Time to Contact for Autonomous Vehicles: Analysis and Estimation from Image Sequences

Andrew Grant
Advisor: Professor Alain Kornhauser

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Engineering
Department of Operations Research and Financial Engineering
Princeton University

May 2016
I hereby declare that I am the sole author of this thesis.

I authorize Princeton University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Andrew Grant

I further authorize Princeton University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Andrew Grant
Abstract

In this thesis, I estimate time to contact values in driving scenarios using image sequences from a camera mounted on a vehicle. I focus on the time to contact between our vehicle and a lead vehicle, when both vehicles are in a straight lane. Accurate estimates of time to contact can be used for a rear-end crash avoidance system for vehicles, and ultimately for fully autonomous vehicles.
Acknowledgements

I would like to thank Professor Kornhauser for being a great advisor. His enthusiasm for and dedication to the development of autonomous cars motivated my interest in this technology. His feedback and encouragement contributed greatly to the positive experience of writing this thesis.
To my family - Annette, Gary, and Brian.
Contents

Abstract ................................................................. iii
Acknowledgements ................................................... iv
List of Tables .......................................................... vii
List of Figures .......................................................... viii

1 Introduction ......................................................... 1

2 Time to Contact for Autonomous Vehicles ................. 3
   2.1 Stationary Lead Vehicle ...................................... 4
   2.2 Moving Lead Vehicle with Constant Velocity ............. 6
   2.3 Time to Contact from Images ................................. 7
      2.3.1 Related Work ........................................... 7
      2.3.2 Distance to Lead Vehicle ............................... 9

3 Time to Contact Estimation ...................................... 13

4 Analysis of Time to Contact Estimation ...................... 18
   4.1 Moving Lead Vehicle ....................................... 18
   4.2 Stationary Lead Vehicle .................................... 27

5 Discussion and Conclusion ...................................... 35
List of Tables

2.1 Comparison of Computed and Ground Truth Distances . . . . . . . . 12
List of Figures

2.1 The Time to Contact is plotted for different following velocities and distances. Acceptable scenarios are colored yellow, while unacceptable scenarios are blue. ......................................................... 5

2.2 Our max acceptable velocity as a function of initial distance and velocity of lead vehicle. ................................................................. 7

2.3 Schematic of distance to lead vehicle which shows the useful properties of the pin-hole camera model. ................................................. 9

2.4 The red pluses indicate the selected points input to the distance calculation. ............................................................... 11

4.1 Selected frames from the sequence. .................................................. 19

4.2 Road classification and vehicle detection. ............................................. 21

4.3 Matched feature points between frames ........................................... 22

4.4 The median difference in the distance between features in each frame. 23

4.5 Distance to the lead vehicle. ............................................................. 24

4.6 Relative velocity between lead vehicle and our vehicle for each frame. 25

4.7 Estimated Time to Contact for each frame of the sequence. ............. 26

4.8 Minimum Time for Appropriate Deceleration with Estimated Time to Contact for each frame of the sequence. ................................. 27

4.9 Selected frames from the sequence. The tracked vehicle is at the top of the right parking lane. ..................................................... 28
4.10 Selected frames from the sequence with the detected vehicle surrounded by a blue bounding box.

4.11 Distance to lead vehicle.

4.12 Velocity of our vehicle.

4.13 Time to contact and the minimum time needed to avoid contact, using camera data only.

4.14 Time to contact and the minimum time needed to avoid contact, using the IMU velocity data.

4.15 Time to contact and the minimum time needed to avoid contact. Using camera data - solid lines. Using IMU velocity data - dashed lines.
Chapter 1

Introduction

A ‘time to contact’ value may be necessary for an algorithm that controls the headway movement of a vehicle. Time to contact is the amount of time it will take for a vehicle to make contact with the rear end of a vehicle in front of it.

In this thesis, I first analyze the time to contact value in the framework of safe driving situations. I then discuss an algorithm that I developed to estimate the time to contact value from image sequences taken by a single camera in a real driving scenario. I highlight the main computer vision techniques and learning methods employed by the algorithm. This algorithm estimates the time to contact value, along with two other metrics. The second metric is the distance to the lead vehicle. The third is the relative velocity between the lead vehicle and our vehicle. The time to contact value is derived from these latter metrics.

I then analyze and discuss these estimates for two sequences of a vehicle driving behind a lead vehicle. In the first sequence, the lead vehicle changes velocities. In the second, the lead vehicle is stationary. These two scenarios are useful for testing our algorithm. In the first scenario, we can only compare the distance metric with the available recordings from the LIDAR sensors. In the second sequence, since the lead vehicle is stationary, the relative velocity is the same as the velocity of our vehicle.
We can therefore compare our estimate of the relative velocity with the velocity of our vehicle recorded by the IMU sensors. The second sequence also involves a lead vehicle in the lane directly next to our vehicle. By assuming that the vehicle is in front of our vehicle, we acquire data that is similar to that of a sequence where contact is made between the vehicles. We can therefore determine at which frame the unsafe situation is entered.

The metrics from both sequences show good performance of the algorithm. In the second sequence, the algorithm detects the occurrence of future contact at an earlier frame than that of the time to contact using metrics from the IMU sensors. This means that by using data from the camera we are able to determine the “point of no return” a few frames before we can from the IMU data. From a safety perspective, this is desirable.

From the accurate distance estimations in both sequences, and the safe time to contact estimations from the second sequence, I conclude that it is possible to use images to estimate time to contact values that are safe for the forward movement of an autonomous vehicle.
Chapter 2

Time to Contact for Autonomous Vehicles

Time to contact:

The time until two objects are in the same position given their current positions, velocities, and accelerations.

\[ x_1 + v_1TTC + \frac{1}{2}a_1TTC^2 = x_2 + v_2TTC + \frac{1}{2}a_2TTC^2 \] (2.1)

The time to contact at any given time enables an autonomous vehicle to determine a course of action in the present and future that will prevent contact with objects ahead of the vehicle. In this section, I illustrate how time to contact can be used to analyze driving scenarios.

First, it is necessary to establish a maximum deceleration rate for an autonomous vehicle. This rate is determined by the desired level of comfort of the trip - any deceleration rate above this maximum would result in an uncomfortable experience for the occupants. I will use 4.6 m/s². Or 10.3 mph/s.

A driving scenario is safe if our vehicle is able to decelerate to the velocity of the lead vehicle before contacting the rear of the vehicle. The deceleration time is the
time it takes our vehicle to decelerate to the velocity of the lead vehicle. The minimum deceleration time is the deceleration time when decelerating at the maximum deceleration rate.

\[
\text{min time decel} = \frac{v_1 - v_2}{\text{max decel}} \quad (2.2)
\]

If the time to contact is less than the minimum deceleration time, then the autonomous vehicle will not be able to avoid contacting the object ahead of it. On the other hand, if the time to contact is greater than the minimum deceleration time, an autonomous vehicle can avoid contacting the object ahead of it by slowing down. A driving scenario is therefore unsafe when the time to contact is less than the minimum deceleration time. And it is safe when the time to contact is greater than the minimum deceleration time.

I focus on two main driving scenarios that occur for an autonomous vehicle operating on current road infrastructure: that when there is a stationary vehicle ahead, and that when there is a moving vehicle ahead. In 2.1 and 2.2, I analyze these two driving scenarios with respect to the time to contact to the lead vehicle - the closest vehicle in front of our vehicle. In 2.3, I discuss existing methods of calculating time to contact along with the motivation for the algorithm presented in Chapter 3.

I will use TTC to represent the time to contact value.

\section{2.1 Stationary Lead Vehicle}

In the case that the lead vehicle is stationary, and our vehicle is non-accelerating, the TTC equation becomes

\[
x_1 + v_1TTC = x_2 \quad (2.3)
\]
Solving for TTC, we get

\[ TTC = \frac{x_2 - x_1}{v_1} \]  

or

\[ TTC = \frac{d}{v_1} \]

where \( d \) is the initial distance between our vehicle and the lead vehicle, and \( v_1 \) is the velocity of our vehicle.

We plot the TTC for \( d = [10\text{m},50\text{m}] \) and \( v_1 = [1\text{m/s},30\text{m/s}] \) below.

Figure 2.1: The Time to Contact is plotted for different following velocities and distances. Acceptable scenarios are colored yellow, while unacceptable scenarios are blue.
From the border of the two colors, we can infer our vehicle’s maximum acceptable velocity for a given initial distance. For example, with a 45 meter headway, our vehicle should not have a velocity greater than 15 m/s or approximately 33.5 mph.

We can also observe from the surface that the smaller the headway, the smaller the effect a change in relatively high velocities has on the TTC. For example, with a 10 meter headway, a reduction of the initial velocity from 30 m/s to 20 m/s results in a minuscule quarter second increase in TTC. However a reduction of similar size from 10 m/s to 1 m/s results in a nine second increase in TTC.

2.2 Moving Lead Vehicle with Constant Velocity

In the case of a moving lead vehicle with constant velocity, the TTC equation is:

\[ x_1 + v_1TTC = x_2 + v_2TTC \]  

(2.6)

Solving for TTC, we get

\[ TTC = \frac{x_2 - x_1}{v_1 - v_2} \]  

(2.7)

or

\[ TTC = \frac{d}{v_1 - v_2} \]  

(2.8)

In Figure 2.2 we plot the maximum acceptable velocity of our vehicle as a function of the initial distance and the velocity of the lead vehicle. Assuming the maximum deceleration rate established earlier and the inability of the vehicle to change lanes, this maximum acceptable velocity is the upper bound for our vehicle’s speed given our distance to the vehicle ahead of us and that vehicle’s velocity. If our vehicle travels above this velocity, then an accident will occur.
2.3 Time to Contact from Images

2.3.1 Related Work

In this thesis I focus on estimating time to contact from image sequences. LIDAR sensors could be used instead but the current price of these sensors is too expensive.

I initially investigated two main ways of calculating time to contact directly from images. The first is that used by Horn et al. (2007), where they calculate time to contact by measuring the change in size of the object. The problem with this method is that it requires a good segmentation of the object from the rest of the image. In the case of vehicles in an outdoor environment, it is extremely difficult to develop a heuristic that can be used to segment the rear-end of the vehicle reliably. This is because of the variability of the appearances of different types of vehicles. The
open, outdoor environment of driving scenarios also introduces noise through light reflection and shadows which make it difficult to acquire accurate segmentations of the vehicle. The inaccuracies of consecutive segmentations are amplified in the time to contact estimations.

Sagrebin et al. (2008) calculate the focus of expansion of an image, and then calculate the time to contact of an object from its movement with respect to the focus of expansion. However the calculation of the focus of expansion requires the tracking of multiple features in the image in order to determine the camera’s main direction of movement in the scene. This can be very difficult in the presence of multiple moving objects in the scene, as is common in driving scenarios. They also only report time to contact estimations for experiments conducted in a controlled, indoor environment.

In this thesis, I estimate the time to contact by estimating the distance to the lead vehicle and then I take the change in distance between consecutive frames in order to calculate the relative velocity between our vehicle and the lead vehicle. With the distance from our vehicle to the lead vehicle and their relative velocity at a given point in time, we can then calculate the time to contact using the equation:

\[
TTC = \frac{d}{v_1 - v_2}
\]  

(2.9)

where \(d\) is the distance to the lead vehicle, and \(v_1 - v_2\) is the relative velocity.

The main information we need from an image is therefore the distance. From this, everything else can be derived (given we have an image sequence). The distance can be observed from other sensors, such as LIDAR, but as mentioned earlier these sensors are too expensive for practical purposes.

I calculate the distance using an algorithm described in the next chapter. In order to make the distance estimations robust, I track features that belong to the rear-view of the lead vehicle. The difference in the distances among the features between
frames can be applied to the distance to the lead vehicle to calculate the distance to
the lead vehicle in the new frame. Pundlik et al. (2011) also tracks features in order
to calculate TTC. However, their experiments do not involve a moving camera and
are only performed in a controlled, indoor environment.

In the next section, I demonstrate how the distance to an object, specifically the
lead vehicle in a driving scenario, can be calculated from an image. I also provide
some preliminary results of distance estimations that show the feasibility of such an
approach for estimating TTC.

### 2.3.2 Distance to Lead Vehicle

The time to contact problem can be simplified for autonomous vehicles, since TTC
only needs to be calculated for vehicles ahead of our vehicle in our lane.

Using the pin-hole model of a camera (the model for most cameras), we are able to
calculate the distance using the camera’s focal length, the pixel height of the optical
center to the bottom of the car, and an estimate of the actual height of the car.

![Figure 2.3: Schematic of distance to lead vehicle which shows the useful properties of the pin-hole camera model.](image)

As demonstrated in the figure above, the relevant triangles are symmetric, and
the distance to the vehicle can be calculated by:

\[
\text{Distance to Vehicle} = \frac{f}{h}H \tag{2.10}
\]

\(f\) is the focal length of the camera, measured in pixels. It is obtained from the camera calibration matrix. \(h\) is the height of the optical center with respect to the bottom of the vehicle, measured in pixels. The bottom of the vehicle, in this case, is the bottom edge of the bumper. \(H\) is the actual height of the bottom of the vehicle to the 3D point in space which corresponds with the optical center. This is measured in meters.

**Estimate of \(H\)**

In our current method, our value of \(H\) is an estimate. In the image sequence used in this section, our vehicle follows a lead vehicle with a rear-end height of about 2.5 feet. This gives the height of the bottom of the bumper to the top of the trunk. \(H\) is then calculated using the respective ratio obtained from the image. This estimate is shown to be adequate for calculating the distance to the vehicle.

**Tests with Ground Truth Distances**

I performed a preliminary analysis of this method of calculating distance. The video sequence chosen contains a lead vehicle initially at a stop light which then accelerates away from the main vehicle, which in turn accelerates as well. The main vehicle is equipped with stereo cameras as well as LIDAR sensors. The ground truth distance to the lead vehicle is acquired from the LIDAR.

I manually pick the points in the image which correspond to the top of the trunk and the bottom of the vehicle, shown in Figure 2.4.
Figure 2.4: The red pluses indicate the selected points input to the distance calculation.
Table 2.1 compares the distance estimated by using the camera geometry and pixel height of the vehicle, with the distances recorded by the LIDAR, for the images shown above.

Table 2.1: Comparison of Computed and Ground Truth Distances

<table>
<thead>
<tr>
<th>Dist. with $H = 2.5$ feet</th>
<th>G.T (lidar)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.80</td>
<td>5.74</td>
</tr>
<tr>
<td>6.73</td>
<td>6.93</td>
</tr>
<tr>
<td>11.83</td>
<td>11.85</td>
</tr>
<tr>
<td>14.66</td>
<td>14.85</td>
</tr>
</tbody>
</table>

As seen from the above results, this method of calculating the distance to the lead vehicle is promising. We therefore would like to find a method which can identify the top and bottom of the lead vehicle in the image automatically. An automatic segmentation of objects on the road in front of our vehicle is a first step towards accomplishing this task.

In the next chapter I present an algorithm which detects and localizes the lead vehicle in the image, and then estimates distances which are in turn used to estimate time to contact values for the sequence.
Chapter 3

Time to Contact Estimation

In this chapter, I describe the algorithm used to estimate the time to contact to lead vehicles. First, I provide an overview of the algorithm and then provide details for important parts.

Algorithm Overview

The algorithm assumes we have located the lane. Hence we therefore know the slope (in pixels) of the lane ahead of us. Given the slope of the left and right boundaries of the lane, the area of the image bounded by the lane is our region of interest. Using a trained classifier of the road texture, the algorithm classifies a small area in the middle of the lane, starting from closest to farthest. If the patch is determined as a non-road patch, then there is a possibility that a vehicle is there. We then use a vehicle classifier to localize a vehicle in this region. Once a vehicle is detected, we then extract feature points from the bounding box of the vehicle. We then track these points in the subsequent frame of the sequence using Lucas-Kanade tracking. With the same features located in subsequent frames, we can compare how the features’ locations have changed in relation to one another. We can determine if the lead vehicle has moved closer or farther away from us, and we can quantify this change.
We then multiply the distance to the lead vehicle in the old frame by this ratio to get the distance to the lead vehicle in the new frame. The time to contact is then calculated.

**Support Vector Machines**

There are two main classification tasks in my algorithm - road and vehicle classification. There are many machine learning techniques that can perform this task of image classification and detection. Currently neural networks perform the best, especially on multi-class detection when the number of possible objects to detect is large. However, since this is a simpler binary classification task, I decided that Support Vector Machines would be sufficient.

**Road Features**

In order to classify the road, I decided that a texture descriptor would provide a good feature space for an image patch. This is because the texture of a given road lane is mostly uniform. I initially used Gabor filters, which provided good classification results, but had too long of a run time. I therefore decided to use the simpler Haralick features. While these were not as good descriptors as Gabor filters, their performance was sufficient and had a much faster run time. For a given image patch, a feature descriptor was created using the 3 values of the color channels (RGB), and 5 Haralick features.

**Vehicle Features**

I used the Histogram of Oriented Gradients (HOG) as the feature descriptor for vehicle detection. The HOG features encapsulate the general shape of a vehicle.
Localizing Vehicles

The algorithm classifies the 10x10 pixel area in the middle of the lane, starting from the bottom of the image. If the Road SVM classifies an image patch as not a road, then that area of the image is scanned for any vehicles. This is done by classifying a rectangular area of the image whose bottom is at the non-road patch. The width and height of this rectangular box is proportional to the width of the lane at that point. If a vehicle is detected, then the size and location of the rectangular box is dynamically adjusted in order to get the best bounding box of the vehicle.

Distance calculation from Bounding Box

Once a bounding box has been determined for the vehicle, the distance can then be calculated using the procedure outlined in the previous chapter. The top and bottom of the bounding box are used to calculate the height of the car in pixels.

After calculating the distance from bounding boxes for several driving sequences, I found that this procedure produced generally accurate estimates of the true distance. However, the distance can change too greatly or have ‘jumps’ between some frames. This is because the bounding boxes returned for each frame may be slightly different from each other. For example, the height of some bounding boxes in some frames were too large due to the presence of a strong shadow below the car, or because of a cluttered environment directly above the car in the background. In these cases, the small increase in height of the bounding box drastically affected the distance calculation, returning a distance that was too small. I attempted to refine the bounding box even more by using the Hough transform to detect lines that represent the top and bottom of the vehicle. However this was subject to the same issues, and did not rectify the problem. The problem was solved, instead, by tracking features within the bounding box, as described next.
Feature Tracking

Tracking features of the lead vehicle between frames can be used to estimate the movement between our vehicle and the lead vehicle. Once the vehicle has been localized, we can extract features from the area of the image within the bounding box, and this ensures that we are tracking features on the vehicle. The difference in the feature locations in each frame provides useful information about the movement that occurred between frames.

For each frame I calculated the distances from each feature point on the vehicle to every other feature point on the vehicle. I then used the difference of the respective distances between frames to calculate the movement between the vehicles. A decrease in the distances signifies that the lead vehicle has moved farther away from our vehicle, and therefore the relative velocity between the lead vehicle and our vehicle was positive during the time elapsed between the frames. If the distances increased, then the relative velocity between the lead vehicle and our vehicle was negative during this time. The magnitude of the change in feature distances is inversely proportional to the magnitude of the relative velocity.

In this implementation of the algorithm, I used Speeded Up Robust Features (SURF) (Bay et al., 2006) to detect feature points that could be easily tracked. I then tracked these points between frames using the Lucas-Kanade algorithm (Lucas et al., 1981).

Time to Contact Calculation

Given the height of an initial bounding box, the height of the vehicle in subsequent frames can be calculated using the ratio obtained by tracking feature points. The distances to the lead vehicle can therefore be calculated from these heights. These estimations will be more robust since distances will not be calculated directly from each bounding box. This reduces the noise created by some bounding boxes that
are not as accurate or ‘tight’ as others. The resulting plot of distance is therefore smoother.

Given the distances to the lead vehicle, we can use the difference between distances in subsequent frames to calculate the relative velocity between the lead vehicle and our vehicle during the time elapsed between the two frames.

The time to contact at the time a frame was taken, can be calculated using the distance and the relative velocity:

\[
TTC_t = \frac{\text{Distance to lead vehicle}_t}{\text{Relative velocity}_{t,t-1}}
\]  

(3.1)
Chapter 4

Analysis of Time to Contact Estimation

In this section, I analyze the results of the various steps of the algorithm described in the previous section. I used the KITTI dataset which has high resolution video sequences obtained from a camera mounted on top of a car (Geiger et al., 2013). The videos are on average one minute long. The vehicle was also equipped with LIDAR sensors and GPS and IMU trackers. These datasets were calibrated, making it simple to compare calculations from the camera with the readings from these sensors.

I analyze the results of the algorithm for two different sequences - one that involves a moving lead vehicle, and the other, a stationary lead vehicle.

4.1 Moving Lead Vehicle

Figure 4.1 shows frames of an interesting sequence that involves a moving lead vehicle. The lead vehicle is initially at a stand still. Our vehicle decelerates as it approaches the lead vehicle. Our vehicle then comes to a stop behind the lead vehicle briefly, until the lead vehicle accelerates, followed by our vehicle.
Figure 4.1: Selected frames from the sequence.
In Figure 4.2 are selected frames that display the classification of potential road patches in the middle of the lane, along with the final detection of the vehicle.

Patches that are classified as a road surface are bounded by a green box. The patches along the middle line (green line) that are not bounded by a green box were not classified as a road surface. We can see that the classifier misclassified a few true road patches as non-road patches - especially in shadowed areas. However in this scenario, the false negative rate is not as important as the false positive rate. We need the classifier to correctly label patches that are on the vehicle as non-road patches - which it does. The road classifier could be improved by providing more training examples, or by dynamically training the classifier in an online setting. However, since the only cost of a false negative is the time it takes to scan that area for a vehicle, I decided that this road classifier was sufficient for the current illustration of detecting a vehicle.

The final bounding box of the detected vehicle is outlined in blue.
Figure 4.2: Road classification and vehicle detection.

Figure 4.3 shows the matched SURF points between two consecutive frames.
Figure 4.4 plots the change in distance between the features for consecutive frames. The change in distance is represented as the following ratio:

$$\text{dist diff} = \frac{\text{dist features}_{t2} - \text{dist features}_{t1}}{\text{dist features}_{t1}}$$ (4.1)

This ratio is calculated for the distance between every pair of features, and the median is taken.

As pointed out in the previous chapter, a positive ratio indicates that the intra feature distance has become larger, and therefore the vehicles are closer than in the previous frame, and vice versa if the ratio is negative. A positive ratio therefore indicates a decrease in the relative velocity, and a negative ratio indicates an increase in the relative velocity (where the relative velocity is: velocity of lead vehicle − velocity of our vehicle).

Therefore, we see that the plot accurately reflects the relative velocity of the two vehicles. At first the feature distance ratio is positive, indicating a negative relative velocity, which accurately reflects the approach of our vehicle to the stationary lead vehicle. At around frame 80, the ratio is close to zero for a few frames which reflects the brief moment when our vehicle is also (almost) stationary behind the
stationary lead vehicle. It is then followed by a sharp decrease in the ratio, which reflects the acceleration of the lead vehicle. It then levels off at a negative ratio for the remaining frames, as the lead vehicle decreases its acceleration and our vehicle accelerates, following the lead.

![Graph](image)

**Figure 4.4:** The median difference in the distance between features in each frame.

By multiplying the ratio of the feature distance difference with an initial height of the vehicle, we can acquire height estimates of the vehicle for each frame. The initial height of the vehicle used in my implementation of the algorithm is obtained from the height of the bounding box in the first frame. Figure 4.5 displays the distances to the lead vehicle which are calculated using these heights. As expected, we see that the distance decreases as our vehicle approaches the stationary lead vehicle. The distance remains the same at around 5 meters when both the lead and our vehicle
are stationary. And then the distance increases as the lead vehicle accelerates away from our vehicle.

Figure 4.5 also displays the distances to the lead vehicle recorded by the LIDAR sensor. We see that our estimates are very close to those of the LIDAR. In fact, in this experiment our estimates appear to be less noisy than those of the LIDAR.

![Figure 4.5: Distance to the lead vehicle.](image)

Now that we have smooth estimates of the distances to the lead vehicle for consecutive frames, we can take the difference of the distances between consecutive frames to get an estimate of the relative velocity between the lead vehicle and our vehicle during the time elapsed between the frames. Figure 4.6 shows the relative velocity with the zero line as a reference. If we assume the lead vehicle is absolutely stationary in the first half of the sequence, then our estimates show that our vehicle is traveling
at around 6 miles per hour initially, and this gradually decreases as we get closer to the lead vehicle. As the lead vehicle accelerates during the end of the sequence, we see that it travels at speeds up to 7 miles per hour faster than our vehicle.

![Figure 4.6: Relative velocity between lead vehicle and our vehicle for each frame.](image)

Given the distance to the lead vehicle and the current relative velocity, we can then calculate the time to contact. Figure 4.7 shows the estimated time to contact. All negative time to contact values were set to -1. We see that after about frame 90, the time to contact is negative, reflecting the positive relative velocity during this part of the sequence. We also observe that the estimated time to contact floats around 5 seconds as our vehicle approaches the lead. We then observe higher time to contact values when our vehicle is almost stationary behind the lead. There is greater variation during this time period which is probably due to small movements between the vehicles.
Using the max deceleration rate of $4.6 \text{ m/s}^2$, and the current relative velocity, the minimum time needed to decrease the relative velocity to zero was calculated for each frame. Figure 4.8 plots this minimum time along with the time to contact estimates. We see that the minimum time needed for appropriate deceleration at each frame is much less than the time to contact. This is expected as the vehicle did not come close to crashing during the sequence!
Figure 4.8: Minimum Time for Appropriate Deceleration with Estimated Time to Contact for each frame of the sequence.

### 4.2 Stationary Lead Vehicle

The analysis of the algorithm’s performance in the scenario that the lead vehicle is stationary was especially useful for my experiments on this data set. It provides us with a second “ground truth” metric: the velocity of our vehicle. Because the lead vehicle is stationary, the magnitude of the relative velocity calculated from the frames should be the same as that of our vehicle’s velocity. We can therefore compare the velocities calculated from our algorithm with those recorded by the Inertial Measurement Unit (IMU).

In the KITTI dataset, I was unable to find sequences where the lead vehicle in our lane was stationary for a long period of time. I therefore had to use a sequence where there was a parked vehicle in the lane directly next to our vehicle.

Figure 4.9 shows some frames of the sequence. Figure 4.10 shows the bounding
box of the detected vehicle.

Figure 4.9: Selected frames from the sequence. The tracked vehicle is at the top of the right parking lane.
Figure 4.10: Selected frames from the sequence with the detected vehicle surrounded by a blue bounding box.
Figure 4.11 shows the calculated distances along with the distances recorded by the LIDAR sensor.

![Graph showing distance to lead vehicle](image)

Figure 4.11: Distance to lead vehicle.
Figure 4.12 shows the velocity of our vehicle calculated with my algorithm along with the velocity recorded by the IMU sensor. We see that our estimates are close to those recorded by the IMU. Our estimates have more noise from frame to frame. However this noise is not detrimental, except for the estimate at frame 22 where the estimate drops drastically for that single frame. This could be viewed as an outlier in our estimates. One possible source of the noise could be the vibration or other minor movements of the vehicle that occur between frames, but which do not contribute to the vehicle’s main forward velocity. The camera data would be more sensitive to these movements.

![Figure 4.12: Velocity of our vehicle.](image)
Figure 4.13 shows the time to contact using the metrics estimated from our algorithm: distance to the lead vehicle and velocity of our vehicle. It also plots the minimum acceptable deceleration time needed to avoid contact. If we assume that the parked lead vehicle was actually directly in front of our vehicle, then the two vehicles would have made contact. We therefore see that the time to contact eventually falls below the minimum acceptable deceleration time. We can interpret the intersection of these two times as the moment that our vehicle enters an unsafe scenario.

Figure 4.13: Time to contact and the minimum time needed to avoid contact, using camera data only.
Figure 4.14 shows the time to contact and the minimum acceptable deceleration time needed to avoid contact using the velocity from the IMU sensors, and the distance to the lead vehicle calculated from our algorithm.

Figure 4.14: Time to contact and the minimum time needed to avoid contact, using the IMU velocity data.
Figure 4.15 superimposes the above plots.

Figure 4.15: Time to contact and the minimum time needed to avoid contact. Using camera data - solid lines. Using IMU velocity data - dashed lines.
Chapter 5

Discussion and Conclusion

From the second sequence, we see that using our time to contact estimates results in an earlier detection of our vehicle entering an unsafe state. In this particular case, our algorithm gives favorable results. With the results obtained, an autonomous vehicle in this scenario could easily avoid an unsafe situation with respect to the lead vehicle. A control algorithm responding to these metrics would decelerate our vehicle appropriately to avoid entering the unsafe state.

However, what is really good about these results is that the time to contact estimates are in a close range with the true time to contact. Improvement of the algorithm could yield results that are even closer to the truth. With close enough estimates, a heuristic could be used in a control algorithm to introduce bias in our estimates in favor of safety.
Bibliography


