# Project VIEW

## Visibility in Elevated Wide Vehicles

# Final Report

Gabriel Butterick, Rebecca Jordan, Elizabeth Sundsmo, Kristyn Walker, Lucy Wilcox

Senior Capstone Project in Engineering
Olin College of Engineering
Spring 2018

Table of Contents

# Summary

In the following pages we describe our senior capstone project at Olin College in collaboration with Volpe National Transportation Systems Center and the Santos Family Foundation. The goal of this project is to characterize blind spots in large vehicles. This paper will cover the work completed to develop the Panoramic Method which can be used rate trucks based on the volume visible through the windshield and passenger window. The process of arriving at the Panoramic Method is broken down into the fall and spring semester. The fall semester covered the research conducted and three different solutions that were explored.
These three methods included using stereoscopic vision to gather computer models of vehicles and generate the field of view with Computer Aided Modelling programs, capturing the field of view with a smartphone and processing the photograph to quantify the visibility, and using laser scanning technology to generate a point cloud of the driver's field of view. We explored each of these solutions abilities to be used to create a safety standard for large vehicles in regards to the driver's visibility through the windows. In the spring semester we narrowed in and explored the smartphone method in greater depth which eventually became the Panoramic Method. We developed a web application to accompany the method as a way to process the photographs taken, and generated a database for visibility information to be stored so vehicles could be easily compared to one another. We found this solution to be the most technically viable solution while also being inexpensive and easy to implement.

*Figure 1: Students performing preliminary blind spot investigation*

# Background

This project was completed by a team of five seniors at Olin College of Engineering enrolled in the Senior Capstone Project in Engineering (SCOPE) program. SCOPE is a collaboration between Olin seniors and liaisons from an external party to develop solutions for real-world problems. We worked with the Volpe National Transportation Systems Center and the Santos Family Foundation. Volpe improves transportation by anticipating and addressing emerging issues and advancing technical, operational, and institutional innovations across all modes. The Santos Foundation is a non-profit organization dedicated to improving transportation safety. With the guidance of Volpe and the support of the Santos Foundation, project VIEW aimed to help save lives by reducing collisions between heavy vehicles and vulnerable road users. By characterizing and quantifying blindspots we would like to increase awareness on the limited visibility in large vehicles and influence truck design as well as fleet policies.

## Project Goals

Our goal was to characterize, model, and rate trucks according to their direct vision blind spots in a way that enabled easy comparison between vehicles. Trucks make up a disproportionately high number of fatal crashes, particularly in cities with large numbers of pedestrians and bicyclists. This is partly because drivers of these large vehicles are faced with limited fields of view. In 2016, 5987 pedestrians were killed in traffic crashes, making up 16% of all traffic fatalities. An additional 840 cyclists were killed[1]. These numbers have been increasing year over year from a low of 4109 pedestrians killed in 2009[2].

Direct vision is the visible area through a transparent plane and disregards mirrors, cameras, lenses, and other vision aids. We focused on direct vision blind spots because research[3] has shown direct vision reaction times are approximately 0.7 seconds faster than indirect vision. Even at the relatively low speed of 10 miles per hour, this would mean the truck travels 10 feet further before the driver can recognize the need to hit the brakes. A method to identify blind spots will educate consumers on visibility from their vehicle and lead them to make informed decisions when buying and driving vehicles. To do this we established a risk assessment method and rating system which insurance companies, fleet managers, and policy makers can use to encourage safer vehicle choices.

Though we could not directly impact which vehicles are on the road, we aimed to create a tool which provided blind spot ratings on different models to inform those who do make these decisions. To do this, we needed to:

- Create a method for capturing blind spot data
- Make assessment criteria for trucks based on the resulting information
- Create a tool for users to add and access blind spot characterizations

These decisions were informed by the needs and wants of several different potential stakeholders:

- Insurance companies: to create premium pricing which correlates to the likelihood of collisions.
- Fleet managers: to make more informed purchasing decisions around which vehicles are the safest on the road.
- Cities and policymakers: to provide cities with a resource on the relative dangers of different truck models and support rightsizing initiatives.

---

[1] http://www.ghsa.org/issues/bicyclists-pedestrians
[2] http://www.ghsa.org/resources/spotlight-peds17
[3] http://content.tfl.gov.uk/road-safety-benefits-of-direct-vs-indirect-vision-in-hgv-cabs-technical.pdf

## Existing Solutions

Our project was inspired by the work Transport for London (TfL) completed to improve road safety in the Greater London area. TfL funded studies, performed by Loughborough University and Transport Research Laboratory, on the risks trucks pose to pedestrians and bicyclists and placed a priority on improving the truck driver's ability to see these vulnerable road users during low-speed maneuvers such as turning and starting from a stop. They also found, using virtual reality simulations and consulting truck drivers, that drivers respond more quickly and effectively to people they can directly see through their windshield and windows than through mirrors or other indirect aids, and decided to focus on direct vision[4].

TfL created a ranking system to quantify the risk posed by different trucks. Trucks with very poor direct vision, including many construction vehicles, were ranked at 0 while newer low-entry, panoramic-view cabs were ranked a 5. Most trucks on the road were ranked at 2 or 3. This ranking is based on the driver's ability to directly see the area where vulnerable road users are usually positioned prior to a fatal crash. The area directly in front and to the curb side of the vehicle is divided into segments and weighted by their relevance to the most common crash types. Using computer aided design software (CAD), the visible portion of each area is calculated and used to generate the ranking. By applying this standard, London is progressively making it harder for the lowest-ranked trucks to receive permits to enter the city, with the goal of reducing the risk posed to pedestrians and cyclists.

We are adapting their methods to the data available on the US truck fleet, and designing our own rating system. UK trucks, in contrast with US fleets, are all of the cab-over design and drive on the left side of the road. An example of a cab-over truck is shown in Figure 2. The blind spots, human metrics, and frequency of different crash types varies between the two fleets. We used their existing work on reaction time, vision quantification methods, and the importance of direct vision.[5] More on the work we referenced from TfL can be found in Appendix A: Literature Review, and U.S. crash correlation information can be found in Appendix B.

*Figure 2: TfL Model of high visibility and low visibility truck cabs and their ability to see other vulnerable road users.*

[4] http://content.tfl.gov.uk/road-safety-benefits-of-direct-vs-indirect-vision-in-hgv-cabs-technical.pdf
[5] http://content.tfl.gov.uk/working-towards-direct-vision-hgvs.pdf

# Summary of Work

## Fall Semester

In the first semester we investigated three different methods to characterize and identify direct vision blind spots. The three different methods vary in tools, fidelity, and cost. The first method viewed a physical truck from the exterior and converted the vehicle into a computer model for further computer aided processing. The second method, a low tech option, captured an image of the interior of the truck from the driver's perspective with a smartphone. The third method collected a point cloud of the driver's surroundings with a LiDAR or Kinect scan of the vehicle to gauge how far the driver could see out through the windows.

| | DEVICE | DATA | ANALYSIS |
|---|---|---|---|
| 1 | ZED<br>Kinect / LIDAR | Capture computer model of truck | Use CAD software to generate field of view |
| 2 | "Diana Ross" | Capture picture of truck windows | Programmatically process image to calculate field of view |
| 3 | Kinect / LIDAR | Capture point cloud of drivers field of view | Programmatically process point cloud |

*Figure 3: Overview of the three different first semester methods. This chart shows the device used and the data is it capturing as well a brief explanation as to how the data will be analyzed. *Diana Ross was renamed to the Panoramic Method*

### CAD Processing

Before pursuing a method, we briefly attempted to model some of the work done by TfL, an example is shown in Figure 4. They used CAD models of vehicles, sourced from the vehicle manufacturers, to model the field of view. To test what we could achieve with CAD, we used a model of a Western Star Truck from GrabCAD[6]. From this we generated the approximate field of view for the truck in Solidworks shown in figure 5 by establishing a reference point at an estimated location of the driver's eyes. From the reference point, we projected the driver's view

[6] Western Star 4900SF https://grabcad.com/library/western-star-4900sf-1

based on the window locations, essentially drawing a line from the reference point to each window edge. Figure 6 shows the possible blind spots of the truck in Solidworks.

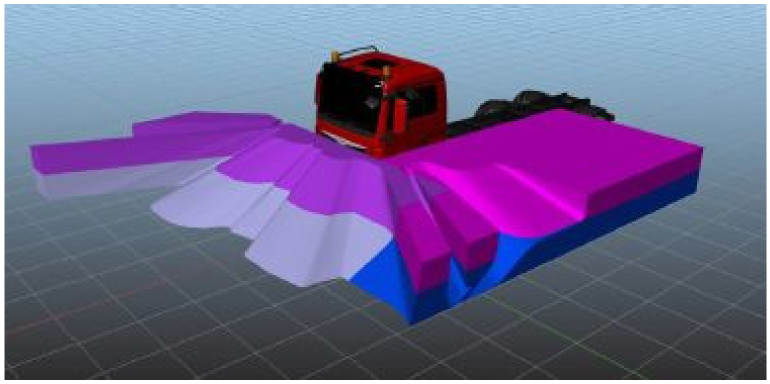


*Figure 4: Truck field of view modeled by TfL*

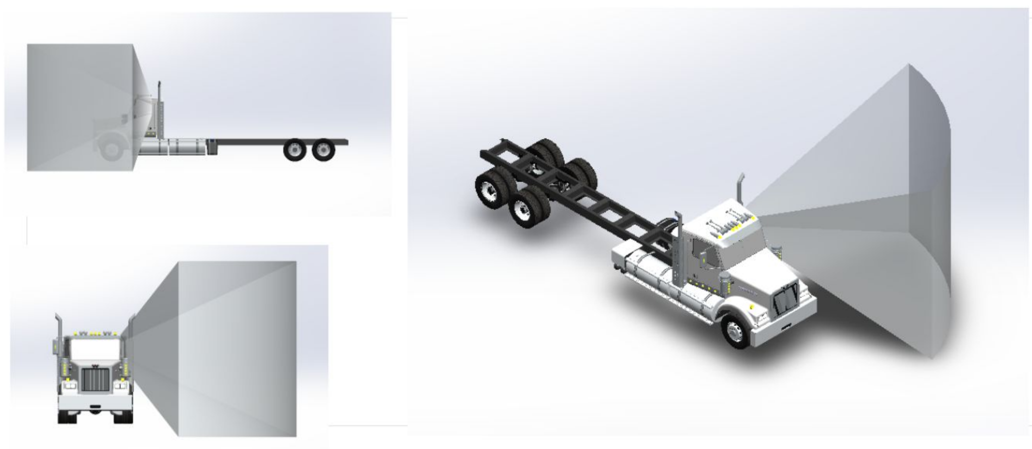


*Figure 5: Example of projected field of view in Solidworks with a pre-existing CAD model of a Western Star truck from GrabCAD. The projections model the visible area through the driver side window.*

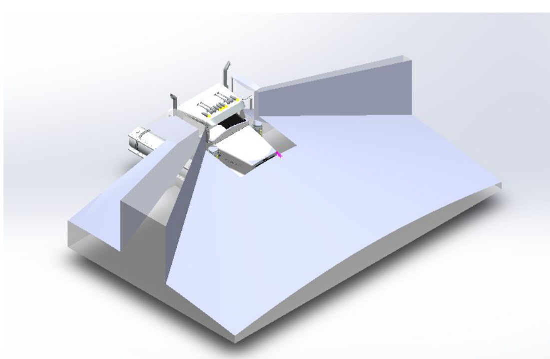


*Figure 6: Example of blind spots in a Western Star truck from GrabCAD*

This model served as a rough estimation of what could be done with a CAD model of a truck and was a great proof of concept but turned out to be time intensive, as it took about an hour per vehicle for an experienced Solidworks user. In addition, Solidworks lacked the tools to create an accurate projection of the visual area from a point, meaning that producing an accurate model of the blind spots with this software would take significantly longer. However, the proof of concept did indicate that modeling the blind spots in CAD would be good for both comparison and visualization purpose. As a result, we decided to explore other pieces of CAD software that might automate some of the window detection and projection of field of view.

We compared the usefulness of Solidworks against a few other CAD tools, the most initially promising of which was OpenSCAD[7]. We were interested in the programmable nature of OpenScad, which we believed would allow us to automate the process of detecting and quantifying the field of view. Ultimately, we decided that while the program is free of charge, determining whether it could achieve the results we needed was not a wise investment of our time, especially when our team members were already comfortable with Solidworks.

We investigated further into which software TfL used to create the figures that inspired us to construct the view in CAD. The Loughborough University team referenced using a software called SAMMIE DHM to construct the field of view of the trucks they were studying in a report from 2015.[8] SAMMIE DHM is a digital human modeling program[9] used to determine whether designs will accommodate people of varying proportions and abilities. This program can be used to determine the eye position of drivers of various heights as well as which areas are visible to them and how much they need to turn their head to see those areas. However, the program is only available through the University and has no free trial period and we were unable to gain access to the program due to its cost.

A more recent report from Loughborough University[10] mentioned using Rhinoceros 3D, a CAD software comparable to Solidworks but aimed more toward design than engineering analysis[11]. The Loughborough team mentioned specific functions in Rhino that generate the visible silhouette of a complicated object, which remove much of the work of manually determining the edges of windows, mirrors, the dashboard, and parts of the hood in the field of view. Rhino also has tools for projecting shapes to a point, which Solidworks lacks, and the Boolean addition and subtraction of volumes required for determining how the field of view intersects with an area of interest. The method used to generate our proof of concept of Rhino is documented in Appendix C.

[7] http://www.openscad.org/
[8] Understanding direct and indirect driver vision in heavy goods vehicles https://dspace.lboro.ac.uk/2134/21029
[9] SAMMIE DHM http://www.lboro.ac.uk/microsites/lds/sammie/
[10] Definition and testing of a Direct Vision Standard for Trucks, Loughborough University
[11] Rhinoceros 3D https://www.rhino3d.com/

## Acquiring CAD Models

We quickly discovered we would need to generate our own model of a vehicle since most truck companies keep their CAD proprietary. To create our own computer models, we planned on accessing a physical truck that we could scan to generate a computer model.

In order to gather truck data to generate CAD models, we purchased a stereoscopic camera, ZED, to create a mesh of a physical truck. The ZED is a 3D camera for depth sensing and motion tracking. This replicates the way humans discern depth by noting the subtle differences between cameras located in a line, with a known distance between them. Using the camera specifications, some basic math, and the disparity between the two images, it is possible to figure out the distance from each pixel in an image to the camera location when the image was taken. This information, combined with an inertial measurement unit (IMU), can be used to create a world coordinate mapping of a room or object. This integrated image processing ability combined with the relatively low price tag of $450 was a very compelling argument for using stereoscopic technology.[12]

It appeared that the ZED would be able to generate clear point cloud and mesh representations of a truck at a low cost. The resulting raw mesh could be imported into a CAD software to create a representation of the truck scanned with the ZED. The process of using stereoscopic technology was appealing because a user could easily plug in the technology, walk around a large object, and collect a mesh model of that object. From the marketing videos, it seemed like the ZED would require little to no experience or expertise to use, but still allow for a large amount of high quality data to be gathered. The supporting software seemed to be exactly what we needed, and appeared process the mesh into a usable form without too much interaction on our part.

As we waited for the ZED we explored other methods of stereoscopic capturing. One of the more successful results from this exploration was the use of smartphone apps[13] to combine a number of pictures of a single object into a 3D model, which could then be moved into CAD in the same way as the mesh files. Unfortunately, the phone wasn't capable of processing the pictures quickly, especially if the object in question was large, and it tended to fail when trying to create truck sized 3D objects.

We decided to pursue more testing on the ZED camera with larger vehicles such as trucks before moving onto a different solution. We located a few trucks in the facilities department at the neighboring campus, Babson College. We were given use of the trucks whenever they were not required by the facilities staff. We scanned a pickup truck and utility vehicle, both of which were large vehicles with truck cabs. Both these scans included suspicious artifacts and signs of

---

[12] https://store.stereolabs.com/
[13] https://play.google.com/store/apps/details?id=com.smartmobilevision.scann3d&hl=en

deformation in the computer model which we attributed to user error and external conditions such as lighting and motion from the wind. Then we went to the City of Boston fleet management center and captured the images in Figure 7 inside their warehouse, which were also of low quality. The edges of the windows in the scans were often left unpopulated due to the reflective nature of the glass when under a bright light such as a the sun or fluorescent lighting.

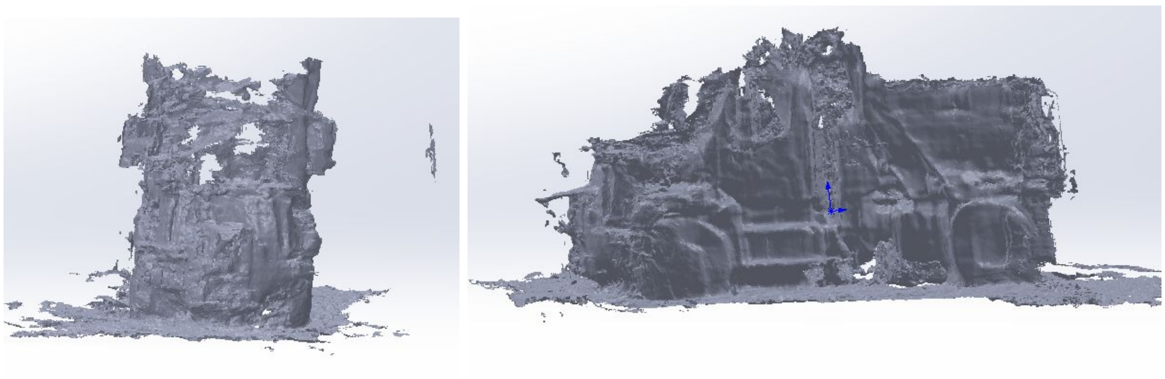


*Figure 7: ZED generated point cloud of a City of Boston fleet vehicle*

We intended to process the truck model gathered from ZED in a CAD software such as Rhino and remove the artifacts and patch up some of the missing areas. However, this required unbroken, accurately-shaped edges in the CAD around each window, and accurately-shaped modeling of areas that were directly visible to the driver, such as the hood and mirrors. With this CAD, we could project the field of view from a driver's eye point outward through the loop of the visual edge of the window and construct the entire area around the truck that is directly visible to the driver as shown in Figure 8. Unfortunately the windows we need to create the projected field of view did not render in the scan, making the model unusable for processing.

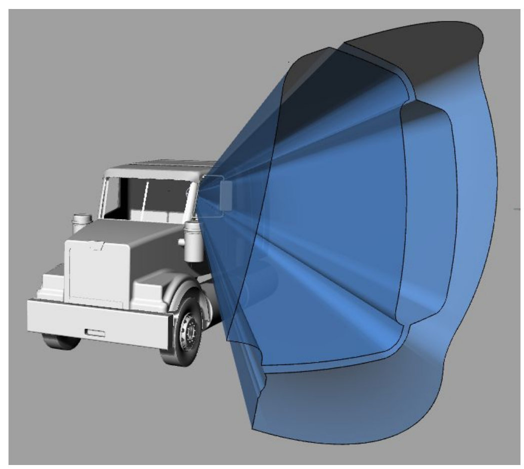


*Figure 8: Field of view from driver through driver side window generated in Rhino. Note the mirror blind spot in the middle of the window field of view.*

We quickly realized that there was no way for us to reconcile the many failings of the ZED technology with the goals of our project, and have chosen to abandon it as a part of the proposed methodology. The distortions and artifacts made it unrealistic for us to move forward with the ZED. In its place, we shifted our focus to the Panoramic Method and the LiDAR or Kinect scans from inside the cab.

## Panoramic Method

For our second method, the Panoramic Method, we wanted to create a process that was reliant on a technology most people carry with them on a day-to-day basis, a smartphone. By capturing the interior view of a truck as a panoramic picture with a standard object for scale, we were able to calculate the field of view at very low cost.

This method slowly evolved through a few distinct moments in the semester. While waiting for the ZED camera to arrive, we explored a low-tech options with the original idea to replicate stereovision, more specifically, depth perception similar to the ZED. This required two photographs of the same object taken with a smartphone to output the depth of the image.

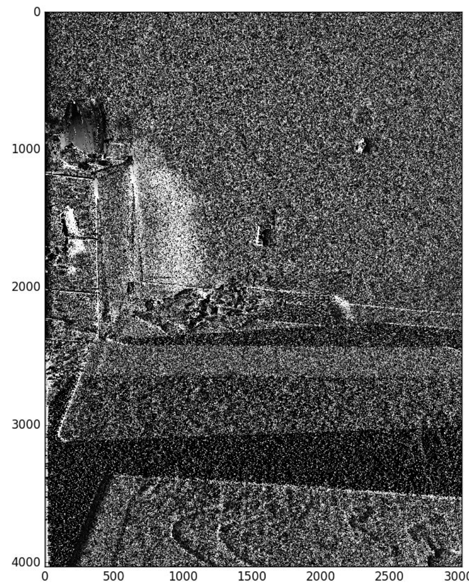



*Figure 9: Image produced with two images taken on a phone of a table, tea kettle, and yoga mat.*

We took a couple test images of our SCOPE classroom to detect the objects in the room including the edge of the table, a tea kettle, and a yoga mat as shown in Figure 9. We learned that going from this disparity map, a map of the change between the two room images, to a point cloud with depth would be difficult because phone cameras are all slightly different and require calibration.

Despite this difficulty, our team was inspired by the information we were able to gather from so few images and continued exploring what else would be possible with a smartphone camera. This evolved into taking photos from the driver's seat looking out each window, and later into a panoramic photo of what is visible from the driver's seat. Figure 10 shows a panorama taken of a truck at Babson. This picture was even more compelling when we put the photo into Google Cardboard, a simple VR viewer. When looking at this image in VR, it felt as if we were sitting in the truck with one of our teammates right in front of us, and from this came the main inspiration of the Panoramic Method.

*Figure 10: We found panoramas particularly compelling because we could capture them and then view them in VR goggles and experience what a drivers visibility would be like in the truck. We then considered adding a measuring stick to quantify the panorama.*

We expanded by adding a measurement pole marked by colors to the process as shown in Figure 11. Each color represents a one foot interval and would later be processed by a computer program we made to detect the heights. Our plan was to have the color stick in every frame at the same distance from the driver as the camera sweeps through the view to provide a scale of what height is visible to the driver. This involved someone walking with the color stick at the same rate as the pan of the camera. This preliminary method is documented in more depth in Appendix D.

*Figure 11: Preliminary testing of the measurement pole.. One team member walks with the measurement pole in a rectangle around the camera, always staying in frame of the camera.*

We discovered it was extremely difficult to keep the color stick in frame with the camera, and were often left with an image that had the pole in less than 25% of the image. The second issue we ran into was detecting the color on the stick. The color captured in the image depended primarily on the lighting of our environment, making the layers indistinguishable to the computer program that would be responsible for determining visibility. In addition, even if we could account for different lighting and detect specific colors in the image, the users would then be required to use a measuring stick with the same colors the program was expecting.

We iterated upon this preliminary method to overcome some of the previous issues. This version, as detailed in Appendix E, uses a single instance of the measurement stick at a set distance from the driver. The color at each height are then propagated across the photo as shown in Figure 12.

*Figure 12: The measurements on the left of side of the photo are propagated across the image.*

The lines propagated across the image denote the height in an arc around the driver with a radius of six feet, the initial distance of the stick away from the driver. This assumes there is minimum distortion in the photo. From this, the lowest visible color through the window on the image can be used to calculate the driver's field of view angle. The results are discussed in Appendix F. This method later evolved into our final procedure which is explained in Appendix J.

## Field of View (FOV) Scanning

The third method, FOV scanning, directly measured the field of view from the perspective of the driver, with a Light Detection and Ranging (LiDAR) or Kinect. We explored this method near the end of the fall semester because we wanted to see whether this would be a valuable method to pursue in the spring. We ended up creating a detailed plan to test the feasibility of applying this technology to our project. Based on our research, we expected this method would produce results with a high resolution due to the removal of possible human error which is present in the Panoramic Method.

The FOV scanning plan was to use LiDAR technology and a panning motor, mounted on a tripod placed on the driver's seat. With this setup, we predicted we could scan a point cloud of the entire field of view from the average driver's eye level. This scan would collect a large set of points that described the nearest visible surface for every angle and distance around the driver.

With some filtering to remove the unimportant areas such as the interior of the cab, we would be left with a map of the area outside the vehicle that could be seen by the average driver. Using this information, we expected to analyze the dimensions of blind spots and available direct vision. Our exploration indicated that LiDAR would be more expensive than other methods, totaling about $2,000 and required assembly by a user with comprehensive technical skill.

We also explored using a Kinect system to collect the visibility around a vehicle as a cost effective alternative with high resolution results. We tested the XBox motionsense Kinect in detecting vehicle interior. It failed to meet our needs because the Kinect could not handle the sunlight, ambient or direct. It was designed for use indoors, and sunlight caused the sensor to become overloaded and washed out useful information. Additionally, a Kinect is a short range device, which was reduced by windshield/ windows and could not scan a useful distance away from the driver’s viewpoint.

## Fall Semester Method Assessment and Comparison

We assessed each of these methods across several factors as shown in the table below. We created specific scales to differentiate levels of skill and fidelity required.

| | External Scan | Panoramic Method | FOV Scan |
|---|---|---|---|
| Needed Resources | Laptop with NVIDIA graphics card, ZED stereoscopic camera, CAD software, physical truck | Smart phone, long stick, marker, laptop, physical truck | LiDAR or Kinect, Pan-tilt system, Tripod, Laptop, electronic components |
| Approximate Cost | $450 - ZED<br>$1000 - laptop<br>$1450 | $5 - materials<br>$500 - computer<br>$505 | $1500/ $100 - LiDAR/ Kinect<br>$60 - Motor shield<br>$50 - Battery<br>$70 - Arduino<br>$330 - Tripod System<br>$1000 - Laptop<br>$1610-3010 |
| Time commitment | 30 min - scan<br>3 hours - processing | 1 hour - image collection and processing | TBD |
| Technical expertise | Run CAD software | Panoramic Photo with smartphone | Arduino knowledge and programming experience; technology dependent |
| Data fidelity | Poor | Poor-Average | Very good |
| Pros | Automation potential post scanning | Low cost, no proprietary software | High fidelity, semi-automatic |
| Cons | Poor scan quality, Battery life limitation | Low fidelity data, high potential for user introduced error | Cost, large quantities of data, and post processing requirements |
| Assumptions | ZED scan sufficient, Phone scan feasible, Future CAD from manufacturers | Manufacturer CAD is unavailable | Higher fidelity data, significantly more expensive despite using cheapest LIDAR |

*Table 1: Decision Matrix*

Of the three methods that we explored, we decided to pursue the Panoramic Method and the FOV scanning. We chose not to continue working with the ZED because the scan quality was too low. The Panoramic Method and the FOV scans would output similar data on visibility which could make them easier to incorporate into a single rating system.

## Spring Semester

During the spring semester we continued testing FOV scanning and the Panoramic Method. We were unable to find a suitable FOV scanning technology, so we continued improving the panoramic method over the course of the semester. As part of this final method we developed instructions for scanning a truck with the Panoramic Method, and a website to perform the required calculations required. Each of these topics, and our results from testing the Panoramic Method and FOV scanning are discussed.

### LiDAR Testing

LiDAR is a technology capable of capturing distance and angle information on a single plane of view using light from a laser and radar principles. By mounting a LiDAR on a pan or tilt mechanism it is possible to scan multiple planes of view from a single point. Those planes can be analyzed to create a map of distance and angle data. This information can be processed to find visibility angles, or get imported into a CAD software as a point cloud with to create a visual representation of the scanned area. We planned to test this technology to see if it was suitable for our use case as shown in Table 2.

<table>
<tr><td rowspan="4">← Time Progression</td><td>Obtain LiDAR</td><td>Outline a bill of materials for needed components</td><td></td><td rowspan="4">Document process noting time and personnel requirements, successes, failures, areas for improvement, and future applications</td></tr>
<tr><td>LiDAR testing for accurate information capture</td><td>Finalize BOM for LiDAR system and order all parts</td><td>LiDAR code writing and debugging</td></tr>
<tr><td>Write code to link together different scans to form a complete map</td><td>Obtain/Assemble pan/tilt-LiDAR system</td><td>Program motion of servo, debugging</td></tr>
<tr><td colspan="2">Scan a vehicle using LiDAR system</td><td>Complete code debugging</td></tr>
</table>

*Table 2: Our planned approach to testing and using LiDAR.*

We thought this method might be feasible because it takes advantage of a more precise data collection method than stereoscopic analysis, which we found did not work in the fall semester. While the LiDAR is significantly more expensive than other options we explored, there are multiple benefits. The process would be more exact than a stereoscopic or manual approach, and there would be little technical experience required to perform the scan. Analysis code could be easily shared, such that a user could put files in an appropriate location and have them automatically analyzed. Additionally, a system could be packaged as a kit and shared between groups reducing the effective price.

We did have some concerns about collecting data with LiDAR:

- Too much data to process in a reasonable amount of time.
- Inaccurate data collection due to refraction/reflection through glass windshield.
- Data corrupted due to direct sunlight (IR), specifically glare off of metallic surfaces.

We obtained a SICK TIM551 from Olin Aquatic Robotics Club and conducted a few tests to address our three concerns. All tests were completed using the TIM551 and a laptop running ROS on Ubuntu 14.04. We immediately ran into several issues upon trying to operate the LiDAR due to missing documentation and cables. With some guidance from SICK Technical Support, we made our own cables and powered up the LiDAR.

We tested for data inaccuracy when scanning through vehicle windows by placing the LiDAR in a stationary vehicle with objects and walls around it. We observed the objects through the windows, and noticed a qualitative change in the LiDAR distance reading when the door was opened to move the window pane out of the way. This shift was also observable when moving the LiDAR around because pixels representing solid surfaces would 'jump' whenever glass was introduced or removed from the plane of view.

We then expanded that test to see how the tinting and angle of incident inherent in vehicle windshields affected the data. We also sought professional advice from the manufacturers of LiDARs. The results from the testing were strongly indicative towards infeasibility. The loss in number of readings was around 90% and those that did get through were erratic and spotty. A brief comparison of LiDAR results through a windshield and LiDAR in the open air shows a drastic difference in both data quantity and fidelity, as can be seen in Appendix G Figure 26 and 27.

The conversation with SICK backed up our initial experimental findings, citing issues with dispersion and refraction that would corrupt our data and shorten our range. We received no information on whether that error would be constant, and therefore mitigable, or semi-random, and insurmountable. Our testing results indicated that it will not be mitigable because the LiDAR only gets readings through tinting when scanning something perpendicular to the LiDAR, which is unacceptable. We tried to get readings through the much more tinted part of the windshield, at the top, to see if there was a noticeable difference. We found that we got roughly the same results regardless of the level of tinting.

We came to the conclusion that as long as there are windshields in the way, a LiDAR cannot be used to scan the field of view from a driver's perspective. The tinting caused the loss of too much data to get FOV scans with high enough accuracy. This could have been resolved by removing the windshield and lowering the windows, but we were concerned that this modification would be too intrusive to the vehicles for it to be done easily and on a widespread basis. If there were an easy way of removing and replacing the truck windshield, this methodology would become possible again.

While working on the LiDAR FOV scans we made plans to automate the LiDAR scanning system with a pan and tilt motor, these can be found in Appendix H and the power plan can be found in Appendix I.

## Panoramic Method Updates

The first update we made to the Panoramic Method in the spring semester validated that objects in an arc around our camera would appear in a straight line on the image we captured. We did this by setting up four identical chairs around the camera such that they were all the same distance away from the camera. In the first capture, we used the Google Cardboard Camera app. We found that all of the chairs were distorted in an arc in the top image shown in Figure 13. We next tried using the  Google Pixel's camera app in panorama mode. This resulted in all the chairs appearing to be the same distance from the bottom of the photo, and corresponded with the height marked from the measurement pole as shown in the bottom image in Figure 13.

*Figure 13: The top image is taken with the Google Cardboard Camera app and does not have a straight line between the chairs that at all the same distance from the camera. The bottom image, taken with the panorama setting on an Android Google Pixel phone has the desired straight line.*

In both instances, it was difficult to keep the camera at the correct height and rotate the camera steadily. We felt a tripod would make this easier. We also decided to make a stand for the measurement pole so that a person is not required to hold the pole.

We attempted to make rotating the smartphone easier with a camera tripod, but the tripod was not easy to work with when placed in a truck chair. It was difficult to extend the legs of the tripod while the tripod was on the driver's seat. There were issues with keeping the phone level on the tripod, since adjusting the legs was the only mechanism to establish a level surface. It was also difficult to keep everything stable, the soft cushion of the seat made small adjustments change the whole system.

To solve these issues we designed our own camera stand. The final design is an assembly consisting of a metal plate, a rectangular, metal tube, and a phone mount. The assembly is designed to sit in the driver's chair and provide a stable base for taking panoramic photos without distortion. The stand has a ball joint at the top of that allows the system to be levelled closer to the phone mount. The phone mount can be oriented in landscape or portrait to accommodate different phone and camera applications. The bill of materials, an image of the camera stand , and the most current method can be found at the end of Appendix J.

We tested our methodology, shown live with the  City of Boston's fleet of trucks that are in frequent use for city needs such as snow plowing, waste removal, and other miscellaneous needs. We looked at the differences in field of view based on the type of cab (cab over vs. conventional), which indicated that certain cab models offer a substantial increase in direct vision. Additionally, having windows the extend further down (like those on school busses) offers even more direct vision in the highest risk area of the truck.

## Rating System Overview

A few important principles our rating system follows are:

- Our rating system should be absolute, not curved to what is on the market.
    - Why: we want to encourage improvements even in the safest vehicles - no vehicle is risk-free.
- Vehicles should be compared within and across classes.
    - Why: Sometimes a large class 7 or 8 vehicle is necessary and we wanted to enable the safest large trucks to be chosen, but we also wanted to encourage rightsizing and make sure that someone choosing a large truck is aware of the risks posed by this choice.
- Enable VIN lookups.
    - Why: Insurance companies wanted to be able to do VIN lookups to see how the make and model of the vehicle rates.

We found TfL's rating system useful in determining what ours might look like. TfL used a five star rating system for several categories of trucks as opposed to assigning ratings on a truck by truck basis. However, they found some issues with this system and have considered looking at each individual truck. TfL also found that only the area in front and on the far side from the driver is worth weighting in their model because relatively few collisions occur on the driver's near side. To generate rankings from this they used a volume based model which accounts only for the area outside of the truck. They then weight this volume based on how low to the ground the area is, corresponding to the percentage of population which would be visible at that height.

We developed our own rating algorithm and website to process the panoramic images and calculate the visibility. This is discussed in the following sections.

## Panoramic Method Calculations

The output of the rating algorithm was the percentage of a volume of interest around the truck which can be seen directly by the driver. This volume of interest extends 15 feet beyond the front and passenger sides of the truck, and is bounded horizontally by the left side of the windshield and the right side of the passenger window. We selected the 15 foot boundary based on Transport for London's research on where pedestrians were located before fatal crashes[14]. Trucks were not penalized for not having visibility within the bounds of their own hood, which was approximated as a rectangle defined by horizontal measurements to the front bumper and the passenger side window relative to the driver's head.

Vehicle measurements and the visibility photo are submitted to the website, which is described in Appendix J. The website then calculates visibility using an algorithm to determine both the total volume of the area of interest around the truck and the portion of that volume which is visible. The height of the bottom edge of the field of view at the distance of the measuring stick is calculated for each given column of pixels in the panoramic image. We construct the horizontal distance to the corresponding point on the ground with similar triangles. The height at the measuring stick is converted into the angle of the sightline relative to vertical. The tangent of its complement is then multiplied by the driver's height off the ground to find the distance to the nearest visible point on the ground.

Based on the nearest visible point distance, the driver's height off the ground, and the size of the truck, we calculated the volume of a wedge centered on the angular position of each pixel measurement. By summing these wedges we approximated the actual volumes of blind and visible space.

[14] http://content.tfl.gov.uk/assessing-drect-vision-in-hgvs-technical.pdf

## Website

We created a website to process the photos and vehicle information collected to output a rating of the vehicle. Our website[15] was built with the Django web framework and is hosted on the Heroku web application hosting service. The source code is publicly hosted on GitHub. It can be replicated by following the instructions on the GitHub page[16]. When using the website, users upload the panoramic image taken in the scanning procedure. For best results, crop the image beforehand so that the majority of the photo is taken up by the window instead of the inside of the cab. Next, users locate each visible height marker on the measurement pole, and the line between the nearest visible ground and truck cab. They then enter each of the four measurements taken in the scanning procedure. This allowed the heights to be propagated across the image at the radius the measurement pole is away from the driver. The website takes the entered data and calculates the percentage visible volume, which is described in the more detail in the following rating calculation section.

If desired the user can enter the VIN number associated with the vehicle and the information will be stored in the site's database. Users can then look up the visibility of different trucks by their VIN, partial VIN, make, weight class, or filter by all trucks with percentage visible volumes above a set limit. This allows users to see the visibility of trucks that they did not scan and compare trucks they scanned to other models.

Our website site architecture is shown in Figure 14.

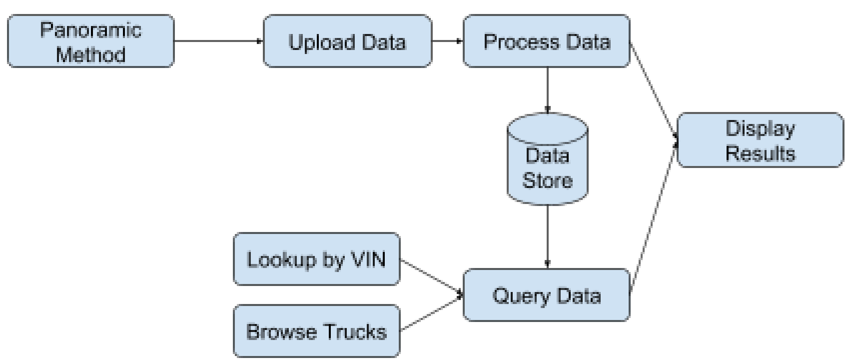



*Figure 14: Architecture for our rating system*

This website is free to be copied and updated with new features or calculations as needed in the future.

[15] https://blindspotcalculator.herokuapp.com/
[16] https://github.com/LucyWilcox/VIEW-site

## Method Validation and Error

In an effort to ensure that the simplicity of the Panoramic Method, described in full in Appendix J, did not detract from precision, we tested our methodology on the same truck with different users. When testers used our camera stand, there was a standard deviation of 2.6%. When users did not use the camera stand and took the panorama with the phone in hand, the standard deviation was 2.7%. From this we could not conclude that the camera stand improves the repeatability of our method. However, the camera stand could be improved to increase repeatability. With the final iteration of the camera stand, it was acceptable to forgo using it and take measurements from a tester's eye position sitting in the cab and have the tester rotate the phone by hand to take the panoramic photo. The downside of holding the camera by hand is that it is more difficult to keep the camera stationary while taking camera location measurements such as the distance from the camera to the ground, passenger window, front of vehicle, and measurement stick.

We found that user error on our website which calculates the percentage visibility also introduces some error. The average standard deviation between results using the website with the same photo and truck data was 0.8%. This could potentially be improved on with more usability updates to the website, but this step introduces less error than the scanning itself as discussed previously.

To put these percentages in perspective, we scanned trucks with percentages visibilities between 51% and 78%. These trucks were of classes three through eight and included conventional and cab over engine styles. From these results we suggest that any difference in visible volume above 3% is a legitimate difference and not simply error. In order to further sanity check our model, we tested trucks with obvious disparities in field of view to ensure our results reflected those differences. We found large differences, like those noted above, that indicate our method produces reasonable results.

From our testing we also examined the time required to scan a truck. Our testing showed that new users took about 30 minutes to perform the methodology the first time, but all users saw a decrease in time required by half or more in subsequent scans with them eventually taking between 6 and 12 minutes. Our testing results indicate that the initial time required was high due to users reading through instructions for the first time to learn the process.

Our method is relatively affordable. The total cost of the assembly is $185.72, which is fairly cheap when compared with other methods, like LiDAR. If the camera stand is not used, the total cost is $35.34 and could likely be done for less with other materials.

As validation, we wanted to test our method against CAD model methods. The added accuracy and precision gained from using such an exact source of data would have been ideal for ensuring that our method is correct. Unfortunately, because we were unable to acquire either manufacturer CADs or collect our own, we were only able rely on human intuition and use the fact that some truck cabs have notably more direct vision area over others as a sort of validation. Our testing also indicated that our initial assumptions and intuition regarding cab vision were correct. We found that cab over engine models, which feel like they have more vision, actually tend to have a percentage visible volume of 70-80% compared to a percentage visible volume of 50-55% in conventional cabs for trucks of similar sizes.

# Future Work

After a year of working on this project, we produced a reasonable way to quantify blind spots using the Panoramic Method. However, we recognize this product is a proof of concept and requires more work to be completed before it can be used to on mass basis.
Some areas for future work we think would be worthwhile are:

- Web Development
    - Upload image and measurements to website so other can verify website usages
    - Include more error checking, for example check for unrealistic or partly duplicated bottom visibility lines
    - Account for overlap between front and passenger windows, for example due to non-vertical window boundaries
    - Continue to improve user interface
    - Have actual degree boundaries for left side of passenger window and right side of the front window instead of boundaries that depend on the window boundaries
- Camera stand
    - Improvements for stability.
    - Standardize height based on torso height of truck driver population.
- Measurement collection
    - Make it possible for one person to complete a scan, for example possibly use lasers to get distances
    - Make video for instructions on how to scan
- Add vehicles to the database
- Get manufacture CAD to validate Panoramic Method
    - Generate field of view through of vehicle using CAD software to compute the theoretical visible volume and compare to the measured volume with the Panoramic Method
- Test Panoramic Method on other vehicles such as passenger cars
- Truck driver human factors and regulations
    - Develop with truck driver regulations in mind
    - Safety Officers

# Appendix A: Literature Review

## Definition of Direct Vision Standards for Heavy Goods Vehicles (HGVs) Technical Report

The July 2016 report from the Transport Research Laboratory (TRL) under contract from Transport for London (TfL) details the process of developing a ranking system for trucks in the UK and EU.

TfL focuses on ranking the direct vision of heavy goods vehicles. While trucks make up a small percentage of the vehicles on urban roads in the UK, they are overrepresented in deadly crashes, especially when vulnerable road users such as pedestrians and bicyclists are involved. A significant portion of these deadly crashes occur when trucks are either starting from a stop and run over a pedestrian directly in front of the truck, or when they are turning and hit a bicyclist traveling between the truck and the curb. In both of these scenarios, the victim is usually located in one of the truck's blind spots. Though they may be visible in one of the truck's mirrors, recognition of the danger is quicker and easier through a direct line of sight. For this reason, TfL seeks to incentivize the adoption of trucks with fewer and smaller blind spots.

TRL considers the benefits and drawbacks of several different methods of determining a direct vision ranking from a CAD model of a truck, and settles on projecting the visible areas from a driver's eye position through the window openings to generate a model of the volume of space around the truck that is visible. They then consider the overlap of this visible area with the area in which vulnerable road users are usually located before a crash, taking into account the driver's reaction time and the rates of travel of the participants. The more completely these key areas are visible, the higher the truck's direct vision ranking.

A significant portion of the paper is devoted to determining how to generate a ranking that is applicable to trucks throughout the EU and accurately represents the danger posed by each level of visibility. Areas which would allow a driver to see any person with a height between that of a 5th percentile woman and a 95th percentile man, and those corresponding to the most common crashes, were weighted more heavily. The ranking system differentiates between trucks that are known to have differing levels of visibility, such as construction vehicles and low-entry cabs.This paper has been our primary inspiration in our project. Its identification of direct vision as an area for a high-impact, narrow-scope project was ideal for us to address.

## Definition and Testing of a Direct Vision Standard for Trucks

This is an update from the University of Loughborough, continuing on TRL's work on TfL's ranking system. The ranking system is based on existing regulation of areas that must be visible through mirrors. No regulation exists in the UK on what areas around a truck must be directly visible through windows, but the minimum area visible through mirrors is regulated. This update was made because crash data rapidly becomes outdated, introducing subjectivity to the ranking system.

The paper cites research showing that direct vision is more beneficial than indirect vision for drivers noticing pedestrians, especially when the head and shoulders of the person are visible. Since lawmakers have already determined the areas of mirror regulation to be important for safety, the new ranking is based on whether the head and shoulders of pedestrians and bicyclists within this area are visible.

## Code of Federal Regulations: School Bus Vision Requirements

The Code of Federal Regulations is a U.S. government source that details specific requirements for school bus driver visibility. It uses a camera and cylinders placed around the bus to measure mirror and direct vision capabilities as listed in S13 to make sure vision standards are met. The cylinders, or poles, are laid out carefully using specific measurements, and the camera is set up to represent the eye location of a 25th percentile adult female. Pictures of the visible poles are taken from a single point within the semicircle created with respect to the specified eye location. Poles only visible in mirrors are located and measured using minutes of arc[17].

---

[17] https://books.google.com/books?id=scPxlviLm7kC&printsec=frontcover#v=onepage&q&f=false p375-379

# Appendix B: Collision Correlation

We tried to determine the relationship between different makes/models and crashes. This would allow us to determine if the visibility ratings we developed correlated to crash risks. We believe showing a correlation would be compelling data for insurance companies to adjust their fees based on our ratings.

The number of trucks of different makes and models in crashes each year can be determined with data from FARS and TIFA detailing truck crashes. The specific model can be determined using the Vehicle Identification Number (VIN). Some manufacturers, like Freightliner[18], provide a record of how their VIN codes map to models. We initially used this to map VINs to model data. However, we later found an API provided by the National Highway Safety Administration[19], which allowed us to determine the manufacturer, make, model, series, weight class, and body class of the vehicle. We used this to generate data on how many trucks of each type were registered in New York state and how many of each type of truck had been in a collision in the past six years.

It was difficult to find the number of registered trucks throughout the US, but we did find this data for New York state[20]. We processed this to get data on all trucks as defined by a weight class of 14,001 pounds and above. To directly correlate this to crashes we can only look at crashes involving trucks registered in New York.

[18] https://vpic.nhtsa.dot.gov/mid/home/displayfile/32488
[19] https://vpic.nhtsa.dot.gov/about.html
[20] https://data.ny.gov/Transportation/Vehicle-Snowmobile-and-Boat-Registrations/w4pv-hbkt/data

# Appendix C: Investigating ZED

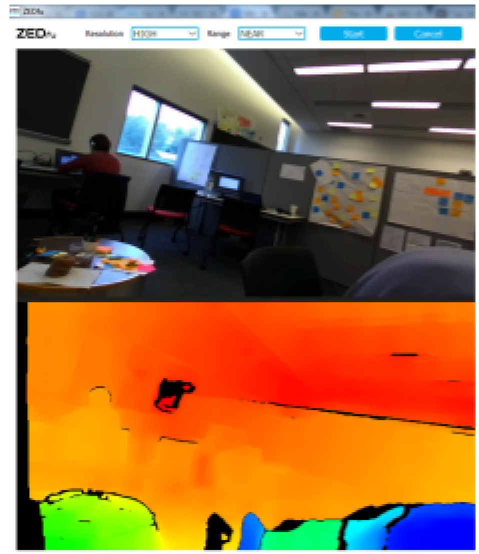


*Figure 15: Real time ZED scanning data*

While determining how we could used ZED scans, we found a tool in Rhino that allowed us to import point cloud data as a CAD file and use it to visualize the field of view of a vehicle. If we could get point cloud data of the field of view, we could generate a CAD of the field of view from the driver's perspective. We thought this second option would be more valuable in terms of information, as well as faster than plotting out the field of view manually. To this end, we found a stereoscopic camera from ZED Labs which is capable of generating point clouds. We planned on using the point cloud generating ability to scan trucks and determine their fields of view in Solidworks.

We ran into some issues with the ZED camera due to a couple troublesome hardware/software requirements. The automatic install wasn't very clear on what it required and regularly failed to download dependencies correctly. The software required a Nvidia graphics card and seemed to only work with Windows 7/8. Through extensive testing, we figured out a way to get around these issues and work with materials we happened to have. To do this we manually installed required software and used a team member who had a Nvidia graphics card by chance, and used the SCOPE desktop, despite it having Windows 10, for the heavier processing. An example of the information that is generated by the ZED is shown in Figure 15.

## Preparing the Hardware

In order to use the ZED stereoscopic camera for scanning, it must be connected to a laptop with a Nvidia graphics card and running Windows 7/10 or Ubuntu 14.04. You must have physical access to the vehicle you want to scan in a still environment. An indoor environment is preferred for lighting consistency, but the ideal environment has no strong directional lighting.

## Performing a Scan

To scan, connect the ZED camera to a laptop then start up the ZEDfu software. Change the settings to allow for a raw scan to be generated, and begin a live scan with the camera aimed at the desired truck. Slowly move around the truck, allowing the software to generate a point cloud. The plot on the right side of the screen will populate with a mesh of the truck. Hold the camera still until a portion of the mesh appears, adjust the angle of the camera to capture more pieces of the mesh. Continue moving around the truck and to areas where a mesh has not been generated. If there is too much motion, or the motion is too fast, the ZED will lose track of where it is, and the user will have to return to a previously scanned point to continue the scan. The software has a maximum memory capacity that is reached after about 15 minutes of continuous scanning, so the scan must be done as efficiently as possible.

## Analyzing a Scan

Analyzing the scan data or a truck CAD produced from it is an in-depth process, though Rhino is the most convenient software we have found for this process. It requires a Windows computer, with minimum requirements listed on the Rhino website[21], as well as an operator somewhat familiar with CAD software. The more familiar the operator is with Rhino, the quicker this process will likely be, but for someone who has done it a few times our time estimate is around an hour or two.

Rhino can take in point clouds[22], STL, Solidworks models, and many other common forms of 3D data. Rhino appears to be significantly faster in displaying and handling point clouds and complicated STLs than Solidworks. Before analyzing the visibility of a truck model or scan, the data must be converted to what Rhino calls a "mesh". Point clouds can be converted to mesh with the Mesh From Points plugin[23], while most other formats can be converted with the Mesh[24] command. Before creating the mesh, unnecessary points or shapes can be deleted, and after creating the mesh the surfaces can be trimmed.

Next, determine the driver's eye position coordinates. Before doing this, it is recommended but not necessary to align the model to Rhino's preferred orientation, so the Front, Top, and Right views will be properly named, and to set the CPlane origin to the driver's eye position. This should allow you to input its coordinates as "0" in the command line, rather than typing out coordinates.

Double click the small box in the upper left of the "Perspective" window to maximize this view. Position the camera at the driver's eye position by inputting the coordinates you determined in the last step into the X, Y, and Z location fields for the Camera in the Perspective tab of the sidebar on the right. Use Control-Alt-Right Click and drag to change the view direction without

[21] https://www.simplyrhino.co.uk/support/hardware-operating-system/rhino-for-windows
[22] https://docs.mcneel.com/rhino/5/help/en-us/fileio/points_file_asc_csv_txt_xyz_cgo_import.htm
[23] https://www.rhino3d.com/download/rhino/5.0/meshfrompointsv5x64
[24] http://docs.mcneel.com/rhino/5/help/en-us/commands/mesh.htm

moving the camera position, and use the Camera Length option in the sidebar to adjust the field of view so that one of the windows is fully visible. Saving this as a named view is recommended.

In the command line, use the command MeshOutline. When prompted for meshes, use the command SelVisible. Drag a box to select only the meshes immediately surrounding the window. If the window itself is a mesh or surface rather than an opening, make sure it is not selected. If the mirror or part of the hood is obstructing view through the window, make sure that the meshes describing that area are also selected. Control-click any unwanted meshes to remove them from the selection. Having more meshes selected significantly slows down performance of this command. Hit Enter when done selecting, and wait for the calculation to complete.

Several closed curves will be created. They will be very small and very close to the camera position, but when viewed from that position they will line up with the visible edges of the window. Delete all but the curve(s) that accurately describe the window outline.

Repeat the camera positioning and mesh outlining for each window.

Create a sphere, centered at the driver’s eye position, to project the view onto. This should be larger than the furthest visible area you are interested in. Since this will surround the truck, you can change its view to “wireframe” independently from the rest of the model using the “Set object shading attributes” button in the Display tab.

Use the Pull command to project the window outlines outward onto the sphere. The reason we use a sphere is that the Pull command projects the curve normal to the surface, which ensures that the view projects outward along the driver’s sight lines. Select first the curves created by the mesh outline, then the sphere. This should produce large curves on the sphere.

Loft from the tiny internal curves to their larger counterparts. Make sure they line up correctly, and use the “straight line” option in the loft. The edges of the lofted surfaces should just touch the window edges.

It should be possible to cap these volumes off and convert them to solids, then find the boolean sum or difference of them and any volumes of interest constructed around the truck, but this has not been tested.

# Appendix D: Panoramic Method V0

## Methodology

The Color Stick is a sixt foot long hollow, rectangular pole. It has six marks of varying colors for scale, at one foot intervals. The markers are as follows:

- 6ft: Green
- 5ft: Orange
- 4ft: Yellow
- 3ft: Pink
- 2ft: Blue
- 1ft: Red

Each colored marker is three inches long and was taped to the pole with the top of each color aligning with the foot intervals as shown in Figure 16.

A 5th percentile woman is 4’11”, therefore, if the orange marker can be seen, the top of a 5th percentile woman’s will be visible. A 95th percentile man is 6’ 2” so if the green marker is visible, a 95th percentile man will be visible to a driver.


*Figure 16: The color stick*

## Measurement

<u>Markers Laid</u>

Person A sits inside of the truck in the driver’s seat, person B and C walk around the truck and using string and chalk mark a path 3ft and 6ft away from the truck’s exterior as shown in Figure 17. Person A tells them where to start and stop based on what they can see. These chalk marks should look something like they do in the image to the left.

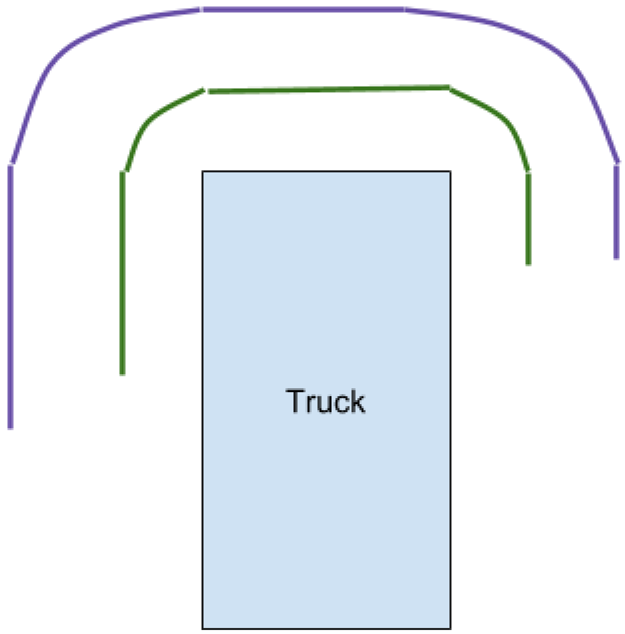



*Figure 17: the green and purple lines are the paths the color stick follows around the truck*

Photo Collection

Person A sits inside of the truck as if they are the driver and uses the Cardboard Camera app to film. They start with the left side of the camera at the edge of the truck's left window. Person B (dressed in dark, muted clothes) stands in the center of the camera's view holding the stick. Person A starts capturing the image and person B moves very slowly and stably along the line keeping the colored side of the color stick facing person A inside of the truck. Person A tracks person B with the camera making sure person B is always in the center of the image. When person B moves out of the visible area, person A stops filming. The process is repeated for each distance. If there are major gaps where the stick is missing or the colors move drastically up and down, redo photo collection process.

Image Processing

Code: [https://github.com/LucyWilcox/VIEW/tree/master/color-stick](https://github.com/LucyWilcox/VIEW/tree/master/color-stick)

Each image is sliced by a width of pixels and OpenCV is used on each of these slices to see what the lowest visible color is. Full points will be awarded if the lowest color (red) is recognized in each slice. The total width the driver can see (total pixel width of the image) will also factor into the score.

# Appendix E: Panoramic Method V1

## Measurement Stick

Any six-foot long object with markers at each foot mark between one and six feet can be used as a measuring stick.

## Photo Collection

With the stick held vertically, six feet to the left of the center of the driver, take a panoramic photo using the Cardboard Camera app on either an Android or iPhone from the point of view of the driver as in Figure 18. The photo should sweep from the leftmost edge of the driver window to rightmost edge of the passenger window at eye height. The purpose is to simulate the visible area to a person sitting in the driver seat looking through each window.

*Figure 18: Panoramic picture taken with the stick held stationary left of the driver*

## Image Processing

Import the image into a computer for further processing. In this program, the the user will identify each foot marker on the measurement stick as a scale to represent the various heights within the image at six feet away. Now that a scale has been identified, the smaller step sizes between the each foot marker can be identified.

The next step is for the user to identify the lowest visible line through the window across the entire image. An example is shown below in Figure 19.

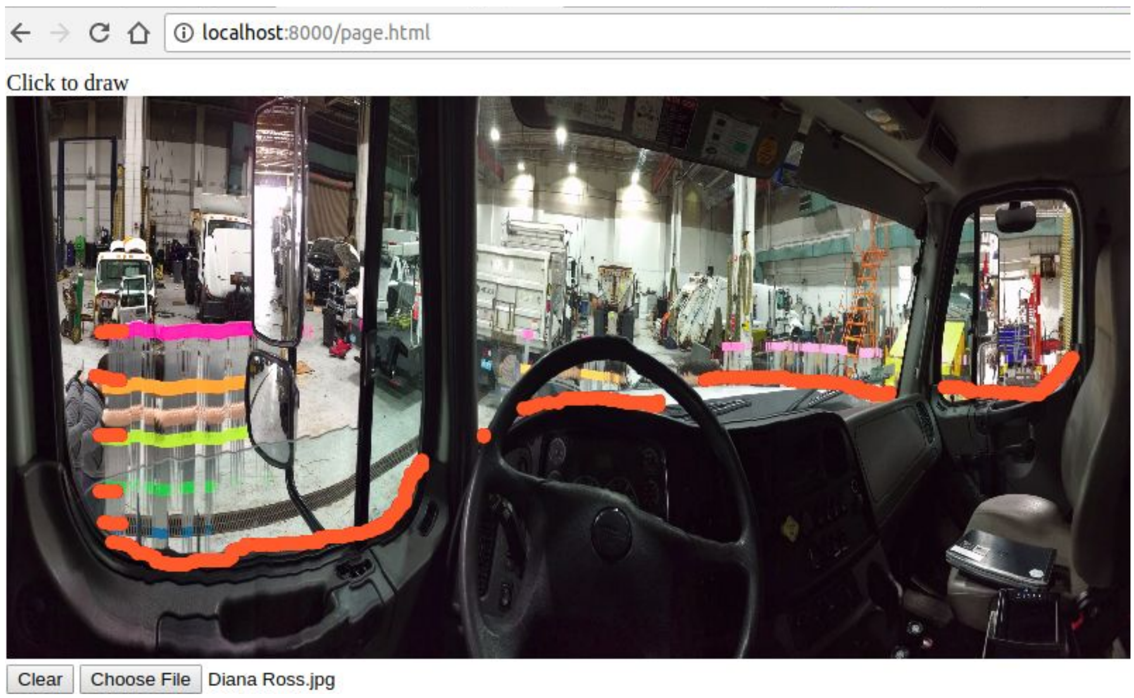


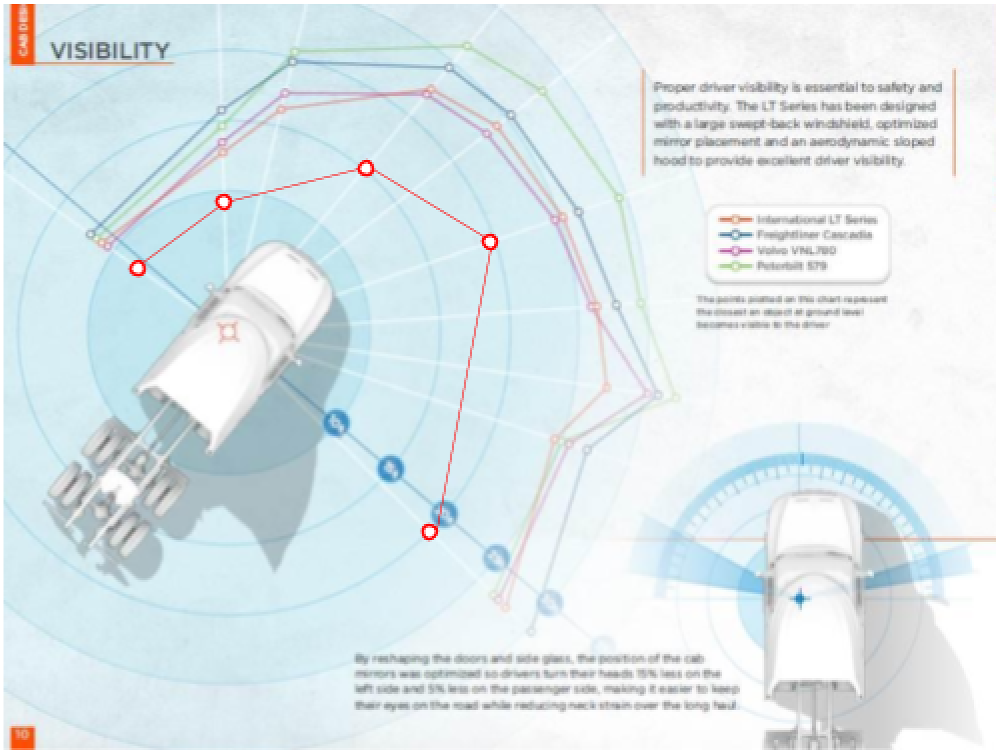

*Figure 19: Panoramic image of a driver's field of view with the lowest visible line marked*

The user drawn line in combination with the scale will identify the lowest height the driver can see. For example, at the leftmost edge of the driver side window, the driver can view the bottom of the blue marker, which correlates with about 21 inches.

If you can imagine the color makers extended throughout the whole image as shown in Figure 20, assuming there is no distortion in the image, a height can be assigned to each instance of the lowest visible height.

*Figure 20: lines from measurement stick on left propagated through the image*

These heights are used to calculate the angle from the driver's eye to ground. The final result is a polar plot of the nearest position on the ground visible to the driver out each window. An example of this same model used in industry is shown in Figure 21 with our measurements of a different truck overlaid.




Red dots represent the nearest floor visibility generated with the color stick method.

*Figure 21: Visibility spider map provided by Navistar with our measurements overlaid in bright red*

The rough estimate in Figure 21 are for angles -90, -45, 0, 45, 90, where -90 is looking out the driver side window, and 90 is looking out the passenger window. At each one of those angles, the distance from the driver at which the ground is visible is calculated with trigonometry. The first step is to find the field of view angle of the driver, which the angle that describes the visible area. This can then be used to find the ground intersection and total blind volume.

The code being used to generate these measurements is shown in Figure 22 and a visual explanation is shown in Figure 23. The color visible is the inch measurement for each corresponding angle.

```
Editor - C:\Users\bwalker\Documents\SCOPE\SemiCircleHeight.m
SemiCircleHeight.m



% Angles are [-90 -45 0 45 90]
% Measured height of the driver eye level.
DriverHeight = [58, 58, 58, 58, 58];
% User detected visible color with the correspoding height measurement
ColorVisible = [10, 18, 42, 42, 36];
Oppositeside = DriverHeight - ColorVisible;
%Distance from the Car
DistFromCar = 72;
%Visible Area Angle from the horizontal
FOVangle = atand(Oppositeside/ DistFromCar);
% Distance from driver at which ground is visible
GroundIntersection = DriverHeight .* tand(90 - FOVangle);
%Convert to Feet
GroundIntersection  = GroundIntersection/ 12
```

*Figure 22: Ground intersection calculation in Matlab.*

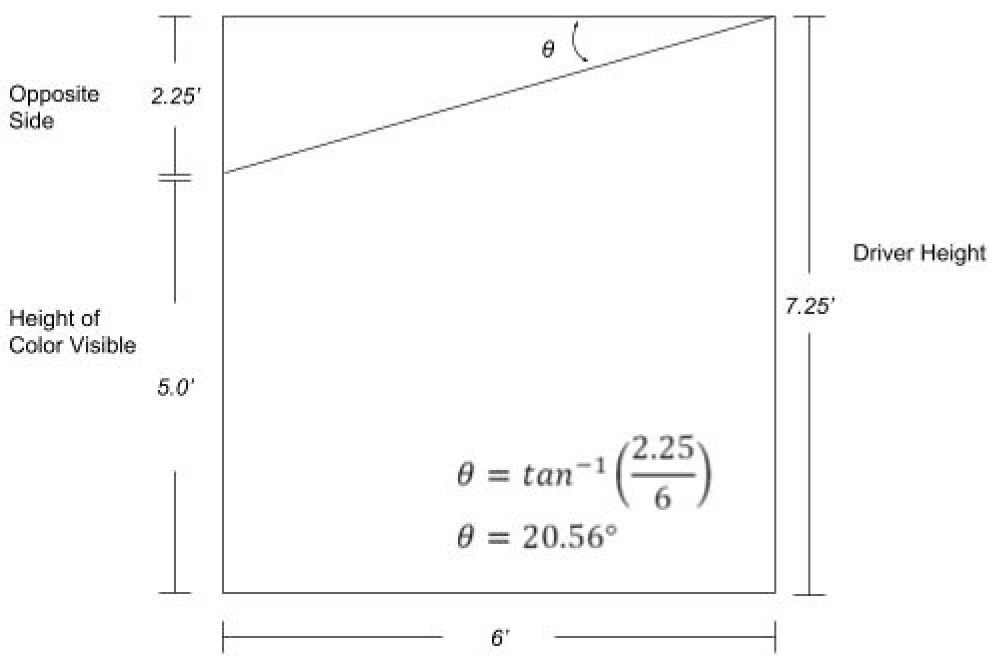



*Figure 23: Geometry of the driver field of view and calculation of the field of view angle.*

# Appendix F: Panoramic Method Proof of Concept

From the initial testing of one truck we were able to verify that determining blind spots this way is plausible, though it needs more verification. We calculated the nearest point to the truck at which the ground could be seen by the driver, which is shown in Table 3. The calculation for the ground intersection is shown in Appendix K.

| Location Angle | Color Visible | Ground Intersection (ft) |
|---|---|---|
| -90 | Blue | 7.9 |
| -45 | Blue | 8.1 |
| 0 | Orange/ Yellow | 15.8 |
| 45 | Orange | 19.4 |
| 90 | Orange | 19.4 |

*Table 3: Ground intersection point calculated from lowest visible color.*

Our proof-of-concept used five angles, with -90 being the view out the driver-side window, 0 is looking forward, and 90 is the view out the passenger window. We then generated a field of view and calculate ground intersection of the driver's view as shown in Figure 24.

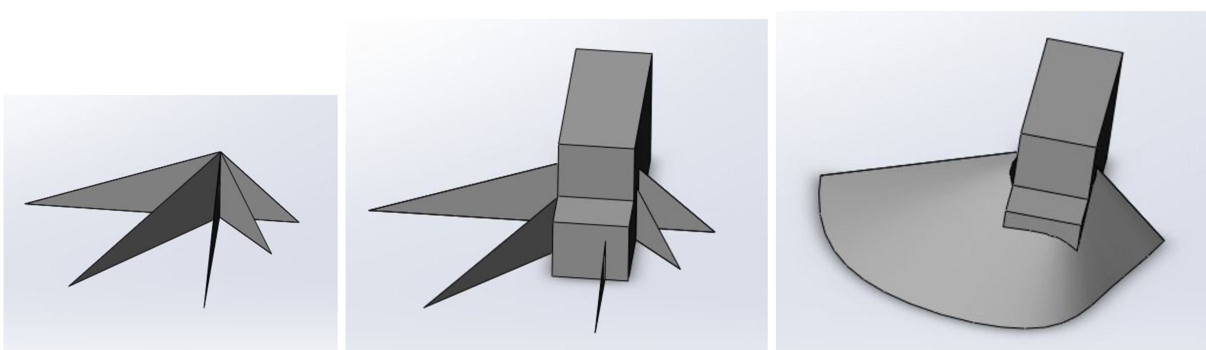


*Figure 24: Generation of a blind spots in Solidworks from Panoramic Method data*

We plotted the ground intersections measured over a field of view map we received from Navistar (shown in Figure 24). We found that our method yielded a similar shape, but that drivers of the truck that we tested (we think it was a Freightliner or International, but we forget) would see the ground closer than in the other trucks. This difference may have been a result of the variation in truck size, assumptions such as driver height, or errors in measurement.

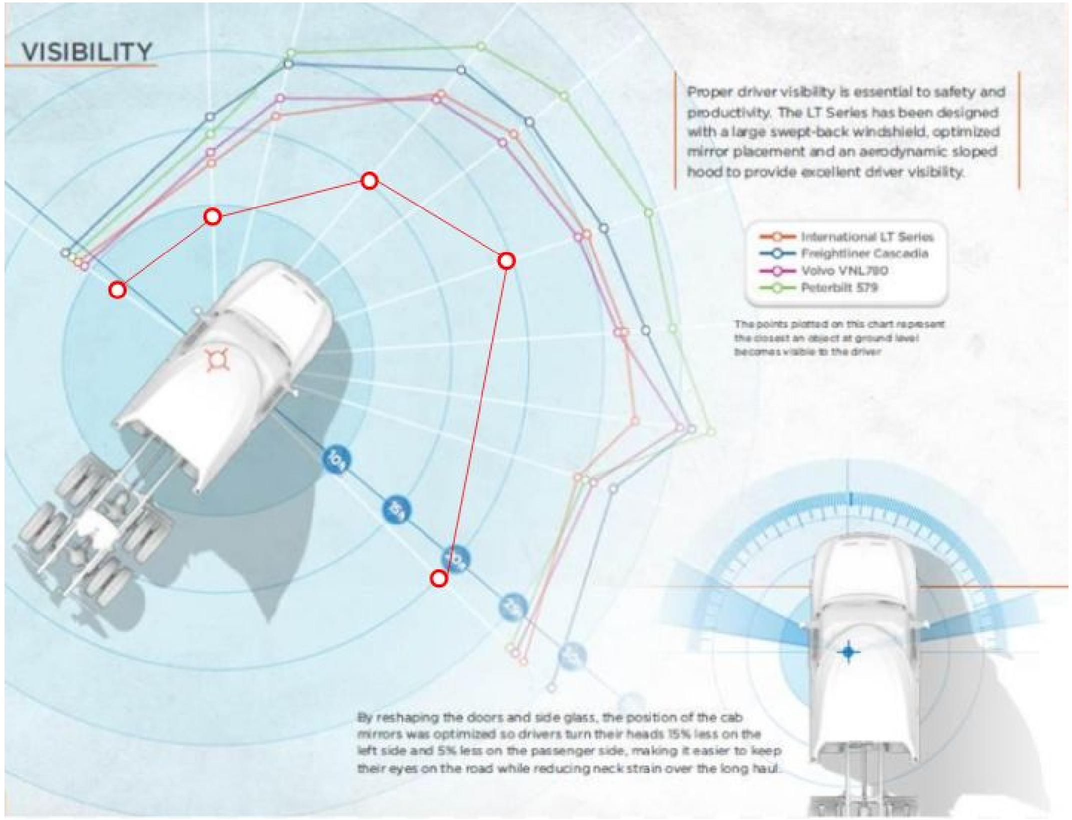



*Figure 25: Field of view map from Navistar with our data overlaid in bright red. The red dots represent the closest distance at which the ground was visible. These locations have been calculated with the “Diana Ross” method.*

# Appendix G: LiDAR Testing Pictures

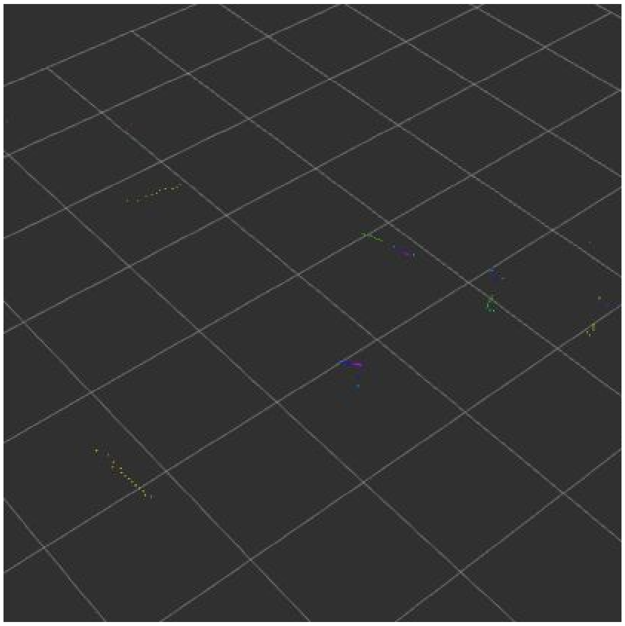


*Figure 26: LiDAR scan of walled corner through windshield tinting.*

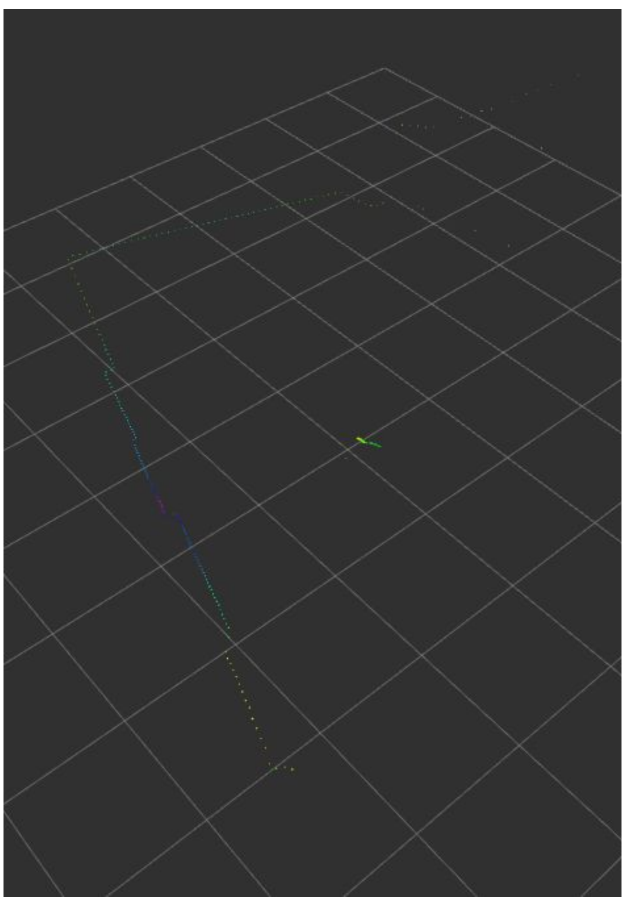


*Figure 27: LiDAR open air scan of walled corner.*

# Appendix H: Automated Scanning

## Required Materials

This methodology requires the following supplies to construct:

- Pan/tilt setup
- Arduino
- Tripod
- LIDAR or Kinect
- Computer (for collecting the data)
- Battery power supply

Alternatively, the entire setup can be bought as a single unit, but doing so substantially raises the total cost.

## Construction

Assemble the pan/tilt setup and attach it to the tripod. Attach the LiDAR or Kinect to the top of the pan/tilt, and attach everything to Arduino and power.

## Data Collection

Place set-up on the driver seat of the vehicle so that it will scan 180 degrees horizontally and about 50 degrees vertically. Activate panning code and allow data collection to take place. Once collected, the data can be processed programmatically to remove unimportant/redundant data points and leave the user with a point cloud of visible points from the driver's perspective. This can then be evaluated in terms of both areas of visibility and proximity to the truck to calculate a visibility metric for the vehicle.

# Appendix I: Automated Scan Power Considerations

## Powered Components - All on 11.1V LiPo power supply

| Component | Quantity | Power (W) | Voltage (V) | Current (A) | Notes |
|---|---|---|---|---|---|
| Dynamixel AX-12A | 1 | N/A | 11.1 | 0.050 standby<br>0.900 max | Stall torque 1.5Nm @ 12V 1.5A |
| ArbotiX-M Robocontroller | 1 | N/A | 11.1 prefered<br>7-12 functional | *0.900* | |
| TiM511 LiDAR | 1 | 4 expected<br>7 max | 12 | <u>0.333</u> expect<br><u>0.583</u> max | |

*Table 4: Power information for electricity requiring components. Note: values that were <u>calculated</u> are <u>underlined</u>. Values that were* assumed *are* italicized.

<u>Battery Choice</u> - 11.1V 3s LiPo (3 cell Lithium Polymer battery (chosen in part because it is commonly used in robot kits with the chosen servo and controller). 900mAh, 2200mAh, and 4500mAh available.

## Assumptions Equations Calculations

- Power consumption for the ArbotiX-M controller was unlisted. Assuming consumption will not exceed the 900mA normal max current for the Dynamixel AX-12A. Currently this is an unvalidated assumption. Can be validated by contacting Trossen Robotics
- Current for the TiM511 LiDAR were calculated assuming zero inductor or capacitance components in the sensor circuitry so that the power factor could be set as 1 in the equation amps = watts / (*PF* × volts) found here on rapid tables: https://www.rapidtables.com/convert/electric/1-watt-to-amp.html
- To find battery life (hours) we divide the battery rating (mAh) by required current (A).

## Results (Max, expected consumption; corresponding battery life)

| Scenario | Required Current (mA) | Battery (mAh) | Active Lifespan (Hours) | Active Lifespan (Minutes) |
|---|---|---|---|---|
| Maximum consumption | 2983<br>(1500+900+583)<br>[ S + RC + L ] | 900<br>2200<br>4500 | 0.37<br>0.92<br>1.89 | 22<br>55<br>113 |
| Expected Consumption | 2133<br>(900+900+333)<br>[ S + RC + L ] | 900<br>2200<br>4500 | 0.42<br>1.03<br>2.11 | 25<br>61<br>126 |
| Standby Consumption | 1283<br>(50+900+333)<br>[ S + RC + L ] | 900<br>2200<br>4500 | 0.70<br>1.71<br>3.51 | 42<br>102<br>214 |

*Table 5: System power consumption in different conditions. Abbreviations stand for: S- Servo, RC - Robo(servo)controller, L - LiDAR*

# Appendix J: Panoramic Method V2, Final Method

## Summary

This procedure will determine the visibility from the driver's seat of a vehicle and locate blind spots to understand and improve vehicle safety. A smartphone and a few measurement devices are required to conduct this test. A measurement stick, used as a scale, is placed in front of the vehicle. Next a panoramic photo is taken with a smartphone from the driver's seat that captures the visibility through the windshield and passenger window. The location of the camera and the measurement stick are recorded. With the photo and known location of the devices, it is possible to calculate the approximate visibility a driver may have for this particular vehicle.

## Requirements

Image Capturing: iPhone or Android smartphone with panorama photo capability.
Image Processing: Computer with internet access.

## Materials

- Vehicle
- Smartphone
- Measurement pole
- Camera stand with smartphone mount
- 2 Levels
- 3 foot long Bar
- 2 Tape measures
- Masking Tape
- Chalk

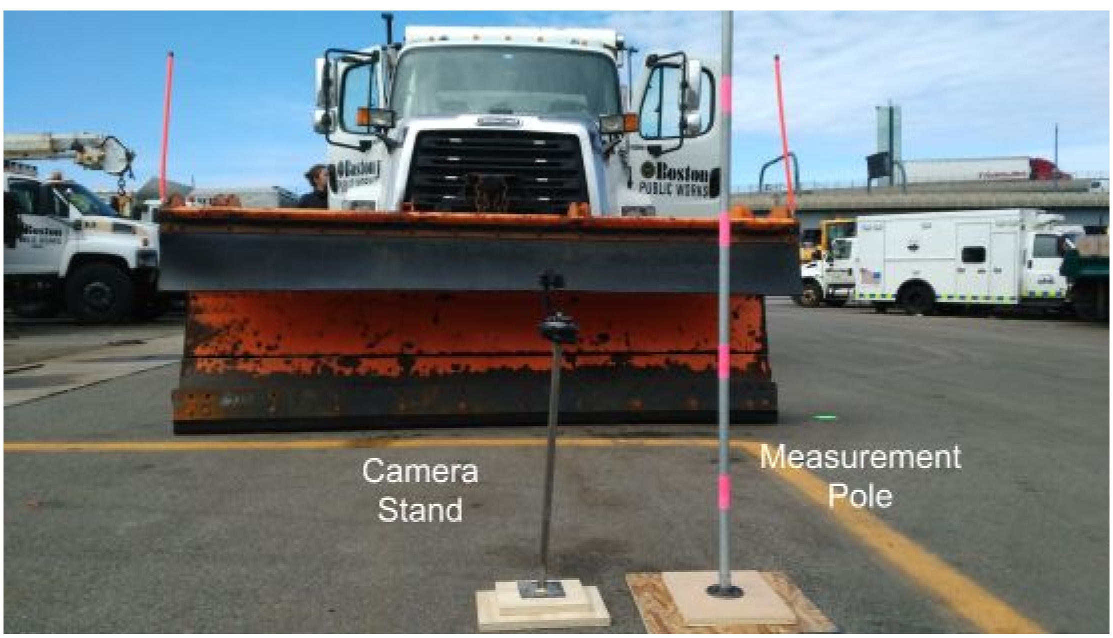



*Figure 28: Measurement pole and camera stand in front of a Class 8 snow-plow truck at City of Boston for perspective.*

## Environment

To perform this test, park the vehicle in a location where the ground is level and the driver has a line of sight to ground directly in front of them. There should be at least 10 feet of clear space in front of the vehicle as well as enough room to comfortable move around the vehicle and open doors.This will increase the accuracy of any measurements taken in the process.

## Set Up

1. Place measurement pole on the ground in front of the vehicle. The pole should be in line with driver's seat as shown in the picture below. From the driver's perspective the base of the pole should not be visible. If the base is in view, move the pole closer to the vehicle until the first mark from the bottom is just visible.


*Figure 29: Cab over truck with the measurement pole placed directly in front of the driver.*

*Figure 30: View from driver's seat with the measurement pole placed correctly. The base is not in sight, and the first mark from the bottom is visible.*

2. Place the camera stand in the seat so the base plate is firmly against the backrest with the camera pointing towards the windshield.

*Figure 31: Camera stand placed in the driver seat with the base firmly against the backrest.*

3. Adjust the leveling table to center the bubble by unhinging the lever and rotating the ball joint. Push the lever closed to lock in place. The mount does not need to be completely level at this stage, a rough estimate will be satisfactory. Fine tune leveling will be done at a later step.

4. Place the phone vertically in the camera stand (avoid pressing the side buttons on the phone with the mount).

*Figure 32: Smartphone in the phone mount in the portrait orientation.*

## Measurement & Recording

Now that the phone mount and measurement stick are in place. There are four measurements that are important to record along with the vehicle information. Each measurement will be taken starting from the camera lens on the smartphone. All lengths will be vertical or horizontal, do not record any diagonal measurements. Finally, it is important to mark a point on the passenger window that is directly to the right of the lens, a 90 degree angle.

A. Lens height
B. Lens to passenger window
C. Lens to measurement pole
D. Lens to front of vehicle

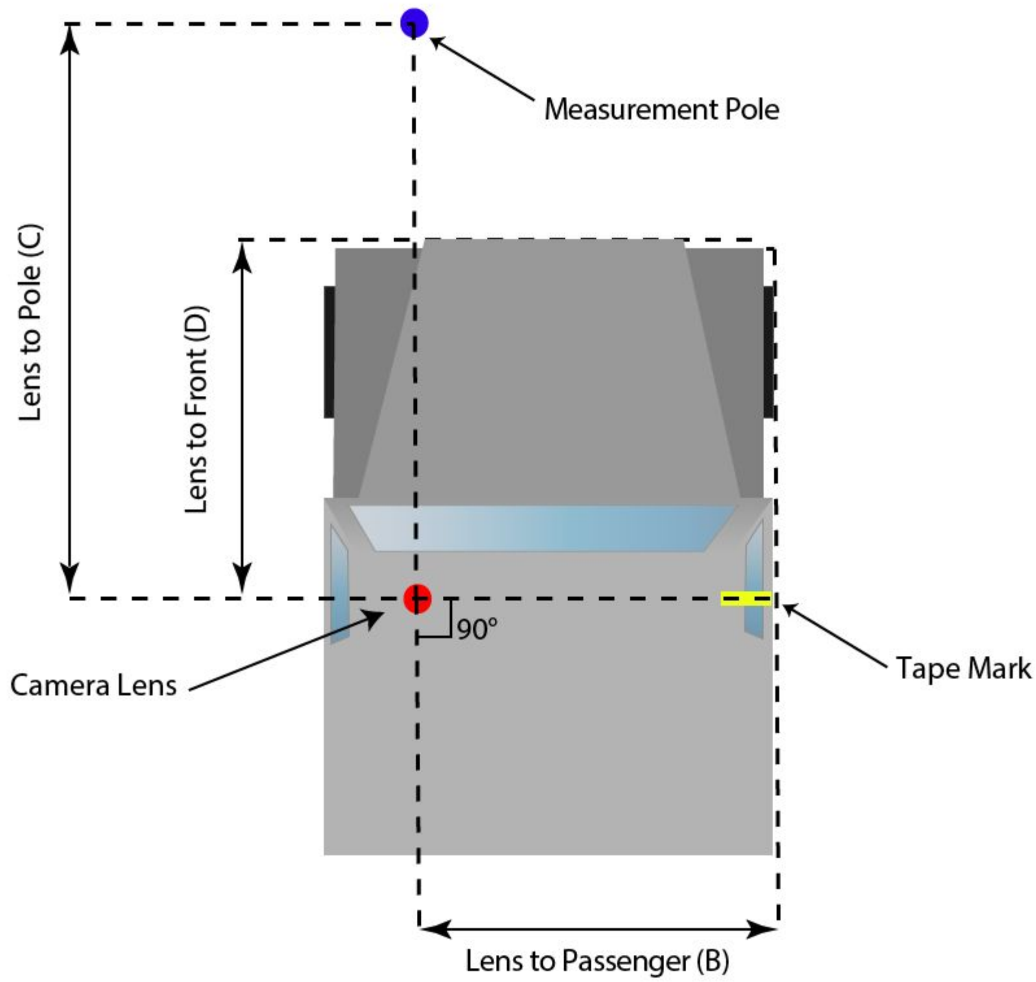



*Figure 33: Top view of important locations, angles, and measurements in the Panoramic Method*
*Note: Not necessarily to scale.*

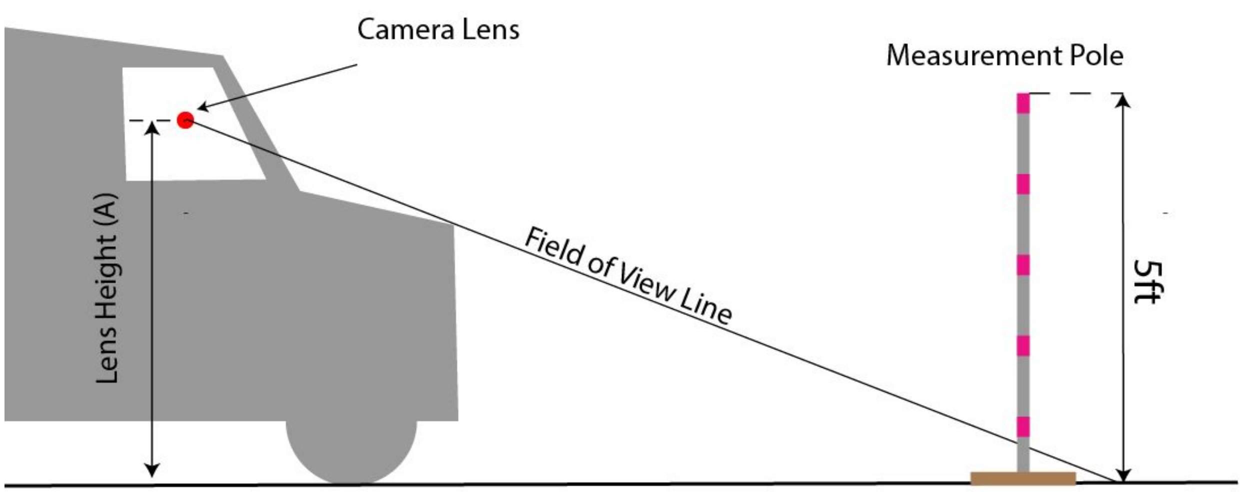


*Figure 34: Side view of important locations and measurements in the Panoramic Method. Note: Not necessarily to scale*

5. Keep note of the vehicle type, and any modifications if present, along with the VIN or partial VIN number. (Optional)

6. This step requires two people. Stand on the driver side of the cab. Hold a 3' bar horizontally at the height of the camera lens. Hold a level with the bar to make sure the bar is level. Measure the height of the level from the ground. Mark the point on the ground with tape or chalk.

*Figure 35: Two people taking the lens height measurement. One person holds the 3' bar level at the lens while the other measures the height of the bar. Mark the point on the ground with tape or chalk.*

7. Measure the distance from the camera lens to the passenger window.
    a. Measure the distance from the back window to the lens and use this to place the tape on the passenger side window as shown in the diagram below.
    b. Next measure the distance from the lens to the tape along the dotted line.

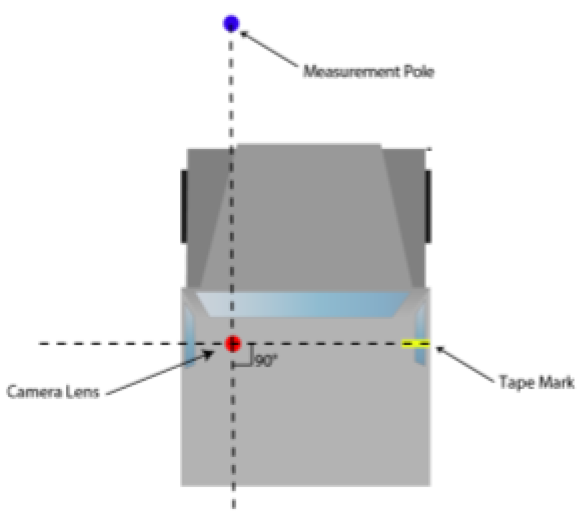



*Figure 36: Place tape on the passenger window in line with the lens on the camera. Use the tape measure place the tape at the same distance away from the back window as the camera lens.*

*Figure 37: Tape on the passenger window marking 90 degrees from the driver forward view.*

8. Measure the distance from the lens to the measurement pole along the ground. Start at the tape/ chalk mark on the ground from step 7. Keep the tape measure parallel to the vehicle as shown in the image below.

*Figure 38: Distance from the tape on the ground to the measurement pole. Keep the tape measure parallel to the vehicle.*

9. Measure the distance from the tape/ chalk mark in step 7 to the front of the cab (front bumper), record this for later.


*Figure 39: Distance from the mark on the ground to the front bumper of the vehicle.*

## Panorama

10. Open the camera app on the smartphone and switch to the panorama setting. Keep the phone mounted on the stand if possible.
11. Level the camera stand by adjusting the lever. Grab a bar level and place line it up with the top of the phone to check how level the phone is. If the phone is not level, use the knob on the back of the phone mount to rotate the phone until it is level.
12. Take a panoramic photo starting at the left edge of the front windshield. Swivel the phone around using the phone mount until the right edge of the passenger window is in view. Do not rotate the leveling table.

*Figure 40: Start image capture at red line sweeping from left to right.*

*Figure 41: Example panorama of the windshield and passenger window.*

13. Save the photo.

This is the end of the image and measurement collection portion. Take care to save your image and keep track of the four measurements taken.

## Processing

1. Navigate to the webpage: https://blindspotcalculator.herokuapp.com/
2. Follow directions in this video tutorial: https://youtu.be/IYucUl6kP3w

## Bill of Materials

| Component | Quantity | Price |
|---|---|---|
| Standard 3/8"-16 Female to 1/4"-20 Male Tripod Thread Reducer Screw Adapter (Brass) Precision Made (2 Pack) | 1 | $6.34 |
| Universal Smartphone Tripod Adapter Cell Phone Holder Mount Adapter, Rotates Vertical and Horizontal, Adjustable Clamp | 1 | $7.99 |
| Manfrotto 438 Compact Leveling Head | 1 | $91.88 |
| 3 ft Square Tube, part #: 6527K254 | 1 | $8.93 |
| Base Plate, part #: 1388K17 | 1 | $30.85 |
| 1/4-20 Bolt, part #: 90128A241 | 1 | $4.39 |
| Marine-Grade Plywood Sheet, part #: 1125T31 *** | 1 | $10.89 |
| 5’ 1/2” OD Pipe *** | 1 | $17.82 |
| 3/4” EMT Set Screw Connector *** | 1 | $0.29 |
| 3/4” Floor Flange with 4 holes *** | 1 | $6.09 |
| 3/ 8” Woodscrews*** | 4 | $0.25 |
| Total: | 14 | $185.72 |

*Table 6: Bill of materials for Panoramic Method. Note: if the camera stand is not used, the measurement pole can be made with the starred (***) components for a total cost of $35.34.*

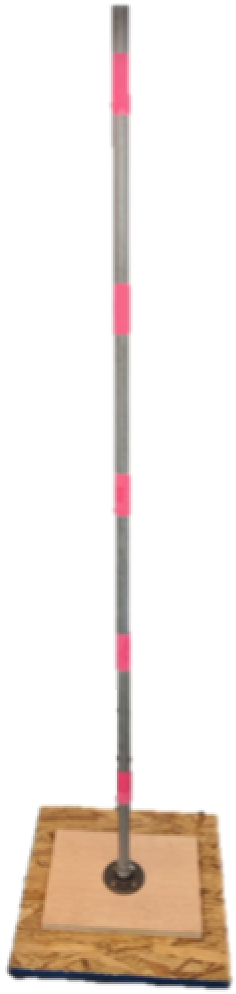


*Figure 42: Measurement Pole*

*Figure 43: Camera Stand*

# K Appendix K: Math

## K.1 Ranking System Overview

The numerical basis of our ranking system is the percentage of a defined volume of interest around the truck that is visible by direct line of sight to the driver. It only addresses the view through the windshield and the passenger-side window.

### K.1.1 Volume of Interest

The volume of interest is the area where direct vision is the most important, the positions around the truck where vulnerable road users tend to be located shortly before being hit by the truck. It is defined as the volume between 0 and 6 feet, 2 inches above the ground, 0 and 15 feet in front of the front bumper, and 0 and 15 feet horizontally from the passenger side of the truck.

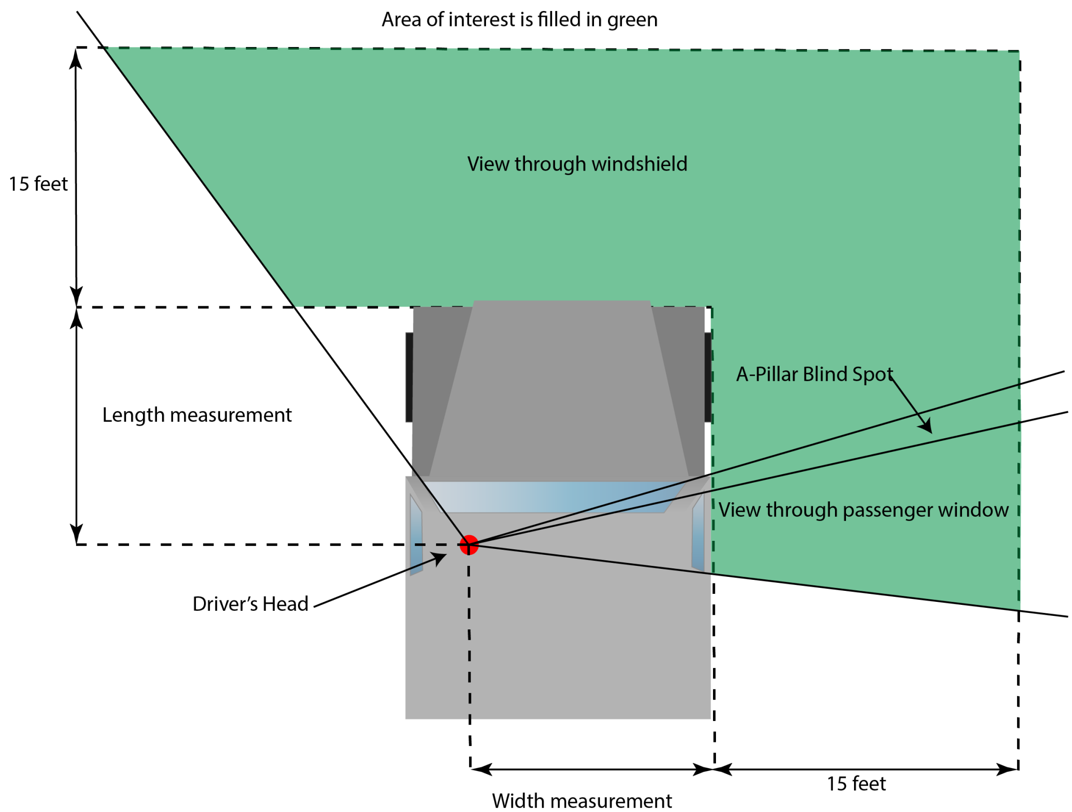



Figure 44: The area of interest around the truck, seen from above. Image is not to scale.

We chose the forward distance of 15 feet very roughly based on Transport for London's research into the positioning of pedestrians and trucks before the low-speed impacts in which direct vision is most important. They found that a truck driver could react to avoid a collision with a pedestrian they could see directly in about 0.75 to 1.5 seconds, around 0.7 seconds faster than a driver who could only see the pedestrian in a mirror. [25] A truck traveling 10mph will cover 15 feet in one second. In the same time, an average pedestrian

[25] http://content.tfl.gov.uk/assessing-drect-vision-in-hgvs-technical.pdf

crossing the road at 3mph will travel 4.4 feet. The space within 15 feet of the truck is therefore an area where the faster reaction time provided by direct vision can potentially save many lives.

In later iterations, TfL reduced their forward area of interest distance to 6 feet. However, drivers of standard US trucks (as opposed to the UK's cab-over trucks) commonly cannot see the ground nearer than around 20 feet ahead, so this extended area is likely to be more important in distinguishing between the US's truck models.

The distance from the passenger side is directly taken from TfL's rating system. For comparison, road lanes vary between 9 and 15 feet.

The upper limit at 6 feet 2 inches corresponds to the 95th percentile height of men in the US. As most people in the U.S. are below this height, our area of interest has a vertical limit to avoid incentivizing direct vision of areas that will not help pedestrians. Trucks are already designed for long-distance vision along roads, and we do not expect any truck to limit its driver's vision to a ceiling below this.

The angular extent of the area of interest stretches from the lower left corner of the windshield to the lower right corner of the passenger window, primarily for ease of calculation. The driver's side window is excluded because TfL's analysis of crash statistics indicates that significantly fewer pedestrians and bicyclists occupy the blind area and are struck on this side. In addition, because the driver is much closer to the window on this side, the corresponding blind spot is much smaller.

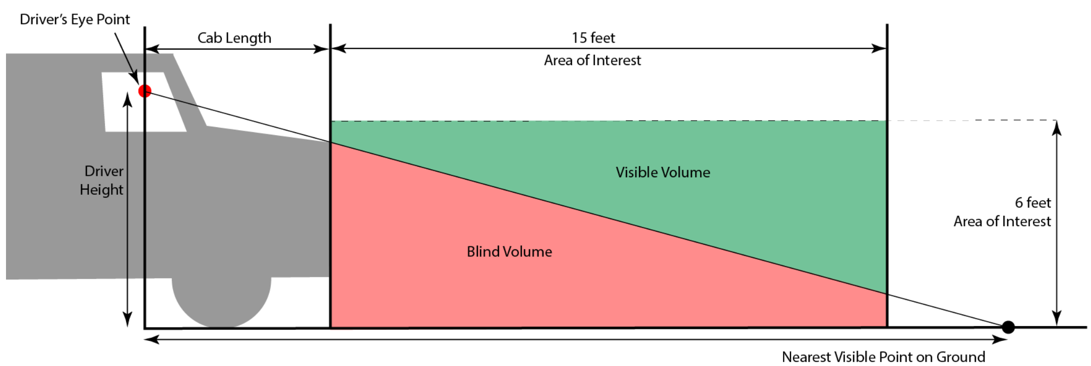



Figure 45: The area of interest in front of the truck seen from the side. Image is not to scale.

### K.1.2 Inputs

In order to determine the volume of the blind spots around the truck, several measurements must be made and entered into the algorithm. When possible, we used the measurements that would be easiest to take.

The measurements required to calculate a ranking via our system are:

- Driver's eye height above the ground (assuming a 50% height driver with the seat adjusted to their usual driving position)
- Horizontal distance from the driver's eye to the front bumper of the truck
- Horizontal distance from the driver's eye to the passenger side of the truck
- The nearest visible point on the ground visible at an array of angles through the windshield and passenger side window.

Rather than directly measuring the nearest visible points at an array of angles, users upload a photo like the one seen in Figure 46 to our website. The website's interface asks the user to draw a line along the lower bound of the field of view for each window. This is interpreted as a series of points corresponding to angles around the truck.


Figure 46: The website's interface asks the user to draw a line along the lower bound of the field of view for each window. This is interpreted as a series of points corresponding to angles around the truck.

The photo also contains the measurement stick, which is placed directly in front of the driver at a known distance $k$. The user marks the position of the stick and its height marks in the photo, as well as an approximation of which direction is straight out the passenger side window. Extrapolating from this information, each row of pixels is assigned a corresponding height off the ground (at distance $k$ from the camera) and each column is assigned an angle. For rows between measuring stick marks, a linear interpolation is used to account for perspective. For rows beyond the measuring stick, the scale of the closest (top or bottom) measuring stick division is used.

We have experimentally confirmed in the "Panoramic Method Updates" section of this paper that a horizontal feature at a constant distance from the camera results in a horizontal line in the panoramic image.

#### K.1.3 Determining Nearest Visible Point

For each given column of pixels in the image, the height of the bottom edge of the field of view at the distance of the measuring stick is known. We construct the horizontal distance to the corresponding point on the ground with similar triangles. As shown in Equations 1 and 2 below, the height at the measuring stick, $h$, is converted into the angle $\phi$ of the sightline relative to vertical. The tangent of its complement is then multiplied by the driver's height off the ground, DH, to find the distance NVP.

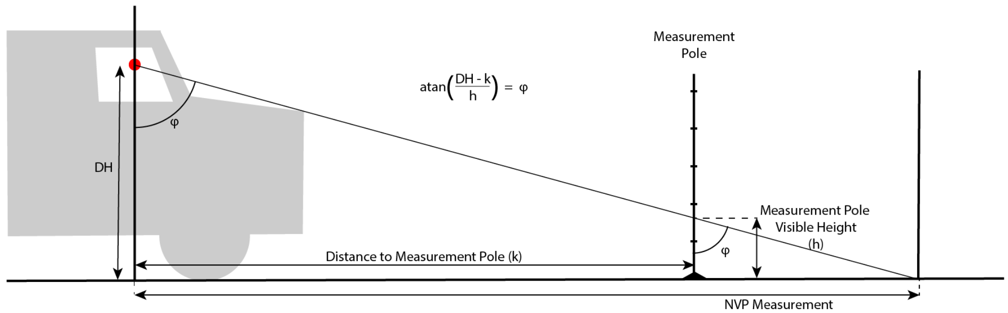



Figure 47

$$\phi = \text{atan}\left(\frac{\text{DH} - k}{h}\right) \tag{1}$$

$$\text{NVP} = \text{DH}\tan\left(\frac{\pi}{2} - \phi\right) \tag{2}$$

### K.1.4 Approximations

The code approximates the total volume inside the volume of interest, excluding the volume that is physically inside the truck, as no pedestrian can occupy this space.

This entire space can be represented by the integral

$$\int_{\theta_p}^{\theta_w} \int_0^C \int_b^R 1r\,dr\,dh\,d\theta \tag{3}$$

where $\theta_p$ is the angular position of the passenger window's lower right corner, $\theta_w$ is the windshield's lower left corner, $C$ is the ceiling of 6 feet 2 inches, $b$ is the radial distance between the driver and the outer surface of the truck as a function of $h$ height off the ground and $\theta$ angle, and $R$ is the radial distance between the driver and the boundary of the area of interest. As these are polar coordinates, an additional coefficient of $r$ is integrated.

This is a very complicated calculation, and as we do not have access to a mathematical definition of $b$, the outer profile of the truck, as in a CAD model, we must make several approximations.

First, we approximate the volume the truck takes up within the area of interest as a rectangular prism defined by the width of the cab from the driver's head to the passenger side and the length from the driver's head to the front of the bumper. As shown in Figure 44 the rectangle representing the vehicle continues indefinitely to the left and rear of the cab, as well as vertically. The vertical extension represents the physical reality that humans should not be located above or below the hood or bumper of a cab. Though some cabs have rounded front bumpers and protruding fenders around the wheels, a rectangle is generally a good approximation, and the measurements are relatively intuitive and easy to take. We leave users to determine what they consider the extent of the front bumper and the side of the truck. As this is only meant to eliminate locations where a pedestrian is unlikely to stand, we can consider a layperson's estimate of the extent of the bumper to be comparable to a pedestrian's.

After this assumption, we can form functions that represent the boundaries of this unique area of interest for each truck. As all our measurements are relative to the driver's eye position, these functions are in cylindrical coordinates $(r, \theta, z)$ where $r$ is the horizontal distance between the driver's eye and a point, $\theta$ is the counterclockwise angle relative to straight ahead, and $z$ is the height off the ground. In the equations below, the width measurement is $W$, the length measurement is $L$, and the driver's eye height is $H$. All length measurements are in feet, though any length unit could be used if the 15-foot area of interest limits are converted to the desired unit. All angular measurements are in radians.

The boundary of the front hood and right side of the truck are represented by the planes

$$R = \frac{L}{\cos(\theta)} \tag{4}$$

$$R = \frac{W}{\cos(\theta + \frac{\pi}{2})} \tag{5}$$

respectively for all H, which intersect at the corner of the truck rectangle at

$$\theta_{co} = \text{atan}\left(\frac{W}{L}\right) \tag{6}$$

Likewise, the forward and passenger side limits of the extent of the area of interest are

$$R = \frac{L + 15 \text{ ft}}{\cos(\theta)} \tag{7}$$

$$R = \frac{R + 15 \text{ ft}}{\cos(\theta + \frac{\pi}{2})} \tag{8}$$

respectively for all H, meeting at

$$\theta_{ci} = \text{atan}\left(\frac{W + 15 \text{ ft}}{L + 15 \text{ ft}}\right) \tag{9}$$

The other boundaries align with a given sightline and represent a plane of constant $\theta$.

$$\theta_{left} = \theta_{data} + \theta \tag{10}$$

$$\theta_{right} = \theta_{data} - \theta \tag{11}$$

$$\theta = \frac{\theta_w}{2} \tag{12}$$

In order to integrate and determine the blind and visible segments of this volume we would need an integrable equation for the edge of the visible space. What our data input method gives us is a list of discrete points, so we perform the 3D equivalent of a midpoint Riemann sum on the volume. As a typical input from the website contains around 200 points along the line, the approximation is relatively good. The number of points gathered depends on the speed at which the user draws, so the angular spacing between them is not necessarily consistent.

The angular width $\theta_w$ of a given slice is the angular distance between the current and preceding measurements. We approximate the "slice" of the area of interest (before subtracting the volume inside the truck hood) around a given angular measurement as a prism with an isosceles triangle base of height $N$, where $N$ is the radius found in Equation 7, angular width $\theta_w$, and a vertical height of $H$. Rather than using cylindrical coordinates, further calculations on the volume will be derived from sums of geometric areas. For ease of calculation, the geometric area calculations will be for one half of the resulting volume, so the base area is a right triangle.

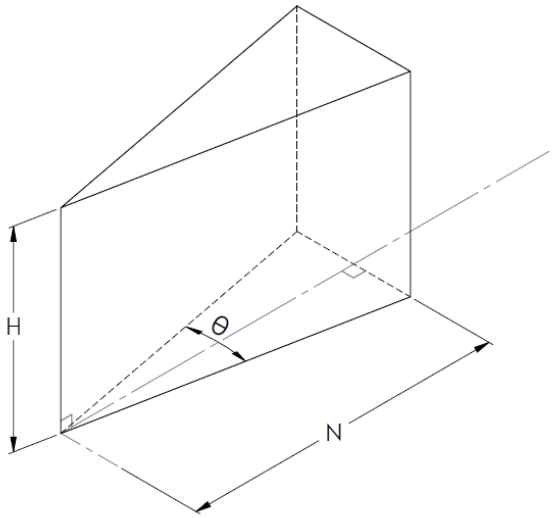



Figure 48: The entire area of interest around a radial measurement along the dotted line. $\theta$ in this diagram represents $\theta_w$. The volume of the hood has not been subtracted from the volume of interest.

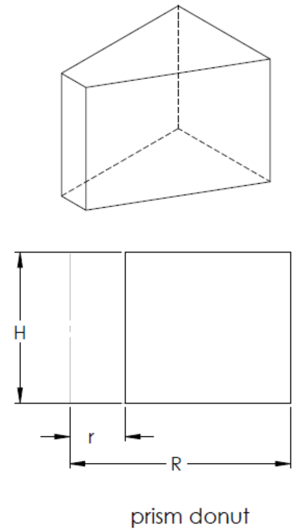



Figure 49: Half of the area of interest, with the volume within the hood subtracted. The horizontal distance to the bumper is $r$. The base angle is $\theta$, half of $\theta_w$

For each measurement we know an angular position and the horizontal radial distance between the driver and the nearest point on the ground that is visible, determined from the photographic input in Section K.1.3. We use this and the dimensions of the hood (from Equation 4 and overall slice at that angle to determine the equation to use for the blind portion of the area.

The total volume of the area of interest minus the volume behind the bumper for every case is

$$V_{total} = \frac{R^2 - r^2}{2} H \tan(\theta) \tag{13}$$

#### K.1.5 Case Breakdown

Our code classifies each slice of the volume into one of eight cases based on the shape of the blind volume being calculated. The flowchart in Figure 50 summarizes the distinguishing characteristics used to categorize slices.

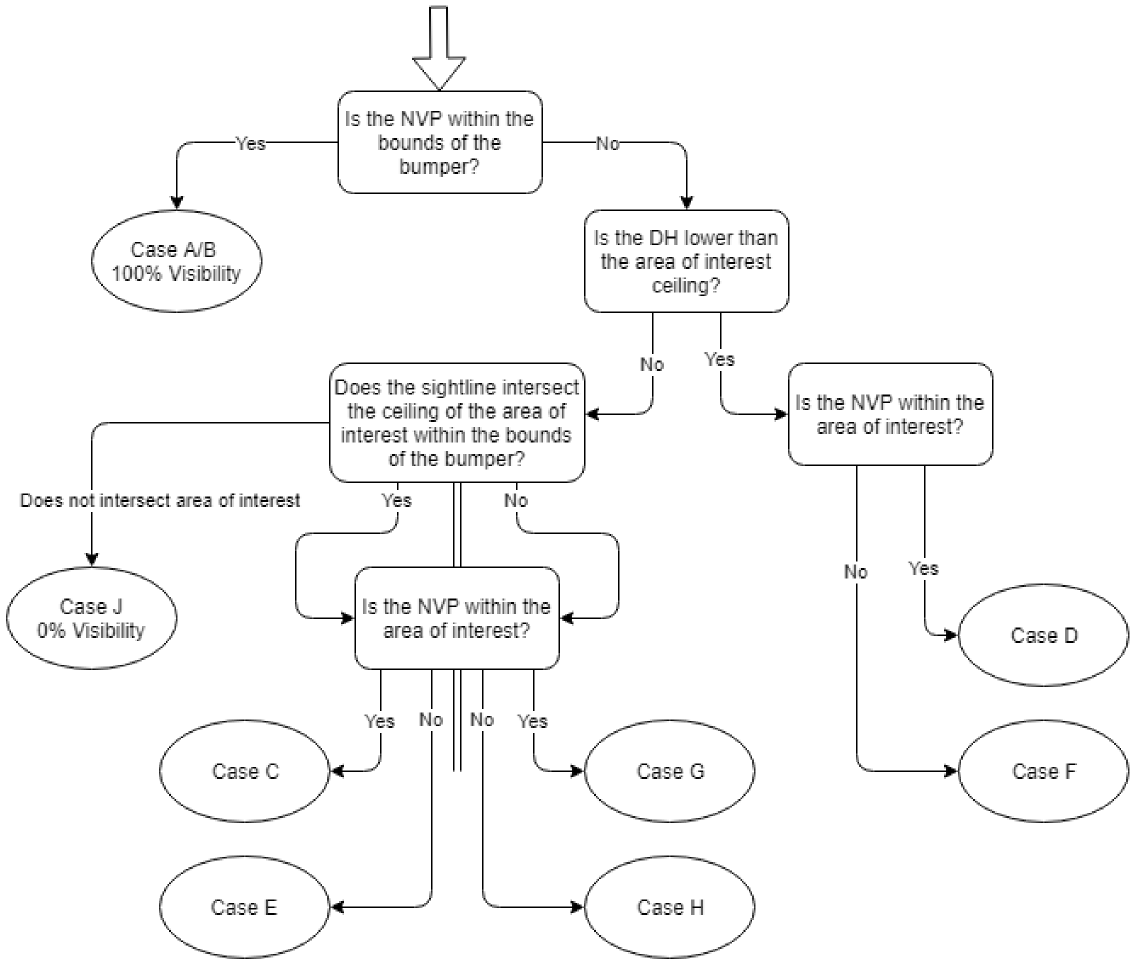



Figure 50: A flowchart summarizing the algorithm's determination of which case a specific slice falls into. NVP refers to the nearest visible point on the ground. DH is driver height.

Though the example images only show the slices at 0 degrees, these same cases are used for the full range of angles and slices. $R$ in these equations is the extent of the area of interest and $H$ is the driver eye height. $R$ varies based on the angle $\theta$.

### K.1.6 Case A and B

Figure 51: Case A

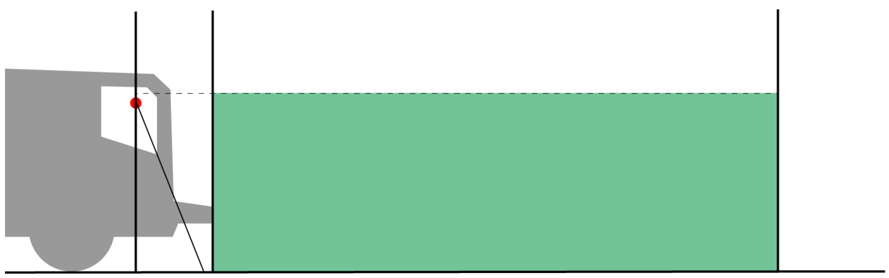


Figure 52: Case B

The entire area of interest, minus the area within the bumper, is visible in this slice. Realistically this requires a transparent lower part of the cab or bumper, or a bumper shape that is not rectangular.

The blind volume is

$$V_a = V_b = 0 \tag{14}$$

### K.1.7 Case C and D

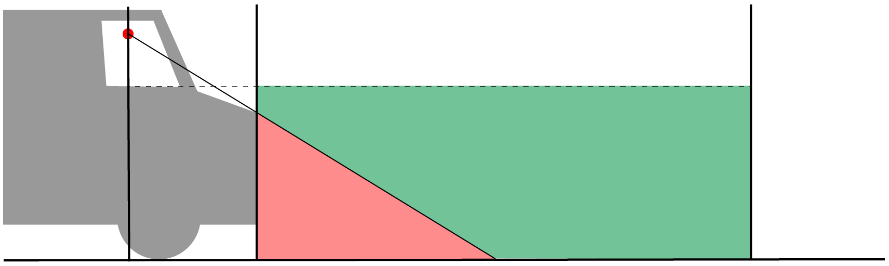


Figure 53: Case C

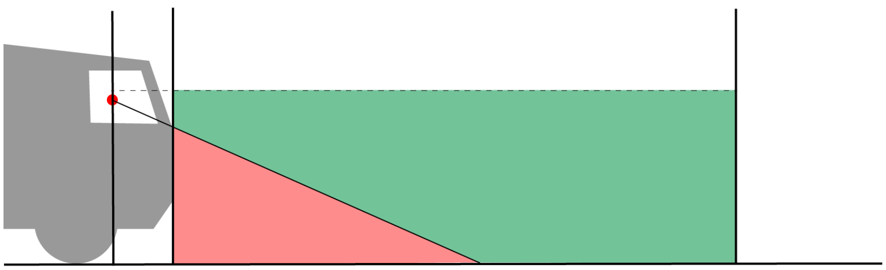


Figure 54: Case D

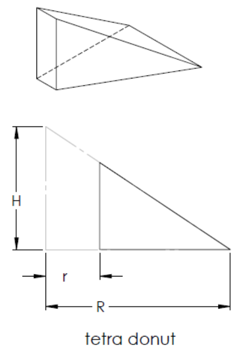



Figure 55: variables for cases C and D

The blind volume in the area of interest is

$$V_c = V_d = \frac{1}{6}\left(H\left(R^2 - r^2\right) - 2hr^2\right)\tan(\theta) \tag{15}$$

### K.1.8 Case E and F

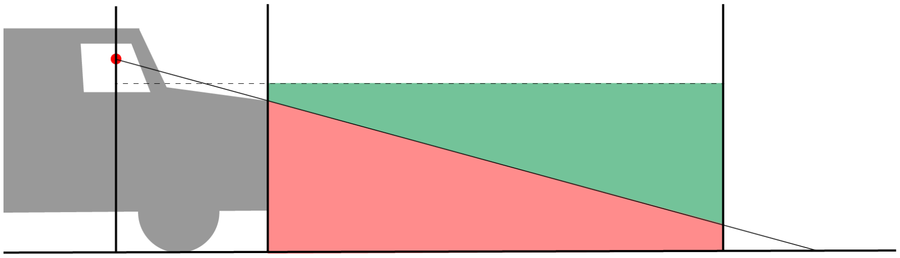


Figure 56: Case E

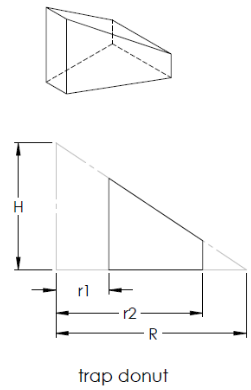



Figure 58: variables for cases E and F

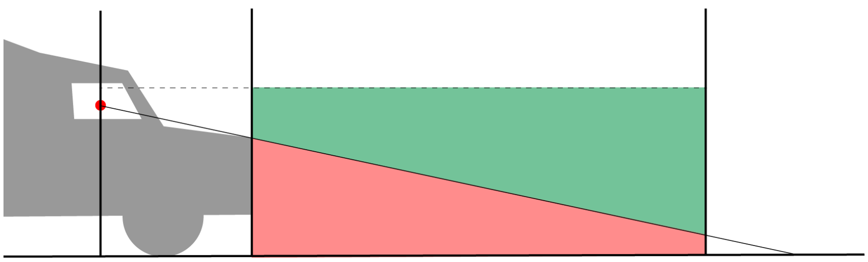


Figure 57: Case F

The blind volume in the area of interest is

$$V_e = V_f = \frac{2H}{3}\tan(\theta)\left(\frac{r_1}{4R}\left(-3r_1^2 + 2r_1r_2 + 3r_2^2\right) + r_1^2 - r_2^2\right) \tag{16}$$

### K.1.9 Case G

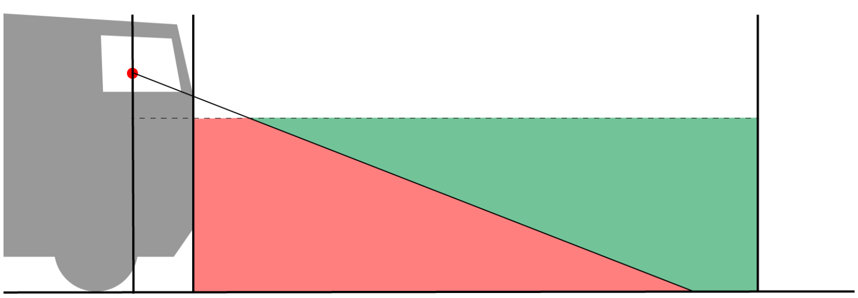


Figure 59: Case G

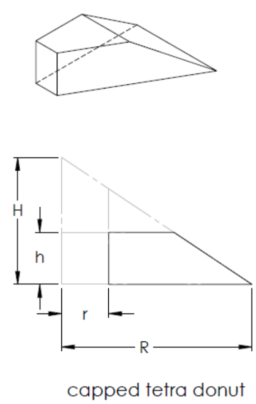



Figure 60: variables for case G

The blind volume in the area of interest is

$$V_g = \frac{h}{6H^2}\tan(\theta)\left(h^2R^2 - 3hHR^2 - 3H^2\left(r^2 - R^2\right)\right) \tag{17}$$

### K.1.10 Case H

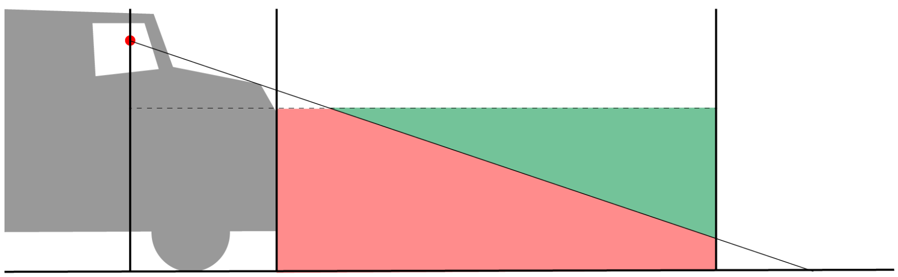


Figure 61: Case H

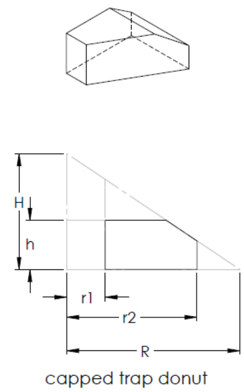



Figure 62: variables for case H

The blind volume in the area of interest is

$$V_g = \frac{\tan(\theta)}{6H^2R}\left(h^3R^3 - 3h^2HR^3 + 3hH^2R\,\left(R^2 - r_1^2\right) + H^3\left(-(R-r_2)^2\right)(R+2r_2)\right) \tag{18}$$

### K.1.11 Case J

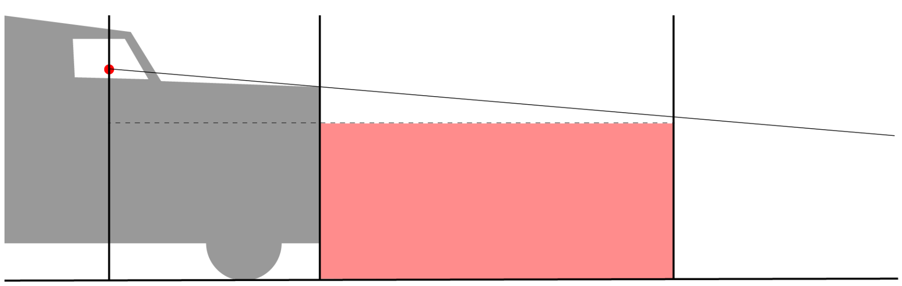


Figure 63: Case J

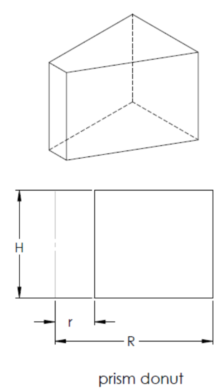



Figure 64: variables for case J

The blind volume in the area of interest is the same as the volume of the area of interest. There is no visibility.

$$V_j = V_{total} = \frac{R^2 - r^2}{2} H \tan(\theta) \tag{19}$$

## K.2 Rating Calculation

The final rating for a truck is the percentage of the volume of interest that is directly visible to the driver. This is the sum of the visible volumes in each slice divided by the sum of the volume of the area of interest. The equations above give the blind volume for a slice, so the visible volume is found by subtracting the blind volume from the total volume. We express the rating as a percentage, so the possible ratings vary between 0 and 100. Most trucks we have measured fall between 40% and 80% visibility.

$$\text{Rating} = \left(100 \times \frac{\text{visible volume}}{\text{total volume}}\right)\% \tag{20}$$

## K.3 Accuracy

Identifying and quantifying sources of error in this method may be key to its adoption in industry. Based on testing in which multiple people measured the same truck we found that the entire process had a standard deviation of 2.6 percentage points in the final rating, and that different people annotating the same photo on the website had a standard deviation of 0.8 percentage points in the final rating.

Future work might encompass quantifying how much the true volume differs from the volume resulting from such assumptions as the truck hood being rectangular, or the geometric modeling of the volume around each data point. As the error with this Riemann sum approximation is greater when fewer points are used for calculation, we might investigate the effect of the number of points used on the accuracy. This would ideally be performed with accurate CAD models of trucks.

After data from a wider range of trucks is collected, it may be advantageous to check for edge cases or unintended geometries that are incentivized by the rating. For example, we define the area of interest based on the edges of the windshield and passenger window, primarily for ease of calculation. This may disadvantage some windshield shapes in a way that does not represent their vision quality. This may be solved by introducing more horizontal bounds or defining the angular extent of the volume of interest. Similarly, the size of the area of interest is dependent on the width and length of the truck as the volume of interest extends a fixed distance from the front and side. An improvement that adds the same volume to the visible area may not add the same amount to the ratings of two different trucks. While this is intended to represent the greater risk and responsibility associated with a larger truck, it may also apply to trucks with a wider windshield in terms of angular spread from the driver's point of view, potentially affecting cab-over trucks and those with large windshields, two categories which tend to have better direct vision.