Verifiable Obstacle Detection

Ayoosh Bansal*, Hunmin Kim†, Simon Yu*, Bo Li*, Naira Hovakimyan*, Marco Caccamo‡, Lui Sha*

*University of Illinois Urbana-Champaign {ayooshb2, jundayu, lbo, nhovakim, lrs}@illinois.edu,
†Mercer University kim_h@mercer.edu, ‡Technical University of Munich mcaccamo@tum.de

Abstract—Perception of obstacles remains a critical safety concern for autonomous vehicles. Real-world collisions have shown that the autonomy faults leading to fatal collisions originate from obstacle existence detection. Open source autonomous driving implementations show a perception pipeline with complex interdependent Deep Neural Networks. These networks are not fully verifiable, making them unsuitable for safety-critical tasks.

In this work, we present a safety verification of an existing LiDAR based classical obstacle detection algorithm. We establish strict bounds on the capabilities of this obstacle detection algorithm. Given safety standards, such bounds allow for determining LiDAR sensor properties that would reliably satisfy the standards. Such analysis has as yet been unattainable for neural network based perception systems. We provide a rigorous analysis of the obstacle detection system with empirical results based on real-world sensor data.

Index Terms—Autonomous vehicles, Vehicle safety, Object detection

I. INTRODUCTION

Autonomous Vehicles (AV) will be among the most significant technological achievements of current times, with the potential to save and improve lives [1], [2]. However, it is unclear at what point AV will be safer than the human-driven vehicles [3]. The imperfection of current AV is evident in the various crashes that have been attributed, in part, to the autonomous control of involved vehicles [4]–[19].

Neural networks and artificial intelligence have enabled capabilities in cyber-physical systems that might have remained unachievable otherwise. Deep Neural Networks (DNN) for object detection, classification, and predictive tracking, are crucial for perceiving an AV's environment in real-time and planning complex maneuvers an AV must execute. But these technologies are inherently unverifiable, *i.e.,* incapable of being verified [20]–[22] that leads to, as yet unsolvable, safety concerns [23]–[29]. Safety-critical software are required to be analyzable and verifiable [30], [31], a role DNN solutions are not yet ready to fulfill [32]–[34].

The impasse caused by the necessity of DNN solutions to enable AV's mission capabilities and their unsuitability for safety-critical tasks can be resolved by decoupling the mission and safety requirements. The disassociated fulfillment of safety and mission requirements has been successful in system architectures like Simplex [35], [36], leading to systems that work with high performance in typical cases but provide verifiable safe behavior in safety-critical conditions. But thus far, such a separation has not been achieved in AV.

In this work, we lay the foundation for decoupling of safety and mission responsibilities for AV. We describe a minimal set of requirements for safety-critical perception with a focus on obstacle existence detection in Section IV. Unlike the mission-critical requirement of perceiving obstacles and predicting their trajectories for complex maneuvers planning, the safety-critical requirement for collision avoidance is to reliably detect the existence of obstacles that the AV may collide with. The safety-critical obstacle existence detection can then be used in various ways, like to detect and correct faults in the mission critical perception systems, fused as an ensemble [37], [38], or make control decisions like brake to stop if all else fails.

In the absence of verifiable DNN based solutions, we turn to classical obstacle detection algorithms, that are verifiable, *i.e.,* capable of being fully analyzed and verified, to determine their ability to fulfill the safety-critical requirements. Traditional LiDAR based obstacle detection algorithms [39], based on geometric properties, are excellent candidates for verifiable safety-critical obstacle detection. LiDAR sensors have shown incredible promises for their use in autonomous driving [40]. This has, in turn, accelerated the improvements in LiDAR sensor technology, increasing the range, scanning frequency, and resolution of the sensor [41]. LiDAR sensors are also superior to stereo cameras for 3D obstacle detection as they only suffer a linear error growth with distance as compared to quadratic for stereo camera [42]. These factors made the LiDAR sensor the natural choice for this work.

Depth Clustering [43], [44] a LiDAR based range image segmentation algorithm is chosen as the example algorithm in this work. The core component of the algorithm analyzed here is the ground removal *i.e.,* separating the flat drivable ground from obstacles that need to be avoided. These bounds are referred to as the *Detectability Model* in this work. Such bounds satisfy the analyzability and verifiability requirements in safety-critical system engineering. The value of such a Detectability Model comes from the predictability of faults in obstacle existence detection and reliable mitigation. Given desired safety standards and properties of obstacles, algorithm parameters and LiDAR sensor parameters can be chosen to meet the safety standards reliably.

The **contributions** of this work can be summarized as:

- Minimized sufficient requirements for safety-critical obstacle existence detection for collision avoidance (§IV).
- Detectability model for an existing classical obstacle detection algorithm, Depth Clustering, with human perceptible bounds on capabilities and limitations (§VI).
- An evaluation[1] based on real sensor data [45] (§VII).

[1]https://github.com/CPS-IL/verifiable-OD

II. MOTIVATION AND OVERVIEW

Obstacle existence detection fault, *i.e.,* False Negative (FN) in perception are a grave safety concern [46]. A survey of fatal collisions involving AV (Table I) points to recurring FN errors. Other fatalities involving AV [8]–[19], excluded from Table I due to unavailability of investigation reports, seem to follow this pattern. These underlying safety concerns are the primary challenge in adopting complete autonomy [47]–[52]. Learning from these incidents and acknowledging the impossibility of all encompassing safety on the road, we limit our focus to FN.

DNN verification, even for simple properties, is an NP-complete problem [53]. Gharib *et al.* [54] describe the need and current lack of verification methods for machine learning components used in safety-critical applications. Empirical risk minimization, a foundational principle of modern statistical machine learning, fails to satisfy the high robustness requirements of safety-critical applications [55]. There is still an open requirement for verification techniques that can validate the behavior of a trained network under all circumstances and not just expected safe input space [56]. The vastness of the input sets for real-world problems, like perception, renders reachability analysis impractical for the DNN used in AV.

Analyzability and verifiability are the crucial components of the certification process of safety-critical systems [30], [31]. Verifiable algorithms, where the *causality* between the input parameters and the algorithm result can be established, are inherently suitable for safety-critical applications. This is in contrast to object detection DNN, trained using supervised learning, which effectively captures *correlations* between training input and labels. Consider the following simple example where y is obstacle height, x is obstacle distance from the AV, a and b are constant parameters based on LiDAR properties:

$$y \geq ax + b \tag{1}$$

If we want to establish that (1) is the *detectability model*, *i.e.,* when the condition in (1) is met, obstacles are always detected by the AV, the following must be defined:

Requirements: A definition of minimal requirements for what it means to *detect* an obstacle (§IV).

Constraints: A set of well defined constraints that must be met for the model to be applicable (§V).

Verification: A deterministic analysis verifying the detectability model (§VI).

Let's assume that an AV safety standard requires that the AV be able to detect all obstacles of a minimum height of $10\ cm$. Further, let's assume all vehicles and structures on the road are also mandated to be taller than this minimum height by a road safety rule. Using LiDAR and AV parameters to determine a and b, the detectability model (1) can be used to determine the minimum distance at which such obstacles can be detected. This minimum distance, in conjunction with the braking capability of the AV, can then be used to determine the max speed at which the AV can safely travel. This example, admittedly simple, shows how a verified detectability model can bring together road safety rules, AV parameters and AV safety policies to provide deterministic collision safety.

Table I: Survey of AV Involved Fatalities

Ref.	Autonomy Response [2]
[4]	A truck in the path of the vehicle was not detected and no evasive actions like braking or steering away were initiated.
[5]	Low confidence, unstable classifications of a pedestrian led to the perception system ignoring the existence of a pedestrian.
[6]	A crash attenuator in the path of the vehicle was not detected.
[7]	A white semi trailer in the path of the vehicle was not detected.

III. RELATED WORK

Classical Obstacle Detection: An approach for long-range obstacle detection based on stereo cameras was proposed by Pinggera *et al.* [57]. While they successfully detect patches on most objects within a long range, it is unclear whether this approach is enough to avoid collisions and whether all obstacles that pose collision risk are detected. While the approach shows promise, its performance under common distortions like bright spots, *etc.,* is not shown, and a further study on its limitations is required to use it in safety-critical tasks. Various LiDAR based geometric algorithms were considered as part of this work [58]–[62]. Each algorithm has a similar flow; identifying points on the ground vs. obstacles, followed by clustering, segmentation, and/or classification. Depth Clustering [43], [44] is chosen as the primary example due to its deterministic explainable behavior, flexible parameters to optimize tradeoffs, and public availability of an efficient C++ implementation with extremely low runtime of $40\ ms$ on an embedded Jetson Xavier platform, using a single CPU core only, for a point cloud with $169,600$ points [63]. The algorithm has also garnered interest in recent works in literature [63]–[68].

Neural Network Verification: Albarghouthi [69] described the challenges for neural network verification, including scale, complexity, and dynamism of the environment, all applicable for their use in Autonomous Driving. Liu *et al.* [70] survey existing verification methods and classify the verification methods into three categories: reachability, optimization, and search. They identify tradeoffs between scale and completeness of the verification methods. Even when the deep networks can be verified, it is done so for small input sets only [71]. Another survey on the safety and trustworthiness of DNNs [72] identified various challenges, including; a physical representation of the verification metrics, completeness of the verifiable properties, and scalability to complex DNN. Verification techniques have also been proposed to explore around available inputs by adding adversarial or context specific distortions [26], [27], improving the input coverage. However, this does not imply complete verification and dependable predictability of behavior. Hardware reliability [28] does not protect against algorithmic faults. Fully verifiable, analyzable and explainable DNNs performing real-world object detection in autonomous vehicles remain an elusive goal.

---

[2]The comments brief the driver assist system's response during the incident and are not necessarily the causal faults for the incident.

Collision Avoidance: This work complements typical collision avoidance systems by potentially providing additional triggers for the emergency braking systems to engage [73]–[75], *e.g.,* when comparing the output from the verifiable algorithm to the DNN, a safety-critical FN in DNN output is determined. Other collision avoidance systems leverage cooperative communication [76], however, our work focuses on all obstacles including human driven vehicles and pedestrians. Motion planning [77]–[79] and risk assessment [80]–[84] based collision avoidance systems would be benefited by using a verifiable algorithm to detect obstacles in the environment.

IV. MINIMAL REQUIREMENTS

In this section, we define the minimally sufficient, though not necessary and sufficient, requirements for safety-critical obstacle detection and collision avoidance. These requirements define a minimized set of features of an obstacle that the AV perception system should perceive to establish the existence of the obstacle. While some of the following observations are already well established, this work is the first to form a minimally sufficient set for safety-critical collision avoidance. Further, some requirements like object classification are typically considered crucial for all aspects of perception in AV, however, we show that while it is valuable for mission-critical requirements, it is not for safety-critical collision avoidance.

A. Classification

We will use the mathematical problem formulation of reach-avoid control to justify why we do not need to classify obstacles for collision avoidance purposes. The current section is motivated by [85], [86] but we simplify the notations and the problem in a deterministic setting while the references consider stochastic problem formulations. Let $x_k \in X$ be the 2D-position of the vehicle at discrete time instance $k \geq 0$, where $X \subseteq \mathbb{R}^2$ is the set of the state space. The set $D \subset X$ is the destination, and the sets $O_i(k) \subset X$ for $i \in M_O \triangleq \{1, 2, \cdots, M\}$ are obstacles, where $O_i(k)$ represents the ith obstacle and M_O is the index set for the obstacles. The set $O_i(k)$ could be time-varying but it is assumed that the obstacles do not overlap with the destination, *i.e.,* $\cup_{i \in M_O}(D \cap O_i(k)) = \emptyset \, \forall \, k$. For brevity, further mentions of $O_i(k)$ are reduced to O_i. Now, the reach-avoid control problem is to drive the vehicle to the destination D while avoiding obstacles O_i for $i \in M_O$ within finite time horizon N, given initial condition x_0. The success of this mission can be characterized by the following index r:

$$r \triangleq \sum_{j=0}^{N} (\Pi_{i=1}^{M}\Pi_{t=0}^{j-1}\mathbf{1}_{X\setminus O_i}(x_t))\mathbf{1}_D(x_j)$$
$$= \begin{cases} 1, & \text{if } \exists j \in [0, N] : x_j \in D \wedge \\ & \forall t \in [0, j-1] : x_t \in \cap_{i=1}^{M} X \setminus O_i \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where $\mathbf{1}_S(\cdot) : X \to \{0, 1\}$ is the indicator function for a set S, and $\wedge$ is the logical AND. In short, $r = 1$ if and only if the objective is achieved. The index r in (2) can be used as a cost function of the optimal control problem as in [85], [86]. Therefore, one is required to evaluate the indicator functions in (2), which means that it is required to know the set D and $X \setminus O_i$ for $i \in M_O$.

However, index r in (2) can be equivalently formulated as

$$r = \sum_{j=0}^{N} (\Pi_{t=0}^{j-1}\mathbf{1}_{X\setminus(\cup_{i \in M_O} O_i)}(x_t))\mathbf{1}_D(x_j) \quad (3)$$

and this expression can be used instead for the optimal reach-avoid control problem. The formulation (3) indicates that one could also address the control problem only with knowing $\cup_{i \in M_O}O_i$, but not individual obstacle sets O_i, *i.e.,* one does not need to classify/distinguish individual obstacles for the reach-avoid control purpose.

It should be noted that classification adds valuable information that supports advanced features like predictive tracking, maneuver planning and improves AV performance. The argument here is only that object classification is not a necessity for obstacle avoidance and therefore not a part of minimal requirements for safety critical obstacle detection.

B. Collision Risk

Collision avoidance involves many factors, from perception to vehicular control. While the dynamics and ethics of collision avoidance are complex [87], the safety-critical obstacle detection system is required to detect all obstacles that can potentially collide with the AV. We utilize a physical model for collision risk from our prior work [88], where the potential risk of collision is determined by the overlap of the existence regions [89] of obstacles and AV within the AV's time to stop.

C. Height

The height of an obstacle is only useful in making a binary determination for collision avoidance, *i.e.,* whether or not the obstacle is completely clear above the height of the AV. For example, in Figure 1 (a) the overhead road sign ① is completely above the AV, its exact height has no implication for collision avoidance. The box ② contains valuable information that is required to identify the obstacle within the box to be a vehicle, however as discussed in Section IV-A, such a recognition of the obstacle class is not a requirement for collision avoidance. Thus for safety-critical collision avoidance ③ contains as much relevant information as ②. Therefore as long as an obstacle's height is not erroneously detected to be above and clear of the AV, we can simply consider the top or bird's eye view of the AV's surroundings. While such a view of obstacles is not traditionally used in perception systems, however, path planning in AV, an inherently 2D problem, regularly uses this representation [90].

D. Distance

The distance to an obstacle must be accurately detected for the collision-free operation of the AV. For collision avoidance, this distance is the minimum distance between the perimeters of the obstacle and the AV. Many safety parameters like safe following distance, time to collision and time to stop, are a function of the distance between the AV and obstacles [73], [75], and therein lies the difference in underestimation and overestimation error in distance detection.
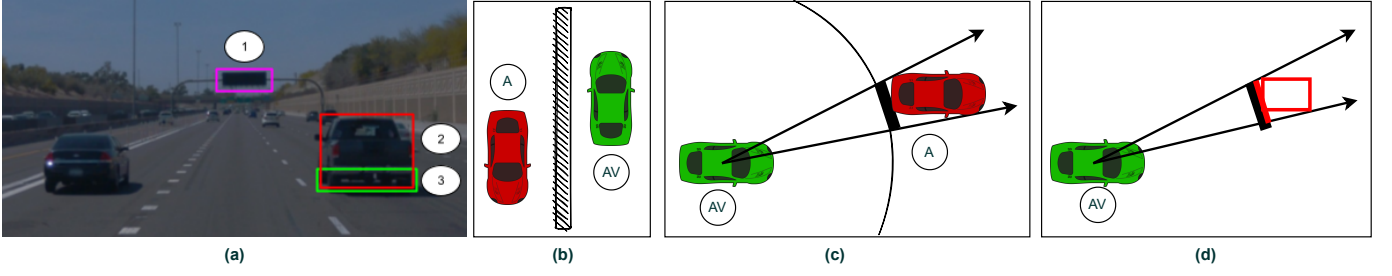
Figure 1: Example scenes. (a) Role of obstacle height. (b) Distance and Occlusion of obstacles. (c) Projection of Obstacles.

1) Underestimation Error: When obstacle distances are underestimated, *i.e.,* obstacles are detected to be closer than they are, the performance of AV suffers, however, there is no negative impact for direct collision avoidance. An implicit effect of acceptance of all detection with a lower distance than the ground truth obstacle is that occluded obstacles become unnecessary to detect as long as the occluding obstacle is detected. An example for this is Figure 1 (a) and (b). A small wall exists to the left side of the AV in both images, separating traffic moving in the opposite direction to the AV. If the wall is detected as an obstacle, the AV would not need to detect the vehicles across the wall to avoid colliding with them. Avoiding collisions with the occluding obstacle (wall here) means avoiding collisions with occluded obstacles. Note that this only applies to completely occluded obstacles. Partial occlusion is discussed in Section IV-E.

2) Overestimation Error: Overestimation of distance is a grave safety concern. Obstacles detected to be further than they are invalidate any safety decisions that would be based on this information. Since it is unrealistic that an obstacle detection system would always have no distance overestimation error, a strict bound on overestimation distance detection error is required. With an established max error bound, any distance-based safety guarantee can assume this error is always present, maintaining the safety guarantee in the worst case.

E. Projection

We determine a projection signifying a line on the road that the AV cannot cross without colliding with the obstacle for each obstacle. Figure 1 (c) shows an example of this. A circle is drawn with its center at the AV's sensor hub and radius equal to the closest point on the obstacle from AV. The projection of the obstacle is then found as a line segment, tangent to the point where the above circle touches the obstacle. This line segment is shown as a thick line segment in Figure 1 (c). This line segment is a representation of the obstacle. Note that as new sensor inputs come in over time, the AV and obstacle move relative to each other, and the projection moves accordingly. So the line segment only represents the obstacle in the current frame to make safety-critical decisions until the next sensor frame is received and processed.

With this minimal representation of the ground truth obstacles, containing only the minimal information about the obstacles as required for safety-critical collision avoidance,

we can now determine when detected obstacles are sufficiently detected to avoid collisions.

F. Coverage

Each detected obstacle that meets the distance criteria (§IV-D) and falls within the direction of an obstacle can now be used to determine if a ground truth obstacle is sufficiently covered by detected obstacles to avoid collisions with the ground truth obstacle. Detected obstacles are projected on the projection of the ground truth obstacle to determine what parts of the ground truth projection are covered by the detected obstacles, as shown in Figure 1 (d). The projected detection must provide enough information to avoid collisions for an obstacle to be considered as detected. As with distance (§IV-D) this coverage may not be perfect but should have bounded errors. A limited proportion of the ground truth projection must be covered. This is equivalent to the traditionally used intersection over union (IOU) limits. It should be noted that the error margin of coverage is less important than that of the distance. The minimum distance always tracks the distance between the closest points between the AV and the obstacle, changing with each sensor input frame.

G. Summary

In this section we have discussed various features of obstacles a perception system may detect and detailed their relevance for safety-critical collision avoidance. Many properties of obstacles that are considered critical parts of perception in AV (*e.g.,* class, 3D dimensions, road sign information), while indeed required for achieving the mission of autonomous driving, are not necessary for safety-critical collision avoidance. In brief, an obstacle is considered detected for safety-critical obstacle avoidance if the following are accurately determined *(a)* the distance of obstacle from AV, within bounded error margins. *(b)* a line on the road that the obstacle makes unsafe for the AV to cross. An obstacle detection system that reliably meets the above requirements can be used to detect safety-critical faults in perception by more complex perception systems, thus providing fault detection and collision avoidance system. The fusion of verifiable algorithms with DNN and the control decisions based on the verifiable algorithms have additional challenges that will be addressed in future works.

Table II: Symbols Summary

Symbols	Description	Example Value
N	Count of lasers in vertical array	64
ξ	LiDAR Beam Angles set	$[-2.4^o, ..., 17.6^o]$
H_L	Height of LiDAR sensor	$2.184\ m$
Ψ	LiDAR horizontal angle step	0.136^o
h_o	Height of obstacle	Variable
α_o	Inclination of obstacle from ground	$[0^o, ..., 8^o]$ [91]
D_{min}	Distance to first LiDAR return	$6.7\ m$
D_{max}	Distance range of the sensor	$75\ m$
α_{th}	Threshold angle for ground removal	10°

V. PARAMETERS AND CONSTRAINTS

A. LiDAR Parameters

A rotating LiDAR contains several lasers stacked vertically. Each laser is given an index $\mathbf{r} \in 1...N$, where $\mathbf{N}$ is the number of lasers. Each laser has an inclination angle ξ_i from the horizontal, positive below the horizontal, the set of which is represented by ξ. The lowest laser below the horizontal is assigned index $r = 1$ and $r = N$ index represents the highest laser, usually inclined above the horizontal. The LiDAR is mounted on the vehicle at a height $\mathbf{H_L}$ above the wheelbase. For rotating LiDAR, the horizontal step angle Ψ is uniform and can be calculated as $360^o/SamplesPerRotation$. The sensor is considered the origin point for the coordinate system. At each rotation, a new column of N samples is recorded and indexed with $\mathbf{c}$. The sensor returns a range image, a 2D depth image of range values $\mathbf{R_{r,c}} \ \forall \ r \in 1...N, \ c \in 1...\Psi$. While we assume the more prolific rotating LiDAR, a solid state LiDAR [41] would have similar parameters. Table II and Figure 2, respectively, summarize and represent some of these parameters, using example values from real world dataset [45].

B. Constraints

We assume the following constraints for the validity of the Detectability model in Section VI:

C1: All LiDAR beams encountering solid obstacles, including ground, within the max range of the LiDAR, return with strong enough intensity from the first obstacle they encounter so that the return is recorded. This assumption holds in real world except when; *(a)* there are physical impediments in the air like dust, smoke, fog or rain; or, *(b)* adversarial objects with extremely reflective, absorbent or transparent surfaces.

C2: The obstacle's width, projected on a plane perpendicular to the LiDAR beams falling on the obstacle, must be enough to have LiDAR points returned from it. For example, a thin sheet aligned parallel to the LiDAR beam's direction can never be detected. According to the formula for circle arc length, a conservative bound can be established:

$$\text{Width Projected} \geq \Psi D \pi / 180^o. \tag{4}$$

Since Ψ is small, the arc length is approximately equal, though always greater than the chord length for the same points. Being a small value, this width is not a prohibitive constraint, *e.g.*, using values from Table II the width constraint at max LiDAR range of $75\ m$ is just $17\ cm$.
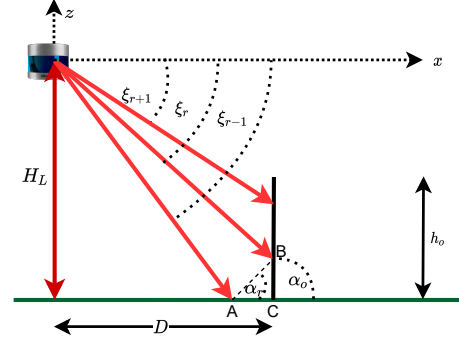

Figure 2: A representation of point segmentation to ground *vs.* obstacle based on α thresholding when two points return from the obstacle.

C3: At least one LiDAR point exists on the ground before the obstacle. This is an algorithmic constraint [44]. While we focus on the primary Top LiDAR only in rest of this work, additional low height limited field of view (FOV) LiDAR sensors can be used for obstacles closer than D_{min}, *e.g.*, Front, Rear, Side-Left and, Side-Right LiDAR in Waymo Open Dataset [45].

Furthermore, the following are assumed initially (§VI-A) and relaxed in later sections:

A1: Obstacle touches the ground at 90°, *i.e.*, $\alpha_o = 90^o$. Relaxed in Sections VI-B and VI-C.

A2: There is no ground inclination change between the vehicle and the obstacle. Relaxed in Section VI-D.

A3: No noise in detected range. Relaxed in Section VI-E.

VI. DETECTABILITY MODEL

A *detectability model* describes, given an obstacle's properties, sensor parameters, and environment parameters, whether a given algorithm can detect the obstacle with a given set of algorithm parameters. The detectability model allows the conversion of safety standards into algorithm and sensor parameter requirements as described with the example in Section II. We now develop the detectability model for obstacle detection using a LiDAR. We use Depth Clustering [43], [44] as the example algorithm, which uses depth discontinuity to segment LiDAR points into ground *vs.* obstacles and then determine bounding boxes for obstacles. Table II contains a summary of the symbols used in this section with example values from Waymo Open Dataset [45] or chosen defaults.

A. Simple Model

We start with a simple system model, as shown in Figures 2 and 3, with Assumptions A1, A2 and A3.

Ground Removal: The primary part of the Depth Clustering algorithm is ground removal, *i.e.*, determining which range image points are on obstacles *vs.* ground. FN errors can occur if points on obstacles are mistakenly considered part of the ground. Ground removal is based on vertical depth discontinuity using the inclination angles α. Assuming the first point ($r = 1$) to always be on the ground, a sharp
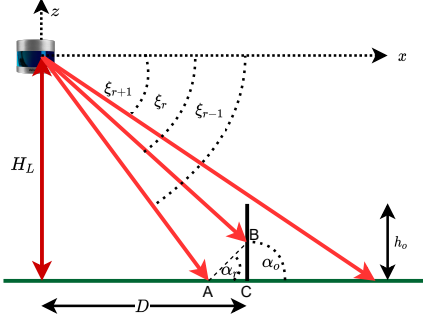
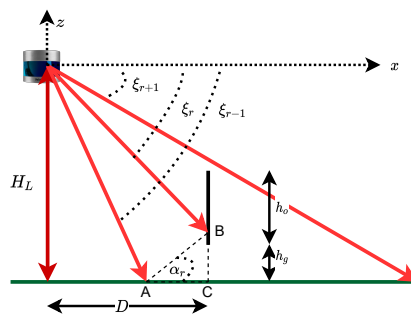Figure 3: Respresentation for α thresholding with only one point on obstacle.

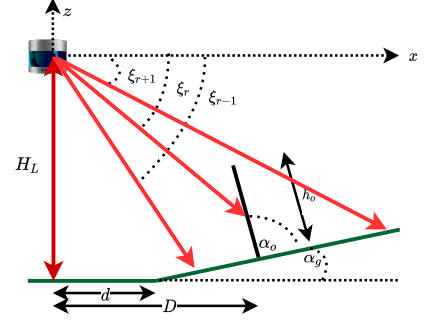Figure 4: Obstacle above ground plane.

Figure 5: Obstacle with ground inclination relative to the AV.

change in α above a threshold α_{th} indicates that a point is on an obstacle. The first point must lie on the ground for the subsequent comparisons to be meaningful (C3). Inclination angle is calculated from the range image, as designed by Bogoslavskyi and Stachniss [44], shown in Figure 2, as

$$\alpha_{r,c} = atan2(||BC||, ||AC||) = atan2(\Delta z, \Delta x) \quad (5)$$

$$\Delta z_{r,c} = |R_{r-1,c}sin\xi_{r-1} - R_{r,c}sin\xi_r| \quad (6)$$

$$\Delta x_{r,c} = |R_{r-1,c}cos\xi_{r-1} - R_{r,c}cos\xi_r| \quad (7)$$

$$\Delta\alpha_{r,c} = \left\{ \begin{array}{l} 0^o, \text{if } r = 1 \\ |\alpha_{r,c} - \alpha_{r-1,c}|, \text{otherwise} \end{array} \right\} \quad (8)$$

The point r, c is considered to not be on the ground if, $R_{r-1,c}$ is on the ground, $\alpha_{th} < 45^o$, and

$$\Delta\alpha_{r,c} > \alpha_{th}, \quad (9)$$

If $R_{r-1,c}$ is not on ground then $R_{r,c}$, is also considered to not be on ground. The lowest row of points $R_{0,c}$ are assumed to be on the ground (§V-B C3). The column index corresponds to a rotational position of the sensor. For brevity, the column index is omitted for points in the same column only.

At horizontal distance D from the sensor, the height of a LiDAR beam r can be calculated as:

$$H_r(D, \xi_r, H_L) = H_L - D * tan(\xi_r). \quad (10)$$

For brevity, $H_r(D, \xi_r, H_L)$ is referred to as $H_r(D)$ from this point as ξ_r and H_L are constant once a LiDAR sensor is chosen and placed on an AV. Let us define $r = \min\{i|H_i(D) > 0^o\}$, i.e., R_r is the first point on the obstacle. With this background, we can determine the minimum height of an obstacle that can be detected when the obstacle is at a distance of D from the sensor (Figure 2). It is required to check three consecutive LiDAR points to detect an obstacle, according to the ground removal algorithm presented in (5) through (9).

Theorem 1. The obstacle is detected at distance D, if and only if one of the following conditions are true:
1) $H_r(D) \leq h_o < H_{r+1}(D)$ AND
   $\alpha_{th} < atan2(H_r(D), |D - \frac{H_L}{tan(\xi_{r-1})}|);$
2) $h_o \geq H_{r+1}(D)$.

Proof. Sufficiency: Consider case 2), i.e., $h_o \geq H_{r+1}(D)$. Then the points R_r and R_{r+1} are on the obstacle, as shown in Figure 2, and thus it holds that $\alpha_{r+1} = 90^o$. Since we have

$$\Delta\alpha_r = \alpha_r - 0^o, \qquad \Delta\alpha_{r+1} = |90^o - \alpha_r|,$$

where $\alpha_{r-1} = 0^o$ by the definition of index r, it follows that

$$\max\{\Delta\alpha_r, \Delta\alpha_{r+1}\} \geq 45^o > \alpha_{th},$$

which guarantees that the obstacle is detected at distance D.

Now consider case 1), i.e., $H_r(D) \leq h_o < H_{r+1}(D)$ as shown in Figure 3. By the assumption, there exists an index $r - 1$ that touches the ground. Since $h_o \geq H_r(D)$, the angle α_{r-1} can be found by (5) as follows:

$$\alpha_r = atan2(H_r(D), |R_{r-1}cos(\xi_{r-1}) - R_rcos(\xi_r)|)$$
$$= atan2(H_r(D), |D - \frac{H_L}{tan(\xi_{r-1})}|), \quad (11)$$

where

$$R_{r-1}sin(\xi_{r-1}) = H_L, \qquad R_rcos(\xi_r) = D.$$

Since $\alpha_{r-1} = 0^o$, the above equation and the angle condition in case 1) imply $\alpha_{th} < \Delta\alpha_r$, i.e., the obstacle is detected at distance D.

Necessity: If the obstacle is detected at a distance D, then at least one point on the obstacle is at distance D, i.e., R_r is on the obstacle. This implies $h_o \geq H_r(D)$. Furthermore, if the obstacle is detected, then

$$\Delta\alpha_r = \alpha_r - \alpha_{r-1} = \alpha_r > \alpha_{th}. \quad (12)$$

There are two cases: $\alpha_{r+1} = 90^o$ and $\alpha_{r+1} \neq 90^o$. If $\alpha_{r+1} = 90^o$, then R_{r+1} is on the obstacle, and thus we have $h_o \geq H_{r+1}(D)$. This implies the case 2). On the other hand, if $\alpha_{r+1} \neq 90^o$, then the inequality in (12) must hold, which renders the condition $\alpha_{th} \leq atan2(H_r(D), |D - \frac{H_L}{tan(\xi_{r-1})}|)$. This implies case 1). This completes the proof. □

Theorem 1 consists of two sets of conditions based on the obstacle's height. The second set indicates that if the height of the obstacle is sufficiently large, then there will be more than two LiDAR points on the obstacle ($\alpha_{r+1} = 90^o$). This allows us to detect the obstacle without further condition on

the angle $\Delta\alpha_r$. The first set is the case that, when the size of the obstacle is small, there is only one LiDAR point on the obstacle, which requires us to have an additional condition for the angle $\Delta\alpha_r$.

The first condition set in Theorem 1 implies that the minimum height to be detected at distance D satisfies

$$H_r(D) = h_o$$
$$\alpha_{th} \le atan2(h_o, |D - \frac{H_L}{tan(\xi_{r-1})}|), \qquad (13)$$

which depends on the distance D and threshold α_{th}.

B. Obstacle at Inclination

Assumptions: We maintain all assumptions from §VI-A except part of A1, *i.e.,* that the obstacle surface inclination angle is $\alpha_o \ne 90^o$. We assume $\alpha_o > \alpha_{th}$, otherwise the obstacle cannot be detected.

In this case, if R_i is at the end of the inclined obstacle, then its height at a distance D is found by

$$H_i(D) = h_o sin(\alpha_o) + h_o cos(\alpha_o) tan(\xi_i).$$

This height can be used to determine whether there are more than two LiDAR points on the obstacle. Further notice that the angle α_r is found by reduced height $\frac{H_r(D)}{tan(\xi_r)/tan(\alpha_o)+1}$ and increased width $D + \frac{H_r(D)}{tan(\xi_r)+tan(\alpha_o)} - \frac{H_L}{tan(\xi_{r-1})}$. This observations induce the following Corollary from Theorem 1.

Corollary 1. The obstacle is detected at distance D, if and only if one of the following conditions are true:

1) $H_r(D) \le h_o sin(\alpha_o) + h_o cos(\alpha_o) tan(\xi_{r+1}) < H_{r+1}(D)$ AND $\alpha_{th} < atan2(\frac{H_r(D)}{tan(\xi_r)/tan(\alpha_o)+1}, |D + \frac{H_r(D)}{tan(\xi_r)+tan(\alpha_o)} - \frac{H_L}{tan(\xi_{r-1})}|)$;
2) $h_o sin(\alpha_o) + h_o cos(\alpha_o) tan(\xi_{r+1}) \ge H_{r+1}(D)$.

C. Obstacle not Touching Ground

Assumptions: We maintain all assumptions in §VI-A except A1, *i.e.,* that the obstacle surface no longer touches the ground. Instead, now the obstacle surface starts at height h_g above the ground as shown in Figure 4.

We are interested in the first beam on the obstacle, where its index is defined by $r_g = \min\{i|H_i(D) > h_g\}$. Noticing the height of the obstacle tip is $h_o + h_g$, we can reformulate Theorem 1 as the following corollary.

Corollary 2. The obstacle is detected at distance D, if and only if one of the following conditions are true:

1) $H_{r_g}(D) \le h_o + h_g < H_{r_g+1}(D)$ AND $\alpha_{th} < atan2(H_{r_g}(D), |D - \frac{H_L}{tan(\xi_{r_g-1})}|)$;
2) $h_o + h_g \ge H_{r_g+1}(D, \xi, H_L)$.

The minimum detectable height for this case is found by

$$H_{r_g}(D) = h_o + h_g$$
$$\alpha_{th} \le atan2(h_o + h_g, |D - \frac{H_L}{tan(\xi_{r_g-1})}|).$$

D. Inclined Ground

Assumptions: We maintain all assumptions in §VI-A except A2, *i.e.,* Ground inclination α_g could be non-zero (Figure 5).

Ground inclination affects the detectability model only when relative inclination changes between the obstacle and the AV. This section assumes that ground inclination starts before the obstacle position, *i.e.,* $d < D$. There are two cases.

Case I: The inclination starts after the beam R_{r-1}, *i.e.,*

$$\frac{H_L}{tan(\xi_{r-1})} \le d \le \frac{H_L}{tan(\xi_r)}. \qquad (14)$$

We can extend Corollary 1 to find the detectability condition. The first beam touching the obstacle must be higher than the ground at distance D. Let us define

$$r_f = \min\{i|H_i(D) > (D - d)tan(\alpha_g)\}.$$

If R_{r_f+1} is at the end of the obstacle, then

$$H_{r_f+1}(D) = h_o cos(\alpha_g) - h_o sin(\alpha_g) tan(\xi_{r_f+1}) + (D - d)tan(\alpha_g),$$

which is the threshold to determine whether R_{r_f+1} is on the obstacle. Furthermore, α_{r_f} is found by increased height $\frac{H_{r_f}(D)-(D-d)tan(\alpha_g)}{1-tan(\xi_{r_f})tan(\alpha_g)} + (D - d)tan(\alpha_g)$ and decreased width $D + \frac{H_{r_f}(D)-(D-d)tan(\alpha_g)}{tan(\xi_{r_f})-cot(\alpha_g)} - \frac{H_L}{tan(\xi_{r_f-1})}$. This observation induces the following Corollary.

Corollary 3. The obstacle is detected at distance D, if and only if one of the following conditions are true:

1) $H_{r_f}(D) \le h_o cos(\alpha_g) - h_o sin(\alpha_g) tan(\xi_{r_f+1}) + (D - d)tan(\alpha_g) < H_{r_f+1}(D)$ AND $\alpha_{th} < atan2(\frac{H_{r_f}(D)-(D-d)tan(\alpha_g)}{1-tan(\xi_{r_f})tan(\alpha_g)} + (D - d)tan(\alpha_g), |D + \frac{H_{r_f}(D)-(D-d)tan(\alpha_g)}{tan(\xi_{r_f})-cot(\alpha_g)} - \frac{H_L}{tan(\xi_{r_f-1})}|)$;
2) $h_o cos(\alpha_g) - h_o sin(\alpha_g) tan(\xi_{r_f+1}) + (D-d)tan(\alpha_g) \ge H_{r_f+1}(D)$.

Case II: The inclination starts before the beam R_{r-1}, *i.e.,*

$$d < \frac{H_L}{tan(\xi_{r-1})}.$$

In this case, the beam point of R_{r_f-1} land on different point from that of the case II. This decreases relative height $H_L - R_{r_f-1}sin(\xi_{r_f-1})$ and increases relative width $\frac{H_L}{tan(\xi_{r_f-1})} - R_{r_f-1}cos(\xi_{r_f-1})$ between R_{r_f-1} and R_{r_f}. This changes α_{r_f} found in Corollary 3. It is also worth to notice that if

$$d < \frac{H_L}{tan(\xi_{r-2})},$$

then $\alpha_{r_f-1} = \alpha_g$, and thus we have $\Delta\alpha_r = \alpha_r - \alpha_g$. Using these facts, one can find an extension of Corollary 3.
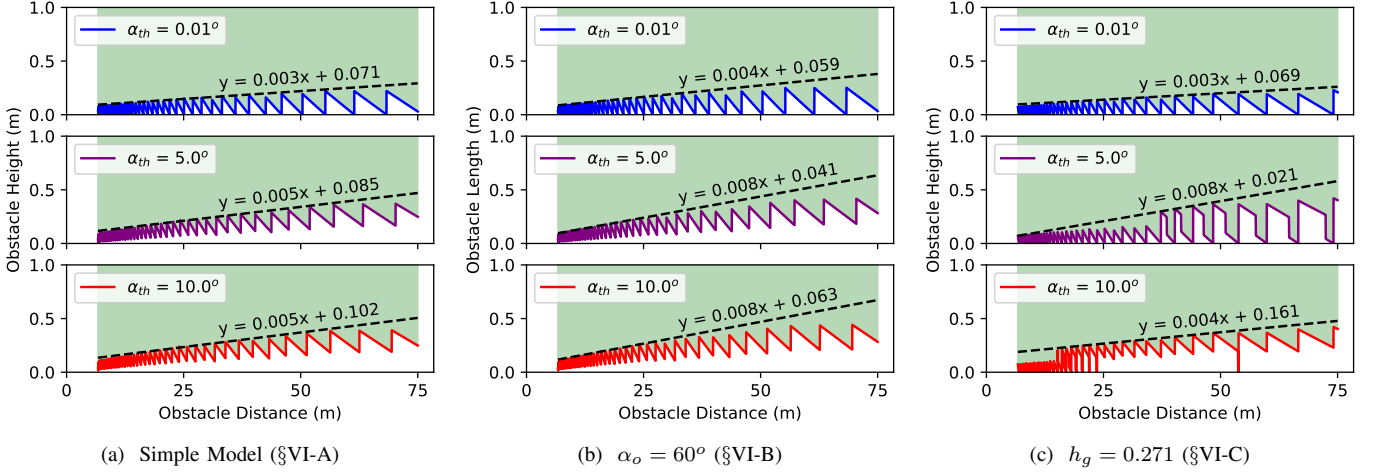
Figure 6: Minimum detectable obstacle height (Y-Axis) at varying distances from the vehicle (X-Axis). Black dashed lines are a linear max fit for the plots, with corresponding equations. Shaded region signifies where the obstacle is always detected.

E. Noise

Assumptions: We maintain all assumptions in §VI-A except A3, *i.e.*, there exists a depth detection error. Minor inaccuracies in depth detection can be a problem when finding $\Delta x, \Delta z$. In particular, the ith beam returns noisy measurement $y_i = R_i \epsilon_i$ [42] instead of its ground truth distance R_i from the sensor to the obstacle where $\epsilon_i \in [1-\epsilon, 1+\epsilon]$ is unknown noise with a known sensor error bound ϵ, *e.g.*, Velodyne HDL-64E S3 LiDAR has a range detection inaccuracy of ± 2 cm [92].

Under mild assumptions, Theorems 2 and 3 provide a bound of the first angle α_r on the obstacle, and a bound of the angle α_i between two consecutive points on the obstacle.

Theorem 2. Assume $y_r cos(\xi_r) \geq y_{r-1} cos(\xi_{r-1})$ and $y_r sin(\xi_r) \leq y_{r-1} sin(\xi_{r-1})$. Then, the angle α_r is lower bounded by

$$\alpha_r \geq atan2(R_{r-1}(1-\epsilon)sin(\xi_{r-1}) - R_r(1+\epsilon)sin(\xi_r),$$
$$R_r(1+\epsilon)cos(\xi_r) - R_{r-1}(1-\epsilon)cos(\xi_{r-1}))$$

and upper bounded by

$$\alpha_r \leq atan2(R_{r-1}(1+\epsilon)sin(\xi_{r-1}) - R_r(1-\epsilon)sin(\xi_r),$$
$$R_r(1-\epsilon)cos(\xi_r) - R_{r-1}(1+\epsilon)cos(\xi_{r-1})).$$

Proof. Angle α_r is found by (5). The assumptions $y_r cos(\xi_r) \geq y_{r-1} cos(\xi_{r-1})$ and $y_r sin(\xi_r) \leq y_{r-1} sin(\xi_{r-1})$ imply that

$$\alpha_r = atan2(|y_r sin(\xi_r) - y_{r-1} sin(\xi_{r-1})|,$$
$$|y_r cos(\xi_r) - y_{r-1} cos(\xi_{r-1})|)$$
$$= atan2(y_{r-1} sin(\xi_{r-1}) - y_r sin(\xi_r),$$
$$y_r cos(\xi_r) - y_{r-1} cos(\xi_{r-1})).$$

Considering the fact that tan is a strictly increasing function in the domain $[0^o, 90^o)$, we can find the lower bound and upper bounds presented in the theorem statement. □

Theorem 3. Assume that two consecutive points R_{i-1} and R_i land on the obstacle, and that $y_i sin(\xi_i) \leq y_{i-1} sin(\xi_{i-1})$. Then, the angle α_i is lower bounded by

$$\alpha_i \geq atan2(R_i(1+\epsilon)sin(\xi_i) - R_{i-1}(1-\epsilon)sin(\xi_{i-1}),$$
$$R_i(1+\epsilon)cos(\xi_i) - R_{i-1}(1-\epsilon)cos(\xi_{i-1}))$$

and upper bounded by 90°.

Proof. The function $atan2$ is upper bounded by 90° in all domains. Similarly, the lower bound can be found with the proof of Theorem 2. Details omitted due to the space limit. □

The assumptions $y_r cos(\xi_r) \geq y_{r-1} cos(\xi_{r-1})$ and $y_r sin(\xi_r) \leq y_{r-1} sin(\xi_{r-1})$ are mild because $R_r cos(\xi_r) \geq R_{r-1} cos(\xi_{r-1})$ and $R_r sin(\xi_r) \leq R_{r-1} sin(\xi_{r-1})$ hold for most of the cases, and ϵ_i is around 1. For the same reason, the assumption $y_i sin(\xi_i) \leq y_{i-1} sin(\xi_{i-1})$ is mild as well.

F. Summary

Using ξ and H_L from the Waymo Open Dataset [45], (13) yields Figure 6a for various α_{th} and D (X-Axis). Figure 6b shows the same when $\alpha_o = 60^o$ as in Section VI-B. Figure 6c is based on Section VI-C, assuming obstacle is 0.271 m above the ground, based on ground clearance of a sedan [93]. The varying nature of the colored plot is due to the discrete nature of LiDAR beams. To get a usable bound, like (1), we determine a linear max curve fit for each case, where the dashed line is always greater than or equal to the original plot. Shaded region is where the obstacle is always detected.

In this section, we have developed detectability model for the Depth Clustering algorithm. We find that not only is such analysis possible, but it also yields human perceptible bounds. This work stands as the first step in establishing the verifiability of classical obstacle detection algorithms and their use in AV for safety-critical obstacle avoidance.
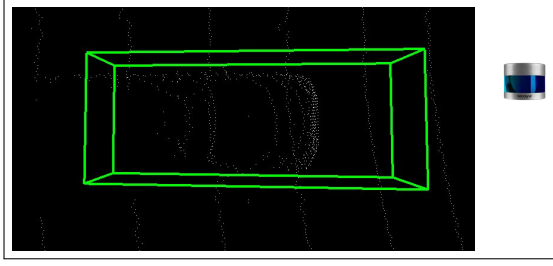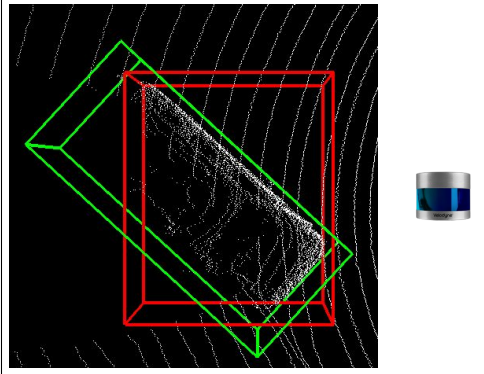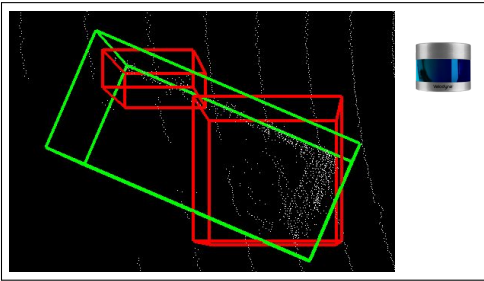
(a) Example showing 3D ground truth box extending too far beyond the obstacle.
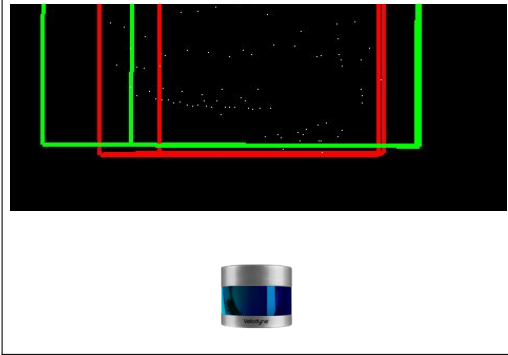
(b) GT extra size causing coverage (73.5% here) to fall just below the threshold (75%).

(c) TP as the detection includes the visible edge of the vehicle towards the AV.

(d) Example of Oversegmentation

Figure 7: Examples from manual inspections of FN candidates. The white points are in 3D space as returned by LiDAR *i.e.,* the point cloud. Ground truth 3D labels are drawn as green bounding boxes, while detection bounding boxes are red. The point cloud is shown from a top view, and the LiDAR direction is indicated with the sensor image.

Table III: Results for False Negative (FN) Evaluation

Count	Category
98224	Total obstacles (Vehicles, Pedestrian, Cyclist, Unknown)
93030	Total without obstacles closer than D_{min} (C3 §V-B)
7565	Obstacles that pose collision risk (§IV-B) [88]
7402	Detected meeting minimally sufficient requirements (§IV)
153	Ground Truth larger than obstacle
10	Oversegmentation
0	Remaining FN count at 75% coverage.

VII. EVALUATION

In this section, we evaluate the obstacle detection algorithm using real-world sensor data from Waymo Open Dataset [45]. We first determine if the algorithm meets the safety-critical requirements proposed in this work (§IV). For obstacle existence fault or FN evaluation we randomly select 16 clear weather scenes from the 202 scenes in the validation dataset. The random downsampling was necessitated by the manual effort involved in analyzing FN candidates. The scenes included various scenarios, including heavy to low traffic, residential with pedestrians, city and highway driving.

Table II summarizes the parameters used. α_{th} was set at 10°. Distance overestimation error was bounded to $10\ cm + 5\%$ of actual distance. The constant was chosen as a small value to keep the error bound low at close distances. However, since the depth perception accuracy reduces with distance an additional percentage-based threshold allows accommodation of sensor limits while having a low impact. A limited coverage threshold of 75% was used as a threshold for True Positive (TP) detection, chosen to parallel the strict metric for IOU from COCO Dataset [94]. All FN candidates were analyzed manually using a visualizer provided by Bogoslavskyi and Stachniss [44]. We enhanced the visualizer to aid the manual inspection. Table III summarizes the results.

1) GT Counts: We first determine the total number of GT in the included scenes. All classes, excluding road signs, were counted to yield a total of **98224**. Obstacles closer than first beam ($R_{1,c}$) on the ground, *i.e.,* closer than D_{min} are removed as per C3 in Section V-B, reducing the count of GT to **93030**. Obstacles that do not pose a risk of collision are also removed (§IV-B), leaving **7565** obstacles.

2) Automated Evaluation: We run the detection algorithm and evaluate results based on requirements described in Section IV. **7402** True Positives and **163** FN candidates were found. The FN candidates were then manually inspected.

3) GT Dimension Error: The most common reason for erroneous FN indication were inaccurate GT labels. This inaccuracy of GT was determined based on visual inspection of point clouds. The GT were visibly larger or offset from the actual obstacle. Figures 7a and 7b show such examples. Drawn as per the provided GT, the green box is clearly either larger or offset from the contained obstacle. Similarly, as shown in Figure 7b, the GT inaccuracy is enough to bring the coverage below the 75% threshold. In some cases, the error is small, *e.g.,* Figure 7c. However when close to the AV, the small GT

error can still be larger than the distance overestimation error allowed. We consider these detections as TP after ascertaining that the detection bounding box meets the requirements. A total of **153** FN candidates were found to fall in this category.

4) Oversegmentation: Figure 7d shows a case where the obstacle was adequately detected but segmented into more than one bounding box. Since the second bounding box did not meet the distance threshold, the automated analysis ignored it. However, given the presence of both bounding boxes, we argue that this detection should be considered True Positive. The points on the obstacle were not erroneously considered to be drivable ground. **10** instances of this scenario on the same vehicle were found in consecutive frames.

5) Obstacle Existence Fault: No FN, *i.e.,* obstacle existence faults were found. This is not surprising given the low minimum height bounds determined in Section VI and Figure 6.The max curve fits in Figures 6a, 6b and 6c are conservative linear approximations. Ignoring the max curve fits, the actual bounds in Figures 6a, 6b and 6c were less than $0.5\ m$ in all cases within the sensor range of $75\ m$.

VIII. DISCUSSION

Generality: The safety-critical requirements for collision avoidance are applicable to all ground based autonomous vehicles. The LiDAR parameters used apply broadly to all LiDAR sensors. The constraints are also generally applicable, except `C3` which is based on the algorithm. Other verifiable algorithms may have different algorithmic constraints. The detectability model in this work is specific to the chosen algorithm. However, the analysis shows that it is indeed possible to derive strict human perceptible bounds on the detectability of these algorithms and serves as guidance for verification of similar algorithms.

Human Comprehensibility: As shown in this work, safety standards, policies and limitations of the example algorithm are in human comprehensible definitions. This makes such policies realistically implementable and enforceable. Human comprehensible limitations also implicitly protect against adversarial objects [95], [96]. This is in contrast to the machine learning based solutions where requirements, faults and adversarial perturbations are not always expressible in human perceptible forms [26].

Requirements: Section IV establishes a minimal set of requirements for collision avoidance in AV. However, additional features of obstacles, if determined in a verifiable manner, can improve the safety envelop, reducing overly conservative behaviors. For example, the collision risk model uses obstacle and AV velocity to determine if there is a potential risk of collision. If velocities cannot be determined reliably, a conservative default high velocity threshold must be used.

Adversarial Objects: The failure modes for analyzable algorithms are well defined and expressible in limitations like minimum height and slopes from ground. Whereas DNN based detectors can have varied failure modes, including fully or partially designed adversarial objects [95]–[98]. The difference in failure modes suggests than an ensemble of the two would be robust against attacks using adversarial designed objects.

IX. FUTURE WORK

Integration: In this work we focus on *detection* of faults in obstacle existence detection. However, the *reaction* to them, *i.e.,* fault handling has its own challenges. The integration of verifiable algorithms within existing AV pipelines and fault handling built upon it are the focus of our future research.

Precision Improvements: The complete Depth Clustering algorithm includes methods for improving the detection precision. Future works will address the expansion of the detectability model to include these methods.

Verifiable Algorithms: Despite the necessity of DNN in AV, this work shows that there is a role for analyzable algorithms. Therefore further research is warranted to improve such physical model backed verifiable algorithms. Improvements like lower physical limitations bounds and lower FP rates within those bounds would make these algorithms have reduced impact to the performance of the AV while maintaining the same safety guarantees.

Weather: Impediments like rain, fog, dust or smoke distort the LiDAR returns and can result in LiDAR beams returning with low enough intensity to not be recorded or causing a false early return [99], [100], *i.e.,* violating `C1` (§V-B). Detectability in the presence of such faults is an avenue for future research.

X. CONCLUSION

This paper identifies requirements for safety-critical obstacle detection and presents a safety analysis of an obstacle detection algorithm. The results encourage a thorough separation of mission and safety-critical requirements in autonomous vehicles. Furthermore, verifiable algorithms could fulfill the critical safety requirements offloading that responsibility from DNN based solutions that remain inherently unverifiable.

REFERENCES

[1] Coalition for Future Mobility, "Benefits of self driving vehicles," (accessed Dec 10, 2021). [Online]. Available: https://coalitionforfuturemobility.com/benefits-of-self-driving-vehicles/

[2] National Highway Traffic Safety Administration, "Automated vehicles for safety," (accessed Dec 10, 2021). [Online]. Available: https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety

[3] P. Koopman, "Autonomous vehicle myths: The dirty dozen," https://www.eetimes.com/autonomous-vehicle-myths-the-dirty-dozen/, 2021.

[4] NTSB, "Collision Between a Car Operating With Automated Vehicle Control Systems and a Tractor-Semitrailer Truck Near Williston, Florida, May 7, 2016," Accident Report NTSB/HAR-17/02 PB2017-102600, 2017. [Online]. Available: https://www.ntsb.gov/investigations/accidentreports/reports/har1702.pdf

[5] ——, "Collision between vehicle controlled by developmental automated driving system and pedestrian tempe, arizona, march 18, 2018," Accident Report NTSB/HAR-19/03 PB2019-101402, 2019 [Online]. Available: https://www.ntsb.gov/investigations/AccidentReports/Reports/HAR1903.pdf

[6] ——, "Collision Between a Sport Utility Vehicle Operating With Partial Driving Automation and a Crash Attenuator Mountain View, California, March 23, 2018," Accident Report NTSB/HAR-20/01 PB2020-100112, 2020. [Online]. Available: https://www.ntsb.gov/investigations/AccidentReports/Reports/HAR2001.pdf

[7] ——, "Highway accident brief HWY19FH008," Highway Accident Brief, Accident Number: HWY19FH008, 2019. [Online]. Available: https://www.ntsb.gov/investigations/AccidentReports/Reports/HAB2001.pdf

[8] D. Ngo, "Dashcam shows fatal Tesla Model S crash in China," https://www.cnet.com/news/dash-cam-showed-fatal-tesla-crash-in-china, 2016.

[9] F. Lambert, "Another fatal Tesla crash reportedly on Autopilot emerges, Model S hits a streetsweeper truck," https://electrek.co/2016/09/14/another-fatal-tesla-autopilot-crash-emerges-model-s-hits-a-streetsweeper-truck-caught-on-dashcam/, 2016.

[10] Edward C. Chen and Joel Greer, "Umeda v. Tesla, Inc., United States District Court for the Northern District of California," Case No. 5:20-cv-2926: Docket No. 1, Complaint for damages, 2020. [Online]. Available: https://regmedia.co.uk/2020/04/30/tesla_complaint.pdf

[11] N. Boudette, ""it happened so fast": Inside a fatal tesla autopilot accident," *The New York Times, Aug*, vol. 17, 2021.

[12] J. Torchinsky, "Tesla model 3 drives straight into overturned truck in what seems to be autopilot failure," *JALOPNIK*, (accessed June 01, 2020). [Online]. Available: https://jalopnik.com/tesla-model-3-drives-straight-into-overturned-truck-in-1843827507

[13] D. Kerman, "Lawsuit: Family blames tesla's autopilot for deadly crash," *KRON4*, 2021. [Online]. Available: https://www.kron4.com/news/bay-area/lawsuit-family-blames-teslas-autopilot-for-deadly-crash/

[14] J. Jernagan, "Fatal tesla crash into fire truck was one year ago today," *Banner Graphic*, 2020. [Online]. Available: https://www.bannergraphic.com/story/2856765.html

[15] AP News, "Feds will investigate deadly tesla crash in california," 2020. [Online]. Available: https://apnews.com/article/technology-business-los-angeles-us-news-california-6eae3986e7d9c1d00db7d52146cddf23

[16] G. RØED, "Tesla on auto-steering when man was cut down [translated]," *MOTOR*, 2020. [Online]. Available: https://www.motor.no/autopilot-nyheter-tesla/tesla-pa-auto-styring-da-mann-ble-meid-ned/188623

[17] C. Alexander, "Two dead in wreck involving tesla that pct. 4 says burned nearly four hours," *KHOU11*, 2021. [Online]. Available: https://www.khou.com/article/news/local/tesla-spring-crash-fire/285-c28a4993-5b5f-43f4-a924-e39638390647

[18] CBS Los Angeles, "Tesla driver killed, 2 seriously hurt in big-rig wreck on 210 freeway in fontana," 2021. [Online]. Available: https://www.cbsnews.com/losangeles/news/tesla-driver-killed-big-rig-wreck-on-210-freeway-in-fontana-good-samaritan-hurt/

[19] N. N. York, "52-year-old fixing flat on long island expressway hit, killed by tesla driver: Cops," 2021. [Online]. Available: https://www.nbcnewyork.com/news/local/52-year-old-fixing-flat-on-long-island-expressway-hit-killed-by-tesla-driver-cops/3175414/

[20] P. Koopman and M. Wagner, "Challenges in autonomous vehicle testing and validation," *SAE International Journal of Transportation Safety*, vol. 4, no. 1, pp. 15–24, 2016.

[21] ——, "Autonomous vehicle safety: An interdisciplinary challenge," *IEEE Intelligent Transportation Systems Magazine*, 2017.

[22] D. Heaven *et al.*, "Why deep-learning ais are so easy to fool," *Nature*, vol. 574, no. 7777, pp. 163–166, 2019.

[23] P. Koopman, "The heavy tail safety ceiling," in *Automated and Connected Vehicle Systems Testing Symposium*, 2018.

[24] B. Biggio *et al.*, "Evasion attacks against machine learning at test time," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2013, pp. 387–402.

[25] D. Amodei *et al.*, "Concrete problems in AI," *arXiv preprint arXiv:1606.06565*, 2016.

[26] X. Huang *et al.*, "Safety verification of deep neural networks," in *International conference on computer aided verification*. Springer, 2017, pp. 3–29.

[27] Y. Tian *et al.*, "Deeptest: Automated testing of deep-neural-network-driven autonomous cars," in *Proceedings of the 40th international conference on software engineering*, 2018, pp. 303–314.

[28] Y. Li *et al.*, "Deepdyve: Dynamic verification for deep neural networks," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 101–112.

[29] O. Willers *et al.*, "Safety concerns and mitigation approaches regarding the use of deep learning in safety-critical perception tasks," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2020, pp. 336–350.

[30] P. Feiler *et al.*, "Four pillars for improving the quality of safety-critical software-reliant systems," CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, Tech. Rep., 2013.

[31] M. Heimdahl *et al.*, "Software assurance approaches, considerations, and limitations: Final report," Technical Report DOT/FAA/TC-15/57. Federal Aviation Administration, US DOT, Tech. Rep., 2016.

[32] C. Wilkinson *et al.*, "Verification of adaptive systems," Technical Report DOT/FAA/TC-16/4. Federal Aviation Administration, US DOT, Tech. Rep., 2016.

[33] E. Jenn *et al.*, "Identifying challenges to the certification of machine learning for safety critical systems," in *European Congress on Embedded Real Time Systems (ERTS 2020)*, 2020.

[34] A. Pereira and C. Thomas, "Challenges of machine learning applied to safety-critical cyber-physical systems," *Machine Learning and Knowledge Extraction*, vol. 2, no. 4, pp. 579–602, 2020.

[35] N. Altman *et al.*, "Simplex TM in a Hostile Communications Environment: The Coordinated Prototype," Citeseer, Tech. Rep., 1999.

[36] L. Sha *et al.*, "Using simplicity to control complexity," *IEEE Software*, vol. 18, no. 4, pp. 20–28, 2001.

[37] P. Wei *et al.*, "Fusion of an ensemble of augmented image detectors for robust object detection," *Sensors*, vol. 18, no. 3, p. 894, 2018.

[38] N. M. Gürel *et al.*, "Knowledge enhanced machine learning pipeline against diverse adversarial attacks," in *International Conference on Machine Learning*. PMLR, 2021, pp. 3976–3987.

[39] Z. Wang *et al.*, "A review of vehicle detection techniques for intelligent vehicles," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[40] Y. Li and J. Ibanez-Guzman, "Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems," *IEEE Signal Processing Magazine*, 2020.

[41] R. Roriz *et al.*, "Automotive LiDAR technology: A survey," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[42] Y. Wang *et al.*, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8445–8453.

[43] I. Bogoslavskyi and C. Stachniss, "Fast range image-based segmentation of sparse 3d laser scans for online operation," in *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[44] ——, "Efficient online segmentation for sparse 3d laser scans," *PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, pp. 1–12, 2017.

[45] P. Sun *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2446–2454.

[46] Q. Yang *et al.*, "Introspective false negative prediction for black-box object detectors in autonomous driving," *Sensors*, vol. 21, no. 8, p. 2819, 2021.

[47] Steven Posada, Office of Defects Investigation, NHTSA, USDOT, "ODI RESUME," Investigation PE 21-020, 2021 [Online]. [Online]. Available: https://static.nhtsa.gov/odi/inv/2021/INOA-PE21020-1893.PDF

[48] N. Kalra, *Challenges and approaches to realizing autonomous vehicle safety*. RAND Santa Monica, 2017.

[49] P. Penmetsa *et al.*, "Effects of the autonomous vehicle crashes on public perception of the technology," *IATSS Research*, 2021.

[50] W. Li *et al.*, "Safety criticism and ethical dilemma of autonomous vehicles," *AI and Ethics*, pp. 1–6, 2022.

[51] A. R Yacoub and B. Briggs, "Liability and regulatory oversight of semi-autonomous and autonomous vehicles," *Available at SSRN: https://ssrn.com/abstract=4042306*, 2022.

[52] P. Jing *et al.*, "Listen to social media users: Mining chinese public perception of autonomous vehicles after crashes," *Available at SSRN: https://ssrn.com/abstract=4011917*, 2022.

[53] G. Katz *et al.*, "Reluplex: An efficient smt solver for verifying deep neural networks," in *International Conference on Computer Aided Verification*. Springer, 2017, pp. 97–117.

[54] M. Gharib *et al.*, "On the safety of automotive systems incorporating machine learning based components: a position paper," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE, 2018, pp. 271–274.

[55] K. R. Varshney, "Engineering safety in machine learning," in *2016 Information Theory and Applications Workshop (ITA)*. IEEE, 2016, pp. 1–5.

[56] C. Urban and A. Miné, "A review of formal methods applied to machine learning," *arXiv preprint arXiv:2104.02466*, 2021.

[57] P. Pinggera *et al.*, "High-performance long range obstacle detection using stereo vision," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 1308–1313.

[58] M. Himmelsbach *et al.*, "Fast segmentation of 3d point clouds for ground vehicles," in *2010 IEEE Intelligent Vehicles Symposium*. IEEE, 2010, pp. 560–565.

[59] D. Korchev *et al.*, "On real-time lidar data segmentation and classification," in *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*. The Steering Committee of The World Congress in Computer Science, Computer . . . , 2013, p. 1.

[60] A. Asvadi *et al.*, "Detection and tracking of moving objects using 2.5 d motion grids," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 788–793.

[61] P. Chu *et al.*, "A fast ground segmentation method for 3d point cloud," *Journal of information processing systems*, vol. 13, no. 3, pp. 491–499, 2017.

[62] D. Zermas *et al.*, "Fast segmentation of 3d point clouds: A paradigm on lidar data for autonomous vehicle applications," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5067–5073.

[63] J. Chen *et al.*, "Lidar cluster first and camera inference later: A new perspective towards autonomous driving," *arXiv preprint arXiv:2111.09799*, 2021.

[64] S. Liu *et al.*, "On removing algorithmic priority inversion from mission-critical machine inference pipelines," in *2020 IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 2020, pp. 319–332.

[65] X. Dang *et al.*, "Sensor fusion-based approach to eliminating moving objects for slam in dynamic environments," *Sensors*, vol. 21, no. 1, p. 230, 2021.

[66] Z. Shen *et al.*, "Fast ground segmentation for 3d lidar point cloud based on jump-convolution-process," *Remote Sensing*, vol. 13, no. 16, p. 3239, 2021.

[67] H. Tian *et al.*, "Unsupervised object detection with lidar clues," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5962–5972.

[68] A. Vobecky *et al.*, "Drive&segment: Unsupervised semantic segmentation of urban scenes via cross-modal distillation," *arXiv preprint arXiv:2203.11160*, 2022.

[69] A. Albarghouthi, "Introduction to neural network verification," *arXiv preprint arXiv:2109.10317*, 2021.

[70] C. Liu *et al.*, "Algorithms for verifying deep neural networks," *arXiv preprint arXiv:1903.06758*, 2019.

[71] H.-D. Tran *et al.*, "Verification of deep convolutional neural networks using imagestars," in *International Conference on Computer Aided Verification*. Springer, 2020, pp. 18–42.

[72] X. Huang *et al.*, "A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability," *Computer Science Review*, 2020.

[73] P. Seiler *et al.*, "Development of a collision avoidance system," *SAE transactions*, pp. 1334–1340, 1998.

[74] J. Funke *et al.*, "Collision avoidance and stabilization for autonomous vehicles in emergency scenarios," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 4, pp. 1204–1216, 2016.

[75] S. Cheng *et al.*, "Longitudinal collision avoidance and lateral stability adaptive control system based on mpc of autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2376–2385, 2019.

[76] C.-F. Lin *et al.*, "Active collision avoidance system for steering control of autonomous vehicles," *IET Intelligent Transport Systems*, vol. 8, no. 6, pp. 550–557, 2014.

[77] H. Wang *et al.*, "Crash mitigation in motion planning for autonomous vehicles," *IEEE transactions on intelligent transportation systems*, vol. 20, no. 9, pp. 3313–3323, 2019.

[78] K. Lee and D. Kum, "Collision avoidance/mitigation system: Motion planning of autonomous vehicle via predictive occupancy map," *IEEE Access*, vol. 7, pp. 52 846–52 857, 2019.

[79] H. Tahir *et al.*, "Heuristic approach for real-time multi-agent trajectory planning under uncertainty," *IEEE Access*, vol. 8, pp. 3812–3826, 2019.

[80] S. Noh and K. An, "Decision-making framework for automated driving in highway environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 58–71, 2017.

[81] S. Noh, "Decision-making framework for autonomous driving at road intersections: Safeguarding against collision, overly conservative behavior, and violation vehicles," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 4, pp. 3275–3286, 2018.

[82] D. Shin *et al.*, "Human-centered risk assessment of an automated vehicle using vehicular wireless communication," *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[83] M.-Y. Yu *et al.*, "Occlusion-aware risk assessment for autonomous driving in urban environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2235–2241, 2019.

[84] G. Li *et al.*, "Risk assessment based collision avoidance decision-making for autonomous vehicles in multi-scenarios," *Transportation research part C: emerging technologies*, vol. 122, p. 102820, 2021.

[85] S. Summers and J. Lygeros, "Verification of discrete time stochastic hybrid systems: A stochastic reach-avoid decision problem," *Automatica*, vol. 46, no. 12, pp. 1951–1961, 2010.

[86] S. Summers *et al.*, "A stochastic reach-avoid problem with random obstacles," in *Proceedings of the 14th international conference on Hybrid systems: computation and control*, 2011, pp. 251–260.

[87] N. J. Goodall, "Ethical decision making during automated vehicle crashes," *Transportation Research Record*, 2014.

[88] A. Bansal *et al.*, "Risk ranked recall: Collision safety metric for object detection systems in autonomous vehicles," in *2021 10th Mediterranean Conference on Embedded Computing (MECO)*, 2021.

[89] C. Schmidt *et al.*, "Research on trajectory planning in emergency situations with multiple objects," in *2006 IEEE Intelligent Transportation Systems Conference*. IEEE, 2006, pp. 988–992.

[90] Y. Ming *et al.*, "A survey of path planning algorithms for autonomous vehicles," *SAE international journal of commercial vehicles*, vol. 14, no. 1, pp. 02–14, 2021.

[91] American Association of State Highway and Transportation Officials, "Policy on geometric design of highways and streets," *American Association of State Highway and Transportation Officials, Washington, DC*, vol. 1, no. 990, p. 158, 2001.

[92] Velodyne Lidar, Inc., "Velodyne Lidar HDL-64E S3 HD Real-Time 3D Lidar," 2018. [Online]. Available: https://velodynelidar.com/products/hdl-64e/

[93] The MathWorks, Inc., "Sedan," 2021. [Online]. Available: https://www.mathworks.com/help/driving/ref/sedan.html

[94] C. Consortium, "Detection evaluation," https://cocodataset.org/#detection-eval, 2021.

[95] Y. Cao *et al.*, "Adversarial objects against lidar-based autonomous driving systems," *arXiv preprint arXiv:1907.05418*, 2019.

[96] J. Tu *et al.*, "Physically realizable adversarial examples for lidar object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 716–13 725.

[97] M. Abdelfattah *et al.*, "Adversarial attacks on camera-lidar models for 3d car detection," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 2189–2194.

[98] Y. Zhu *et al.*, "Can we use arbitrary objects to attack lidar perception in autonomous driving?" in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 1945–1960.

[99] R. Heinzler *et al.*, "Weather influence and classification with automotive lidar sensors," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1527–1534.

[100] A. M. Wallace *et al.*, "Full waveform lidar for adverse weather conditions," *IEEE transactions on vehicular technology*, 2020.