

The software, including motion planning, visual perception and virtual simulation, are developed on the Robot Operating System (ROS) environment, and we use Gazebo for performing simulations¹. We use the Gazebo simulator for testing and planning our strategy and for customizing our hardware and software. The visual perception component carries out target (heliport) localization of the moving vehicle, and we plan the efficient approaching and landing strategies based on the motion of both the UAV and the vehicle. Since we use Nvidia TX1 embedded processor for fast computations on GPU, our algorithms for *task 1* and *task 3* are developed in CUDA-C, C/C++ and Python.

0.2 General Approach

We use the heliport model to train a linear classifier for detection of the landing region. Since the heliport model is known, it is used as an *a priori* for learning. Once the heliport is detected, a visual object tracker running in real time on-board is autonomously initialized to start tracking the target region. We use a robust tracking algorithm with efficient drift compensation to avoid lost of target when the UAV is in motion. Our visual tracking algorithm is also capable of recovering the target even if it went completely out of view. Once the target is localized, the UAV uses pose information from the visual tracker to navigate towards the target and plan out a landing strategy.

0.3 Results Achieved to Date

In this task, we have completed target detection, tracking and autonomous landing in a Gazebo simulator. Moreover we have also tested our tracking algorithm on an outdoor video sequence of a heliport marker affixed to a vehicle travelling around a track at various speeds. The video was taken from a DJI Phantom 4 and processed offline to test the stability, robustness and speed of tracking.

0.4 Future Plans

The future work on hardware platform for task 1 includes the design of landing gear which will enable the UAV to attach itself onto a moving heliport. A sturdy and light protector for the UAV is also needed to avoid the landing impact with the truck from damaging the UAV. The future work on software for task 1 involves testing the completed software on the customized UAV which is currently under development. This involves fine tuning the current simulator version of our software. Considering the challenges in outdoor environment such as abrupt changes in image space, winds speeds etc. the landing strategy might vary significantly from the simulator version. One very important aspect of autonomous systems which we like to implement is the ability of the UAV to recover from erroneous decision, false positives that might result in highly cluttered and unstructured scene. To achieve this we plan to generate a map of the environment at the beginning for efficiently localizing the vehicle and for trajectory mapping. The idea is that, the UAV can use the constructed map, to eliminate regions that produces false positives. However, the current limitation is the time required for generating the map which we aim to reduce using the idea of task oriented solution.

WOLFGANG