Vision Based object detection and avoidance

A. González[1] , Á. López[1]
Supervised by R. Leconte[1], V. Lecrubier[2] and J. B. Chaudron[1]
[1]ISAE [2] Sterblue

Abstract **– The present project aims to research into the potential development of Unmanned Aerial Vehicles autonomous capabilities. Particularly, the project focuses on an algorithm for obstacle detection and avoidance, obtaining the data by means of a stereo camera, able to compute a three-dimensional map of the surroundings of the UAV.**
The system is meant to work in real time and on board the vehicle. High performance will be a critical aspect of the project due to this latter features. In order to achieve the specified targets, the embedded device in charge of all the data processing will be a General Purpose Graphic Processing Unit.
Several algorithms have been developed to create an autonomous obstacle avoidance system. This software is thought to be operate in connection with a ROS environment so that multiple data and devices can be interconnected. To test its performance, various tests were performed, both in theoretical and experimental environments. The results were satisfactory, obtaining a performance up to 60 FPS; although some possible improvements were detected and thus, listed for pending future work.

1. Introduction

Nowadays the field of Unmanned Aerial Vehicles is experiencing an important development not only in the cutting-edge devices but also in the way they are used. The performance of these devices is a current object of study as the final goal is always to make this vehicles a hundred per cent automatic and reliable.
Many enterprises have taken advantage of the current situation in the automatic vehicles field and have started several ambitious projects. One of these enterprises is Sterblue, a French start-up, which aims to carry out high performance inspections in power lines and wind turbines, not only offshore but also onshore. To do so, automatic systems are used as not only does the quality but also the efficiency of the services are noticeably improved. For example, the current price of an inspection offshore would be around 800 euros and with this type of system only two or three inspections per day would be possible. In contrast, the objective of Sterblue is to fully develop an entire innovative automatic aerial system of inspection, so that the price could be reduced to 400 euros and the number of inspections per day could be increased up to 16.

2. State-of-art

Some elements play a very important role within this project. This is why some research has been done into them. Firstly, a Stereo Camera is necessary thanks to their capacity to detect depth. This computation is carried out by computing the disparity maps of the video stream and triangulating the position of the obstacle points in the three-dimensional space. The two tested models are Intel RealSense R200 and Zed Stereo Camera.
Secondly, due to the large amount of computations that need to be done in addition to the high performance desired, a powerful processing unit is needed. GPGPU (General Purpose Graphic Processing Unit) present truly significant characteristics regarding the parallel processing that could significantly improve the computational performance. Plus, to deal with GPU environment and configure it, we make use of the CUDA (Compute Unified Device Architecture) [1] library, developed by Nvidia.
Last, but not least, some existing algorithms are interesting for the purposes of the present project. In terms of the trajectory computation, there is a large variety of reliable algorithms [2]. After studying their possible implementation, an algorithm based on Covariant Hamiltonian Open Motion Planning (CHOMP) was developed [3].
Moreover, we went deep into techniques and libraries as well as possible programs that could be used all along the project like OpenCV [4] or Point Cloud library. ROS [5] (Robotic Operative System) is the software allowing the interactions between all the algorithms and their appliance to some physical devices, like drones or robots.

3. Obstacle avoidance

The present report focuses on the development of the first objective: obstacle detection and avoidance. The algorithm for obstacle avoidance is based on Covariant Hamiltonian Open Motion Planning algorithm [3]. The fundamental idea of the algorithm is that the optimum trajectory will be following the minimum cost. In the version developed for this project, the cost factors taken into account are: the distance to the obstacle and the curvature of the trajectory (so that the trajectory has not really steep curves and the accelerations of the drone are too strong). The steps of the avoidance algorithm will be: voxelization (discretization of the space), obstacle avoidance cost function, computation of gradients regarding this function and finally the gradient descent optimization loop (including the computation of the discretized curvature and its gradient).
The voxelization is a process of discretization of space in order to reduce the size of the computations. The basic idea is that the points in a region in space can be reduced to a single point, that is a single value, situated in the center of a box that occupies that region. The result is a matrix that

distinguishes the regions with obstacles and without them.
The obstacle cost function's goal is to assign a value to each element of the three-dimensional matrix that represents how far is this voxel from the closest object. For that, the algorithm will compute the distances from a voxel to the rest of voxels containing an obstacle and will take the closest one. Then, voxels' values will be compared with their neighbours' to check the minimum values and increase the value of the "worse options" voxels. The convergence is reached when, for all voxels, the previous and current minimum value is the same: all the voxels have the values corresponding to the minimum distance to the closest object. Afterwards, a safety factor (distance drone-object) is defined to exclude some areas around the obstacle.
Finally, the optimization loop searches for the minimums in the cost function. Being initialized with the previously computed trajectory from the current position to the target, trajectory way-points will be displaced during the optimization taking into consideration both the obstacle gradient and curvature gradient and checking if the new position is a better match. Local curve is also evaluated in comparison with the neighbour way-points. In terms of the convergence criteria, a threshold value is set to represent the magnitude under which the gradient must be in order to be considered small enough to be close enough to the function minimum. If this value is reached, the optimization is considered to be converged.

4. Results

Several tests were carried out, both in theoretical and experimental environments. The static tests of the system were performed in a three-dimensional space, with more than satisfactory results. On the other hand, in picture 1 the results of a moving surface vehicle test (in 2D) are displayed. The tests demonstrated that the system performance meets the requirements, the speed that the algorithm reaches is 60 Hz. Furthermore, the tests also demonstrated the possible improvements that the system could have, explained later in section *Future Work*.

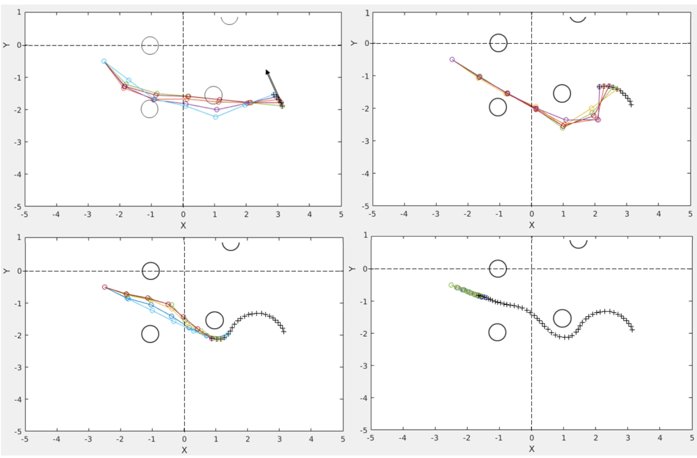

Figure 1: Results from experimental test performed with the avoidance algorithm.

5. Conclusions

First of all, after explaining the algorithm, it is clear that the process is computationally expensive, but easily parallelizable. Then, it is truly necessary to have a powerful parallel processing unit available. With the GPU all these computations can be handled in parallel, making it possible to obtain good results and a good performance.

Regarding the results, the trajectories have proven to be smooth and perfectly avoid the objects in the conditions tested. However much more testing would be necessary to assure the proper functioning of the algorithm under every condition.

6. Future work

The future guidelines for this project lead to tests in three dimensions, with aerial vehicles.
Moreover, mapping algorithms [6] are thought to be be implemented in CUDA as their utility is undeniable. This will allow the system to fully be aware of the environment and not to "forget" past obstacles. On the other hand, a deeper study in parallel computing could possibly improve the performance of the algorithm and optimize its capabilities. Other aspects could also be improved in the present project. For example, the convergence criteria could be better optimized. Plus, the robustness of the algorithm could be improved by an enhancement of the obstacle cost function with smarter, more sophisticated algorithms that may take into account every possible obstacle configuration and guarantee a collision free trajectory.

7. References

[1] C. Nvidia, “Nvidia cuda c programming guide,” *Nvidia Corporation*, vol. 120, no. 18, p. 8, 2011.

[2] M. Nieuwenhuisen and S. Behnke, “3d planning and trajectory optimization for real-time generation of smooth mav trajectories,” in *Mobile Robots (ECMR), 2015 European Conference on*, pp. 1–7, IEEE, 2015.

[3] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, “Chomp: Gradient optimization techniques for efficient motion planning,” in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 489–494, IEEE, 2009.

[4] “Opencv.” http://opencv.org/. Accessed: 01-2017.

[5] “Ros wiki.” http://wiki.ros.org/ROS. Accessed: 03-2018.

[6] J. Engel, T. Schöps, and D. Cremers, “Lsd-slam: Large-scale direct monocular slam,” in *European Conference on Computer Vision*, pp. 834–849, Springer, 2014.