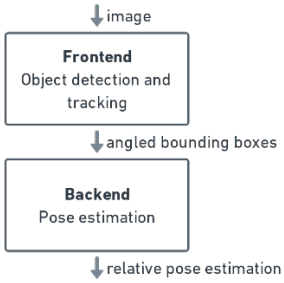
Vision-based Localization Solution for a Team of Quadrotors in Formation

Théo Gieruc

Professor : Alcherio Martinoli
Assistant(s) : Kagan Erünsal

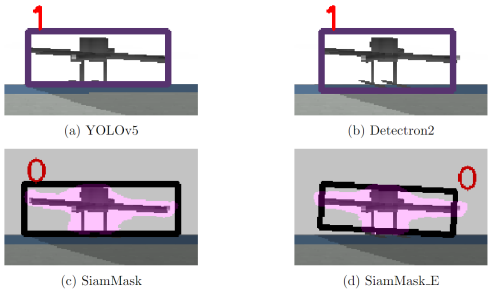
Reliable and accurate relative localization is crucial when performing formation of quadrotors. The main objective of this project is to design a relative position estimator tailored for UAVs using an RGB camera. The solution must be fast, lightweight in order to be computed onboard, wrapped in ROS packages and easy to use. The architecture is divided in two parts: the frontend, which computes the bounding boxes around the detected objects from the camera, and the backend, which estimates the relative position of each object from its bounding box.



Project architecture

The frontend is composed of two object detection libraries, YOLOv5 and Detectron2, one multiple object tracker (MOT), ByteTrack, as well as a single object library (SOT), PySOT. Their modularity allows up to 220 different combinations. For this project, I decided to focus on Detectron2's Faster R-CNN R50-FPN for its inference time, YOLOv5's yolov5m for its compromise between accuracy and speed, PySOT's SiamMask and SiamMaskE for their ability to output angled bounding boxes. Their inference time, memory use as well as bounding box quality has been tested on a RTX 3060 in simulations on Webots 2020a.

Detector	Tracker	Inference time [ms]	Memory use [MiB]
YOLOv5	ByteTrack	43	1949
Detectron2	ByteTrack	120	2205
YOLOv5	SiamMask	52	2373
Detectron2	SiamMask	57	2687
YOLOv5	SiamMaskE	58	2419
Detectron2	SiamMaskE	63	2921



(a) YOLOv5 (b) Detectron2

(c) SiamMask (d) SiamMask_E

YOLOv5 is faster than Detectron2, and PySOT's trackers memory usage increase with the number of tracked objects, but their use smooths out the difference.

MSL-RAPTOR is used as backend. It estimates the relative position from the angled bounding boxes provided by the backend and fuses them through time with an Unscented Kalman Filter (UKF). The tracking accuracy has also been tested in simulations on Webots 2020a. The ground truth and estimation of the position of a quadrotor flying in front of a camera has been compared.

Error	Mean [m]	Std [m]	Mean [%]	Std [%]
Depth	-1.23	0.411	-28.43	5.89
Horizontal	-0.026	0.16	-0.7	3.9
Vertical	0.1	0.11	2.4	2.8

While the horizontal and vertical accuracy is high, the depth estimation performs poorly. This is indeed the core problem of estimating position with an RGB camera. This could be improved by initializing MSL-RAPTOR with a precise 3D model of the object rather than its rough dimensions or by replacing the camera by an RGB-D camera, where the depth measurements would be fused in the UKF.

The frontend is fast, accurate and provides up to 220 different combinations of object detectors and trackers. The backend has good accuracy in the vertical and horizontal dimensions but underperforms in depth estimation. The overall inference speed and memory use makes it tailored for UAVs applications.