Actual Traffic in a Virtual Reality

Michael Han, Dr. Cheng Liu, Dr. Kwai Wong

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.

1 Background

Virtual reality consists of computer-generated 3D environment that can be explored by a headset, which itself has two screens viewing said environment similar to how the human eyes view their surroundings. This is not only a breakthrough in programming but it is also an up-and-coming artistic medium capable of conceiving new ways of playing games and telling stories. It can also potentially help others immerse themselves in the real world. One such possibility could be the use of traffic surveillance footage to generate a more practical driving simulation.

2 Concept and Implementation

Traffic footage cannot be translated directly to a 3D context, so the process of making the transition must be split into several steps. The first order of business is writing code to detect the vehicles in any video files we receive. We chose OpenCV as our programming libraries for said analysis.

Any computerized image analysis takes advantage of the fact that images and, by extension, videos, are stored as a trio of numerical matrices, representing the amounts of light in red, green and blue. Furthermore, computerized video analysis takes advantage of the fact that videos are made of countless images that are referred to as frames. Knowing these facts, in order to detect our vehicles we wrote code that stores two frames at once – the frame that our program is currently reading and the one right beforehand. Then we used a technique called image differing, which subtracts the numbers of one frame's matrices from the numbers of another frame's matrices, leaving static visual elements as black and other dynamic visual elements as non-black colors. We can conclude that the latter of these visual elements are the vehicles we want to detect. Then, using a few functions allowed by the OpenCV libraries, we can visually highlight the vehicles by drawing rectangles around the shapes.

3 Future Plans

Our work is far from done. The next goal is to create a virtual road for the detected vehicles and the coordinates of their generated rectangles, which will ease the transition from 2D to 3D. We were thinking of collecting each individual point of a coordinate and putting them all on a grid. The result should be a linear mass of dots, which we could then conduct and interpret a correlation on, generating several lines that can represent the interstate in the footage.

Then when we have our virtual lanes, we plan on translating the entire scene into a 3D setting, because our traffic footage is in 2D and therefore only showing the traffic from a single perspective. We plan on generating a plane 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.