Circumspectly Crash of Autonomous Vehicles

Yair Wiseman and Ilan Grinberg

1 Computer Science Department, Bar-Ilan University, Ramat-Gan 52900, Israel
            wiseman@cs.biu.ac.il

Abstract. Sometimes an autonomous vehicle realizes that unfortunately a crash will come about; however there are several possible crashes and the autonomous vehicle can still make a decision what the least destructive option is. In this paper a technique for a real-time assessment of crash potential damages is presented with the aim of facilitating the vehicle's embedded computer to come to the best decision.

Keywords: Autonomous Vehicle, Car Accident, Embedded Real-Time System.

1 Introduction

Moral dilemmas similar to the well-known "Trolley Problem"[1] can take place in circumstances where an autonomous vehicle should decide and select between several damaging actions in the course of an inescapable crash. Such morals questions like "Who should die the driver of the car or a pedestrian in the vicinity of the car?" have been debated by many philosophers, religions and law makers.

In this paper we do not intend to find the answers for these moral dilemmas. We would like to focus in the eminent subject of the passenger safety [2]; accordingly, we would strive for giving techniques for assessing the potential damages that possibly will happen in each course of action.

Fig. 1: Simulation Model of a geometry shape consists of basic polygons
One of the most widespread techniques for effectively employing computational geometry functions is creating an intelligent simulation model for each geometry structure consists of simple polygons. Fig. 1 is an example for such a simulation model for a vehicle after a crash. The right picture is the original picture of the damaged vehicle. The left picture is the simulation model consists of simple polygons.

Spatial Data Structures are employed in two main approaches. The first approach is diminishing the number of intersection detections of vehicle models in a given scenario [3,4]. For n structures, there will be O(n²) possible structures that possibly will be intersected. This number is clearly too high; therefore diminishing the number of intersection checks achieved by Spatial Data Structures is imperative.

The second approach is diminishing the number of intersection checks of pair of simple polygons in a crash probe of two vehicle models. In this approach, the Spatial Data Structures are generated in a preprocessing step and remain static because the vehicles are rigid and their models do not changed.

Spatial Data Structures are often employed for Space Partitioning [5] and Bounding Volumes [6]. Space Partitioning is a sub-partitioning of a space into convex regions called cells. Each cell keeps a list of objects that it contains. Using these structures, a computer can sift out numerous pairs of objects.

Bounding Volume is generated by a split of an object set into several subsets and finding for each one of the subsets tight bounding volume; so as a result when the computer checks intersections of each of the subsets, it will straightforwardly sift out these subsets because it will only have to find out which bounding volumes are not overlapping. Hardware-software codesign [7] can perform this even better.

Some research have been conducted on approaches of representing Bounding Volumes like Bounding Spheres [8], K-DOPs - Discrete orientation polytopes [9], OBB - Oriented Bounding Boxes [10], AABB - Axis Aligned Bounding Boxes [11] and Hierarchical Spherical Distance Fields [12].

We will employ the most widespread approach, the AABB approach. In this approach the bounding volume is denoted by minimum and maximum values of the vehicle model in each one of the axes. The disadvantage of the AABB approach is that its representation is more memory consuming than the "Bounding Sphere" approach; however, nowadays memory chips are much larger and much more inexpensive and furthermore AABB has an important advantage – its objects can more tightly enclose the vehicle model than Bounding Spheres can, which will generate less intersection checks.

Another advantage of AABB is the quick construction of bounding volumes [13]. This advantage is very important in a case of an autonomous vehicle accident when the vehicle's computer does not have much time to make its decisions. The computer just needs to check each element of the basic elements that the bounding volume consists of and projecting it on each of the axes. After that, just finding the minimum value and the maximum value for each axis and the construction is done.

In view of that, we employed the AABB approach. The creation of the bounding box tree has been recursive. First, the computer computes a bounding box for the set of the remaining triangles. Then, the computer splits the set of the triangles into two sub-meshes. At last, the computer executes the recursive process on the two new split sub-meshes.
2 Triangles Split Algorithm

Each triangular mesh with a corresponding bounding box can be split into two sub-meshes. The Triangles Split Algorithm is described herein below:

- Let min_sum be the maximum value that a float variable can represent.
- For each of the box axes:
  - Select a positive direction for the axis.
- For each triangle:
  - Find the maximal valued vertex on the projected axis.
  - Sort the triangles by their maximal vertex value.
- For each triangle from the minimum to the maximum:
  - Tag the triangle as a "split triangle" (This tag indicates that the first sub-mesh will contain the triangles from the minimal to the split triangle; whereas the second sub-mesh will contain the rest of the triangles).
  - Calculate the sum of the relative segments of the two sub-meshes. (A relative segment is the length of the projection of a sub-mesh onto the box axis divided by the original mesh projection length).
  - If the relative segments sum is less than the min_sum:
    - Let min_sum be the new relative segments.
    - Tag the current axis as the split axis.
    - Tag the current triangle index as the split index.
  - Split the triangles according to the latest split axis and the latest split triangle index.

Fig. 2: triangles' set arranged from left to right on the projecting axis with splitting index 4
The incentive of the algorithm is guaranteeing that there is the smallest possible overlapping between the two sub-meshes' bounding boxes.

Fig. 2 and Fig. 3 show a paradigm of two different split indices. Fig. 2 shows a possible split at triangle index 4. Such a split will generate a larger overlapping between the two divided segments than a split in triangle index 2. Fig. 3 shows this possible split in triangle index 2 and actually this explains why the algorithm would select triangle index 2 to be the split triangle if this was the case.

![Diagram of splitting index and overlapping segments](image)

Fig. 3: triangles' set arranged from left to right on the projecting axis with splitting index 2

### 3 Conclusions

The emerging concept of autonomous vehicles gave a boost to the embedded vehicular and transportation computing systems [14,15,16]. The autonomous vehicles motivated many researchers to revisit well-known subjects of computer science [17,18,19].

Primitive intersection is a well-known technique for real time computer graphics implementations like 3D game engine [20]. We suggested in this paper how to adapt this very general concept into a specific assessment tool for potential vehicle crash damage. Such a tool's aim is an automatic decision maker for autonomous vehicles that will decide in an inescapable accident scenario, which sort of accident is the least harmful crash.
References