Otto - Testing Operations

Scott Ryvola - Software Quality Manager

Daily Road Testing

Otto trucks drive the highways surrounding San Francisco on a daily basis. The goal of this testing is to evaluate our (constantly improving) software against real-world driving situations and identify the current system's strengths and weaknesses.

Otto is very deliberate in our testing efforts. We test at all times of the day, morning rush hour traffic, throughout the day, and at night to capture every highway situation imaginable. Trucks are routinely sent back to known problem areas as well as into new conditions on a daily basis. We collect as much information as possible to ensure not only that we will be able to handle anything thrown at us but also to use everything that is thrown at us towards improving the system. This is achieved through advanced machine learning models that our perception system employs. Although not every component of a self-driving vehicle can employ a machine learning model, we use problem scenarios encountered on the road as a baseline of what our system must be able to handle.

In a typical testing scenario, there are two passengers inside the cabin. One is a licensed Commercial Driver responsible for the operation of the truck itself, getting us to the highway, exercising active physical monitoring at all times, and taking over full driving responsibilities when necessary. This driver is constantly monitoring the road and giving input when necessary, which is vital to safe operation of the vehicle. The co-driver in the passenger seat is responsible for monitoring the output of the self driving system and alerting the driver of any anomalies, as well as instructing when the driver should "disengage" from self-driving mode. Drivers can disengage by grabbing the steering wheel, applying the brake, applying throttle, flipping an "engage" button on the dash, or hitting a large red button next to the steering wheel. In an emergency, any one of these inputs instantly restore full driving control to the driver.

There are two ways through which the Otto trucks stay on course and within a marked lane on the freeway. The first is through "seeing" the marked lane lines on the freeway using our perception system to stay in the middle of the lane. The second is through a highly-detailed gps map. These maps are created by sending a truck (driven by one of our truck drivers) along a specified route without engaging the self-driving system; the goal is strictly to collect data about the specific road and lane conditions. Once a map has been generated for a section of highway, our trucks can localize to it the next time they drive that same section of road in self-driving mode. By connecting to the national RTK system, we can localize the truck's exact location on the map to a few centimeters of accuracy. These two systems (perception software and map localization) work in concert with each other and are not mutually exclusive.

Before testing begins on any given day and route, the co-driver will first evaluate the current road conditions and system readiness before calling for an "engage." The driver must

then manually press a button on the dashboard to engage the self-driving system. While the self-driving system is engaged, the driver must be extremely attentive and ready to take back full control whenever necessary. In situations where the driver decides to take back full control, the logging software in the self-driving system remains active and records the corrective action the driver took. Later, our engineers review these logs to obtain an accurate representation of what a human truck driver would do in that particular scenario. At the moment when the driver is taking over full manual control, the co-driver will leave detailed notes describing what was happening on the road at the time, as well as the reason for the particular disengage.

Dashboard and Log Analysis

Once a road-test is complete, we present the collected information to our engineers via a web dashboard that enables them to see the location of disengages, the comments left by the co-driver, and the exact log file that they can download to review and test new software upgrades against (see screenshot below, which displays two weeks' worth of testing).

Disengages are vital to evaluating the performance of our system. Each disengage is reviewed closely after the fact to determine the root cause of the shortcoming and to assign a severity level to each event. All disengages are important and taken seriously, but the various severity levels serve as a signal to our software engineers as to which bugs to prioritize.

Disengage Severities

The following categories are assigned to disengagement events after the logs have been carefully analyzed:

Comfort - a case where drivers felt uncomfortable with the decision the self-driving system made but, in reality, pose a near-zero safety threat to themselves or anyone else on the road. This category exists because we train our drivers to be extremely cautious and defensive.

Public Perception - a case where the behavior of the truck would be considered odd from an onlooker's perspective. This is a real software bug, but has near-zero safety implications. An example might be minimal weaving within the lane while traveling in slow-moving traffic.

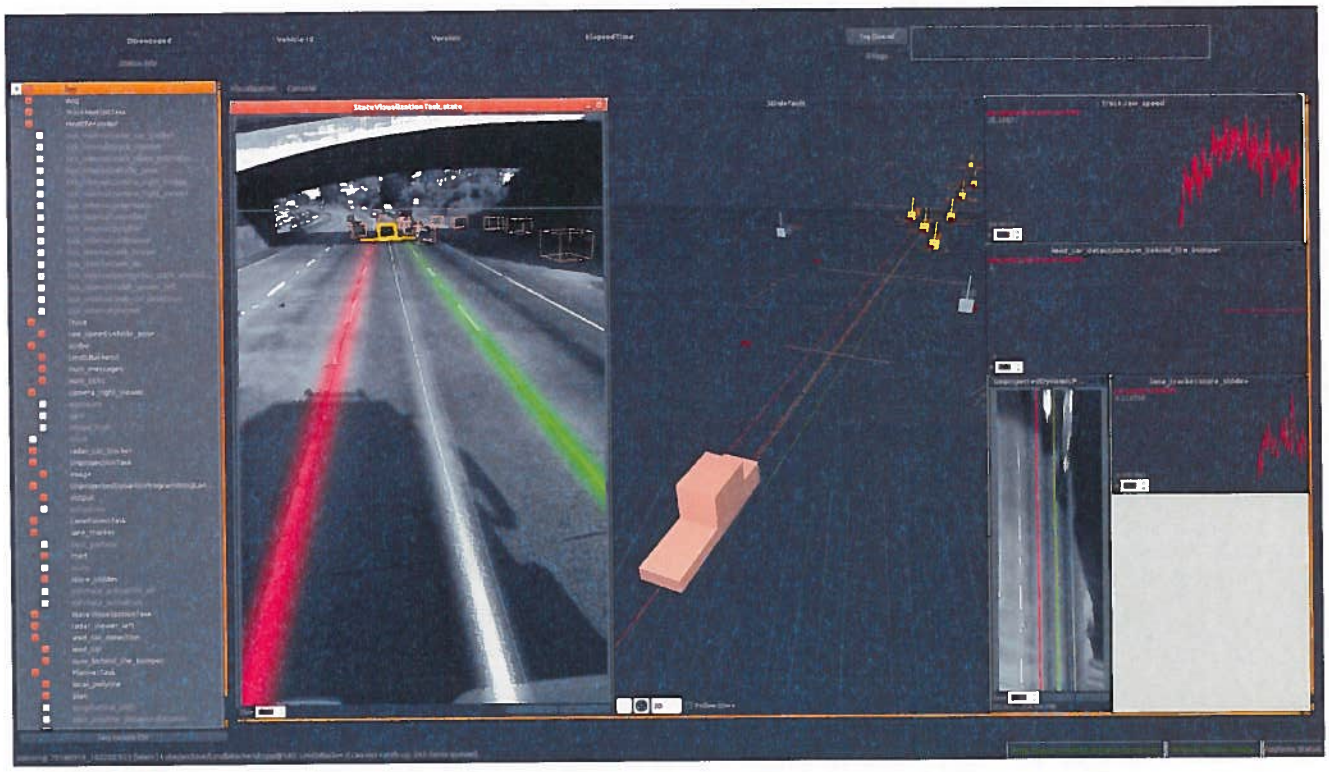
Major - a scenario that could have had safety implications had there not been driver intervention. An example would be braking on the freeway when there was no reason to do so. "Major" scenarios are ones that could have resulted in a safety risk if there had been a different set of actors surrounding the truck; but they would not have actually resulted in a collision or near-collision even if the human driver had not taken back full control during the particular environment of road conditions and surrounding vehicles when the disengage occurred.

Critical - a scenario where the truck's self-driving actions put it or any surrounding actors in actual danger that requires the driver to take back full control at that time. An example could be a situation where the driver takes over because the system did not command enough braking in order to stop in time for a lead vehicle.

Triage

Once a disengage event has a severity level assessed, it is assigned to a tracking bug. These bugs capture specific behavioral responses from the truck without drilling down to the exact line of code where something went wrong. While deep analysis is undertaken by the responsible engineer, the tracking bugs serve as a marker for us to monitor general system performance through each iteration of the software. At Otto, we refer to this process of post analyzing the logs as "triage" to highlight our need to have quick analysis and to shine a spotlight on the most important issues. During this process, further triage notes are left for each event highlighting exactly what occurred on the road, what the system's response was, and what was the root cause of the system's response.

When engineers and software quality analysts review logs they do so using our playback system nicknamed "DogTV", which is the same visualization tool that co-drivers use for active monitoring while on the road (see screenshot below).



In DogTV, we can see the video captured during testing overlaid with our perception system output. The perception system takes in camera images, radar measurements, as well as lidar returns to see the lane and all objects around the truck. For example, in the image above, we are outputting the system's interpretation of the left lane (red), right lane (green), centerline (white), and surrounding vehicles (yellow boxes). The camera view is the most interpretable from an outsider's perspective, but there are many more visualization options. Within DogTV we have a 3D visualization that shows raw perception and planner output (seen to the right of the camera image above). In this view, engineers can diagnose exactly what the system "saw" as well as what it planned to do with that information in terms of trajectory, braking response, etc.

Simulation

The advantage to this system is that our engineers have the ability to test new code on the exact events we struggled with in the past. Software simulation of this nature is a crucial component to ensure truck safety on the road. It gives us the ability to recreate previous drives and test how they would play out without having to put a truck back into the same situation. For instance, if we momentarily lose tracking of a lane when we go under a bridge, we can replay that same issue with different versions of software and see which one fixes the issue. The same process can be applied to not only what we see, but also how we react. The planner team can go back to events where drivers reported we braked too much or too little to ensure the future commanded braking will space the truck appropriately.

As much as we would like to write every line of code perfectly on the first try, bugs do arise and we are continually updating our software. In order for an engineer to submit a code change they first must pass a regression test. This test ensures that the change will improve the software and not make it worse than it was before. An example case is testing that our lane perception improves or remains the same as it was before. This is accomplished by comparing the system output to images where the lane lines have been labeled by humans. These images are referred to as ground truth, if the change does not increase the deviation from the ground truth lane lines it is one step closer to being accepted. The final step in submitting a change is to have the code reviewed by another engineer that owns that section of the code. Once the code passes the review it is submitted to the master version of our system.

Even though the new code has gone through thorough testing we have another step before it is released to our entire fleet. We bundle many code changes into "releases", which are comparable to software versions of other products. Once a release is bundled up we have one final test before the software is pushed to all the trucks. The software is pushed to one truck and it is sent out on the road to do an initial test run. During this run, the driver and co-driver are extremely cautious and send a report to the entire software team after the run is complete. If the testing goes well, the release is accepted and pushed to the entire fleet. If the testing does not go well, the code is re-evaluated and the whole process starts over.

Colorado Preparations

Before road testing begun in Colorado, members of our operations team analyzed over 80 possible routes around the country using satellite imagery to determine the best option for our 100-mile demonstration. Once we determined that the route met our system specifications, an initial truck was sent out in early August to confirm our findings. We decided that we were ready to proceed and immediately added the Colorado logging data into our machine learning pipeline to tune for the differences between Colorado highways and those in California. With this new information another truck was sent out late August, along with our head of software and our lead perception engineer. They validated that the I-25 route between Fort Collins and Colorado Springs would work very well for this demonstration. During this trip our trucks collected the data necessary for us to make a detailed map of the route without having to engage the autonomous system.

In the meantime, we have been performing hardware upgrades that we will use for the demonstration. These upgrades include a new computer, new LiDAR sensors, and a more robust wiring harness. We have already tested these upgrades on California roads for several weeks as well as at a private testing facility called GoMentum Station. We have been doing stress testing with a fully loaded trailer similar to the one we will be using for the actual commercial delivery in Colorado.

As of now, we have a truck in Colorado testing extensively. A driver and co-driver take the day shift, and they are replaced by another driver and co-driver for the night shift. These 2-shift rotations will occur every day and night until the demonstration in October. For this demo, we will be running at night, but the day testing gives us more chances to see traffic in case there is traffic the night of our delivery. In addition to our usual testing with the operations team, crew engineers from both the Control team and the Perception team in San Francisco are flying out to work shifts as co-drivers.

Criteria for Executing the Self-Driving Delivery

As we prepare for the first ever self-driving commercial delivery, one key metric we monitor is MPI, miles per intervention. This statistic, monitored by our safety engineers, is the number of miles our truck travels autonomously on a given route before the driver intervenes for any reason. The requirement to ensure safety is that we must be at 5x the MPI necessary to complete the demonstration. So, for our Fort Collins to Colorado Springs route of 125 miles, we will not do the demonstration unless we achieve a MPI of 625 miles. This means no disengages of any kind (Comfort, Public Perception, Major, or Critical) for 625 miles or, equivalently, 5 consecutive runs of the self-driving delivery route without a disengage.

Safety Precautions During the Demonstration

To give an extra layer of safety on behalf of the national shipper we are working with, we will have chase cars and lead cars driven by Otto employees near the truck, monitoring for any unusual road condition. This could include things like a broken bumper in the middle of the road, a CSP officer pulled over to the side, or a reckless or drunk driver.

All of us at Otto want to thank you for working with us on this project. We are very excited for the next couple weeks and hope this demonstration will help you understand the full capabilities of our autonomous system.