TRACKING TO IMPROVE DETECTION QUALITY IN LIDAR FOR AUTONOMOUS DRIVING

Jennifer Tang[⋆,†] *Atulya Yellepeddi*[†] *Sefa Demirtas*[†] *Christopher Barber*[†]

[⋆] Massachusetts Institute of Technology
[†] Analog Devices, Inc. (Analog Garage)

ABSTRACT

Enabling Lidar systems to detect objects at very long ranges has the potential to be extremely valuable for autonomous driving applications, but is challenging due to noise. In this work, we leverage information from multiple consecutive frames to improve the detection capabilities of Lidar systems. We develop a mathematical model whose solution gives a low memory and low computation algorithm that detects some fraction of the objects present while keeping the number of false positives small. Performance of the proposed method is characterized using simulations of a realistic Lidar chain.

***Index Terms*—** Lidar, autonomous vehicles, object detection, hypothesis testing

1. INTRODUCTION

Light detection and ranging (Lidar) is likely to be a primary sensing modality on autonomous vehicles [1]. As shown in Figure 1, Lidar works by measuring the time-of-flight of photons to various points in the scene, which are illuminated either by scanning [2] or by capturing return signals from a large part of the scene at once (so-called "Flash Lidar" [3]). A "frame" is defined as a complete 3D image of a scene taken by a Lidar sensor. State-of-the-art sensors have a frame rate of up to 10-30 frames per second.

For so-called direct-detection Lidar systems, the processing scheme for measurements consists of a filter matched to the shape of the outgoing laser pulse. The matched filter output is compared to a threshold, which sets a particular detection and false positive rate (due to noise) at each distance. Noise in Lidar is a combination of thermal noise in the receive chain, shot noise in the detector, and background light in the scene; and can be reasonably modeled as Gaussian. A more detailed description of the Lidar processing chain may be found in [4].

Figure 2 shows sample matched filter outputs for the signal received from objects with 10% reflectivity[1] at different ranges from the sensor[2]. At near ranges, the SNR is high, leading to easy detection, but as the range increases, it becomes challenging to perform detection without correspondingly increasing the false positive rate.

Moreover, objects the size of a car or road debris will only occupy a few voxels (volume pixels) in the Lidar system's field of view when far away. For instance, a typical resolution for a Lidar system is $0.1^o \times 0.1^o$ which would lead to a car 200m away appearing in only about 5 voxels width and 4 voxels height. This means there is a high probability of missing the object entirely, especially if (as is typical), we are highly sensitive to false positives (and thus set a relatively high threshold).

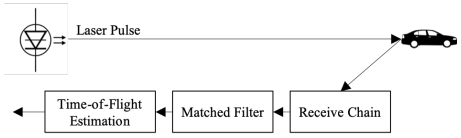
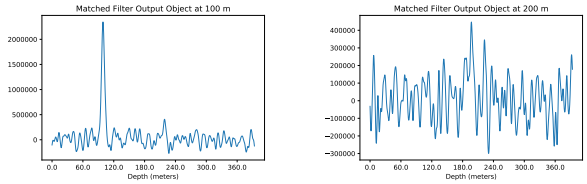


Fig. 1: Principle of operation of Lidar



(a) 100m (b) 200m

Fig. 2: Matched filter outputs for a 10% reflective object at different depths.

Therefore, sophisticated processing is necessary to detect the presence of relatively small objects far away from the autonomous vehicle. In this work, we apply information from multiple consecutive frames to improve the detection limits of a Lidar sensor. In particular, by ensuring that detections must appear in a physically consistent manner (i.e., that physical objects cannot move too fast or accelerate too quickly), we are able to allow more detections from the matched filter output, eliminating a substantial number of false positives.

To simplify the problem, we will work with objects moving at constant and reasonable velocity (i.e., a few hundred miles per hour at the most). To justify the constant velocity assumption, note that for an object accelerating from 0 to 60 miles per hour in 2 seconds, the change in velocity in 3 frames will be ~ 0.2 pixels per frame, which is so small, it can effectively be treated as noise.

1.1. Previous Works

In the literature, Lidar has often been used to recognize vehicles on the road [5, 6, 7, 8, 9]. In this context, several works such as [10, 11], attempt to match objects in consecutive Lidar frames. In [12] motion is used as a means of detection. Deep learning based approaches, such as those of [13, 14, 15, 16, 17], also attempt to correlate object information across frames. Finally, while our work is restricted to

Patent Pending.

The authors would like to acknowledge collaborators in Analog Devices' Autonomous Transport and Safety Business Unit, and in particular Miles Bennett and Ron Kapusta for helpful discussion and simulation.

[1] 10% reflectivity is assumed to be the lowest reflectivity of objects of interest.

[2] All the data in this paper are generated from a circuit-realistic simulation of a particular Lidar receive chain. The specific SNR values obtained at different ranges will, of course, depend on the specific receive chain used.

Lidar only, there is also much literature on combining Lidar data with video data, for instance [18, 19, 20, 21].

The goals of the above works are primarily to create and fine-tune accurate depth maps of objects in order to aid with detection and classification tasks. At the ranges of interest in this work, detailed depth maps are challenging to obtain. Thus, our more modest goal is to use tracking to simply detect the presence of a faraway object, rather than to obtain a depth map, in which respect it differs from the current literature. Additionally, as the low-level processing to perform detection would likely need to run in real-time in embedded processing, we require that it be efficient in both time and memory.

There exists work on detecting trajectories of moving objects using 3D matched filters[22]. However, this method requires preforming intensive computations if the shape and velocity of the objects are unknown[23]. Some of the processing for Lidar used in non-vehicular applications has similar objectives as our work. In [24], to suppress noise, the authors remove small regions of pixels and in [25], the authors apply spatio-temporal interest points [26] to depth images. Another relevant area of Lidar processing which is concerned with noise from long ranges is in atmospheric applications[27, 28, 29, 30]. However, these techniques do not easily translate to the autonomous driving space.

1.2. Summary

Our method looks at consecutive frames and builds trajectories of various objects over time. We can then make a decision if a specific trajectory is a real object or noise using hypothesis testing. We begin with a mathematical model which idealizes this problem and use it to show that basic physical constraints suffice to classify a trajectory as an object or as noise.

2. MATHEMATICAL PROBLEM: THE FIREFLY PROCESS

To formulate the mathematical problem, we describe a "firefly" process. In a bounded 3D box, imagine there are fireflies which move with constant, relatively small velocity. At each time step t, the positions of all the firefly "flashes" are recorded as a pair (t, y), where $y = (y_w, y_h, y_d)$ is the 3D position of the flash. Two types of noise are present in such recordings.

1. The position of the firefly flashes have noise, modeled as Gaussian with mean 0 and variance σ^2 in each dimension.
2. There are ambient flashes which do not correspond to any fireflies, modeled as a Poisson distribution with parameter λ per unit volume. These, of course, correspond to false detections from the matched filter output.

The fireflies model the center of the objects in our Lidar problem. The Gaussian distribution approximates the error in processing a cluster of matched filter detections into a representative center point (we will describe how to do this in 3.1). The Gaussian model is not exact, but gives reasonable results in practice.

After seeing frames $1, ..., n$, the goal is to detect which flashes in frame n are from fireflies (we do not need to estimate current position). A false positive in this context occurs if we falsely determine that a certain sequence of ambient flashes belong to a firefly.

2.1. Estimating Parameters

We begin by ignoring the constraint that fireflies cannot move too fast – we will return to this in the next subsection.

Consider the problem of determining whether a particular sequence of n flashes $(t_1, y_1), ..., (t_n, y_n)$ are all from a firefly or all from noise [3]. Given the model described above, the optimal decision is given by Neyman-Pearson Hypothesis testing, but we do not know the actual positions and velocities to determine the probability of observing the sequence. Thus, we will use the Generalized Likelihood Ratio Test (GLRT).

Let $\hat{\beta}_1$ represent our estimate for velocity and $\hat{\beta}_0$ represent our estimate for the initial position (arbitrarily chosen to occur at time 0). Under the Gaussian model, the best estimates for these parameters are given by

$$\hat{\beta}_1 = \frac{n\sum_{i=1}^n t_i y_i - \sum_{i=0}^n t_i \sum_{i=1}^n y_i}{n\sum_{i=1}^n t_i^2 - \left(\sum_{i=1}^n t_i\right)^2} \tag{1}$$

$$\hat{\beta}_0 = \frac{\sum_{i=1}^n t_i^2 \sum_{i=1}^n y_i - \sum_{i=0}^n t_i \sum_{i=1}^n t_i y_i}{n\sum_{i=1}^n t_i^2 - \left(\sum_{i=1}^n t_i\right)^2} \tag{2}$$

Now, let

$$\beta = \left[\beta_0, \beta_1\right]^\top, \quad Y = \left[y_1, y_2, ..., y_n\right]^\top, \tag{3}$$

and

$$T = \begin{bmatrix} 1 & 1 & ... & 1 \\ t_1 & t_2 & ... & t_n \end{bmatrix}^\top . \tag{4}$$

To apply the GLRT, we need to compute the probability of trajectory $(t_1, y_1), ..., (t_n, y_n)$ occurring given that $\hat{\beta}$ corresponds to the true position and velocity. This is given by

$$p_t(\hat{\beta}) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{|T\hat{\beta} - Y|^2}{2\sigma^2}\right) . \tag{5}$$

If all the flashes are from ambient noise, the probability of each flash occurring at a position in space is uniform and each frame is independent from each other. Let p represent this probability. The GLRT metric $L(T, Y)$ is the log-ratio of $p_t(\hat{\beta})$ to p, i.e.,

$$L(T, Y) = \log \frac{1}{p} \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{|T\hat{\beta} - Y|^2}{2\sigma^2}\right) \tag{6}$$

$$\propto C(n, p, \sigma^2) - |T\hat{\beta} - Y|^2 \tag{7}$$

where $C(n, p, \sigma^2)$ is some constant dependent on the characteristics of the ambient flashes and the variance of the Gaussian noise of the firefly flashes. Thus, to distinguish between a firefly and noise, it is only necessary to compare $|T\hat{\beta} - Y|^2$ to some appropriate constant η^2, i.e.,

$$|T\hat{\beta} - Y|^2 \leq \eta^2 . \tag{8}$$

2.2. 3-frame Acceleration Constraint

For the important special case of $n = 3$, the comparison (8) is equivalent to comparing the magnitude of the acceleration of 3 points to a constant. To understand this, suppose that in three consecutive frames, the positions of the 3 flashes in a sequence are y, $y + v$, and $y + 2v + a$, where v is the velocity measured in pixels per frame and $a/2$ is the acceleration measured in pixels per frame-squared. Then, it is simple to show that $|T\hat{\beta} - Y|^2 = \frac{1}{6}|a|^2$. This is interesting as it shows that the GLRT reduces to a simple physical constraint for this case, which corresponds to the intuition that objects in the real world should not "jitter" too much.

[3] The situation where some of the flashes in a sequence are from fireflies and others are from noise comes up rarely in our algorithm

2.3. False Positives and the Bounded Velocity Constraint

It can be shown that the probability a sequence of ambient noise points will pass the threshold test (8) is upper-bounded asymptotically for some constant C by

$$\mathbb{P}\left[\min_{\beta} |X\beta - Y|^2 < \eta^2\right] \leq C\eta^{n-2} \tag{9}$$

Applying the union bound, we find that if there are at most M flashes in each frame, for large n, the number of false positives is small provided $\eta < \frac{1}{M}$. This limitation on η is quite severe and could lead to poor detection performance.

However, using the additional constraint that fireflies cannot move too fast, we can eliminate sequences where $\frac{y_i - y_{i+1}}{t_i - t_{i+1}} > \delta$ for some chosen δ greater than the maximum velocity of the fireflies. Then, under the Poisson model with parameter λ per unit volume, the number of sequences of length n we have to search (in consecutive frames) is expected to be $V\lambda(\frac{4}{3}\pi\lambda\delta^3)^{n-1}$ where V is the volume of the total space. Then, increasing the sequence length will reduce the expected number of false positives so long as $\eta < \frac{4}{3}\pi\lambda\delta^3$.

3. ALGORITHM

Based on the analysis above, we now present a practical algorithm for Lidar returns. The steps of the proposed algorithm are:

1. Preprocess the Lidar receiver output into clusters
2. Search for sequences of 3 clusters (triples) meeting constraints
3. Extend cluster sequences to longer lengths n to test against (8)

Focusing on 3 clusters helps reduce the complexity of the algorithm while the extension to longer sequences of n gives a stricter criteria for passing the threshold, allowing a trade-off between detection and false positive rates.

3.1. Preprocessing Received Signal

We first apply spatial filters in each dimension (width, height, depth) to the matched filter outputs in each frame – this is to reduce the noise on the assumption that objects span a few voxels in one or more dimensions. Then, we threshold the filtered matched filter outputs to produce detections. At this stage, we keep the threshold low to include many candidate detections, since we will have a later stage which eliminates some.

We then group the points that pass our threshold into clusters by using DBSCAN. Figure 3 gives an illustration of the discovered clusters for a 2D scene for 3 consecutive frames (overlaid) from a bird's-eye view. A 2D scene would correspond to one line scanned by a Lidar system. In practice a 3D map would be produced, but all the methods translate over directly– we have shown a 2D map in this figure for simplicity. We represent the position of each cluster by its center of mass (which is the average position in width, height, and depth).

3.2. Finding Cluster Triples

We define a triple as a sequence of 3 clusters in consecutive frames. Based on the analysis of Section 2, triples must satisfy the following:

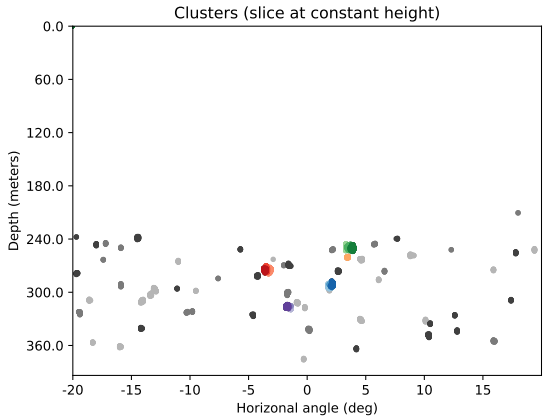


Fig. 3: Clusters found in a 2D scene with 4 objects in 3 consecutive frames overlaid and seen from a bird's-eye view. Lighter colors are clusters from the earlier frames. Clusters that form a triple which passes the velocity and acceleration constraints are shown in color while all other clusters (from noise) are shown in greyscale. Note that all 4 objects satisfy the constraints, whereas none of the noise clusters do and are eliminated.

1. **Velocity Constraint** If cluster A moves to cluster B in the next frame, the difference in position cannot be too large in magnitude.
2. **Acceleration Constraint** Change in velocity of a cluster in 3 consecutive frames must be small as discussed in Section 2.2.

For each cluster in frame i, we find all clusters in frame $i+1$ which meet the velocity constraint. We additionally use a cluster metric which tests for similarities between clusters (based on size, widths, distance, etc.) to filter out clusters. We call the five most similar matches within the velocity constraint the *candidates*.

Once candidates from two frames are processed, we search through triples (A, B, C) where A, B, C are clusters from frames $i, i+1, i+2$ respectively; B is a candidate of A and C is a candidate of B. We store all triples that meet the acceleration constraint and discard the rest. Figure (3) shows an example of triples found that demonstrates that this method already successfully rejects most of the noise.

3.3. Extending Triples

For every triple with clusters from frames $i, i+1, i+2$, we check to see if there is a cluster in frame $i+3$ which is along its trajectory. We calculate if adding the cluster to a trajectory meets the threshold given by (8). If the threshold is met for length n, we store the trajectory in memory. The value of n for a final decision is chosen based on the false positive rate desired.

4. EXPERIMENTS

4.1. Setup

Experiments are performed on synthetic data. Testing on real world data is the next step left for future work. To characterize the algorithm, we generate random scenes with 4 objects each. Each object is a randomly-sized box between 1 to 3 meters in height and width, and has reflectivity 10%. The objects are placed on an invisible horizontal plane representing the road between 200 and 350 meters away from the Lidar and move with a random velocity up to 3 meters per

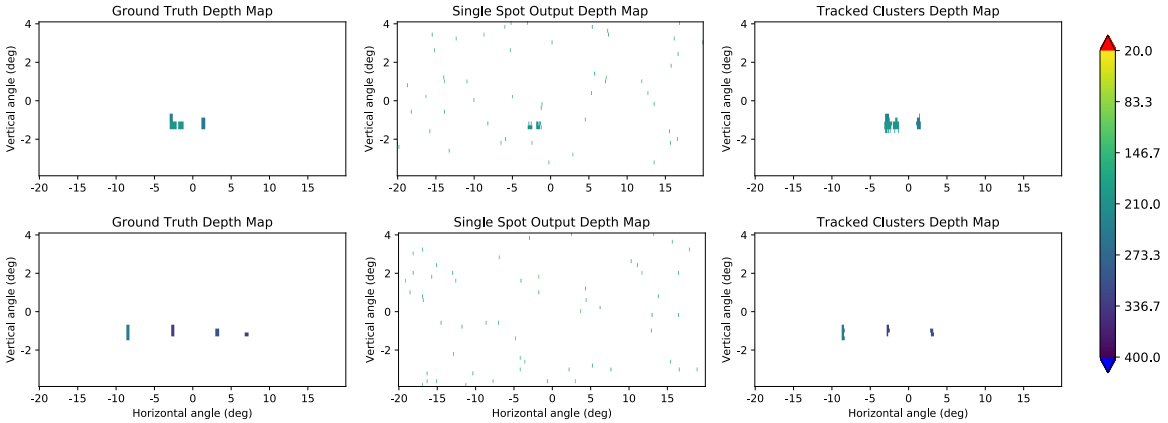


Fig. 4: Examples of one frame viewed from the perspective of the autonomous vehicle (i.e, from in front of the Lidar), where depth is encoded by color. The images on the left show the true location of objects. The images in the middle are the reconstruction of the frame using peaks from the matched filter output with the threshold chosen so that point is 99.9% likely to be an object. The images on the right are the reconstructions of the frame using the Firefly algorithm with sequence length $n = 4$.

	Cluster Size	Length	Detect %	False Pos.
Firefly	-	3	67.7	3.14
	-	4	65.6	0.12
	-	5	61.5	0.04
	-	6	58.8	0.02
	-	7	55.1	0.03
	-	8	51.6	0.02
	-	9	46.9	0.00
	-	10	40.6	0.00
MF Peak 99.9%	1	-	19.1	52.00
	10	-	18.0	14.34
	20	-	13.1	0.00
	50	-	6.7	0.00
MF Peak 99.0%	1	-	36.3	622.90
	10	-	34.8	261.46
	20	-	29.2	2.08
	50	-	17.8	0.00
MF Peak 95.0%	1	-	49.5	2712.41
	10	-	48.4	1731.09
	20	-	43.3	200.61
	50	-	33.9	0.02

Table 1: Performance characteristics of different methods and parameters. Detected percentage is number of correctly detected objects out of total possible objects. False positive rate is number of incorrect detection per frame.

frame in each direction[4]. The field of view of the Lidar is 40 vertical spots by 400 horizontal spots with 0.1 degree resolution. The Lidar receives signals from objects up to 360 meters away. For the Firefly algorithm, we set the maximum velocity to 4.5 meters per frame, 3 meters per frame squared for the acceleration constraint, and 1 per cluster threshold for (8) on cluster sequences longer than 3.

4.2. Results

Our experiments use 16 different scenes with 10 frames each. We consider a cluster sequence "detected" if at least one voxel in the cluster of the last frame in the sequence overlaps with the position of the true object.

For $n = 3$, our algorithm detected 67.7% of objects. There were also a total of 402 false positives found which is about 3.14 false positive per frame[5]. For $n = 4$ (recall this criteria is harder to pass, so both detection and false positive rates will fall), our algorithm detects 65.6% of objects whereas the number of false positives drops dramatically to 0.12 per frame. As we continue to increase n, the detection percentage decreases, but the false positives stays very small, as shown in Table 1. Note that some objects may have such weak signals that no clusters representing it are in the frame. Additionally, occlusion will affect detection rate.

The results are compared to a typical matched filter based detector[4]. The threshold for the matched filter output is set based on probabilities (i.e. confidence that the voxel is an object) and on size of the clusters. Only keeping points which are in the 99.9% confidence level with a minimum cluster size of 1, we can detect an object 19.1% of the time, but we have on average 52.0 false positives in each frame. We can get better detection percentage and lower false positives by using 95.0% confidence level and ignoring any detected clusters smaller than 50 voxels. This gives us 33.9% detection of true objects and negligible false positives, but it is outperformed by the Firefly algorithm. Overall, Table 1 shows that our algorithm is able to achieve better detection percentage with fewer false positives than traditional methods.

5. CONCLUSIONS

We have explored a method for improving Lidar depth images by detecting long range objects by leveraging information between frames and using physical contraints. This improves the range at which a Lidar system can detect an object, which possibly allows for a longer decision horizon for an autonomous vehicle. Future work will involve characterization with real-world data and implementation.

[4]Note that no frame rate is being assumed, which is why the velocity is reported per frame.

[5]The Firefly algorithm has a detection delay of n frames. Reported rates are thus only averaged over frames which have a sufficient number of frames preceding them. The increase in false positive rates between $n = 6$ and 7 in Table 1 is an artefact of this.

6. REFERENCES

[1] E. Ackerman, "Lidar that will make self-driving cars affordable [news]," *IEEE Spectrum*, vol. 53, no. 10, pp. 14–14, October 2016.

[2] R. Halterman and M. Bruch, "Velodyne HDL-64E lidar for unmanned surface vehicle obstacle detection," in *Unmanned Systems Technology XII*, Grant R. Gerhart, Douglas W. Gage, and Charles M. Shoemaker, Eds. International Society for Optics and Photonics, 2010, vol. 7692, pp. 123 – 130, SPIE.

[3] R. Stettner, "Compact 3D flash lidar video cameras and applications," in *Laser Radar Technology and Applications XV*, Monte D. Turner and Gary W. Kamerman, Eds. International Society for Optics and Photonics, 2010, vol. 7684, pp. 39 – 46, SPIE.

[4] R. Richmond and S.C. Cain, "Direct-detection ladar systems," 2010.

[5] K. Takagi, K. Morikawa, T. Ogawa, and M. Saburi, "Road environment recognition using on-vehicle lidar," in *2006 IEEE Intelligent Vehicles Symposium*, June 2006, pp. 120–125.

[6] P. Lindner and G. Wanielik, "3d lidar processing for vehicle safety and environment recognition," in *2009 IEEE Workshop on Computational Intelligence in Vehicles and Vehicular Systems*, March 2009, pp. 66–71.

[7] J. K. Kim, J. W. Kim, J. H. Kim, T. H. Jung, Y. J. Park, Y. H. Ko, and S. Jung, "Experimental studies of autonomous driving of a vehicle on the road using lidar and dgps," in *2015 15th International Conference on Control, Automation and Systems (ICCAS)*, Oct 2015, pp. 1366–1369.

[8] A. N. Catapang and M. Ramos, "Obstacle detection using a 2d lidar system for an autonomous vehicle," in *2016 6th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, Nov 2016, pp. 441–445.

[9] Z. Wang, B. Dai, and H. Fu, "A fast approach for vehicle-like region proposal based on 3d lidar data," in *2015 7th International Conference on Intelligent Human-Machine Systems and Cybernetics*, Aug 2015, vol. 2, pp. 508–512.

[10] J. Choi, S. Ulbrich, B. Lichte, and M. Maurer, "Multi-target tracking using a 3d-lidar sensor for autonomous vehicles," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, Oct 2013, pp. 881–886.

[11] R. Domnguez, E. Onieva, J. Alonso, J. Villagra, and C. Gonzlez, "Lidar based perception solution for autonomous vehicles," in *2011 11th International Conference on Intelligent Systems Design and Applications*, Nov 2011, pp. 790–795.

[12] A. Petrovskaya and S. Thrun, "Model based vehicle detection and tracking for autonomous urban driving," *Autonomous Robots*, vol. 26, no. 2, pp. 123–139, Apr 2009.

[13] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3d lidar using fully convolutional network," *ArXiv*, vol. abs/1608.07916, 2016.

[14] A. Asvadi, L. Garrote, C. Premebida, P. Peixoto, and U.J. Nunes, "Multimodal vehicle detection: fusing 3d-lidar and color camera data," *Pattern Recognition Letters*, vol. 115, pp. 20 – 29, 2018, Multimodal Fusion for Pattern Recognition.

[15] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner, "Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 1355–1361.

[16] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," *CoRR*, vol. abs/1711.06396, 2017.

[17] D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection," *CoRR*, vol. abs/1804.05132, 2018.

[18] A. Rangesh and M.M. Trivedi, "No blind spots: Full-surround multi-object tracking for autonomous vehicles using cameras & lidars," *ArXiv*, vol. abs/1802.08755, 2018.

[19] C. Premebida, G. Monteiro, U. Nunes, and P. Peixoto, "A lidar and vision-based approach for pedestrian and vehicle detection and tracking," in *2007 IEEE Intelligent Transportation Systems Conference*, Sep. 2007, pp. 1044–1049.

[20] M. Mahlisch, R. Schweiger, W. Ritter, and K. Dietmayer, "Sensorfusion using spatio-temporal aligned video and lidar for improved vehicle detection," in *2006 IEEE Intelligent Vehicles Symposium*, June 2006, pp. 424–429.

[21] L. Huang and M. Barth, "Tightly-coupled lidar and computer vision integration for vehicle detection," in *2009 IEEE Intelligent Vehicles Symposium*, June 2009, pp. 604–609.

[22] I. S. Reed, R. M. Gagliardi, and L. B. Stotts, "Optical moving target detection with 3-d matched filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, no. 4, pp. 327–336, July 1988.

[23] A. D. Stocker and P. Jensen, "Algorithms and architectures for implementing large-velocity filter banks," in *Signal and Data Processing of Small Targets 1991*, Oliver E. Drummond, Ed. International Society for Optics and Photonics, 1991, vol. 1481, pp. 140 – 155, SPIE.

[24] S. Mehta and B. Prabhakaran, "Region graph based method for multi-object detection and tracking using depth cameras," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, March 2016, pp. 1–8.

[25] L. Xia and J. K. Aggarwal, "Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, June 2013, pp. 2834–2841.

[26] I. Laptev, "On space-time interest points," *International Journal of Computer Vision*, vol. 64, no. 2, pp. 107–123, Sep 2005.

[27] B. Sun, D. Huang, and Hai-Tao Fang, "Lidar signal denoising using least-squares support vector machine," *IEEE Signal Processing Letters*, vol. 12, no. 2, pp. 101–104, Feb 2005.

[28] H. Fang and D. Huang, "Noise reduction in lidar signal based on discrete wavelet transform," *Optics Communications*, vol. 233, no. 1, pp. 67 – 76, 2004.

[29] X. Zeng, W. Guo, K. Yang, and M. Xia, "Noise reduction and retrieval by modified lidar inversion method combines joint retrieval method and machine learning," *Applied Physics B*, vol. 124, no. 12, pp. 238, Nov 2018.

[30] M. Sarvani, K. Raghunath, and S. Vijaya Bhaskara Rao, "Lidar signal denoising methods- application to narl rayleigh lidar," *Journal of Optics*, vol. 44, no. 2, pp. 164–171, Jun 2015.