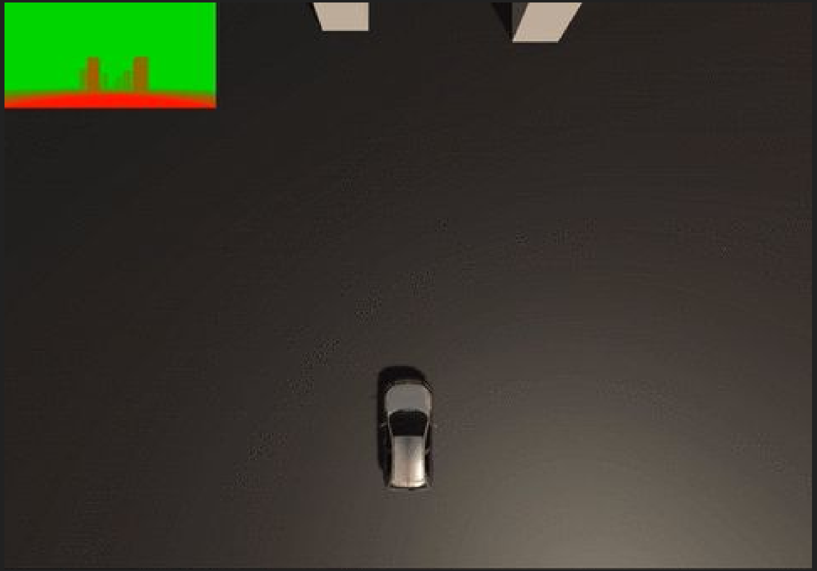
Looming VR

By: Inbar Donag and Alaa Khier

Supervisors: Yaron Honen, Daniel Raviv, Boaz Sternfeld, Alon Zvirin

Project Goals

1. Implementing a simulation of a vehicle with a camera that uses visual looming cue to navigate and avoid obstacles.
2. Visualizing the looming in different ways and compare 2 methods for looming calculation.
3. Developing a VR experience of driverless car ride in the city.



Background

Visual looming is related to an increasing projected size of an object on a viewer's retina as the relative distance between the viewer and the object decreases.

To use looming for navigation we need to know how much the obstacles in view are threatening, and move the car away from the most threatening obstacles.

To calculate the amount of threat (or looming) we used 2 approaches:

1. Using ranges from the car's camera to each rendered pixel.
2. Using texture densities calculated from a sequence of images.

The Ranges Approach

The rages approach is accurate but it can be applied in a simulation only.

We calculated in a fragment shader the range between the car’s camera and the fragment, and used it to calculate looming value for each fragment with this equation:

$$L = \frac{V \cdot R}{R \cdot R}$$

Where L is looming value for the fragment, V is the camera’s velocity vector, and R is the range.

The Texture Densities Approach

This approach can be applied in real world but is less accurate.

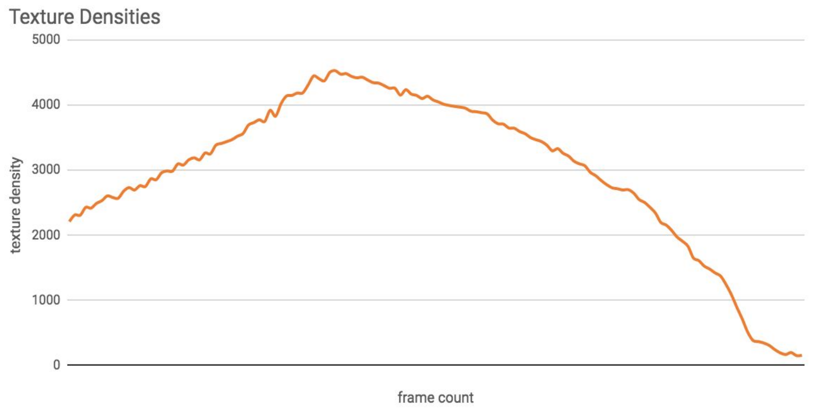
Using a sequence of images we can get from the camera we calculated how the density of the texture on the obstacle is changing over time and used it to calculate looming.

Texture gets less dense -> the obstacle is getting closer -> looming gets higher.

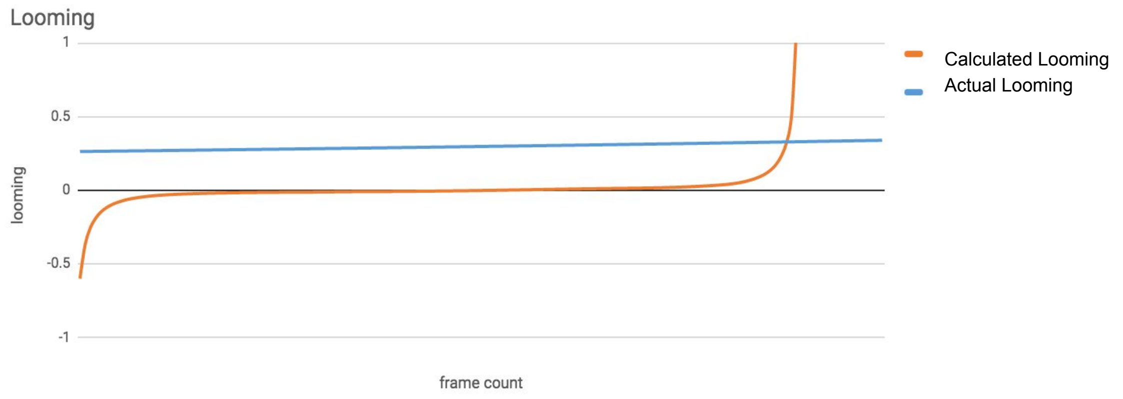
The calculated looming with this method is getting exponentially higher when a camera is getting closer to the obstacle in constant speed.

The actual looming is getting higher linearly.

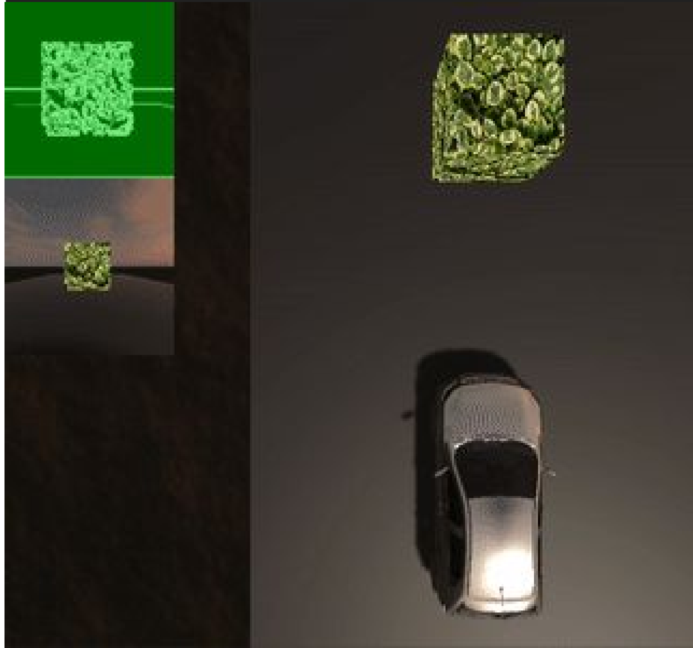
Comparison



Texture Densities
5000
4000
3000
2000
1000
0
texture density
frame count



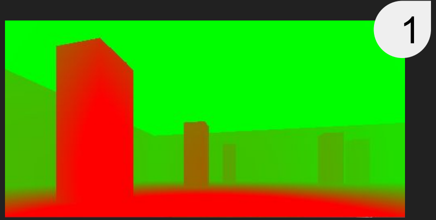
Looming
1
0.5
0
-0.5
-1
looming
frame count
Calculated Looming
Actual Looming

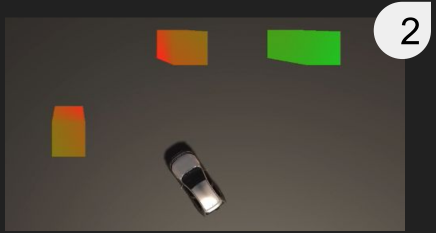


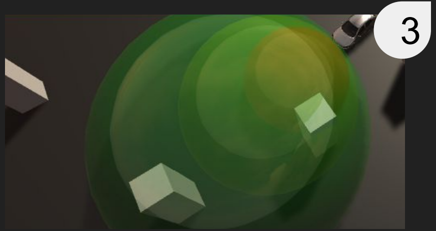
Looming Visualization

We implemented the visualization in 3 ways:

1. Image effect shader - the image from the camera is colored.
2. Material shader - the obstacles are colored.
3. Spheres that grow when the car's speed gets higher and shrink when the speed gets smaller.





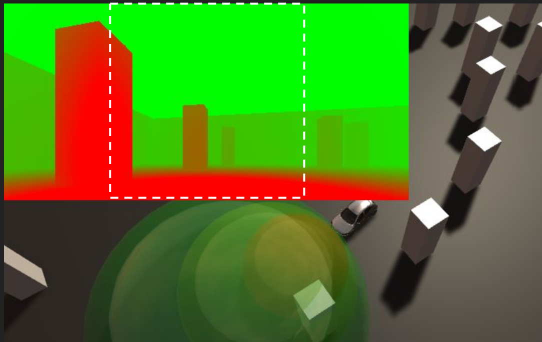


Navigation Decisions

In each frame, after the Image effect shader was applied, we had to decide whether the car should turn left or right, or keep going straight.

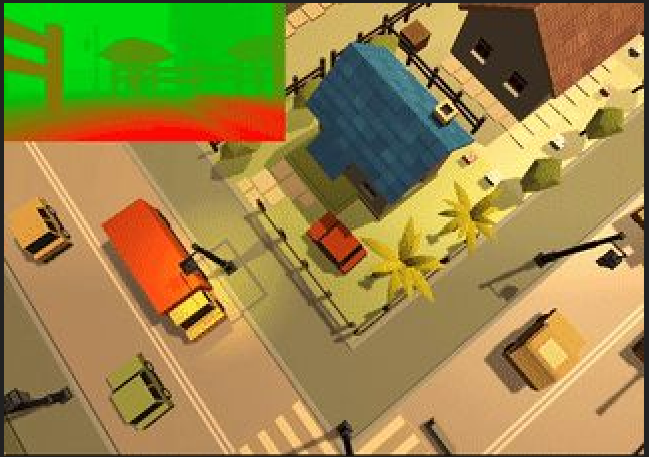
We calculated the percentage of red in the middle rectangle - if it passes a threshold we also calculate it in right half and left half of the image and make a decision: If the red percentage is higher on the right, the car will turn left, And vice versa.

If it didn’t pass the threshold - the car will go straight.



Navigation in the City

This is a result of our navigation decision algorithm.



Virtual Reality

We implemented some VR features with Oculus and touch controllers including:

1. The ability to fly.
2. The option to go back to your car if you lost it.
3. Settings panel for changing speed, and view looming visualizations.

Conclusion

A simulation in VR was implemented using the ranges approach for calculating looming and driving the car in a city.

Also the texture density algorithm was implemented and compared to the ranges algorithm.

Work Cited

- The Visual Looming Navigation Cue : A Unified Approach
  Daniel Raviv and Kunal Joarder
- A Novel Method to Calculate Looming Cue for Threat of Collision
  Daniel Raviv and Kunal Joarder
- Alglib - Numerical analysis and data processing library.
  http://www.alglib.net/
- Unity3D - Game engine.
  https://unity3d.com/
- Hover-UI-Kit - User interfaces for immersive VR/AR experiences.
  https://github.com/aestheticinteractive/Hover-UI-Kit
- Oculus - VR.
  https://developer.oculus.com/

Future Work

- The paper “The Visual Looming Navigation Cue : A Unified Approach” mentions more ways to calculate looming. The more methods implemented and combined, the better the results will be.
- Use deep learning.