Estimation of Looming from LiDAR

Juan D. Yepes and Daniel Raviv

Florida Atlantic University
Electrical Engineering and Computer Science Department
Boca Raton, FL, USA
jyepes@fau.edu, ravivd@fau.edu

Keywords: Looming, Obstacle Avoidance, Collision Free Navigation, LiDAR, Threat Zones, Autonomous Vehicles.

Abstract: Looming, traditionally defined as the relative expansion of objects in the observer’s retina, is a fundamental visual cue for perception of threat and can be used to accomplish collision free navigation. The measurement of the looming cue is not only limited to vision and can also be obtained from range sensors like LiDAR (Light Detection and Ranging). In this article we present two methods that process raw LiDAR data to estimate the looming cue. Using looming values, we show how to obtain threat zones for collision avoidance tasks. The methods are general enough to be suitable for any six-degree-of-freedom motion and can be implemented in real-time without the need for fine matching, point-cloud registration, object classification or object segmentation. Quantitative results using the KITTI dataset shows advantages and limitations of the methods.

1 INTRODUCTION

Collision avoidance continues to be one of the greatest challenges in unmanned vehicles (Yasin et al., 2020) and autonomous driving (Roriz et al., 2021), where the demand for increasingly robust, fast and safe systems is crucial for the development of these industries. It is closely tied with the perception of threat. For creatures in nature, this task seems to be more fundamental than the recognition of shapes (Albus and Hong, 1990). Studies in biology have shown strong evidence of neural circuits in the brains of creatures related to the identification of looming (Ache et al., 2019). Basically, creatures have evolved instinctive escaping behaviors that tie perception directly to action. In this way they can avoid imminent threat from predators that project an expanding image on their visual systems (Evans et al., 2018) (Yilmaz and Meister, 2013).

The visual looming cue, defined quantitatively by (Raviv, 1992) as the instantaneous change of range over the range, is related to the increase in size of an object projected on the observer’s retina. It is an indication of threat that can be used to accomplish collision avoidance tasks.

The looming cue is not limited exclusively to vision and can be computed from active sensors like LiDAR (Light Detection and Ranging). LiDAR systems are very popular in autonomous vehicles since they are less sensitive to lighting conditions, have wider field of view (FOV) than cameras and require less processing time and computation power (Wang et al., 2018).

In general, architectures for autonomous vehicles consist of three sub-systems: Perception, Planning and Control (Sharma et al., 2021). The approaches for implementing the Perception sub-system can be mainly divided subsequently into three categories: camera-based, LiDAR-based and sensor fusion-based approaches (Wang et al., 2018).

Traditionally the task of obstacle avoidance is exclusively related to the Planning and Control sub-systems where the action can be categorized further into four major approaches: Geometric, Force-Field, Optimized, Sense and Avoid (Yasin et al., 2020). The Planning and Control sub-systems demand from the Perception sub-system, as a prerequisite, the detection and classification of objects in 3D or 2D space prior to any global or local path planning action (Mujumdar and Padhi, 2011).

In contrast, it has been shown that obstacle avoidance tasks can be accomplished without 3D reconstruction (structure from motion), using the Active Perception paradigm. The idea is to estimate normal flow from derivatives of image intensity and from there an estimation of time-to-contact with features in the scene allows for local navigation actions that steer the robot away from the hazard (Aloimonos, 1992).

In this paper we follow an approach similar to (Raviv and Joarder, 2000). It uses 2D raw data to directly
get quantitative indication of visual threat. We present methods for obtaining looming values using raw LiDAR data and threat zones. This allows for real-time collision avoidance tasks without 3D reconstruction or scene understanding. In the proposed approach further processing of the LiDAR point cloud is not required for obtaining threat zones.

2 RELATED WORK

2.1 Visual looming

The visual looming cue is related to the relative change in size of an object projected on the observer’s retina as the range to the object varies (Figure 1). It is quantitatively defined as the negative value of the time derivative of the range between the observer and a point in 3D divided by the range (Raviv, 1992):

\[ L = -\lim_{\Delta t \to 0} \frac{(r_2 - r_1)}{\Delta t} \]

(1)

where \( L \) denotes looming, \( t_1 \) represents time instance 1, \( t_2 \) represents time instance 2, \( \Delta t \) is \( t_2 - t_1 \), \( r_1 \) is the range to the point at time instance \( t_1 \), and \( r_2 \) is the range at time instance \( t_2 \).

In the limit \( L \) can also be expressed as:

\[ L = -\left( \dot{r} \right) \]

(2)

where dot denotes derivative with respect to time. The reason for the negative sign in equations (1) and (2) is to associate image expansion with positive values of looming.

The looming \( L \) can also be expressed in vector form as:

\[ L = \frac{\mathbf{t} \cdot \mathbf{r}}{r \cdot r} \]

(3)

where \( \mathbf{t} \) is the instantaneous translation velocity vector of the vehicle and \( \mathbf{r} \) is the relative position vector of the point with respect to the vehicle origin.

Figure 1: Visualizing looming cue: relative expansion of 3D object on 2D image sensor.

2.1.1 Looming properties

Note that the result for \( L \) in equation (3) is a scalar value which is dependent on the vehicle translation component but independent of the vehicle rotation. Also, \( L \) is measured in \([\text{time}^{-1}]\) units.

It was shown that points in space around the vehicle that share the same looming values form equal looming spheres with centers that lie on the instantaneous translation vector \( \mathbf{t} \) and intersect with the vehicle origin. These looming spheres expand and contract depending on the magnitude of the translation vector (Raviv, 1992).

Since an equal looming sphere corresponds to a particular looming value, there are other spheres with varying values of looming with different radii. A smaller sphere signifies a higher value of looming as shown in Figure 2 (\( L_3 > L_2 > L_1 \)).

Regions for obstacle avoidance and other behavior related tasks can be defined using equal looming surfaces. For example, a high danger zone for \( L > L_3 \), medium threat for \( L_3 > L > L_2 \) and low threat for \( L_2 > L > L_1 \).

2.1.2 Advantages of looming

Sensory data from LiDAR systems provide point clouds that can be used for 3D reconstruction. Using information about ego-motion combined with 3D reconstruction it can help in scene understanding (using mainly machine learning and AI techniques) followed by path planning to achieve obstacle avoidance. Note that additional processing is needed when dealing with moving objects.

Using the same LiDAR point cloud, it is possible to get looming directly without 3D reconstruction and without knowing the ego-motion of the Li-
DAR sensor. Looming is measured in time units and provide information about imminent threat to the observer. There is no need for scene understanding such as identifying cars, bikes, or pedestrians. In addition, looming provides threats for moving objects as well. Extracting looming for obstacle avoidance using point cloud raw data from LiDAR is the main contribution of this paper.

2.1.3 Measuring visual looming

Several methods were shown to quantitatively extract the visual looming cue on a 2D image sequence by measuring attributes like area, brightness, texture density and image blur (Raviv and Joarder, 2000).

Relative change in image area: Visual looming is calculated from the relative temporal change in image area and the surface orientation by assuming that the camera can be fixated on a planar patch of the object.

Relative change in image brightness: Visual looming can also be calculated from the relative rate of change of image brightness by assuming Lambertian surfaces, i.e., surfaces that are equally bright irrespective of the angle of view.

Relative change in texture density: Visual looming can be calculated from the texture density and its temporal changes by assuming a small area around the point of fixation to be locally planar (and the density of texture primitives on the 3D surface to be constant in that region).

Relative change in image blur: The relative change in image blur or defocus is used to calculate the visual looming. It is assumed that the camera’s depth of field is very small. Looming can be measured by studying the relative change in the radius of the blur circle.

The Visual Threat Cue (VTC), just like the visual looming, is a measurable time-based scalar value that provides some measure for a relative change in range between a 3D surface and a moving observer. The VTC is computed from relative variations of a global image dissimilarity measure called the image quality measure (IQM) that is extracted directly from grayscale level images. The method has been shown to be robust allowing for closed loop control of vehicles (Kundur and Raviv, 1999).

Event-based cameras were shown to detect looming objects in real-time from optical flow using clustering of pixels. The method is fast, taking only microseconds to process each event assuming that a single object is moving in the scene (Ridwan, 2018).

2.2 LiDAR for obstacle detection and avoidance

In general, collision avoidance approaches require 3D reconstruction of the environment where collision free paths can be computed and executed.

Several deep learning and path planning techniques for autonomous driving were mentioned in (Grigorescu et al., 2020). The path planning process considers all possible obstacles in the environment in order to calculate a trajectory along a collision-free route.

General approaches for collision avoidance that use reactive, deliberate or hybrid planning where identification of objects is a requirement before planning an evasive action were mentioned in (Yasin et al., 2020).

An online 2D LiDAR-based object detection and tracking approach using cylinders to model obstacles in the environment is proposed in (Wang et al., 2018). Point clouds, provided by LiDAR sensors, are the preferred representation for many scene understanding related applications. These point clouds are processed to build real-time 3D localization maps, using SLAM (Simultaneous Localization and Mapping) and related techniques. A LiDAR-only SLAM method called LOAM that estimate vehicle velocity by odometry and map the environment in real-time using fine matching and point cloud registration was developed by (Zhang and Singh, 2014).

Several deep learning methods for 3D understanding, including 3D shape classification, 3D object detection and tracking were presented by (?). Methods for ground segmentation and identification of driveable space were mentioned by (Roriz et al., 2021).

In general, most methods use 3D object detection, segmentation and scene understanding. In addition, ego-motion is required for obstacle avoidance and autonomous navigation.

In contrast, our approach provides relevant information for the task of collision avoidance in the form of a real-time looming threat map. This allows the direct transition from perception to action without the need of object identification, finding ego-motion or 3D reconstruction.
3 METHOD

3.1 Derivation of looming from the relative velocity field

We will derive a general expression for Looming ($L$) for any six-degree-of-freedom motion that involves velocity and range using spherical coordinates. Then we can apply this expression with LiDAR data.

Vehicle motion Consider a vehicle undergoing a general six-degree-of-freedom motion in a 3-D space relative to an arbitrary stationary reference point. At any given time, the vehicle will have an associated translation velocity vector $t$ and a rotation vector $\Omega$ relative to the world frame as shown on Figure 3.

LiDAR coordinate system Consider a local coordinate system centered at the LiDAR sensor, fixed to the moving vehicle. We chose the $x$-axis to be aligned with the forward direction of the vehicle and the $z$-axis to be aligned with the vertical orientation.

Refer to Figure 4: In this frame any stationary feature on the 3-D scene can be represented by spherical coordinates $(r, \theta, \phi)$ where $r$ is the radial range to the feature point $F$, $\theta$ is the azimuth angle measured from the $x$-axis and $\phi$ is the elevation angle from the XY-plane. Rectilinear and spherical coordinates conversions are given by:

$$
x = r \cos \phi \cos \theta, \quad r = \sqrt{x^2 + y^2 + z^2}$$
$$
y = r \cos \phi \sin \theta, \quad \theta = \arctan(y, x)$$
$$
z = r \sin \phi, \quad \phi = \arctan(z, \sqrt{x^2 + y^2})$$

Note that in order to eliminate quadrant sign confusion we take advantage of two argument $\arctan$ function for the computation of $\theta$ and $\phi$.

Relative velocity field In our analysis the vehicle is undergoing a generalized six-degree-of-freedom motion in the world frame. This is equivalent to interpreting the motion as the vehicle being stationary and the feature point $F$ moving on the LiDAR frame with opposite velocity vector $-t$ and rotation $-\Omega$.

In this LiDAR frame the feature $F$ has relative velocity vector:

$$
V = (-t) + (-\Omega \times r)
$$

Notice that for a given feature $F$ the relative velocity vector $V$ can be interpreted as the relative velocity field in 3-D due to ego-motion. ($r$ in bold refers to the range vector and the scalar $r$ to its magnitude, i.e $r = |r|$).

We can conveniently express $V$ using rectilinear or spherical unit vector components. For this purpose we can apply the derivative with respect to time of $x, y, z$ in equation set (4), shown here in matrix notation:

$$
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{z}
\end{bmatrix} =
\begin{bmatrix}
\cos \theta \cos \phi & -\sin \theta & -\cos \theta \sin \phi \\
\sin \theta \cos \phi & \cos \theta & -\sin \theta \sin \phi \\
\sin \phi & 0 & \cos \phi
\end{bmatrix}
\begin{bmatrix}
\dot{r} \\
\dot{\theta} \cos \phi \\
\dot{\phi}
\end{bmatrix}
$$

Since the inverse of the transformation matrix above is also its transpose we can easily find the corresponding components in spherical coordinates:

$$
\begin{bmatrix}
\dot{r} \\
\dot{\theta} \cos \phi \\
\dot{\phi}
\end{bmatrix} =
\begin{bmatrix}
\cos \theta \cos \phi & \sin \theta \cos \phi & \sin \phi \\
-\sin \theta & \cos \theta & 0 \\
-\cos \theta \sin \phi & -\sin \theta \sin \phi & \cos \phi
\end{bmatrix}
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{z}
\end{bmatrix}
$$
Consider also spherical unit vectors $e_r, e_\theta, e_\phi$ associated with a stationary feature $F$ in the 3-D scene. Note that there are conversions from spherical unit vectors $(e_r, e_\theta, e_\phi)$ to/from rectilinear unit vectors $(i, j, k)$:

\[
\begin{bmatrix}
e_r \\
e_\theta \\
e_\phi
\end{bmatrix} =
\begin{bmatrix}
\cos \theta \cos \phi & \sin \theta \cos \phi & \sin \phi \\
-\sin \theta & \cos \theta & 0 \\
-\cos \theta \sin \phi & -\sin \theta \sin \phi & \cos \phi
\end{bmatrix}
\begin{bmatrix}
i \\
j \\
k
\end{bmatrix}
\tag{8}
\]

\[
\begin{bmatrix}
i \\
j \\
k
\end{bmatrix} =
\begin{bmatrix}
\cos \phi \cos \theta & -\sin \phi \cos \theta & \sin \theta \\
\sin \phi \cos \theta & \cos \phi \cos \theta & 0 \\
\sin \phi & -\sin \theta \sin \phi & \cos \phi
\end{bmatrix}
\begin{bmatrix}
e_r \\
e_\theta \\
e_\phi
\end{bmatrix}
\tag{9}
\]

We can now decompose $V$ by its components using the convenient unit vectors $e_r, e_\theta, e_\phi$:

\[V = i e_r + r \phi \cos(\phi) e_\theta + r \phi e_\phi\]  \tag{10}

Note that equations (5) and (10) refer to the same relative velocity field $V$.

Looming from normalized velocity field: We can get another expression for looming ($L$) by normalizing the relative velocity field $V$ by $r$. Dividing equation (5) by $r$ and expanding $t$ and $\Omega$ using spherical unit vectors yield:

\[
\frac{V}{r} = \frac{-t}{r} + \left(\frac{-\Omega \times r}{r}\right)
\]

\[
= \frac{-(t, e_r + t_\theta e_\theta + t_\phi e_\phi)}{r}
\]

\[
- (\Omega, e_r + \Omega_\theta e_\theta + \Omega_\phi e_\phi) \times e_r
\]

\[
= \left(\frac{-t_r}{r}\right) e_r + \left(\frac{-t_\theta}{r}\right) e_\theta + \left(\frac{-t_\phi}{r}\right) e_\phi
\]

\[
+ (-\Omega_\phi) e_\theta + \Omega_\phi e_\phi
\]

where:

\[
t_r = t \cdot e_r, \quad \Omega_r = \Omega \cdot e_r
\]

\[
t_\theta = t \cdot e_\theta, \quad \Omega_\theta = \Omega \cdot e_\theta
\]

\[
t_\phi = t \cdot e_\phi, \quad \Omega_\phi = \Omega \cdot e_\phi
\]

In addition, by dividing equation (10) by $r$ we obtain:

\[
\frac{V}{r} = \left(\frac{r}{r}\right) e_r + \theta \cos(\phi) e_\theta + \phi e_\phi
\]

\tag{12}

By equating (11) and (12) and by isolating only the resultant $e_r$ components on both sides of these equations we get:

\[
\left(\frac{-t_r}{r}\right) e_r = \left(\frac{r}{r}\right) e_r, \quad \tag{13}
\]

or

\[
\frac{t \cdot e_r}{r} = - \frac{r}{r} \quad \tag{14}
\]

Therefore, by substituting $L$ from (2) into (14) we obtain another expression for looming in spherical coordinates:

\[
L = \frac{t \cdot e_r}{r} \quad \tag{15}
\]

Notice that by knowing the instantaneous translation vector of the vehicle $t$ and range $r$, we can compute the looming value ($L$) for any point along the directional unit vector $e_r$.

### 3.2 Looming from LiDAR

We propose two ways to extract looming from LiDAR data.

#### 3.2.1 Using LiDAR data only

Looming can be estimated using LiDAR point clouds. Each full scan of the LiDAR results in a range image grid. Using two consecutive scans we get two range image grids as obtained from the same 3D environment. Theoretically, if the range value of each LiDAR pixel in each grid corresponds to the same 3D point in space, we obtain two range values from which the looming can be estimated, where the only error is due to the range values as obtained from the LiDAR measurements.

However, this is practically not the case: the problem with this approach is that due to the vehicle motion the assumption may hold only when the changes from one image to another are infinitesimally small. To minimize the effect of incorrect looming calculations and improve the robustness of the approach, we use interpolation/decimation and discretization of the data in the grid allowing for range values that are closer to the real ones.

We understand that this method adds error to the calculation of looming, but as can be seen in the results section it is possible to get some crude estimation of the looming values.

Equation (16) shows the specific calculations.

\[
L_{ij} = \frac{\left(\frac{r(\theta_i, \phi_j)}{r(\theta_i, \phi_{j+1})} \right) \Delta r}{r(\theta_i, \phi_j)} \quad \tag{16}
\]
where $i = [1, 2, ..., n]$, $n$ is the number of points on the grid, $j + 1, j$ correspond to the current and previous point cloud sample sets. The resulting $L_{ij}$ can be interpreted as a LiDAR-based looming image from where threat zones can be obtained for the purpose of collision avoidance actions.

### 3.2.2 Using LiDAR + IMU

Another method for obtaining looming is using LiDAR and IMU (Inertial Measurement Unit). Even though this is not the primary focus of this paper, its purpose is to demonstrate that LiDAR combined with IMU increase the robustness of computation of looming. However, this approach has its drawback because it gives incorrect values of looming for moving objects. Note that it requires only the translation vector and not the rotation component of the LiDAR sensor. It uses ego-motion only partially, and there is no need for 3D reconstruction and image understanding.

We can compute looming ($L$) for every measured stationary point by using equation (15). The range $r$ is provided by the LiDAR sensor for every point on the point cloud. The instantaneous translation vector $t$ of the vehicle can be obtained from the IMU or by other odometry methods.

For special cases where vehicles have little side motion, like most automotive applications, the translation vector can be assumed to be equal to the vehicle forward motion component, and the magnitude (speed) can be obtained from the vehicle odometer or from the speed encoder of the wheel.

### 3.2.3 Threat maps

From either method, threat regions can be obtained from looming values by assigning specific thresholds. For example, High, Medium, or Low threat zones. These threat maps were obtained directly from measurements and are sufficient for identifying location threats. Object identification or 3D reconstruction is not necessary.

Figure 5.a shows the looming sphere as a function of time for a given constant speed. Figures 5.b and 5.c show equal looming spheres using two threat zones. An increase in vehicle speed, as can be seen in Figure 5.c (compared to Figure 5.b) causes expansion of the time-based high threat zone and thereby includes new obstacles in these zones which were not deemed to be obstacles in Figure 5.b.

This is in contrast to raw LiDAR data that provides range only and not threat values.

### 4 RESULTS

We present results of the methods for obtaining looming and threat zones using synthetic and real data.

#### 4.1 Ground truth looming with synthetic data

A simulation of a moving vehicle in a stationary environment was performed using the Unity3D game engine. In this simulation the vehicle is translating...
Figure 7: LiDAR data from the KITTI dataset: a) Original image from a color camera mounted on the vehicle, b) 360 degrees LiDAR data, the yellow rectangle corresponds to the color camera field of view, c) Top view of point cloud in 3D, d) Bird’s eye 3D view of point cloud.

Figure 8: Estimated looming from LiDAR only. (Refer to text for details).
Figure 9: Looming from LiDAR (red is potential threat): a) Top view, b) Bird’s eye view.

Figure 10: Looming from Lidar: a) Original image, b) Depth from LiDAR, c) Looming from LiDAR + IMU, d) Estimated looming from LiDAR only.
forward at a constant speed between some stationary objects. The top part of Figure 6 shows an actual image of the scene. The bottom part of Figure 6 shows ground truth looming computed using equation (2) and shown from the observer point of view. A pixel shader filter was implemented to visualize the looming value as levels of red color. Brighter red color corresponds to higher value of looming.

4.2 LiDAR data from KITTI dataset

We processed real data using a particular city drive from the well-known KITTI dataset (Geiger et al., 2013). The KITTI dataset includes raw data provided by a Velodyne 3D laser scanner (LiDAR) along with velocity vector from a GPS/IMU inertial navigation system. Color camera images were only used as reference as shown in Figure 7.a. The LiDAR sensor specifications are: Velodyne HDL-64E rotating 3D laser scanner, 10 Hz, 64 beams, 0.09-degree angular resolution, 2 cm distance, accuracy, provides around 1.3 million points/second, field of view: 360 degrees horizontal, 26.8 degrees vertical, range:120 m.

Shown in Figure 7 are the original images from one of the vehicle color cameras and gray-scale representations of the raw LiDAR data. A yellow rectangle indicates where the LiDAR data matches the field of view of the color camera. Also shown are top and bird’s eye views of the data for a specific time instant of the video.

4.3 Estimation of looming using LiDAR

In Figure 8, original images from two consecutive frames (frames 22 and 23) are shown along LiDAR views (360 degrees) for two time instants. The resultant looming estimation using equation (16) is also shown at the bottom of Figure 8. Notice positive values of looming are shown in red and corresponds mainly to points along the forward direction of motion (around the center of the image).

Due to the effect of occlusions of points, there are sudden changes in range between frames, causing an “edge-effect” around objects with incorrect looming values, shown in intense red or blue colors. However an important advantage is that the method incorporates values of looming as obtained from moving objects. For example, the bicycle in the middle of the image is correctly portrayed with dark colors, meaning low values of looming as expected for this kind of motion since the bicycle velocity almost matches the velocity of the vehicle, and the rate of change in range is close to zero.

4.4 Looming from LiDAR and IMU sensors

We use equation (15) to compute looming for each point. From the IMU sensor we obtained the translation velocity vector $\mathbf{t}$ and from the LiDAR sensor we obtained the range $r$. Figure 9 shows top and bird’s eye views for two time instants. Color intensity was assigned to each point representing the obtained
looming value: red for positive values, and blue for negative values.

We can see that the looming is very similar to the expected shown in simulation. This method adequately registers looming for stationary objects; however, it is imprecise for moving objects since the relative speed is not considered.

In Figure 10 a comparison between both methods is shown.

4.5 Threat zones for collision free navigation

In Figure 11 several threat zones are computed using looming from section 4.4: red zone - high threat, orange zone - medium threat and yellow zone - low threat. Notice how the threat zones change sizes between frames, this is due to the change in speed of the vehicle.

5 CONCLUSIONS

Researchers and practitioners use LiDAR technology for autonomous navigation tasks. Sensory data from LiDAR systems provide point clouds that can be used for 3D reconstruction. Combined with information about ego-motion it can lead to scene understanding (using mainly machine learning and AI techniques) followed by path planning to achieve obstacle avoidance.

In this paper we demonstrate how to compute looming directly from raw LiDAR data without 3D reconstruction. There is no need for scene understanding such as identifying cars, bikes, or pedestrians. Two approaches are shown, one uses LiDAR data only, and the other uses LiDAR data combined with IMU.

The approach shows that looming, which is measured in time units, provides information about imminent threat that can potentially be used for navigation tasks such as obstacle avoidance.

When using only LiDAR for obtaining looming (i.e., using multiple LiDAR range images at multiple time instants) relative motion between the vehicle and the environment can yield good approximation of looming even in the presence of moving objects. However, when using instantaneous LiDAR information and ego-motion information (using, for example, GPS data of the moving LiDAR sensor) the looming values that are obtained are better when moving relative to a stationary environment but are incorrect when moving objects are present. The reason is that in the latter case motions of moving objects are not accounted for.

It appears that combining the two methods can further improve the results.

The paper shares highly encouraging initial results and should be considered as “work in progress” as more looming from LiDAR-related methods are being explored.

ACKNOWLEDGEMENTS

The authors thank Dr. Sridhar Kundur for many fruitful discussion and suggestions as well as very detailed comments and clarifications that led to meaningful improvements of this manuscript.

REFERENCES

Ache, J. M., Polsky, J., Alghailani, S., Parekh, R., Breads, P., Peek, M. Y., Bock, D. D., von Reyn, C. R., and Card, G. M. (2019). Neural basis for looming size and velocity encoding in the drosophila giant fiber escape pathway. Current Biology, 29(6):1073–1081.

Albus, J. S. and Hong, T. H. (1990). Motion, depth, and image flow. In Proceedings., IEEE International Conference on Robotics and Automation, pages 1161–1170. IEEE.

Aloimonos, Y. (1992). Is visual reconstruction necessary? obstacle avoidance without passive ranging. Journal of Robotic Systems, 9(6):843–858.

Evans, D. A., Stempel, A. V., Vale, R., Ruehle, S., Lefler, Y., and Branco, T. (2018). A synaptic threshold mechanism for computing escape decisions. Nature, 558(7711):590–594.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. The International Journal of Robotics Research, 32(11):1231–1237.

Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. Journal of Field Robotics, 37(3):362–386.

Kundur, S. R. and Raviv, D. (1999). Novel active vision-based visual threat cue for autonomous navigation tasks. Computer Vision and Image Understanding, 73(2):169–182.

Mujumdar, A. and Padhi, R. (2011). Evolving philosophies on autonomous obstacle/collision avoidance of unmanned aerial vehicles. Journal of Aerospace Computing, Information, and Communication, 8(2):17–41.

Raviv, D. (1992). A quantitative approach to looming. US Department of Commerce, National Institute of Standards and Technology.

Raviv, D. and Joarder, K. (2000). The visual looming navigation cue: A unified approach. Comput. Vis. Image Underst., 79:331–363.

Ridwan, I. (2018). Looming object detection with event-based cameras. University of Lethbridge (Canada).
Roriz, R., Cabral, J., and Gomes, T. (2021). Automotive lidar technology: A survey. IEEE Transactions on Intelligent Transportation Systems.

Sharma, O., Sahoo, N. C., and Puhan, N. (2021). Recent advances in motion and behavior planning techniques for software architecture of autonomous vehicles: A state-of-the-art survey. Engineering applications of artificial intelligence, 101:104211.

Wang, M., Voos, H., and Su, D. (2018). Robust online obstacle detection and tracking for collision-free navigation of multicopter uavs in complex environments. In 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), pages 1228–1234. IEEE.

Yasin, J. N., Mohamed, S. A., Haghbayan, M.-H., Heikonen, J., Tenhunen, H., and Plosila, J. (2020). Unmanned aerial vehicles (uavs): Collision avoidance systems and approaches. IEEE access, 8:105139–105155.

Yilmaz, M. and Meister, M. (2013). Rapid innate defensive responses of mice to looming visual stimuli. Current Biology, 23(20):2011–2015.

Zhang, J. and Singh, S. (2014). Loam: Lidar odometry and mapping in real-time. In Robotics: Science and Systems, volume 2, pages 1–9. Berkeley, CA.