International Journal of xxxxxx
Vol. x, No. x, xxxxx, 20xx

Automatic Alert System for Worn Out Pipes in Autonomous Vehicles

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

Abstract

Autonomous vehicles will be used without a driver. This will make them less inspected because the driver often listens to the engine and its sounds; however, a passenger in an autonomous vehicle will do other things and even sleep, so there will be no effective inspection on the engine and particularly on the pipes system. This paper suggests a technique to alert about worn out pipes before they are malfunctioning and the vehicle is unable to move anymore.

Keywords: *Autonomous Vehicle, Vehicle Pipes, Embedded Real-Time System.*

1. Introduction

Any vehicle has several kinds of pipes for several kinds of materials like gas, air, brake fluid, oil and more [1]. Like every mechanical parts, these pipes are also liable to wear and tear [2]. A worn out pipe can cause various problems depending on the pipe that has been torn [3].

There are pipes that even a large tear will not be a problem in specific cases like the pipe that supplies hot air to the carburetor in hot countries; whereas there are pipes that are very critical like the pipes of the brake fluids [4].

The goal of this paper is a system alerting about torn pipes. Autonomous vehicles are less supervised because there is no driver to pay attention for the engine and its sound. Also, many of the autonomous vehicles will be shared by more than a few users [5,6]. These users will not be familiar with the typical sounds and the typical performance of the engine, so they will not be able to notice different sounds; nor will they be able to become aware of performance degradation.

The concept of automatic detection of faulty parts of means of transportation has been suggested in the past. E.g. fuselage damage detection [7,8], damaged tires detection [9,10,11] or SkyTran tracks computerized inspection [12,13]. This paper suggests an automatic system that will be able to detect a torn pipe with the aim of warning the passenger that a pipe is going to be unusable before the pipe is indeed torn and the vehicle is out of use.

2. Methods

Polygons are simple shapes that can simulate the real objects. It is very common to generate models of real objects using simple polygons. This practice is usually called Spatial Data Structures [14].

When it comes to simulation of vehicle pipe systems, Spatial Data Structures are implemented in order to find the weak spots that are about to crack and realizing which polygons contain these weak spots. There are several methods to reduce the number of polygon checks when using Spatial Data Structures [15,16].

Spatial Data Structures are the basis for Space Partitioning [17] and Bounding Volumes [18]. Space Partitioning is a method of space sub-partitioning into convex regions. These convex regions are named "cells". Each of these cells maintains a list of objects that it comprises of. By employing these cells, the algorithm knows how to sift out polygons that have no connection to a pipe.

The other method is Bounding Volume. This method breaks an object into small components; then the algorithm finds a fitted bounding volume for each of the small component. After that, the algorithm checks for suspected components. It should be noted that the sifting out is less demanding in this method, because the algorithm just have to detect the at least partly cover bounding volumes.

Bounding Volumes applications have been intensely studied over the years and many variations of the method have been suggested: Bounding Spheres [19], K-DOPs - Discrete orientation polytopes [20], OBB - Oriented Bounding Boxes [21], AABB - Axis Aligned Bounding Boxes [22] and Hierarchical Spherical Distance Fields [23].

In this paper we have used the AABB approach which is one of the most well-known approaches. In AABB each of the bounding volume in the object model is represented by its minimum and its maximum values [24]. Compared to the "Bounding Sphere" approach, AABB has an advantage and a disadvantage. AABB encompasses the components of the model more tightly which probably yield less checks and also the split of the object into its bounding volumes is faster [25]. The algorithm first checks each of the basic elements that a bounding volume consists of and projects the element on the axes and then finds the minimum and the maximum values for each axis. The fast operation is very essential in real time systems like checking pipes of autonomous vehicle.

However, AABB has also a disadvantage. Saving the data for AABB takes more memory space which in the past was very costly and it has to be even on a remote machine [26], but nowadays, memory space is much less costly and even simple computers have a plenty of memory space [27,28], so this disadvantage is not so acute; therefore, we have chosen the AABB approach.

Since our system is a real time system and the computation time is very essential we have decided to implement the AABB approach. We generated the bounding volume tree in a recursive manner. In each step, the algorithm generates bounding volumes for the remaining triangles and splits the triangle set into two sub-graphs. Then, it recursively calls itself to do the same for each of these sub-graphs.

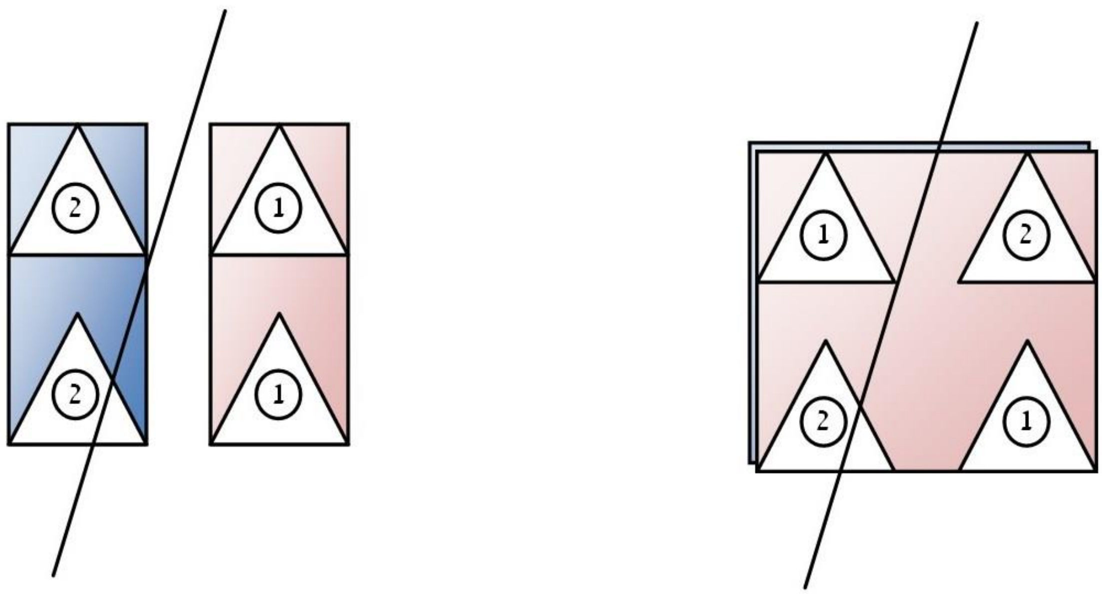
3. Bounding Volume hierarchies

Bounding volume hierarchies are actually a tree symbolizes a model of an object [29]. The basic components are the leaves of the tree and each sub-tree rooted by an internal node represents a segment of the model.

Such trees have only one leave for each basic component, so the size of the storage needed for each vehicle model is linear in the number of the basic components. This also impacts on the check time which is therefore quite fast.

The construction time however is longer. This can be a significant disadvantage when the model is flexible and a reconstruction is frequently needed whenever the object changes its shape; however a vehicle is a rigid object and no changes are made, so the construction can be done only once and the tree will fit for the entire life of the vehicle, so this disadvantage is irrelevant for the objective of this paper.

When the algorithm checks for cracks, it will begin to check the roots of the model trees and then in a recursive manner it will go down the trees.



(a) **(b)**

Figure 1. Triangle split (**a**) Example of split that causes 2 checks; (**b**) Example of split that causes 4 checks.

4. Implemntation

As was explained above, the implementation of the method was in a recursive manner. We used triangles as it is a very common in such implementations [30].

Initially, the algorithm finds a bounding volume for the remaining triangles. Then, the algorithm splits the set of triangles into two sub-graphs. Finally, the algorithm calls itself in a recursive manner to handle the two new sub-graphs that have been generated in the previous step. When a sub-graph has just one triangle, the recursive call will stop and the algorithm will not call itself any longer

The incentive for the triangles split into several sub-graphs is the formation of as small as possible bounding volumes so the model will as accurate as possible.

Figure 1 explains by an example how four triangles can be split in two different ways. The numbers written in the triangles denote which sub-graph contains the triangle after the suggested split. This figure clearly shows that a hierarchical checking with a specific segment can generate more triangle checks in the right side figure because the generated bounding volume remains bigger. This attribute motivated us to implement a split algorithm that uses better splits as in the left side, so as to minimize the triangle checks.

Actually, the bounding volume algorithms and the triangle split algorithms greatly shape the bounding volume tree generation algorithm and its efficiency. We have employed "Fitting points with Gaussian distribution" [31] to as the basis for the algorithm for generating the bounding volumes.

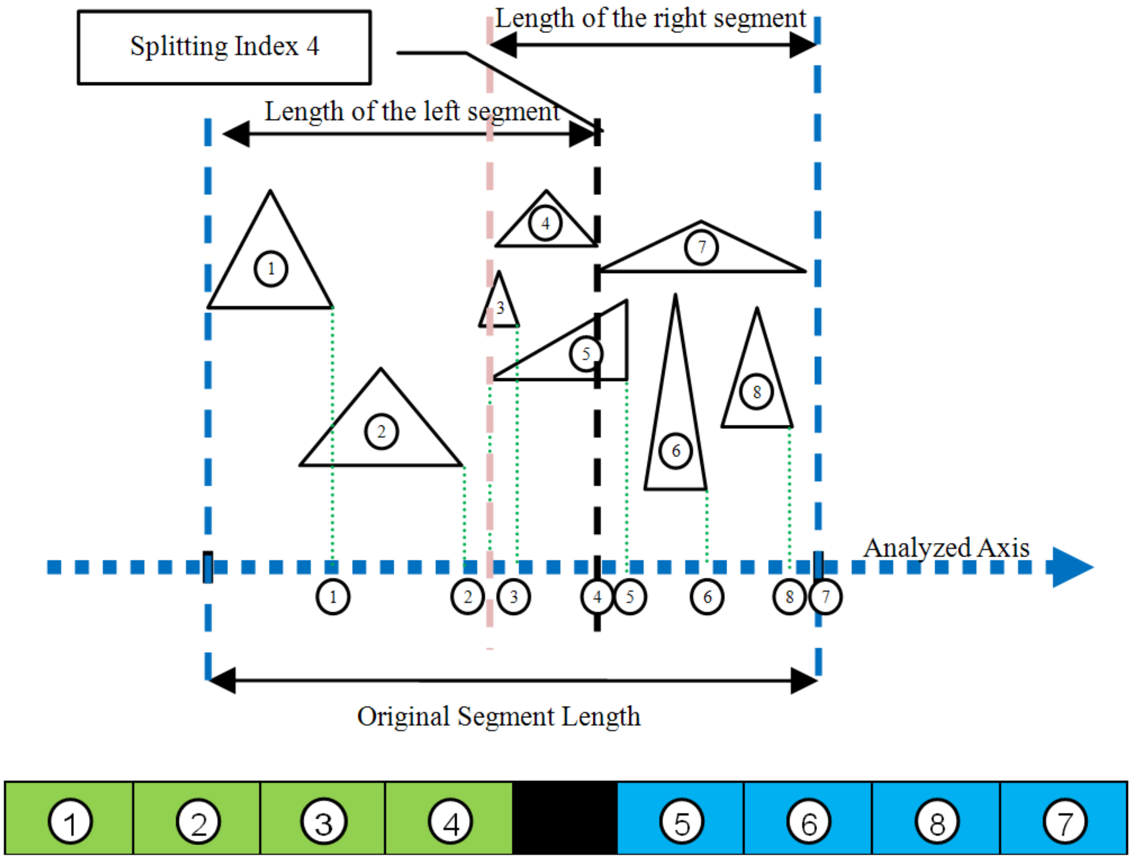


Figure 2. Example of triangles' split on the projecting axis.

5. Triangle Split Algorithm

Each graph of triangles has a corresponding bounding volume that can be split into two sub-graphs. The pseudo code of this split algorithm is:

- For each axis of any volume select a positive direction.
- For each triangle find a vertex with the highest value on the projected axis.
- Sort the triangles vector by their vertex values.

• For each triangle in the triangles vector compute the sum of the projection lengths on the axis divided by the original graph projection lengths of the two sub-graphs and split this triangle in order to get a smaller sub-graphs.

The algorithm strives to do its best in order to generate the least possible overlapping sub-graphs.

Figure 2 and Figure 3 illustrate two different potential index split. Figure 2 shows one potential split at triangle index 4. If indeed the algorithm chooses to split at this index, the split will ineffectually generate a larger overlap between the two new graphs. Figure 3 shows a better option which is a split in triangle index 2 which generates a smaller overlap between the two new graphs. This example shows a case when the algorithm would select a better split over an inferior split. In this particular example it was a split at triangle index 2 rather than a split at triangle index 4.

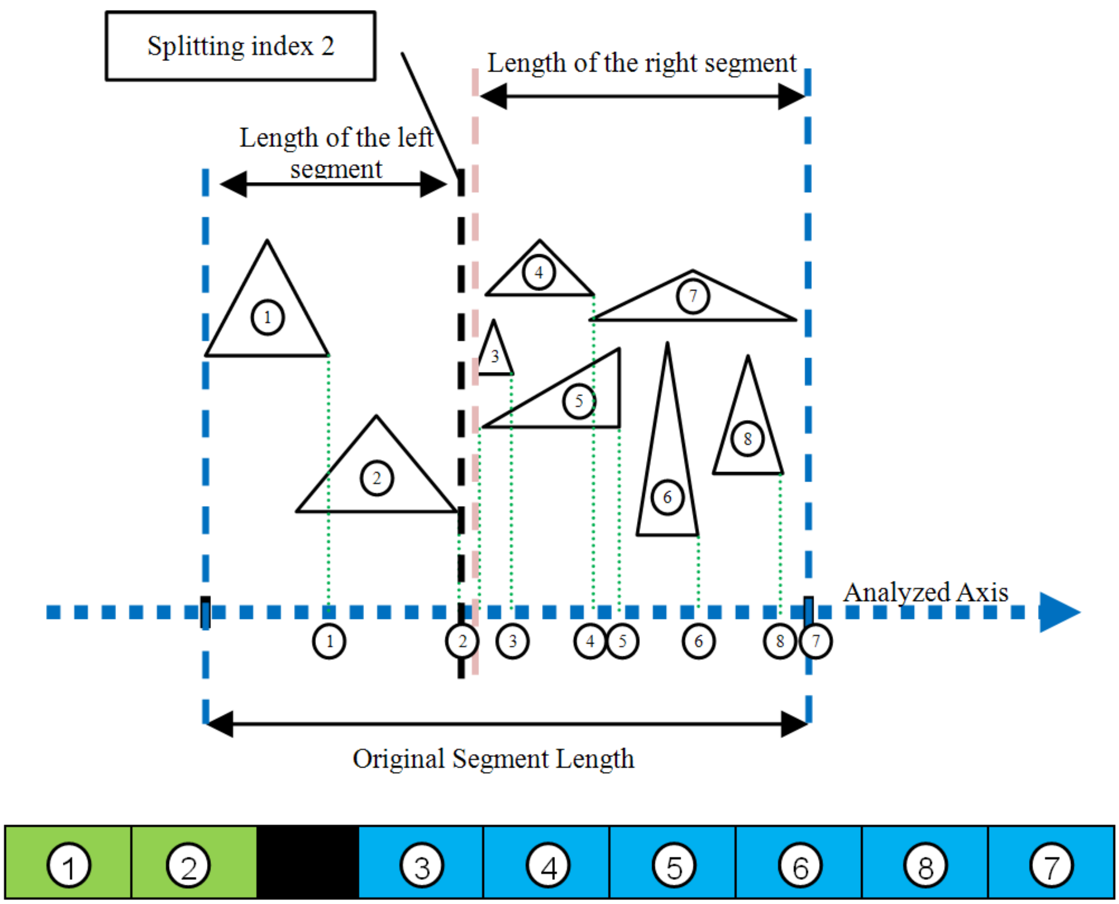


Figure 3. Yet another example of triangles' split on the projecting axis.

6. Results

The proposed system has been tested on several photos of pipes. An example can be seen in Figure 4.

Figure 4. Example of a torn pipe

The photo shows a pipe that has been torn in the left side. This rift can be easily noticed by human eyes; however, computers need a clear-cut denotation and the triangular split can facilitate this.

Figure 5 shows a triangulated drawing of Figure 4 that can be better handled by a computer.

Figure 5. Triangulated drawing of Figure 4

The technique also has a limitation when false positive circumstances occur. The pipe in Figure 6 is actually unflawed; however, some sorts of liquids have been leaked on this pipe and because their color is very different, the stains are very noticeable and the algorithm considers this pipe as a torn pipe.

Because of these false positive occurrences, any alert of the system should be checked by a human so as to observe whether the alert is genuine or just a false positive warning.

Figure 6. False positive case of an unflawed pipe

In Figure 7, a triangulated drawing of Figure 6 can be seen. Many triangles' colors have been changed because of the stains. These color changes can be misinterpreted by the system as a torn pipe whereas this pipe is just very uncleanly.

Figure 7. Triangulated drawing of an uncleanly pipe

7. Conclusions

Continuous checkup is an ordinary task of computers along with real time decision making. Autonomous vehicle is no exception and continuous checkup along with real time decision making is certainly done by a computer [32,33,34].

This paper presents a technique to handle the inspection of damaged tires by an automatic system. This system is very important for autonomous vehicles where the passengers are not so aware to the vehicle reverberations and also typically often changed.

The proposed system is designed to detect the damage before it becomes tangible and the vehicle cannot go on. An earlier alert can avoid unpleasant circumstances like vehicle that is abruptly unable to move.

8. References

[1] W. Durfee and S. Zongxuan, "Fluid power system dynamics", mini-book published by The Center for Compact and Efficient Fluid Power in Department of Mechanical Engineering University of Minnesota, **(2009).**

[2] Y. Zhou and Y. Zhang, "Applications of Bayesian Network in Fault Diagnosis of Braking Deviation System", In IEEE Fourth International Symposium on Computational Intelligence and Design (ISCID), pp. 170-173, **(2011).**

[3] T. Zhang and X. Liu, "Reliability design for impact vibration of hydraulic pressure pipeline systems", Chinese Journal of Mechanical Engineering, Springer-Verlag Berlin Heidelberg, Vol. 26, No. 5, pp. 1050-1055, **(2013).**

[4] D. Antanaitis, M. Riefe and J. Sanford, "Automotive brake hose fluid consumption characteristics and its effects on brake system pedal feel", SAE International Journal of Passenger Cars-Mechanical Systems, Vol. 3, no. 2010-01-0082, pp. 113-130, **(2010).**

[5] Y. Wiseman, "Self-Driving Car - A Computer will Park for You", International Journal of Engineering & Technology for Automobile Security, Vol. 1, No. 1, pp. 9-16, **(2017).**

[6] Y. Wiseman, "Remote Parking for Autonomous Vehicles", International Journal of Hybrid Information Technology, Vol. 10, No. 1, pp. 313-324, **(2017).**

[7] Y. Wiseman, "Device for Detection of Fuselage Defective Parts", Information Journal, Tokyo, Japan, Vol. 17(9(A)), pp. 4189-4194, **(2014).**

[8] Y. Wiseman, "Fuselage Damage Locator System, Advanced Science and Technology Letters, Vol. 37, pp. 1-4, **(2013).**

[9] Y. Wiseman, "Take a Picture of Your Tire!", Proc. IEEE Conference on Vehicular Electronics and Safety (IEEE ICVES-2010) Qingdao, ShanDong, China, pp. 151-156, **(2010).**

[10] Y. Wiseman, "The Effectiveness of JPEG Images Produced By a Standard Digital Camera to Detect Damaged Tyres", World Review of Intermodal Transportation Research, Vol. 4, No. 1, pp. 23-36, **(2013).**

[11] Y. Wiseman, "Camera That Takes Pictures of Aircraft and Ground Vehicle Tires Can Save Lives", Journal of Electronic Imaging, Vol. 22, No. 4, 041104, **(2013).**

[12] Y. Wiseman, "Safety Mechanism for SkyTran Tracks", International Journal of Control and Automation, Vol. 10, No. 7, pp. 51-60, 2017.

[13] Y. Wiseman, "Automatic Persistent Inspection of SkyTran Track System", http://u.cs.biu.ac.il/~wiseman/skytran1.pdf , **(2017).**

[14] K. Keul, P. Lemke and S. Müller, "Accelerating spatial data structures in ray tracing through precomputed line space visibility", In Proceedings of the 24th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, Plzen, Czech Republic, pp. 17-26, **(2016).**

[15] I. Grinberg and Y. Wiseman, "Scalable Parallel Simulator for Vehicular Collision Detection", International Journal of Vehicle Systems Modelling and Testing, Inderscience Publication, Vol. 8, No. 2, pp. 119-144, **(2013).**

[16] I. Grinberg and Y. Wiseman, "Scalable Parallel Collision Detection Simulation", In Proceedings of Signal and Image Processing, Honolulu, Hawaii, pp. 380-385, **(2007).**

[17] R. Vuillemot, and J. Boy, "Structuring Visualization Mock-ups at the Graphical Level by Dividing the Display Space", To appear in IEEE Transactions on Visualization and Computer Graphics, **(2017).** Available at: http://romain.vuillemot.net/publis/infovis17-visualization-mockups.pdf

[18] J. S. Park, C. Park and D. Manocha, "Efficient probabilistic collision detection for non-convex shapes" In proceedings of IEEE International Conference on Robotics and Automation (ICRA), pp. 1944-1951, Singapore, **(2017).**

[19] T. Larsson, "Exact bounding spheres by iterative octant scan", In Proceedings of SIGRAD 2015, Stockholm, Sweden, pp. 9-12, **(2015).**

[20] Y. Wang, Z. X. Luo, J. C. Liu, X. Fan, H. Li, and Y. Wu, "Real-time estimation of hand gestures based on manifold learning from monocular videos", Multimedia tools and applications, Vol. 71, No. 2, pp. 555-574, **(2014).**

[21] Q. Fu, X. Chen, X. Su, J. Li and H. Fu, "Structure-adaptive Shape Editing for Man-made Objects", In Computer Graphics Forum, Vol. 35, No. 2, pp. 27-36, **(2016).**

[22] P. Cai, C. Indhumathi, Y. Cai, J. Zheng, Y. Gong, T. S. Lim and P. Wong, "Collision detection using axis aligned bounding boxes", In Simulations, Serious Games and Their Applications, pp. 1-14. Springer Singapore, **(2014).**

[23] R. Weller, "New geometric data structures for collision detection and haptics", Springer Science & Business Media, **(2013).**

[24] I. Grinberg and Y. Wiseman, "Scalable Parallel Simulator for Vehicular Collision Detection", Proc. IEEE Conference on Vehicular Electronics and Safety (IEEE ICVES-2010) Qingdao, ShanDong, China, pp. 116-121, **(2010).**

[25] U. Schwesinger, R. Siegwart, and P. Furgale, "Fast collision detection through bounding volume hierarchies in workspace-time space for sampling-based motion planners", In proceedings of IEEE International Conference on Robotics and Automation (ICRA), Seattle, Washington, USA, pp. 63-68, **(2015).**

[26] Y. Wiseman, K. Schwan and P. Widener, "Efficient End to End Data Exchange Using Configurable Compression", Proceedings of The 24th IEEE Conference on Distributed Computing Systems (ICDCS 2004), Tokyo, Japan, pp. 228-235, **(2004).**

[27] P. Weisberg and Y. Wiseman, "Using 4KB Page Size for Virtual Memory is Obsolete", Proc. IEEE Conference on Information Reuse and Integration (IEEE IRI-2009), Las Vegas, Nevada, pp. 262-265, **(2009).**

[28] P. Weisberg and Y. Wiseman, "Virtual Memory Systems Should Use Larger Pages rather than the Traditional 4KB Pages", International Journal of Hybrid Information Technology, Vol. 8(8), pp. 57-68, **(2015).**

[29] D. Meister and J. Bittner, "Parallel Locally-Ordered Clustering for Bounding Volume Hierarchy Construction", To appear in IEEE Transactions on Visualization and Computer Graphics, IEEE Xplore Digital Library, **(2017).**

[30] K. J. Mei, and R. S. Lee, "Collision detection for virtual machine tools and virtual robot arms using the Shared Triangles Extended Octrees method", International Journal of Computer Integrated Manufacturing, Vol. 29, No. 4, pp. 355-373, **(2016).**

[31] L. Maarten, and J. M. Phillips, "Shape fitting on point sets with probability distributions" In Algorithms-ESA 2009, pp. 313-324, Springer Berlin Heidelberg, **(2009).**

[32] Y. Wiseman and I. Grinberg, "When an Inescapable Accident of Autonomous Vehicles is Looming", International Journal of Control and Automation, Vol. 9 No. 6, pp. 297-308, **(2016).**

[33] Y. Wiseman and I. Grinberg, "Autonomous Vehicles Should Not Collide Carelessly", Advanced Science and Technology Letters, Vol. 133, pp. 223-228, **(2016).**

[34] Y. Wiseman and I. Grinberg, "Circumspectly Crash of Autonomous Vehicles", Proceedings of IEEE International Conference on Electro Information Technology (EIT 2016), Grand Forks, North Dakota, USA, pp. 382-386, **(2016).**

Authors

Yair Wiseman

Dr. Yair Wiseman got a Summa Cum Laude M.Sc. and a PhD from Bar-Ilan University and completed two Post-Doc - one at the Hebrew University of Jerusalem and one in Georgia Institute of Technology.

Dr. Wiseman's research interests include Computational Transportation Science, Intelligent Transportation Systems, Process Scheduling, Hardware-Software Codesign, Memory Management, Computer Clusters, Data Compression, JPEG, Embedded Systems, Real-Time Systems and Operating Systems.

Dr. Wiseman is on the editorial board of several journals, a member of dozens of conference committees and a reviewer of many scholarly journals. Dr. Wiseman authored two books as well.

In addition, Dr. Wiseman has been teaching in many institutes including Bar-Ilan University, The Hebrew University of Jerusalem, Israel Aircraft Industry, Holon Institute of Technology and Jerusalem College of Technology.

Dr. Wiseman has been supervising many graduate students and an interesting point is that Albert Einstein is Dr. Wiseman's academic great-great-grandfather (i.e. the advisor of the advisor of the advisor of Dr. Wiseman's advisor).

Dr. Wiseman has collaborated with other partners and received research grants to run an active laboratory from inter alia Sun Microsystems, Intel, Polak Foundation and the Open University.

Dr. Wiseman is an international expert who has reviewed and evaluated several large projects of the European Union, Israel Science Foundation, MB Logic and more. Dr. Wiseman's papers have been published in many venues around the world.