedom for the drone. The drone all in all has 6 degrees of freedom about the x, y and z axis which can be clearly be seen in the figure below.



There are namely three kinds of motion that the drone follows around the x, y and z axis which are Roll, Pitch and Yaw respectively.

- Roll -> drone left-right tilt. Floating point value in range of -1 to1.
- Pitch-> drone front-back tilt. Floating point value in range of -1 to1.
- Yaw -> drone angular speed. Floating point value in range of -1 to1.

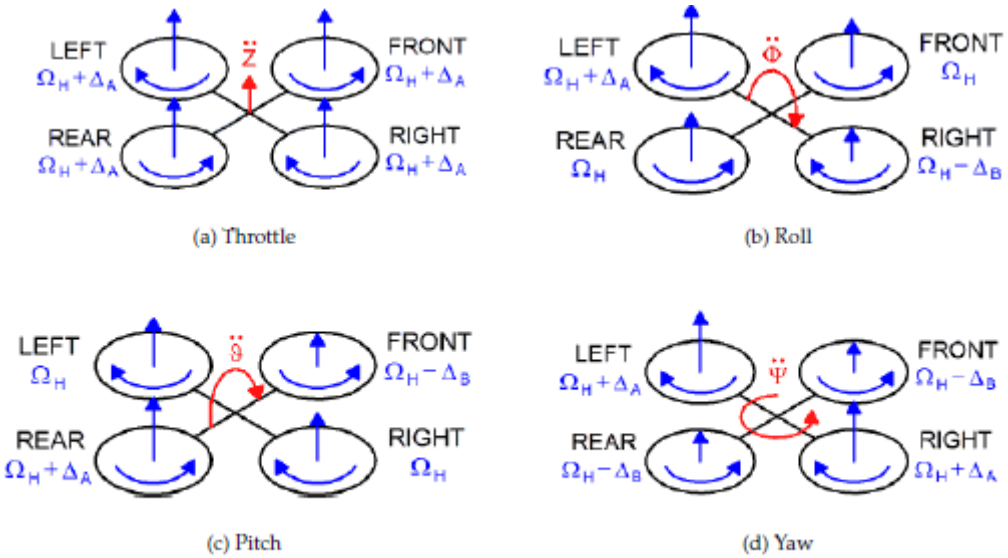


Figure 2.1: Drone movements

Roll argument values

Roll argument value sets the percentage of the maximum inclination as configured in the drone parameters for drone's left-right tilt.

The direction is determined by the sign of the number:

- Negative percentage value will make the drone tilt to its left, thus flying leftward;
- Positive percentage value will make the drone tilt to its right, thus flying rightward;

How to represent the value:

- Convert the float number [-1, 1] to hexadecimal signed value. The explanation for the conversion will be given in future posts;
- Convert back from hex to signed 32-bit decimal number;

The following values are mostly used:

- -1082130432: The minimum converted value -1 will make the drone fly leftward with full power;
- -1086324736: The converted value -0.75 will make the drone fly leftwards with 3/4 power;
- -1090519040: The converted value -0.5 will make the drone fly leftwards with half power;
- -1098907648: The converted value -0.25 will make the drone fly leftwards with 1/4 power;
- 0: This value will make the drone fly horizontally;
- 1048576000: The converted value 0.25 will make the drone fly rightwards with 1/4 power;
- 1056964608: The converted value 0.5 will make the drone fly rightwards with half power;
- 1061158912: The converted value 0.75 will make the drone fly rightwards with 3/4 power;
- 1065353216: The maximum converted value 1 will make the drone fly rightward with full power;

Pitch argument values

Pitch argument value sets the percentage of the maximum inclination as configured in the drone parameters for drone's front-back tilt.

The direction is determined by the sign of the number:

- Negative percentage value will make the drone lower its nose, thus flying frontwards;
- Positive percentage value will make the drone raise its nose, thus flying backwards;

How to represent the value:

- Convert the float number [-1, 1] to hexadecimal signed value. The explanation for the conversion will be given in future posts;
- Convert back from hex to signed 32-bit decimal number;

The following values are mostly used:

- -1082130432: The minimum converted value -1 will make the drone fly frontwards with full power;
- -1086324736: The converted value -0.75 will make the drone fly frontwards with 3/4 power;
- -1090519040: The converted value -0.5 will make the drone fly frontwards with half power;
- -1098907648: The converted value -0.25 will make the drone fly frontwards with 1/4 power;
- 0: This value will make the drone fly horizontally;
- 1048576000: The converted value 0.25 will make the drone fly backwards with 1/4 power;
- 1056964608: The converted value 0.5 will make the drone fly backwards with half power;
- 1061158912: The converted value 0.75 will make the drone fly backwards with 3/4 power;

Yaw argument values

Yaw argument value sets the percentage of the maximum angular speed as configured in the drone parameters for drone's right-left spin.

The direction is determined by the sign of the number:

- Negative percentage value will make the drone spin left;
- Positive percentage value will make the drone spin right;

How to represent the value:

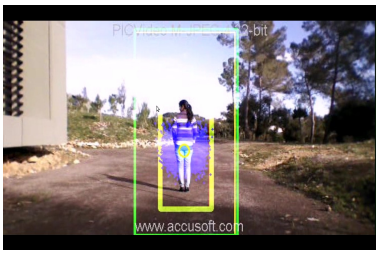
- Convert the float number [-1, 1] to hexadecimal signed value. The explanation for the conversion will be given in future posts;
- Convert back from hex to signed 32-bit decimal number;

The following values are mostly used:

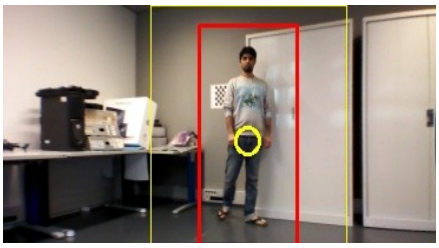
- -1082130432: The minimum converted value -1 will make the drone spin left with full power;
- -1086324736: The converted value -0.75 will make the drone spin left with 3/4 power;
- -1090519040: The converted value -0.5 will make the drone spin left with half power;
- -1098907648: The converted value -0.25 will make the drone spin left with 1/4 power;
- 0: This value will make the drone stay at the same position horizontally;
- 1048576000: The converted value 0.25 will make the drone spin right with 1/4 power;
- 1056964608: The converted value 0.5 will make the drone spin right with half power;
- 1061158912: The converted value 0.75 will make the drone spin right with 3/4 power;
- 1065353216: The maximum converted value 1 will make the drone spin right with full power;

Results and Discussions

outdoor



indoor



Our method has been tested on both indoors and outdoors with all possible kinds of illumination and occlusion effects.

With the original video frames coming from the drones, the algorithm implementation is quite slow as the HOG detection window is dependent on the image size. Hence in order to improve the computation performance, the image sizes were re-sized to an appropriate width and height over which we apply our algorithm. In order to improve the detection or its computation, one can play with the adjust the parameters to the function “detectMulti-scale” and accordingly set the desired values. However a trade needs to be set between the computational performance of the algorithm and the detection accuracy.

Indoor

Due to the limitation of the drone to navigate inside the lab room because of the small room size, only the yaw motion of the drone was focussed upon and the pitch was set to very low (0.05 or 0) while testing indoor. Under the lab room luminations the drone was able to track the person quite well and make the angular motions accordingly. However due to bad limunations sometimes it falsely tracks pillars or black CPU's which needs to be taken care of. Also it was even noticed that there is a problem of horizontal drift during the yaw motion of the drone, hence for precise movement, this needs to be nullified in the future work. Altitude of the drone was set to a constant height of 1 meter.

The angular motion of the drone is based upon the difference between the position of the tracking circle (its x coordinates) with the centre of the image (called the offsetpixels). If this difference is greater than a certain threshold, the drone set an angular motion with regards to the angle delta which is the arc tangent of the offset pixel and the distance (focal length) of the image screen from the camera's centre of focus.

Outdoor

While testing for the motion of drone Outside, we have mainly checked for three specific tasks. Firstly how the drone tracks a person moving straight without any deviation. Secondly how the drone reacts to 90 degree turns on corners. And thirdly what is the effect of occlusion on the tracking of the drone. During the entire process the drone is prone to drastic climatic conditions like winds, rains as a future work the effect of which needs to be nullified.

Straight motion and adjusting the forward speed

During the straight motion it was observed that the drone loses track of the person, in case the person is moving too fast and moves much faster relative to the drone. This limitation is mainly due to the resizing of the window size, due to which the tracking is not quite optimal for larger distances between the drone and the person. Hence one of the limitation of the current algorithm is that the drone cannot track a very fast moving person. (We cannot resolve this limitation since a trade off has to be maintained between the computational complexity and the tracking accuracy, so in other words if you increase the tracking accuracy, you increase the computation time due to which the reaction time of the drone reduces).

Here it would be important to note that the forward speed of the drone has been set in accordance with ratio of the height of the detection window to the height of the image. As the distance between the person and the drone increases, this ratio decreases and beyond a certain threshold the drone is ordered to move forward with a constant speed (0.1). The drone is asked to stop (forward speed = 0) in case it finds no detection window or it gets too close to the person. As a future work, the drone can be asked to make a 360 degree rotation in case it doesnt finds a person for more than a couple of seconds.

90 degree sharp turns over the corners

The drone tracks and follows the person quite well around the corners again with a certain drift while it is making a turn. There is also some effect of the wind outside, which needs to be countered as a future work.

Occlusion effect

The drone deals quite effectively with the occlusions from other persons. In times of occlusion, it doesnt looses the track of the person it had been following, and once the people who had been occluding go out of the scene, it tracks back the person and keeps following it. However sometimes, if the occlusion by a person is for quite some time, it starts to follow the person (the one who is occluding). This problem needs to be resolved in the future work.

Conclusion

We propose here a real time People detection in 2-D images in AR-Drones. The implementation is mainly composed of an Adaptive Histogram of Oriented Gradients (HOG) feature, a linear SVM classifier and an adaptive Rao Blackwellised Particle filter. The drone was tested in all the extreme conditions both indoor and outdoor with an without occlusion and with sharp turns. Experiments have showed that we could achieve a hgh degree of accuracy for detection and tracking and at the same time obtain quite good response time from the drone as well.

References

- **Stefano Rosa ; Marco Paleari ; Paolo Ariano ; Basilio Bona; Object tracking with adaptive HOG detector and adaptive Rao-Blackwellised particle filter. Proc. SPIE 8301, Intelligent Robots and Computer Vision XXIX: Algorithms and Techniques, 83010W (January 23, 2012); doi:10.1117/12.911991.**
- **Dalal, N.; Triggs, B., "Histograms of oriented gradients for human detection," *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* , vol.1, no., pp.886,893 vol. 1, 25-25 June 2005**
- **http://wiki.ros.org/ardrone_autonomy**
- **Benjamin Ranft, Jean-Luc Dugelay, Ludovic Apvrille, "3D Perception for Autonomous Navigation of a Low-Cost MAV using Minimal Landmarks", Proceedings of the International Micro Air Vehicle Conference and Flight Competition (IMAV'2013), Toulouse, France, 17-20 Sept. 2013.**