

C. Scene Segmentation for the Identification of On-road Obstacles

The saliency map of each frame is used to categorize different regions of the scene. For illustration purposes the regions are visualized in different colors:

- Blue: The safe area of the road beyond the view of the driver.
- Yellow: Be-aware areas representing negative obstacles.
- Cyan: Hazardous areas in the range of the road representing positive obstacles.
- Purple: Dangerous areas outside of the range of the road.
- Red: Recognized obstacles in the range of the road (e.g., potholes).

To define the vehicle's moving direction steering data are used received by internal sensors of the vehicle. The direction of the vehicle specifies which part of the scene in the field of view is in front of the vehicle and is used as a parameter, in addition to saliency mapping, for the segmentation of the point cloud. The more critical regions are the ones that lie within the limits of the road. A segmentation example is illustrated in Fig. 5

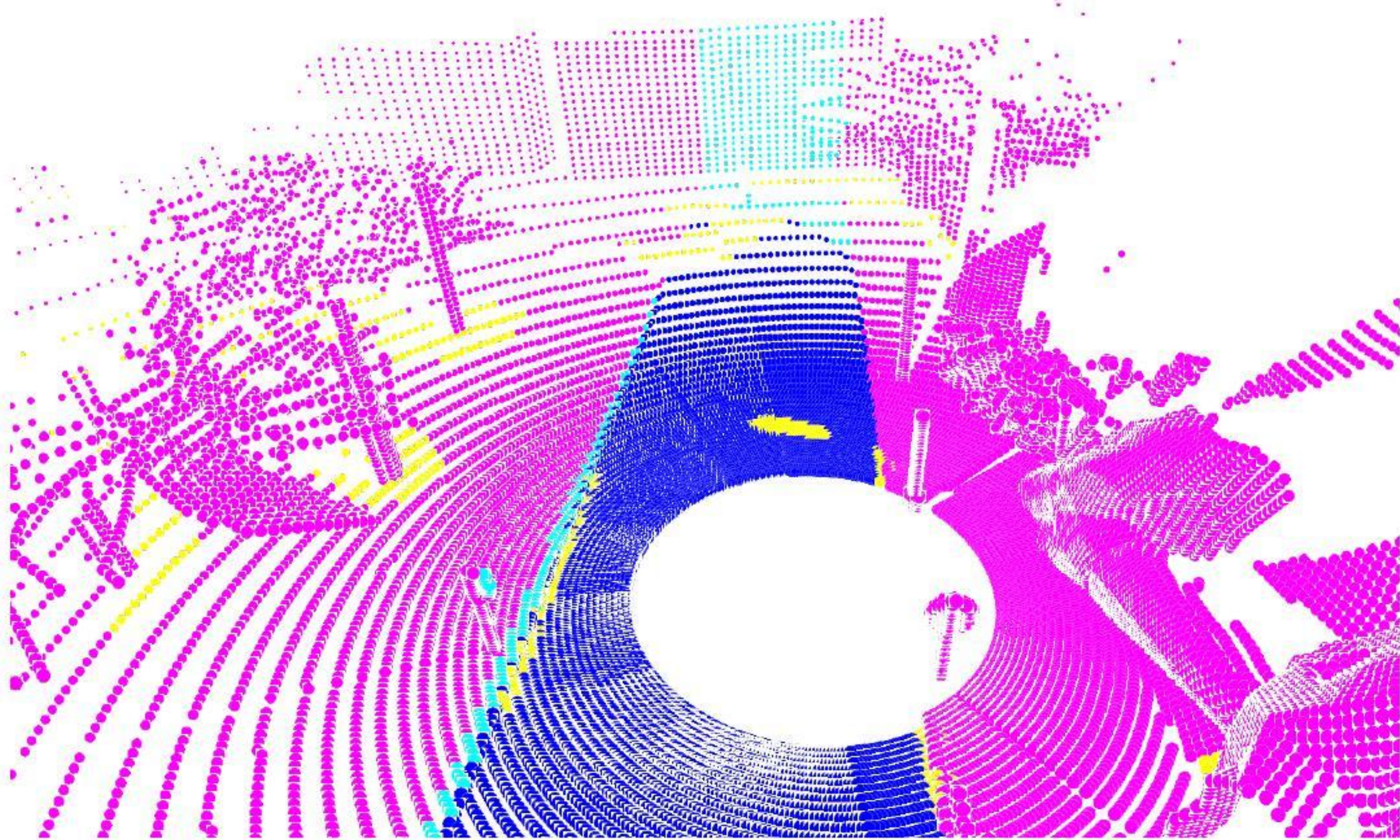


Fig. 5. Segmentation of the point cloud scene based on the saliency map in Fig. 4 and the vehicle's moving direction.

D. Data simulations

For evaluation of our methodology, we created a rich dataset using CARLA, an open-source autonomous driving simulator [16]. CARLA is based on a server-client system, in which the server is responsible for running the simulations including the calculation of physics, weather conditions, collision detection and sensor readings. It operates on the OpenDRIVE specification [58] for defining junctions, traffic lights, etc, and is used by CARLA for simulating independent agents, such as other cars and pedestrians. This makes CARLA ideal for creating complex scenarios and realistic driving conditions for our tests.

The server running the simulations is powered by Unreal Engine. Clients can connect and request changes to almost any element in the world being essential for the creation of scenarios. They also receive sensor data and manage input to the vehicle controlled by the user. CARLA supports a wide

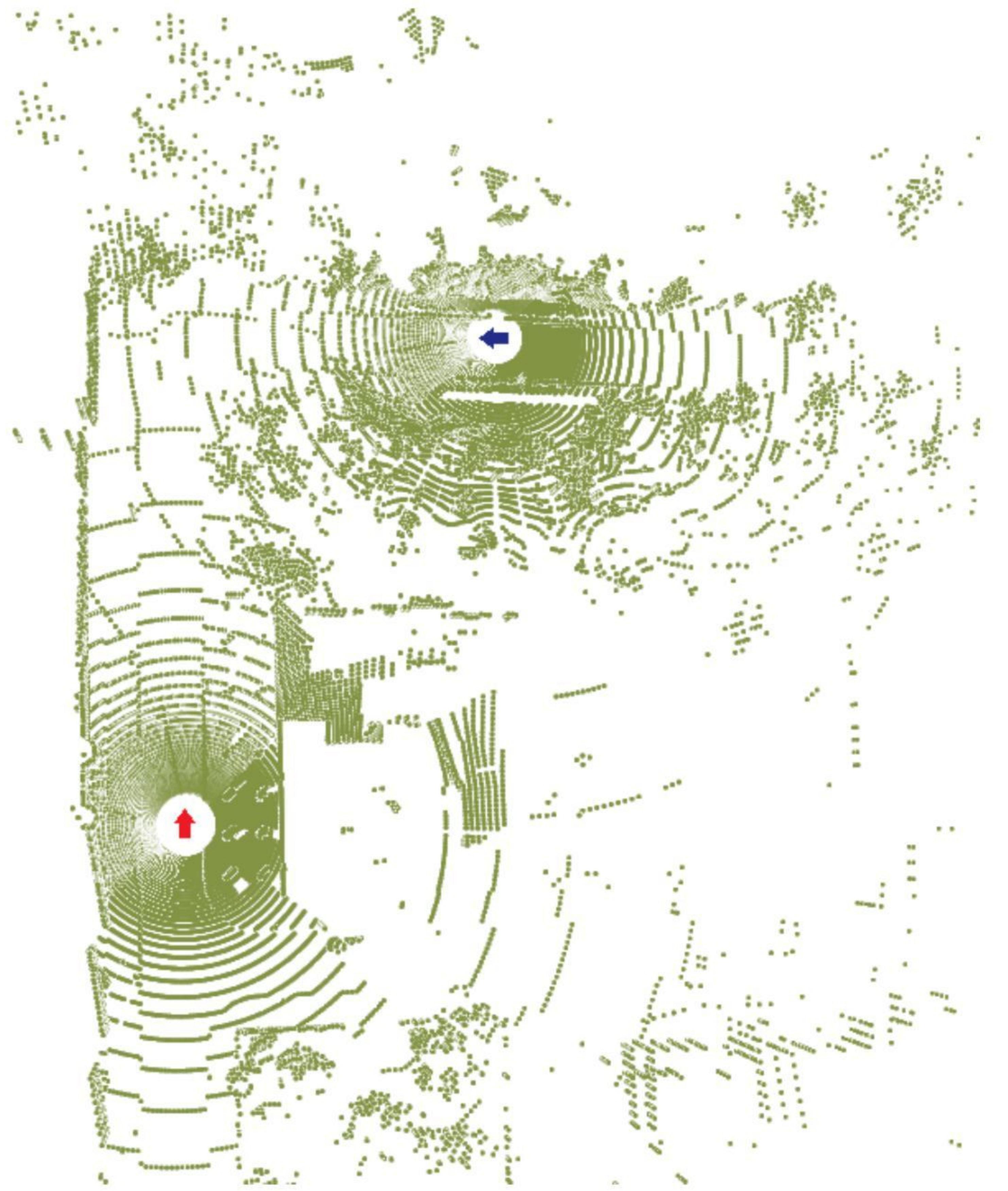


Fig. 6. Point cloud map of both two vehicles (ego1 and ego2).

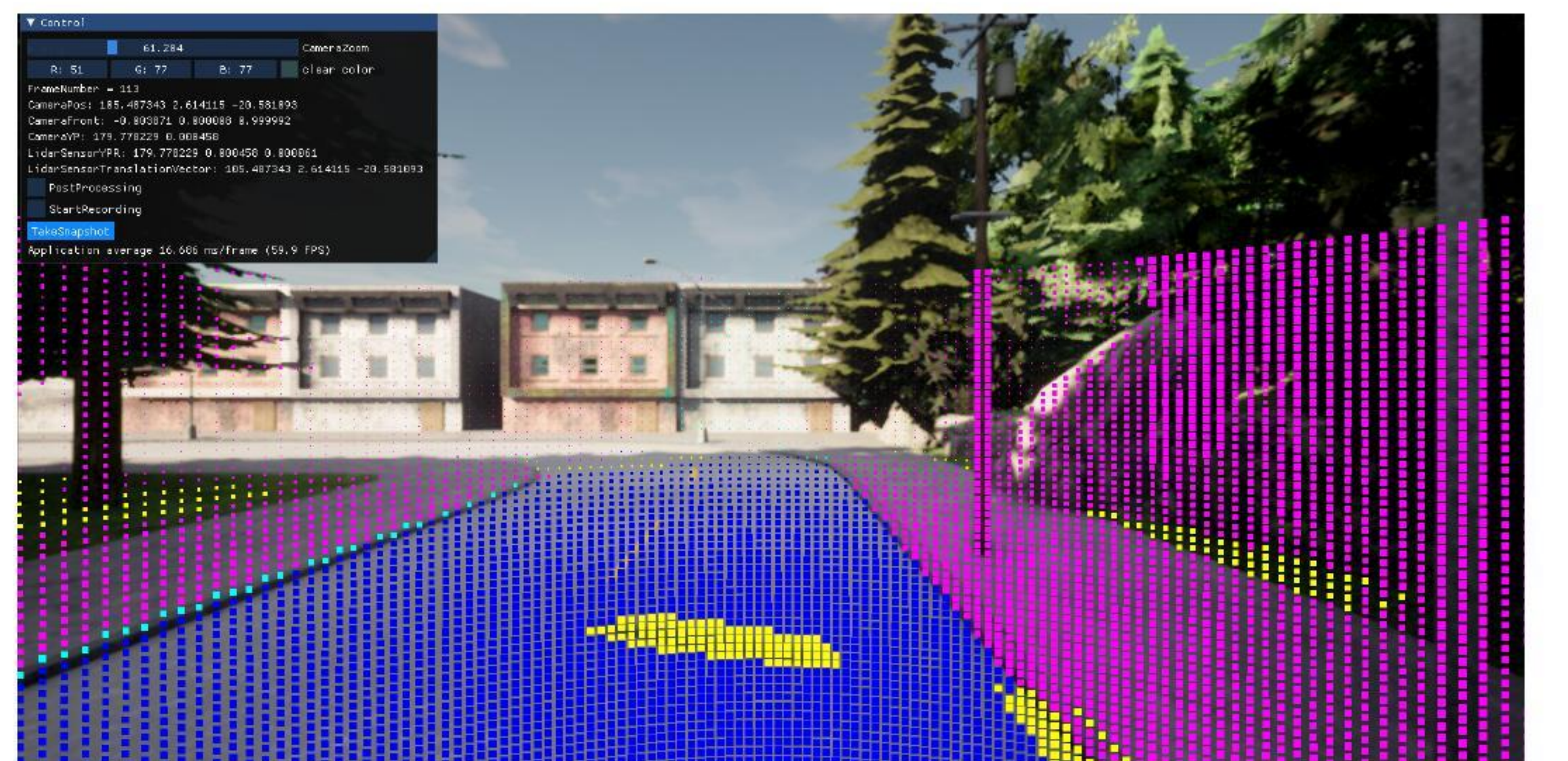


Fig. 7. Example of segmentation of the point cloud projected to the AR interface (in the view of ego1).

range of sensor suites with extensive configurability to its intrinsic parameters. In our work, we use a LiDAR sensor on top of the vehicle and a monocular RGB camera, placed in the front part of the car, for simulated data collection. By placing these sensors in an autonomous car and initiating its navigation in the virtual environment, we were able to create a very large dataset for evaluating our algorithms. In the future, we plan to assess the AR visualization effectiveness, with respect to reaction time and awareness increase, in a real environment with a driver manually controlling a vehicle.

Contributions in CARLA simulator: Due to lack of benchmark point clouds datasets representing real road scenes with obstacles (potholes and bumps), we used the CARLA simulator to create obstacle-free environment data, in which