In: Advances of Machine Learning … ISBN: 978-1-68507-211-7
Editors: Pandian Vasant et al.

Chapter 9

WARNING SYSTEM FOR CRACKED PIPES IN AUTONOMOUS VEHICLES

Yair Wiseman*

Computer Science Department, Bar-Ilan University,
Ramat Gan, Israel

ABSTRACT

The looming autonomous vehicles will be driven by the vehicle itself; therefore, the autonomous vehicles will be less inspected because human drivers often listen to the engine and its noises. The passengers in autonomous vehicles are usually bothered with other things and sometimes even sleep, so there will be no effective human inspection on the engine and specifically on the pipe system. This paper suggests an automatic system that warns about cracked pipes before they are out of use.

Keywords: autonomous vehicle, vehicle pipes, embedded real-time system

* Corresponding Author's Email: wiseman@cs.biu.ac.il.

1. INTRODUCTION

Vehicles have a number of pipes for several uses like pipes for gas, air, brake fluid, oil and other uses [1]. Any component of vehicles is potentially subject to wear and tear and these pipes are no exception [2]. Worn out pipes may bring about various malfunctions depending on the worn out pipe [3].

In some cases torn pipe even on the whole is not a problem; whereas other torn pipes can be an acute problem. Pipes supplying hot air to the vehicle's carburetor in hot countries is actually quite redundant and if this pipe is torn, the vehicle can continue going; whereas the pipes of the brake fluids are very essential and cannot be relinquished [4].

This paper aims at developing a system for torn pipes detection. Many times, drivers' attention is distracted even in non-autonomous vehicles [5]; however, autonomous vehicles [6] are even less supervised because no driver is needed; therefore, many times no one will pay attention to the vehicle's engine and its noises, because the vehicle user's attention can be paid for other issues and this user can even take a nap [7]. In addition, many autonomous vehicles will be shared by several users [8, 9]. These users will not be au fait with the typical noises and the typical functioning of the vehicle's engine, so they will not be aware of changes in the vehicle's noises; nor will they notice changes in the vehicle's functioning.

Several researches about automatic detection of malfunctioning components in transportation have been proposed during recent years - fuselage damage detection [10, 11], damaged tires detection [12, 13, 14] and SkyTran tracks automatic inspection [15, 16]. This paper aims at proposing an automatic device detecting a nearly torn pipe and warning the vehicle's user that this pipe is almost torn before the pipe is indeed torn, so the user can take care of this torn pipe, while the vehicle still functions.

2. AXIS ALIGNED BOUNDING BOXES

We employed polygons. Polygons are minimal shapes that are able to approximately create real objects. It is widespread practice to create models of real objects employing plain polygons. Such a polygons model is called Spatial Data Structures [17]. Several approaches with the aim of reducing the number of polygon checks when employing Spatial Data Structures have been suggested [18].

We have employed Spatial Data Structures to simulate vehicle pipes with the intention of locating frail areas that have the potential to crack and spotting the polygons containing these frail areas.

Actually, Spatial Data Structures are the starting point of Space Partitioning [19] and Bounding Volumes [20]. Space Partitioning is a technique of space sub-partitioning into convex shapes. These convex shapes are called "cells." Every cell maintains a list of several objects that it comprises. The algorithm can sift out polygons that are unrelated to a pipe by employing these cells.

Bounding Volume is a similar technique. The algorithm breaks each object into small components. Next, the algorithm finds out the close-fitting bounding volume for each small component. Then, the algorithm detects the components that might contain a crack. It should be noted that in this technique the sifting out is less time-consuming, because the algorithm can detect only the minimal partly cover bounding volumes.

Applications using Bounding Volumes have been researched over the years. Various alternatives of implementation have been proposed: Bounding Spheres [21], K-DOPs - Discrete orientation polytopes [22], OBB - Oriented Bounding Boxes [23], AABB - Axis Aligned Bounding Boxes [24] and Hierarchical Spherical Distance Fields [25].

We have employed the AABB approach which is one of the most widespread approaches. When using AABB, each bounding volume of an object is represented by its minimum and its maximum values [26]. There are pros and cons for employing AABB. The main advantage of AABB is more compactly encompassing the model components which will generally produce fewer checks. In addition, an object split into bounding volumes is typically faster [27]. Initially, the algorithm analyses each basic element that a bounding volume consists of. Then the algorithm projects the element on the axes and, so as to find out the minimum and the maximum values in each axis. This fast procedure is obviously important in real time systems like a device for automatic pipe check in autonomous vehicles.

On the other hand, AABB has a disadvantage. AABB needs more memory space for saving the information. Some years ago, memory space was limited and sometimes the information was even on remote machines [28]; however, nowadays, memory space is much less limited and even everyday computers have an abundant memory space [29, 30], so the limited memory space disadvantage is not critical; therefore, we could chose the AABB approach, because the advantages that have mentioned above. Specifically, the pipe's warning device is a real time system, so the computation time is very crucial. Therefore, we have come to a decision to adopt the AABB approach.

The bounding volumes tree was produced in a recursive method. In each step, the algorithm produces bounding volumes for the remaining triangles. Then, the algorithm splits the triangle set into two sub groups, followed by recursive calls with the aim of processing each of the sub groups in the same way.

We have employed Bounding Volume Hierarchies which are a tree representing an object [31]. In Bounding Volume Hierarchies each sub-tree is rooted by an internal node and represents a segment of the object.

The trees of Bounding Volume Hierarchies have only a single leaf for each basic component, so accordingly the memory size required for each object is linear in the number of the basic components. This feature also has an effect on the checking time which will be quite faster.

3. Assumptions

The Bounding Volume Hierarchies construction is time-consuming. This drawback can be critical if the object is malleable and a deformation is repeatedly required. Therefore, in detection of moving vehicles usually different techniques have been chosen [32, 33, 34, 35].

However, pipes are statically secured in the vehicle and twists are hardly ever made, so the Bounding Volume Hierarchies construction is performed on the odd occasion, so this drawback is beside the point of this paper's aim.

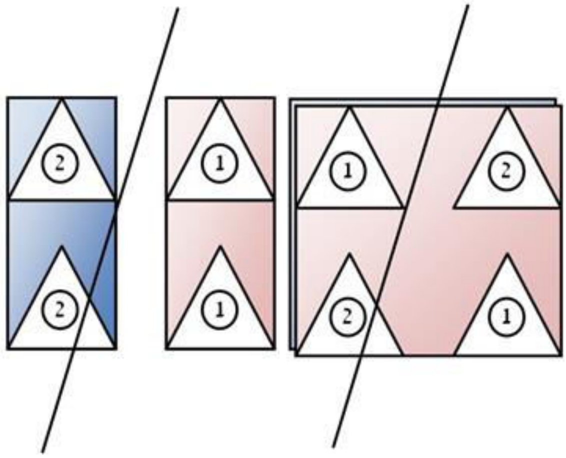
The algorithm scans the tree with the aim of finding possible cracks. The algorithm checks the trees from their roots and in a recursive manner it will continue all the way down to the leaves of the trees.

4. IMPLEMENTATION

As was explained above, the system has been implemented in a recursive manner. We have employed triangles as many other implementations do [36]. Firstly, the algorithm sets a bounding volume for the current group of triangles. Next, the algorithm splits this group of triangles into two sub-groups. Lastly, the algorithm calls itself in a recursive manner to process the two latest sub-groups. If a sub-group contains only one triangle, the recursive call to the algorithm will come to an end.

The reason for the triangles split into many sub-groups is the creation of as small as possible bounding volumes so the object model will be as perfect as possible.

For example, four triangles can be split by the algorithm in two different ways shown in Figure 1. The numbers written in a circle within the triangles denote which sub-group contains each triangle after each split. Figure 1 visibly points out that generating bigger bounding volumes can cause more triangle checks as it is shown in Figure 1(b). This property of the hierarchical checking motivated us to implement a better split algorithm making use of better splits as in Figure 1(a), with the aim of minimizing triangle checks [37].



(a) (b)

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

Essentially, both the triangle split algorithm and the bounding volume algorithm form the bounding volume tree generation algorithm and control its efficiency. In our implementation, we have employed "Fitting points with Gaussian distribution" [38] as the core of the bounding volume generation algorithm.

The algorithm splits every group of triangles having an analogous bounding volume into two sub-groups. The pseudo code of this algorithm is:

- For each axis of each bounding volume, find a positive direction.
- For each triangle, find the highest valued vertex on the projected axis.
- Sort the triangles' vector by the triangles vertex values.
- For each triangle in the triangles vector:
  – Divide the sum of the projection lengths on the axis by the sum of the original projection lengths of the two sub-groups.
  – Split this triangle with the aim of generating smaller sub-groups.

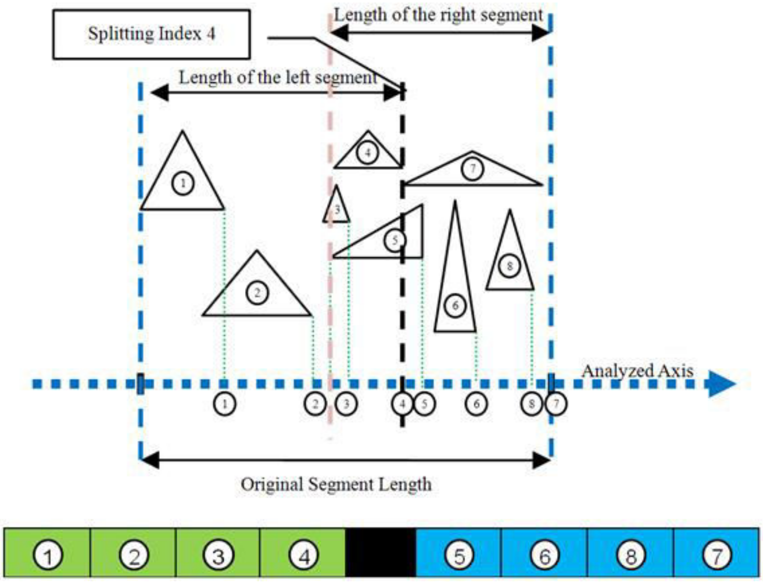


Figure 2. Example of triangles' split on the projecting axis.

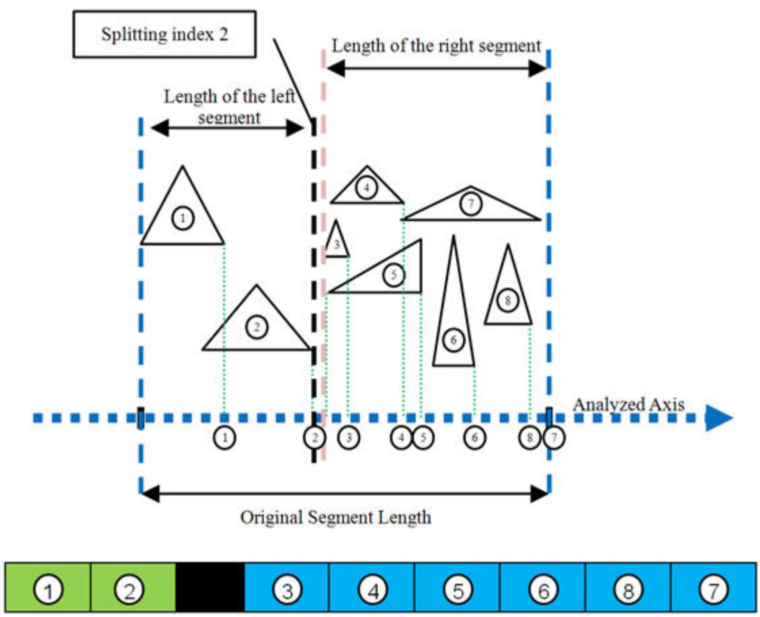


Figure 3. Yet another example of triangles' split on the projecting axis.

The algorithm endeavors to generate as few overlapping sub-groups as possible.

Figure 2 and Figure 3 exemplify two different possible splits. Figure 2 suggests one possible split at triangle index 4. If the algorithm takes this split, a larger overlap between the two generated groups will be created. Figure 3 suggests a better possibility which is a split in triangle index 2. Such a split will generate a much smaller overlap between the two generated groups. Figure 2 and Figure 3 exemplifies when the algorithm should take the best split.

5. RESULTS

The warning system for cracked pipes has been tested on several pipes. The pipe in Figure 4 has been torn in the left side. It is a noticeable rift that can be easily noticed by human eyes; however, an automatic electronic system needs a clear-cut denotation and the triangular split can facilitate such a denotation.

Figure 4. Example of a torn pipe.

Figure 5. Triangulated drawing of Figure 4.

Figure 5 is a triangulated sketch of Figure 4. Figure 5 can be handled by the warning system for cracked pipes and the considerable difference of the colors can hint on a potential rift.

6. Limitations

The warning system for cracked pipes is not accurate at all times. False positive cases are a weakness of the warning system for cracked pipes. Figure 6 exemplifies such a false positive case. Some liquids have been dropped on the pipe in Figure 6. Since the color of the original pipe and the color of the dripping are very different, the stains are quite noticeable and the algorithm has considered this pipe as a torn pipe and warned the user.

A triangulated sketch of Figure 6 is shown in Figure 7. Because of the stains, the colors of many triangles have been distorted. These distortions are misrepresented by the warning system for cracked pipes as rifts whereas this pipe is just dirty.

Because of such false positive cases, any warning of this system should be checked by a human in order to verify that the warning is valid and not a mere false positive warning.

Figure 6. False positive case of an unflawed pipe.

Figure 7. Triangulated drawing of an uncleanly pipe.

7. Future Research Directions

There is an extra execution time when employing OBB (Oriented Bounding Boxes) compared to the AABB technique we have employed. However, the OBB technique can fit the shape of the pipes tighter and do not have to be fully updated if the pipes are repositioned.

In embedded systems it is not common to make use of the most expensive and the fastest CPU; however, if we can afford a much better and faster CPU, choosing the OBB technique might be a more appropriate choice as a bounding volume than the AABB technique. As a future research direction, it would be interesting to check with some experiments, how fast the CPU should be in order to make the OBB technique a reasonable choice.

Conclusion

This paper presents a device that is able to automatically conduct a continuous inspection of potential damaged pipes. This device is especially essential for autonomous vehicles where the passengers are commonly unaware of the vehicle noises, because they do other things. Furthermore, if the vehicles are shared, the users are often changed and so they are not familiar with the vehicle noises.

Continuous inspection along with real time warnings is a common task of computers and autonomous vehicles also adapt this tactic [39, 40, 41, 42]. The suggested device can detect damages before they become tangible and the vehicle is out of use. A timely warning can help to alert the user so as to avoid all of a sudden vehicles coming to a standstill.

Such an intelligent device can be very helpful because detecting damages in pipes before the vehicle stops in nowhere, can avoid annoyance and frustration for the passenger.

References

[1] Johnson, T. and Joshi A., (2018), "Review of Vehicle Engine Efficiency and Emissions," *SAE International Journal of Engines,* 11(6), 1307-1330.

[2] Pentapalli, P., Siddiqui, N. A., Mondal, P., and Nandan, A. (2020). "Impact of Speed Breakers on Air Contamination and Mileage of Vehicles," In *Advances in Air Pollution Profiling and Control* (pp. 211-215). Springer, Singapore.

[3] Li, X. D., Yu, Z. W., and Xu, X. L. (2018). "Failure Analysis on High Pressure Fuel Pipes Used in Truck Diesel Engine," *Failure Analysis and Prevention*, 02.

[4] Nadanasabapathy, S., and Kumar, S. R. (2020). "Analysis of fluid behaviour inside the brake lines in non-anti-lock braking system," *Materials Today*, Volume 33, Part 7, pp. 2690-2696.

[5] Alam, L., & Hoque, M. M. (2018). "Vision-Based driver's attention monitoring system for smart vehicles," *International Conference on Intelligent Computing & Optimization*, Springer, pp. 196-209.

[6] Wiseman, Y., (2020), "Autonomous Vehicles," *Encyclopedia of Information Science and Technology*, Fifth Edition, Vol. 1, Chapter 1, pp. 1-11, 2020.

[7] Chui, K. T., Zhao, M., & Gupta, B. B. (2020), "Long Short-Term Memory Networks for Driver Drowsiness and Stress Prediction," *International Conference on Intelligent Computing & Optimization,* Springer, pp. 670-680.

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

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

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

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

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

[13] Wiseman, Y., (2013), "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.

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

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

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

[17] Howari, F. M., and Ghrefat, H. (2021). "Geographic information system: spatial data structures, models, and case studies," In *Pollution Assessment for Sustainable Practices in Applied Sciences and Engineering* (pp. 165-198). Butterworth-Heinemann.

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

[19] Chen, Z., Tagliasacchi, A., and Zhang, H. (2020). "Bsp-net: Generating compact meshes via binary space partitioning," In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 45-54.

[20] Wei, L., Zhou, H. and Nahavandi, S. (2017), "Haptic collision detection on disjoint objects with overlapping and inclusive bounding volumes," *IEEE transactions on haptics*, 11(1), 73-84.

[21] Sung, K. and Smith, G. (2019). "Distances and Bounding Spheres." In *Basic Math for Game Development with Unity 3D*, pp. 87-114, Apress, Berkeley, CA.

[22] Zhou, Y. W., Hu, Z. Z., Lin, J. R., and Zhang, J. P. (2019). "A review on 3D spatial data analytics for building information models." *Archives of Computational Methods in Engineering*, 1-15.

[23] Pijnacker Hordijk, L. (2020). "From Axis-aligned to Oriented Bounding Boxes: An optimization method to reconstruct Oriented Bounding Boxes from Axis-aligned Bounding Boxes of vehicles in low-altitude aerial imagery," *Delft University of Technology.*

[24] Muntoni, A., Livesu, M., Scateni, R., Sheffer, A., and Panozzo, D. (2018). "Axis-aligned height-field block decomposition of 3D shapes. *ACM Transactions on Graphics (TOG)*," 37(5), 1-15.

[25] Funfzig, C., Ullrich, T., and Fellner, D. W. (2006). "Hierarchical spherical distance fields for collision detection. *IEEE Computer Graphics and Applications,*" 26(1), 64-74.

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

[27] Melero, F. J., Aguilera, Á., and Feito, F. R. (2019). "Fast collision detection between high resolution polygonal models," *Computers & Graphics*, 83, 97-106.

[28] Wiseman, Y., K. Schwan and P. Widener, (2004), "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.

[29] Weisberg, P. and Wiseman Y., (2009), "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.

[30] Weisberg, P. and Wiseman Y., (2015), "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.

[31] Jaillet, F., Zachmann, G., Erleben, K., & Andrews, S. (2018). "SIMDop: SIMD Optimized Bounding Volume Hierarchies for

Collision Detection," In *Workshop on Virtual Reality Interaction and Physical Simulation (VRIPHYS-2018)*, pp. 1-3.

[32] Wiseman, Y., "Real-Time Monitoring of Traffic Congestions," *IEEE International Conference on Electro Information Technology (EIT 2017)*, Lincoln, Nebraska, USA, pp. 501-505, (2017).

[33] Wiseman, Y., "Tool for Online Observing of Traffic Congestions," *International Journal of Control and Automation*, Vol. 10, No. 6, pp. 27-34, (2017).

[34] Wiseman, Y., "Computerized Traffic Congestion Detection System," *International Journal of Transportation and Logistics Management*, Vol. 1, No. 1, pp. 1-8, (2017).

[35] Saha, R., Debi, T., & Arefin, M. S. (2020), "Developing a Framework for Vehicle Detection, Tracking and Classification in Traffic Video Surveillance," *International Conference on Intelligent Computing & Optimization*, Springer, pp. 326-341.

[36] Li, Y., Shellshear, E., Bohlin, R., and Carlson, J. S. (2020, May). "Construction of Bounding Volume Hierarchies for Triangle Meshes with Mixed Face Sizes." In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9191-9195.

[37] Wiseman, Y., (2017), "Automatic Alert System for Worn Out Pipes in Autonomous Vehicles," *International Journal of Advanced Science and Technology*, Vol. 107, pp. 73-84.

[38] Duan, Z., Wang, N., Fu, J., Zhao, W., Duan, B., and Zhao, J., (2018). High precision edge detection algorithm for mechanical parts. *Measurement Science Review*, 18(2), 65-71.

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

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

[41] Wiseman, Y. and Grinberg I., (2016), "Circumspectly Crash of Autonomous Vehicles," *Proceedings of IEEE International*

Conference on Electro Information Technology (EIT 2016), Grand Forks, North Dakota, USA, pp. 382-386.

[42] Wiseman, Y., (2018) "Ancillary Ultrasonic Rangefinder for Autonomous Vehicles," *International Journal of Security and its Applications,* Vol. 12(5), pp. 49-58.