Vehicle Back Camera Simulation

Initially it is important to convert the real world length to the pixels, to replicate the actual simulation of car in reverse gear.

To implement the model here considered the Toyota Prius with length with width about 5 feet and length about 14 feet.

Step 1: Conversion of distance to pixel

Purpose: To indicate the real world length on camera to that it will be accurate reverse gear projection

Approach: Initially the factors need to me considered are 1. Rear camera capturing frame size 2. Actual real world distance covered by camera lenses

In Implemented example conversion of camera frame is done to 1024 * 512 irrespective of original source video frame size considering in general screens sizes in car for best visual quality.
Depending upon the camera lenses quality picture quality at screen will be varying.

Then assumption is considered that camera covers 12 feet width and 50 feet depth of real world scenario.

width_conversion_factor=width/coverage_width_feet

depth_conversion_factor=height/coverage_depth_fee

by using above formulas conversion is done from real world distances to pixels.

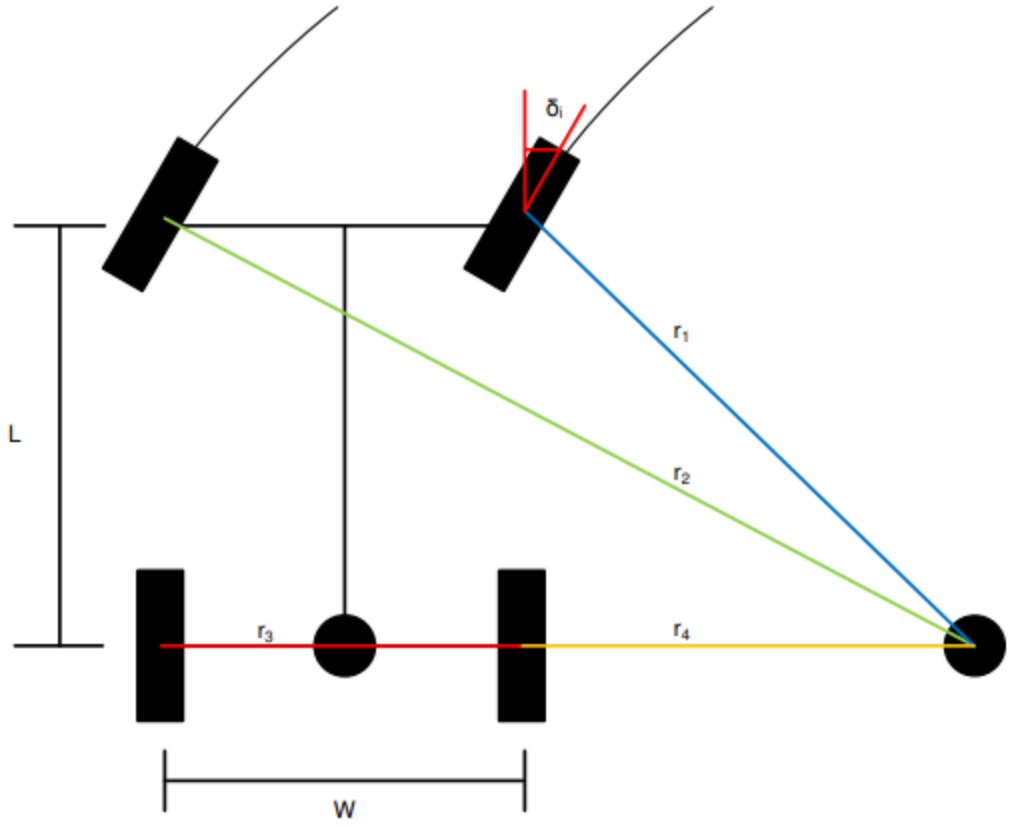
Step 2: Projections on each frame of video (Static)

In this one real world length of vehicles are projected. For vehicle length and width are considered and there lengths are find in the pixel by using conversation factors derived in the step 1 such that multiplying length with depth conversion factor and multiplying width by width conversion factor.

Now considering that camera is fitted at the rear center left and right tires projection are draw as a line and factor of 100 is used to in program to give little bit inclination so that it will appear to be giving as perception of depth while keeping middle line straight.

Now about the full turn projections:

Here considered the full turning radius of vehicle which is in general of full radius covered by front wheels but while projecting rear wheels its needed to be consider the radius of the rear wheels. For finding the rear wheels radius can refer the below image using the turning radius:



By using simple Pythagoras theorem, radius for rear wheel can be find and for extreme left and right both will have same radius just centers will be on opposite site. And to make the representation with perception certain bias is added to the center and arc of radius is projected.

Projections on each frame of video (Dynamic)

Dynamic projection can be done on rear camera view by using the streeting angle. As angle changes r1 will change dynamically and can be implemented by using dynamic input value of streeting angle. depending upon the steering angle i.e. delta in above figure we can find the r1 by using the formula:

$$r_1 = \frac{L}{\tan(\delta_i)}$$

And depending on that we can find the respective r3 and r4. Depending upon the angle we can decide right or left turn and depending on that r3 and r4 will be calculated and will we draw on each frame at that instance.

Implementation:

1. **Measuring Units:**
   In the program measuring units are considered in feet. Which can be easily changes by putting all inputs such as coverage_width_feet, coverage_depth_feet, vehicle_width, vehicle_length in same unit.
2. **Input Video Source:**
   For testing purposes, the input video should be placed in the same directory as the notebook or script. This setup simplifies the process of loading the video into the simulation.
   In real-world scenarios, the camera input would typically be streamed directly into the system. You can adapt the code to read from a camera by setting cv2.VideoCapture with the appropriate camera ID or network stream URL.
3. **Exiting the Simulation:**
   To exit the video simulation during testing, press the "q" key on your keyboard. This command triggers a break in the main loop, effectively stopping the video feed and closing the OpenCV window.