Analyzing wheels of vehicles in motion using laser scanning

Møgelmose, Andreas; Moeslund, Thomas B.

Published in:
Computer Vision and Pattern Recognition Workshops 2016

DOI (link to publication from Publisher):
10.1109/CVPRW.2016.199

Publication date:
2016

Document Version
Accepted author manuscript, peer reviewed version

Analyzing wheels of vehicles in motion using laser scanning

Andreas Møgelmose and Thomas B. Moeslund
Visual Analysis of People Lab, Aalborg University
Rendsburggade 14, 9000 Aalborg, Denmark
[am,tbm]@create.aau.dk

Abstract

In this paper, we discuss a lidar-based wheel-width measuring system. Trucks are used for an enormous part of day to day freight delivery across the world, and cause significant wear to the road they use. Road planners and construction engineers rely on traffic statistics to properly design roads. Currently, no system is able to provide them with numbers on the wheel-widths of the vehicles using a particular stretch of road.

We present a system which uses a horizontal lidar measuring in a plane close and nearly parallel to the road surface. Input from this is used to detect and analyze tires. A vertical lidar detects passing vehicles so individual tires can be combined into full vehicle models. The system detects 58% of passing vehicles, but correctly counts the number of axles on 85% of detected vehicles. More than 90% of of the axles are correctly classified according to the number of mounted tired (single or dual).

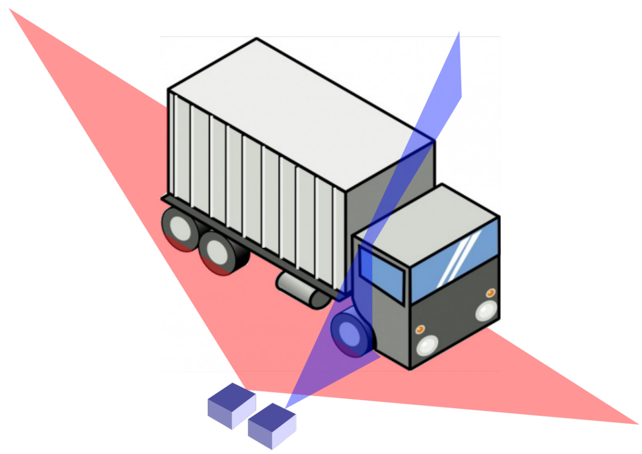

Figure 1: Sketch of the system's working principle. Two laser scanners are deployed. One is measuring in a horizontal plane, pointed at the lower portion of the wheels, and the other is measuring in a vertical plane and used to count vehicles, as well as grouping individual wheels by vehicle.

1. Introduction

Trucks handle the majority of transportation of freight. While rail, water, and air transportation all move large quantities of goods, trucks accounted for 69.6% of all freight transportation in the US in 2013[8]. In a competitive market, companies continually attempts to lower costs. One way of doing this is by lowering fuel consumption of trucks. Upgrading the truck fleets is one way to go about this, but it incurs significant costs. Another way is to use tires with less rolling resistance - or simply have fewer wheels on the truck. To that end, tire manufacturers have released high pressure tires, which can handle a larger weight than conventional tires[4]. This means that a semi-trailer which used to have a triple bogie (three axles, six wheels) can now haul about the same with only a tandem boogie.

While lowering fuel consumption is good for a whole host of reasons, lowering the number of load-bearing wheels results in additional wear of the road surface. Already, a single regular semi-trailer truck wears the road at a level equivalent to 9600 cars[13]. In other words, for every truck driving down a road, 9600 cars can pass, before the same level of damage has been reached. If a truck has fewer tires mounted - as made possible by the new, economical tires mentioned above - the pressure per area from the truck is even higher than usual, and the road damage is naturally worsened.

For this reason, governments are interested in statistics about wheel configuration of trucks. If new trucks put more wear on the roads, measures might have to be taken to construct roads differently, or restrict trucks from driving on certain roads. As discussed in section 2, plenty of sensors exist, which can count the number of axles on passing vehicles. Some can even measure the individual axle weight. However, at present, no commercially available sensor can measure the width of the tires, nor whether certain axles have dual tires mounted.

This paper describes a tire analysis system with the following properties:

1. Tire widths can be measured and duel-tire configurations can be detected. Presently no other system can do this.

2. The system works for vehicles driving naturally on ordinary roads. This is crucial for gathering real-world statistics.

3. Tires belonging to the same vehicle are grouped to provide per-vehicle statistics.

4. The system requires no construction work at the roadside, no cutting into the road surface, and not even that technicians cross the road. This makes deployment easier and safer.

The system is implemented using laser scanners (lidars). The working principle is shown in fig. 1. A horizontal laser scanner is set up in a plane nearly parallel to the road, a few centimeters above the road surface. The aim is to shoot under vehicles, except where tires obstruct the view. By combining multiple scans of tires on passing cars, a robust model of each tire can be obtained. The horizontal main scanner works in concert with a vertical scanner, which is used to detect and count vehicles.

The remainder of this paper is organized as follows: In section 2, any relevant related work is discussed. This is followed by the technical design in section 3, which covers both hardware and software. System performance is evaluated in section 4, and we wrap up the paper with a few concluding remarks in section 5.

2. Related work

Automated traffic counting has been performed for decades, and is at a point where products are generally so well-established that recent research contributions are nearly non-existent. There are four main solutions for automated traffic counting[9]:

1. Pneumatic tubes

2. Infrared sensors

3. Piezo-electric sensors

4. Inductive loops

Tubes and infrared sensors are portable systems, while piezo sensors and inductive loops are placed permanently in slots cut in the road surface.

Tube counting works by mounting an air-filled tube across the road and detecting the rise in pressure whenever wheels hit the tube[6]. These systems can count vehicles accurately, but are unable to weigh vehicles and unable to gauge tire width. Infrared sensing, which is the technology closest to the system described in this paper, works in a way very similar to the pneumatic tubes, except with light beams to replace the physical tubes. In general, systems such as TIRTL (The Infra-Red Traffic Logger)[1] rely on having a device on both sides of the road, so to set it up, the technician must cross the road. This can be an issue on freeways and other places, where it is dangerous for people to go. There are, however, simpler units like the TRAFx[12], which can count vehicles using a single device. Like tubes, IR sensing can count vehicles and axles, and provide speed measurements, but they are not able to measure tires or weigh vehicles.

Inductive loops are just that: Coiled wire embedded in the road, in which a current is induced when a large metal mass passes over. These sensors are often used in signalized intersections to trigger a change of signal. They can count vehicles and do have some classification capabilities, but are not able to count axles or weigh vehicles. Due to their use in signalized intersections, they are the most common technology. Finally, piezo sensors are pressure sensitive sensors placed in a groove in the road[9]. They can count - and in some cases weigh - axles, but are unable to provide a measurement of tire-width. Piezo sensors are used in weigh-in-motion stations.

Lidars are often used in autonomous cars[3, 10, 14] and general traffic surveillance[5, 11], but we have been unable to find others specifically looking at wheels or tires. As with the above approaches, lidar-based systems have their pros and cons: They are portable and easy to setup, and they do not interfere with the road. They also provide many measurements relevant to statistical purposes, but are unable to weigh passing vehicles. Thus, ideally, a lidar-system should be combined with a weigh-in-motion system to give the full spectrum of measurements.

3. Technical design

As briefly mentioned in the introduction, the system consists of a couple of laser scanners, a computer, and obviously software for analysis. While the algorithms are the most interesting, this section also touches on the choice of sensors and the physical setup.

The overall software design (fig. 2) follows the hardware-setup: There is a vehicle detection component attached to the input from the vertical lidar and a tire analysis component attached to the horizontal input. If only counting was desired, the vertical lidar could work on its own, but since the main purpose of this system is measuring tire widths, the horizontal lidar is the most critical component. The vertical lidar has been added to ease the grouping of axles into cars.

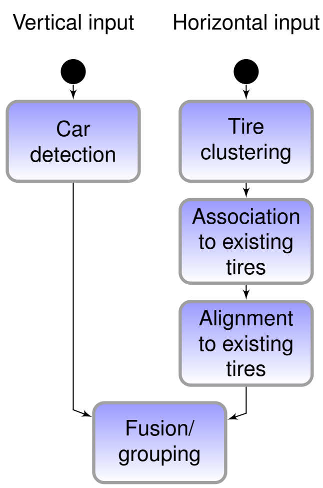

Figure 2: The flow of the system. The two input streams are used for vehicle- and tire-detection, respectively. They are then fused at the end.

Figure 3: A view of the temporary sensor setup at the roadside. On this picture, only the horizontal scanner is mounted. A bike path runs between the sensor and the road.

3.1. Hardware

The horizontal lidar is a Sick LMS511-20100 PRO (from here on just referenced as "LMS511", though other sensors with different specifications exist in that product range). See fig. 3. It was chosen due to a number of reasons:

High power: Black rubber is the worst possible surface to measure against for a lidar, since it returns very little light. Thus, a high powered laser is required. The LMS511 is rated for use at distances up to 80 meters or 40 meters at just 10% object remission.

High distance resolution: Fine measurement accuracy is necessary to measure the relatively small differences in wheel widths. The LMS511 has a systematic error of +/- 25 mm at distances of 1-10 m, which is not ideal, but as good as it gets for a sensor which fulfills the remaining requirements.

High angular resolution: Vehicle wheels can be rather narrow relative to the distance from the sensor. To ensure multiple measurements on each wheel, a high angular resolution is necessary. The LMS511 can have as little as $\frac{1}{6}^{\circ}$ between measurements.

High scanning frequency: As the trucks can be moving quickly past the sensor and multiple readings are necessary to obtain a reliable measurement, a fast scanning frequency is necessary. Depending on the chosen angular resolution, the LMS511 can scan at up to 100 Hz (a full sweep each scan).

Wide field of view: It has a 190°field of view, meaning that it should be possible to see both the front and rear of vehicles using only one sensor. This minimizes cost and also sidesteps any synchronization issues across sensors.

Weatherproofing: The sensor must be usable outside. LMS511 is IP67 rated, which is the highest possible weather resistance rating (dust proof and waterproof for submersion up to 1 m for 30 minutes).

The vertical lidar is a Sick LMS 111-10100. It is not quite as powerful as the 511, but does not have to be since its main purpose is to determine whether or not there is a vehicle in front of it. It is not mainly bouncing beams off of black rubber, and all passing cars are within a few meters of the sensor. This justifies using a smaller and cheaper lidar for this purpose.

The lidars have been mounted on an adjustable stand which allows for precise tuning of the measurement direction. It communicates via ethernet to a Panasonic Toughbook computer, which is also relatively weather resistant.

3.1.1 Sensor output

The output of the horizontal scanner is a side view under a car (if a car is present). A sample scan is seen on fig. 5 and the corresponding real-world view on fig. 4. The lidar is marked with a gray square and arrows showing the scanning direction. The road is sketched out with gray lines. In this particular scenario, the lane closest to the lidar is a through-lane and the topmost lane is a left turn lane for an upcoming traffic signal. Then there is a raised lane division and a single lane going in the opposite direction across that (not shown).

On the plot in fig. 5, a blue line is seen. This line is created by connecting the dot on each measurement across the scan. The sensor is tilted slightly forward, so that a relatively straight blue line is created around the center lane marker at a y-distance of 6000 mm from the sensor. This is

Figure 4: A car passing by the sensor. See fig. 5 above for a view of the resulting scan.

where the scan-plane hits the road. Approximately centered around 0 mm on the x-axis are two very distinct protrusions in the blue line. These are where the laser pulses hit the right hand wheels of the passing car. Looking closely at the protrusions, even the contours of the left-hand wheels are visible as small steps in the protrusions. Toward the right side of the plot, the blue line turns approximately 90°. This is simply due to the curvature of the road and the position of the sensor, which may be leaning a little to the right. Adjusting the tilt of the sensor is an easy job, if done while monitoring the sensor output.

Fig. 6 shows the output from the vertical scanner. The top plot shows the output while no vehicle is present and the lower plot shows the side of a car being scanned. We are looking at the car as if it is driving towards us, not from above as with the horizontal scanner.

3.2. Vehicle detection and counting

This module is tasked with detecting the presence of vehicles. Since the output of the system is supposed to be types of wheels (single or dual) grouped by car, it is necessary to be able to distinguish cars from each other. This module operates exclusively on the output of the vertical sensor. It is based on a simple background subtraction procedure combined with a voting scheme to rule out false positives.

Consider the output from the vertical sensor as a set of points, $\mathbf{V^i}$, where $\mathbf{i}$ denotes the temporal position of the scan. We use a sensor setting with a field-of-view of 170°and an angular resolution of 0.5°per measurement, so we know that

$$|\mathbf{V^i}| = \frac{170}{0.5} + 1 = 341 \tag{1}$$

An initial scan without a vehicle defines the background state, $\mathbf{B} = \mathbf{V^0}$. For each new scan, we define a set of points $\mathbf{V_{mv}} \subset \mathbf{V}$ which have moved more that a certain threshold, α:

$$\mathbf{V^i_{mv}} = \{x^i \in \mathbf{V^i} | x^i - x^{i-1} < \alpha\} \tag{2}$$

x^i are the indivudual measurements, and α is a value in millimeters, which can be set empirically based on what is a large enough motion to rule out measurement noise. We set $\alpha = 500$.

We now introduce a second threshold, $\beta = 10$. This is a count of points. If $|\mathbf{V^i_{mv}}| \geq \beta$, we consider $\mathbf{V^i}$ as describing a passing vehicle. Now, it is possible that a single triggering of this might arise from noise. Thus, this event will cast a vote for a car being detected. For each scan in which $|\mathbf{V^i_{mv}}| \geq \beta$, another vote is cast. Conversely, if $|\mathbf{V^i_{mv}}| < \beta$, a vote is subtracted. The votes are tallied in a variable b, which is clamped to be between 0 and 5. A vehicle is considered detected only if $b = 5$. This ensures that at least 5 consecutive observations must agree that a vehicle is present.

Whenever $|\mathbf{V^i_{mv}}| < \beta$, the background model is updated: $\mathbf{B} = \mathbf{V^i}$.

3.3. Single axle analysis

Parallel to the vehicle detection, a tire analysis is run on the input from the horizontal scanner. As outlined in fig. 2, this consists of 3 stages:

1. Tire clustering
2. Association to existing tires
3. Aligning to existing tires

First step is to **cluster** the input points. This is done using DBSCAN[2]. The advantage of this is that it is not necessary to specify the desired number of clusters - something which would be impossible as we do not know the number of tires present in a scan. Instead, clusters are defined based on the internal distance between points, something that is easy to set for this domain, as points on the same tire are generally close.

Next step is to **associate** each cluster with existing tires, if relevant. For this purpose we use the Kuhn-Munkres algorithm[7]. A cost matrix is populated with the cost for associating a cluster with any given previous tire. The cost is the Euclidean distance between the cluster centroids. n bounding boxes gives an $n \times n$ matrix. The Kuhn-Munkres algorithm gives the possibility of having inequal numbers of inputs and outputs by padding the cost matrix with "infinity" until it is square. With a cost matrix of this type, all clusters would be assigned to tires. This is unfortunate in many cases, because we never want to assign cluster to tires very far away (they will most likely be tires that belong to other axles). To combat this, we add n columns of close-costs. The close-cost is simply a threshold over which we decide that it is better to close a tire than to keep assigning clusters to it.

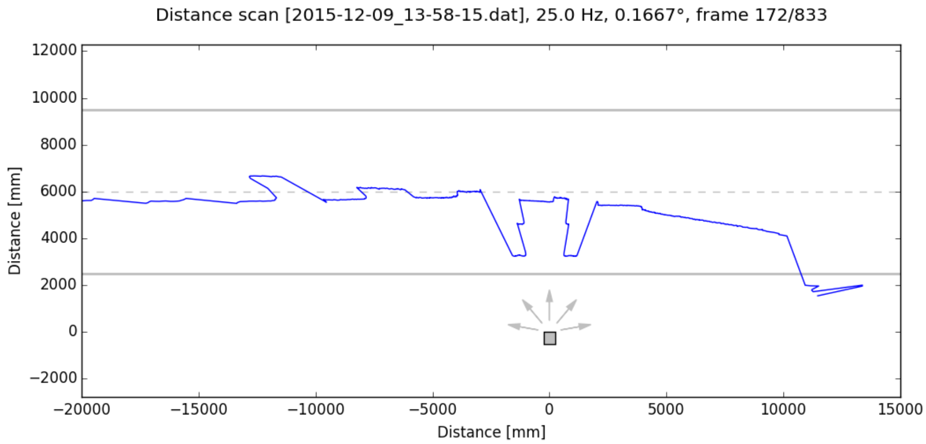

Figure 5: An example measurement which could correspond to the view in fig. 4. Note that this measurement is just an example – it was not taken at the same time as the photograph, but it fits well.

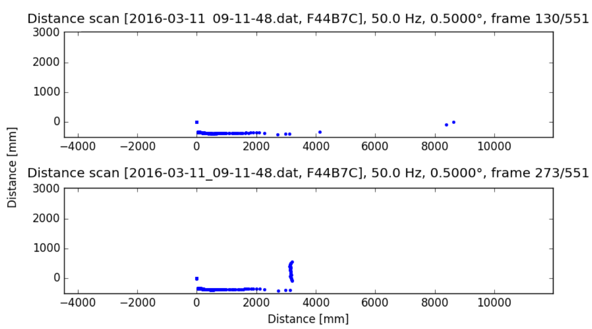

Figure 6: The view from the vertical sensor. This is seen from the front end (cars coming towards the viewer). The sensor is in (0,0). On top, we see the view with no car present. There is a dense cluster of points immediately below and to the right of the sensor, where the scan hits the surface of the bike path next to the road. On the lower plot we see the view with a car passing by.

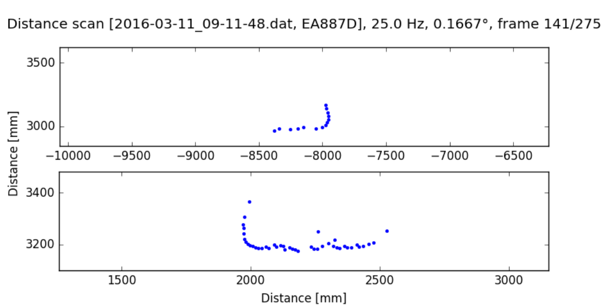

Figure 7: When a tire first enters view, we see only the front and side of it (top plot). As the vehicle moves along, the rear of the tire becomes visible (bottom plot).

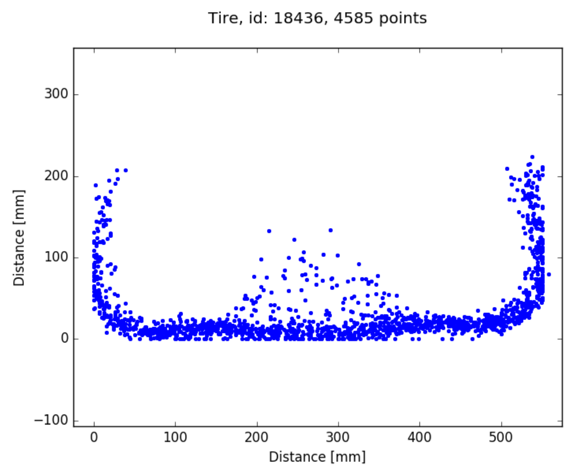

Figure 8: After a while, many scans of the same tire have been combined into a robust model of the tire.

Finally, we **align** the clusters to the tires they have been assigned to. When a tire first comes into view, only the front and side is visible. Later, the rear and the side is visible, see fig. 7. It is our goal to merge multiple of these views into a coherent single model of the tire. That means that the new scan must be aligned properly to the tire model. The points cannot be naively merged, as the width of the observed clusters are not guaranteed to be the same across observations. A simple heuristic is used to determine if a particular scan covers the front edge or rear edge of a tire: If the median of the x-coordinates in a cluster is larger than mean of the x-coordinates, is it the front edge. If the median is smaller than the mean, it is the rear. The points are then shifted to fit the tire and merged with the existing points. Fig. 8 shows the final output of this stage.

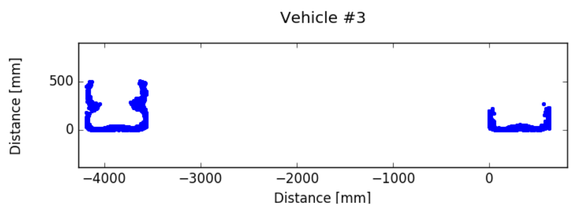

Figure 9: At the final stage, wheels are combined into full vehicles. The dual rear tires are clearly visible in this example.

3.4. Merging the output

Last major stage of the processing is to merge the output of the two processes. This is simple. Whenever the vehicle detector determines that a vehicle is present, a *currentVehicle* is created. At first this vehicle model has no associated wheels. Even though the tire analyzer has been keeping track of the tires for a while, it is not until they pass the sensor (that is, goes from a negative x-coordinate to a positive) that we can know which car they belong to. As soon as that happens for a tire, it is associated with the *currentVehicle*.

When the vehicle detector determines that a car has left its view, *currentVehicle* is moved to a list of vehicles and set to zero. Whenever a vehicle in the list has only closed tires associated with it - tires that will no longer receive updates, because they have been closed by the association step - the vehicle is done, and reported in the output with its number of axles with single-tires and number of axles with dual-tires. The output of this stage is visualized on fig. 9.

4. Evaluation

The desired output from the system is a list of vehicles with information on number of axles and number of tires. We test the system on our own dataset (see below) in multiple stages: First test is to determine if the vehicle count is right. Secondly, we test if the axle count is correct. Thirdly, we test whether the tire count is correct. For each of these measures there will be precision/recall numbers.

4.1. Dataset

The dataset consists of 14 separate captures, with a total of 65 vehicles. They have in total 161 tires on 149 axles. See also table 1. All captures were taken over the course of a single day in the same location (as seen on fig. 3 and 4). In total, the dataset contains 25973 individual scans, with about two-thirds coming from the vertical sensor, which has twice the sample rate as the horizontal scanner.

4.2. Results

The results from running the system on the dataset can be seen in table 2. There are three levels of detail the system can be used at. The first is simply counting (detecting cars). We detect 38 of 65 cars, but with only 2 false positives. That results in a good precision score, but poor recall at 0.58.

One level deeper, the system can report how many axles each vehicle has. We get that count correct on 32 of 65 possible vehicles. However, the system can obviously not record axles on vehicles it did not recognize, so instead of comparing the 32 to 65, it should really be compared to the detected 38 vehicles. That means we count axles correctly on 84.21% of vehicles.

At the finest level of detail, the system reports the exact tire configuration of a vehicle. So for a regular dump truck with single front wheels and dual tires on a tandem bogie in the rear, the system would report 1-2-2. We get the configuration of tires correct on 29 vehicles. That is 90.63% of those where the axles are properly recognized.

Clearly, the performance can be improved. It is interesting to note that it is actually the detection which hampers the performance the most. The background modeling and voting scheme is not strong enough in its current version. On a more positive note, the wheel configuration is rela-

Table 1: Dataset statistics

Number of vehicles:	65
Total number of axles:	149
Total number of tires:	161

Table 2: Evaluation results

Correct detections:	38
Missed vehicles:	27
False detections:	2
Precision:	0.95
Recall:	0.58
Correct number of axles:	32
Correct axles in total:	49.23%
Correct axles on detected:	84.21%
Correct configurations:	29
Correct configurations in total:	44.62%
Correct configurations on detected:	76.32%
Correct configurations on correct axles:	90.63%

Table 3: Example of ground truth vs. output on 4 consecutive misses. Start and end numbers refer to frame numbers.

Ground truth			System output		
Start	End	Config	Start	End	Config
143	151	1-1	145	384	1-1-1-1-1-1-1-1-1-1
223	234	1-2			
286	294	1-1			
374	383	1-1			

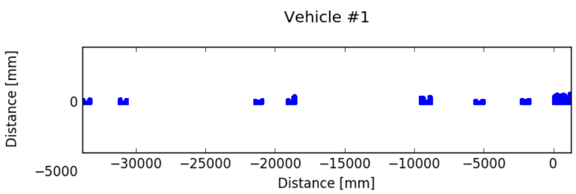

Figure 10: If the detector is unable to detect the end of vehicles, wheels from multiple vehicles are erroneously merged into one. This leads to wrong numbers all the way through the pipeline, even if the wheels are detected correctly.

tively accurate, as long as the subsequent stages perform correctly.

4.3. Qualitative analysis

Multiple issues are worth analyzing further. The lack of detection performance comes down to the background model not being sufficiently robust. A recurring pattern in missed detections is that a vehicle is seen as is enters the vertical field of view, but its end is never detected. A good example is shown in the output reproduced in table 3. Four cars pass the lidar. The first one starts around frame 143, and is picked up fine by the system. It is associated with two tires, and then things go haywire. The vehicle ends on frame 151, but the end is not picked up by the system until frame 383, when it is actually vehicle number 4 which ends. That means that all axles/tires are erroneously absorbed into one giant vehicle. Even though the first vehicle was actually detected correctly initially, is is still considered a miss, since its end was incorrect. The result of such an erroneous merge is shown on fig. 10. Solving this issue would improve system performance a lot, as it is the cause of the majority of errors throughout the pipeline.

Another issue which shows up is the erroneous merge of tires. An example is seen on fig. 11. The correct configuration of this truck is 1-1-2-1. The rear wheels (on the left side of the plot) look about right; the dual tire is well-defined and the rearmost wheel is - if not perfect - fine for our purpose.

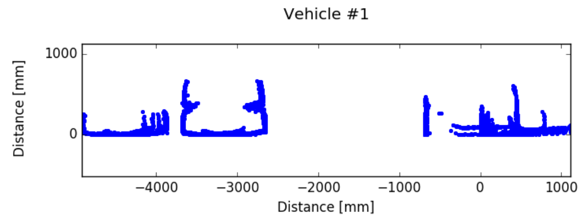

Figure 11: A truck which has been assembled incorrectly due to faulty clustering of wheels.

The front of the truck is a mess. This seems to be mostly caused by rubber flaps hanging below the wheels. This trips up the clustering and alignment, so measures should be taken to prevent this.

5. Concluding remarks

In this paper, we presented and evaluated a lidar-based system for measuring wheel widths of vehicles. The vehicles are driving naturally and no action is required on their part. This system is relevant, since the size of a vehicle's footprint says a lot about the damage in incurs to the road. The system works by having a long range horizontal radar look directly at the tires, and a vertical lidar determine the start and end of vehicles.

The system works as a proof-of-concept, but the performance is still not at the desired level. We detect 58% of vehicles, and this is the cause of most of the problems with the system, as detection issues trickle down through subsequent stages.

Apart from fixing the detection issues, future work includes trying to assemble a 2.5D view of passing vehicles using combined scans from the vertical scanner. These could be used for vehicle classification, as existing classification schemes have issues with specialty vehicles such as farming and construction equipment, and thus fail to handle these.

References

[1] CEOS. TIRTL. http://www.ceos.com.au/index.php/products/tirtl, 2016. 2

[2] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996. 4

[3] E. Galceran, R. M. Eustice, and E. Olson. Toward integrated motion planning and control using potential fields and torque-based steering actuation for autonomous driving. In *Intelligent Vehicles Symposium (IV), 2015 IEEE*, pages 304–309. IEEE, 2015. 2

[4] Goodyear. Goodyear Introduces High Load Truck Tires with Interlaced Strip Technology and 10-tonne Trailer

Tire. http://www.fleetfirst.eu/ff_home_en/news/goodyear-introduces-high-load-truck-tires-with-interlaced-strip-technology.jsp, 2011. 1

[5] M. Litzenberger, B. Kohn, G. Gritsch, N. Donath, C. Posch, N. Belbachir, and H. Garn. Vehicle counting with an embedded traffic data system using an optical transient sensor. In *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 36–40. IEEE, 2007. 2

[6] P. McGowen and M. Sanderson. Accuracy of pneumatic road tube counters. In *Proceedings of the 2011 Western District Annual Meeting, Anchorage, AK, USA*, volume 1013, 2011. 2

[7] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957. 4

[8] L. X. Nguyen, M. Chambers, J. Goworowska, C. Rick, J. Sedor, et al. Freight Facts and Figures 2015. Technical report, 2015. 1

[9] S. L. Skszek. State-of-the-Art Report on Non-Traditional Traffic Counting Methods. Technical report, Arizona Department of Transportation, 2001. 2

[10] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, et al. Stanley: The robot that won the DARPA Grand Challenge. *Journal of field Robotics*, 23(9):661–692, 2006. 2

[11] C. Toth and D. Grejner-Brzezinska. *Vehicle classification from LiDAR data to support traffic flow estimates*. Taylor & Francis, London, 2007. 2

[12] TRAFx. Vehicle Counter. https://www.trafx.net/products.htm#vehicle-counter, 2016. 2

[13] US Government Accountability Office. Excessive Truck Weight: An Expensive Burden We Can No Longer Afford. Technical report, 1994. 1

[14] J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar, and B. Litkouhi. Towards a viable autonomous driving research platform. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 763–770. IEEE, 2013. 2