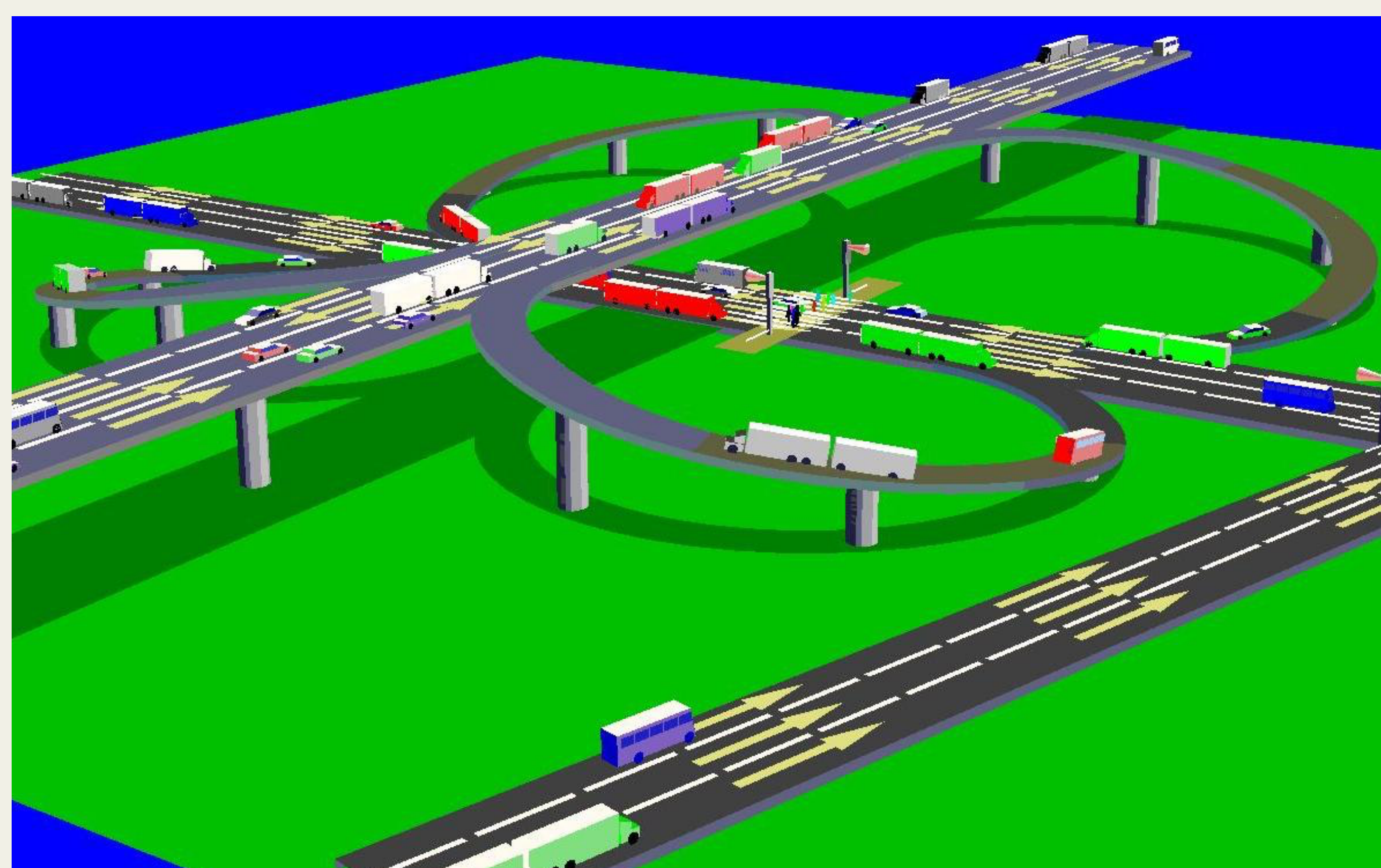
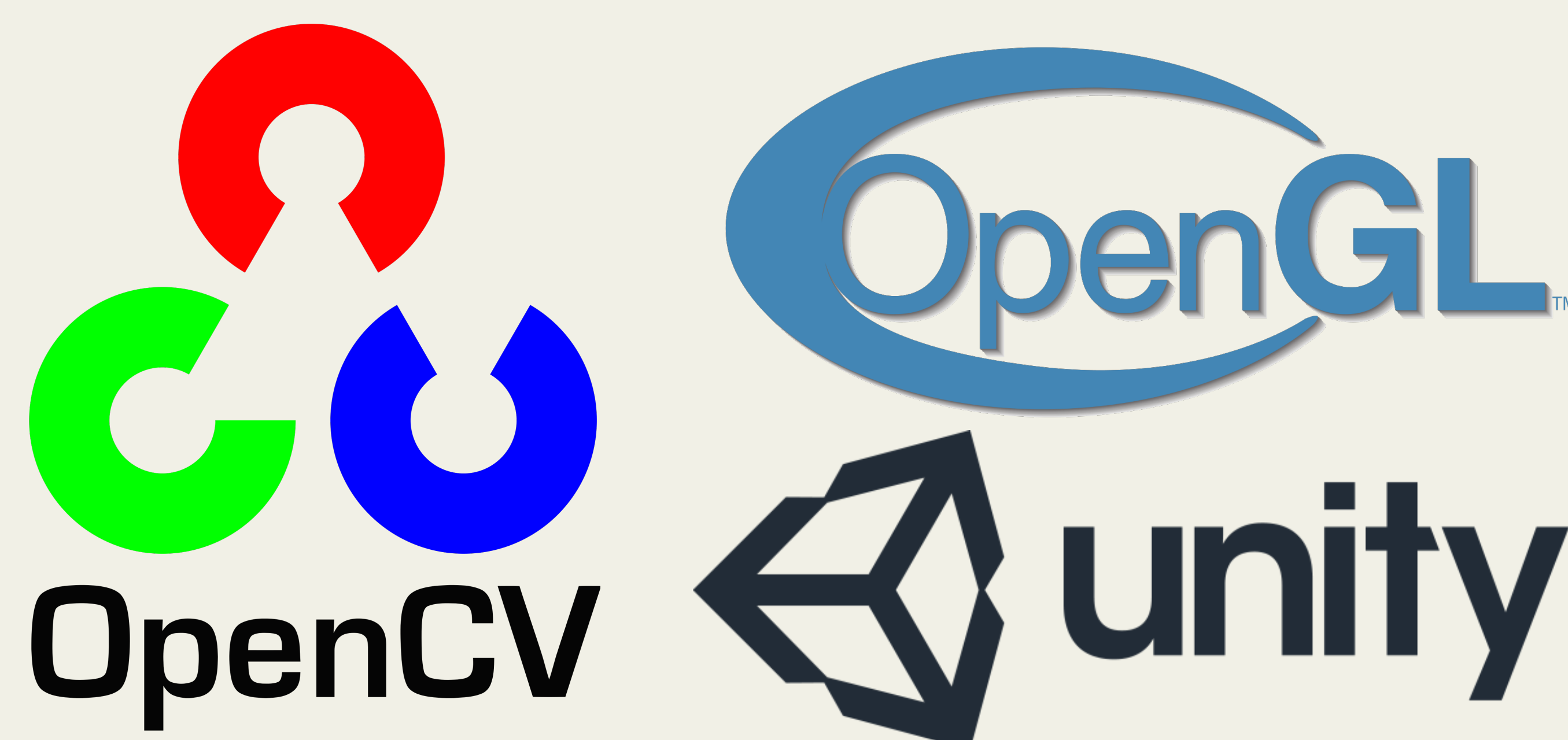


## Abstract

The purpose of this project is to integrate real-time traffic surveillance footage into a virtual reality setting. This can not only help with better projecting where any given car could go, but can also potentially improve driving simulations by giving their 3D environments some real-world traffic, making them more realistic. We are hoping to achieve this goal using primarily OpenCV/OpenGL and Unity.



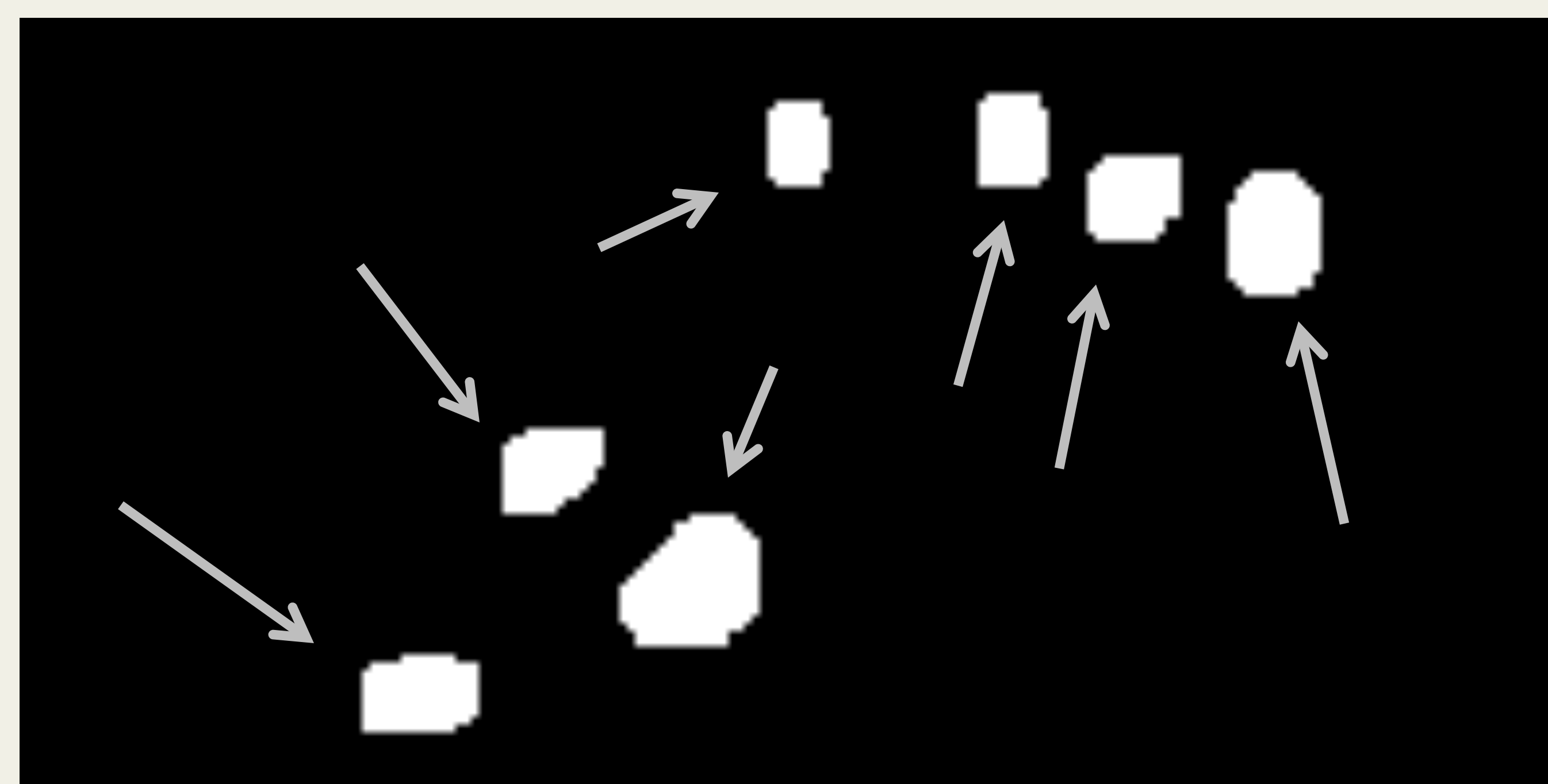
## The Plan

1. Write code that detects vehicles, more importantly, details the exact coordinates where they are on the traffic footage.
2. Write code that generates 3D cubes representing the vehicles and locking on to their coordinates.
3. Write code that generates a 3D plane matching the orientation of the road on the traffic footage.
4. Create the full 3D environment using the objects.

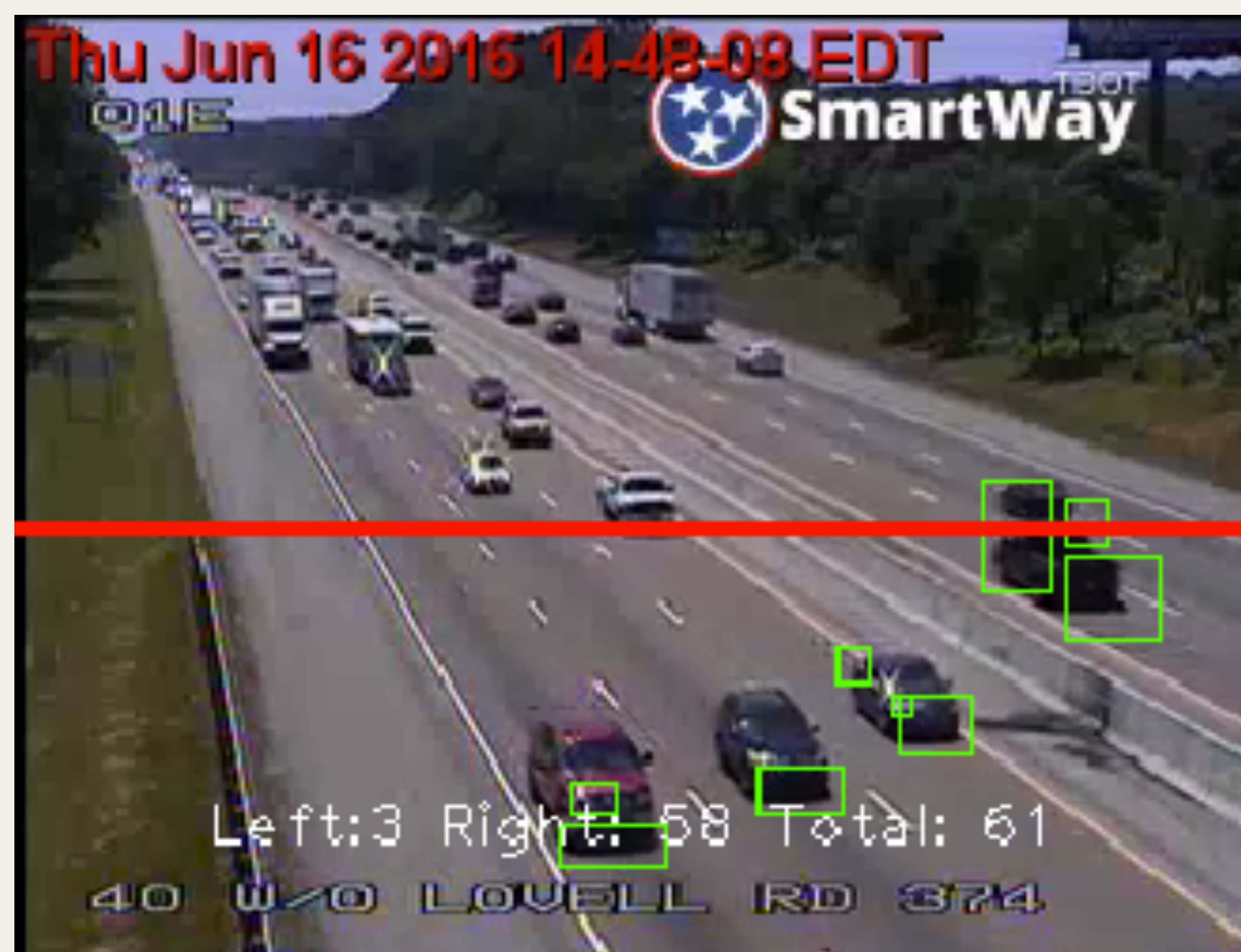
## Current Results

As of yet, step 1 is done. The method of analysis we ended up writing for this project involved OpenCV and OpenGL.

By blurring two frames within a very short period of time, we can find the changes in light between the both instantaneous pictures, allowing us to detect the moving vehicles. The result is usually faint, grey smudges on a black screen, so using dilation and threshold filters brings out the differences between the frames.



We can then conclude that the resulting white shapes are the cars, and then we draw green rectangles over them. Keeping track of the coordinates of the corners of the rectangles allows us to count the cars that have passed either side of the interstate.



## What Now?

Now that we have our footage analyzed, we can start integrating 3D into the project. We need to generate shapes representing the cars and trucks.

We have thought of two ways of doing this:

- Generate shapes using OpenGL in the video analysis's OpenCV code. This would put the cubes directly on the traffic footage.
- Generate shapes in Unity. This would involve feeding the coordinates of the detected cars from the OpenCV code into Unity.

Either way, we would have to make sure that the cubes follow the coordinates generated by the traffic analysis code, and remain the same size (because the size of the green squares fluctuate wildly in our analyzed footage).

Our next goal is to translate the positions of the cubes into a 3D setting, because our traffic footage is in 2D and therefore only showing the traffic from a single perspective. We are thinking about generating a plane/grid representing the road, and writing an algorithm to tilt and rotate the "road" until it matches the lanes in our footage. We can then use the coordinates of our cubes with their relative positions on the plane/grid to be able to generate a 3D environment where one can view the traffic from any perspective.

## Acknowledgements

This project is supported by the National Science Foundation.

Thanks to Dr. K. Wong for letting me onboard on this project and thanks to Dr. Cheng Liu for giving me three books on Unity and OpenCV.