LiDAR and Camera Sensor Fusion for Autonomous Applications

Dieter Balemans*, Simon Vanneste†, Jens de Hoog†, Sigfried Mercelis† and Peter Hellinckx†
*Department of Applied Engineering - Electronics-ICT
University of Antwerp
Antwerp, Belgium
Email: dieter.balemans@student.uantwerpen.be
†University of Antwerp - imec
IDLab research group
Department of Applied Engineering - Electronics-ICT
Sint-Pietersvliet 7, 2000 Antwerp, Belgium
Email: {simon.vanneste, jens.dehoog, siegfried.mercelis, peter.hellinckx}@uantwerpen.be

***Abstract*—Perception of the world around is key for autonomous driving applications. To allow better perception in many different scenarios vehicles can rely on camera and LiDAR sensors. Both LiDAR and camera provide different information about the world. However, they provide information about the same features. In this research two fusion methods are proposed to combine camera and LiDAR information to improve what we know about the world around, and increase our confidence in what we detect. The two methods work by proposing a region of interest (ROI) and inferring the properties of the object in that ROI. The output of the system contains fused sensor data alongside extra inferred properties of the objects based on the fused sensor data.**

I. Introduction

Modern autonomous vehicles use a large number of sensors. Besides LiDAR and radar, many also use on-board cameras to sense the environment. Every sensor has its own use case. However, sensors often provide information about the same objects. For example, with data from a LiDAR sensor we can determine a distance between an object and the front of the vehicle. In addition, camera sensors provide more information about what object it is. Knowing this, we can combine multiple sensors to provide more information about the detected objects. This is something that is not possible with only one sensor. Furthermore, multiple sensors can also be used for noise reduction, thus increasing our confidence in our sensor readings. Every sensor has its own advantages and disadvantages. The purpose of sensor fusion is to combine sensors in such a way the advantages of each sensor are combined and the disadvantages are suppressed.

This concept is not new as living organisms have the capability of using multiple senses to learn about the environment. The brain fuses all the information in order to perform a decision task. This concept is called multisensor data fusion. Multisensor data fusion can thus be used in systems of different sensors to enrich the information about the environment, or, if sensors of the same type are used, for noise reduction. [1]

In this work multisensor data fusion will be used on LiDAR and camera data. The main purpose is to optimally use the information of both sensors to detect objects. The methods proposed in this research will detect objects and shall infer properties about these objects. As the Joint Directors of Laboratories (JDL) [2] proposed sensor fusion can be established on different levels:

- **Level zero - data alignment:** Level zero processing involves synchronizing individual data streams and mapping their individual coordinates onto each other.
- **Level one - object refinement:** Level one processing involves object refinement. In this level locational, parametric, and identity information is combined to achieve refined representations of individual objects.
- **Level two - situation refinement:** Level two processing is situation refinement. It develops a description of current relationships among objects and events. It is evident the data from level one can be used for this.
- **Level three - threat refinement:** Level three processing deals with threat refinement. This is often seen as a prediction function. It projects the current situation into the future and draws inferences about threats and opportunities. A non-military example of this is the prediction of collisions in autonomous vehicles.
- **Level four - process refinement:** Level four processing involves process refinement. This means its objective is to refine the fusion process and monitor its performance.

In this research two different fusion system are proposed. The main purpose is to evaluate which system performs best based on the KITTI benchmark. In the next section the related work in the field is briefly discussed. Afterwards a general view of sensor fusion and the main system are given. In section IV and V the two fusion systems are discussed in detail. And in the last sections the evaluation method is discussed along with results and the conclusion.

II. Related work

A lot of research has already been done in the field of LiDAR and Camera fusion, with mixed results. Zhang, Clarke, and Knoll [3] use these sensors for a vehicle detection system.

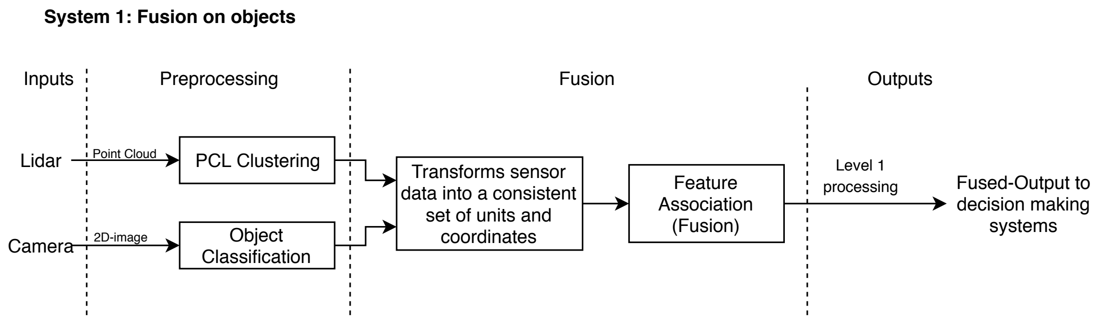

Fig. 1: Global System overview

They developed a system that proposes a hypothesis and verifies this hypothesis using a classification stage. Their classification stage is embodied by a support vector machine (SVM). Their main conclusion was the proposed method has a lower false positive rate then other detection systems.
Liang et al. [4] from Uber Advanced Technologies Group and the University of Toronto developed a 3D object detector using LiDAR and camera data to perform very accurate localization. Their approach consists of continuous convolutions and uses LiDARs and bird eye view images. The results of their research are very promising.
In other research of Xu et al. [5] a more machine learning approach is used. They use a fusion network to fuse the preprocessed image and point cloud data. This is in contrast with this research where a more algorithmic method is proposed. On the market side of things, interesting patents can also be found. In the United States a patent [6] exists for a method for fusing radar/camera object data and LiDAR scan points. The patented system is based on the hypothesis generation/verification principle earlier discussed. In the system the LiDAR data is used in response to an object being detected by the radar or camera. This is a clear example of level 1 processing of the JDL model since the method uses object files to represent the position, orientation and velocity of the detected objects.
This research is focused on the fusion of LiDAR and camera data. Both the LiDAR and camera will be used to infer properties of detected objects. A comparison will be made between a fusion system that uses the LiDAR data as hypothesis and a system that uses the camera data. The proposed systems are mainly situated on level 0 and level 1 of the JDL model. The proposed methods are different from the state of the art methods as they are more focused on analytic data association algorithms. This work tries to improve the state of the art methods in execution time and detection accuracy.

In the next section we will firstly dive deeper into general sensor fusion concepts. Afterward the two fusion variants are discussed.

III. General Sensor Fusion

The main purpose of sensor fusion is to combine the advantages of different sensors, and suppress their disadvantages. The ultimate goal is to achieve a higher level of confidence in the detects objects or features. As said earlier we can divide sensor fusion in different levels. This research is mainly focused on level 0 and level 1. Thus the data alignment and object refinement level.
In the proposed methods data alignment is accomplished by two major components: a time synchronization block and a transformation block. The time synchronization block makes sure the individual data streams are synchronized for the fusion algorithm itself. This is accomplished by a node that listens to each sensor stream and outputs a combination of the sensor streams based on matching time stamps. In order for this time synchronization to be very accurate each individual sensor needs to run on a shared clock. The transformation block will map certain features from one sensor to the coordinate system of the other. In this research a mapping between the LiDAR point cloud space to the camera' image space is used mainly.

The proposed system can be seen in Fig. 1 and consists of two major stages. A preprocessing stage and a fusion stage. The preprocessing stage is where the individual sensor streams are processed to detect individual objects. Thus a feature based fusion is implemented. The preprocessing stage is developed using off the shelve components. More details can be found in the following sections IV and V. After the preprocessing step the actual fusion processing takes place. In this stage the individual sensor features are combined and associated with each other. Using these combined features we can infer the objects properties. In the current state of the system a 2D image bounding box and a 3D LiDAR point cloud bounding box is inferred. These bounding boxes alongside the fused features are the output of the system. This output is an object refined output, thus a level 1 output. For the fusion step two different approaches are proposed: a camera based and a LiDAR based approach.
Furthermore is it worth noting this research was developed using the Robotic Operating System (ROS). The ROS middleware is used for communication between the sensors and program nodes.

IV. Fusion 1: camera hypothesis

The first approach is focused around the camera features. In this method the camera preprocessing is the essential part,

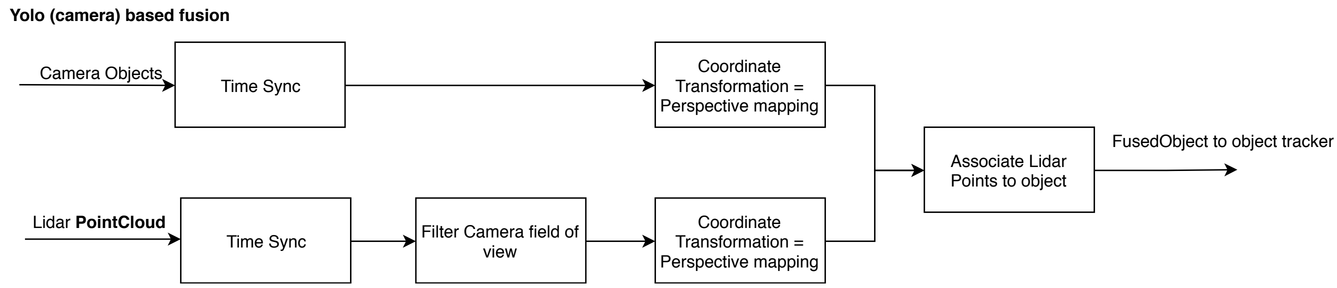

Fig. 2: Fusion 1: camera hypothesis

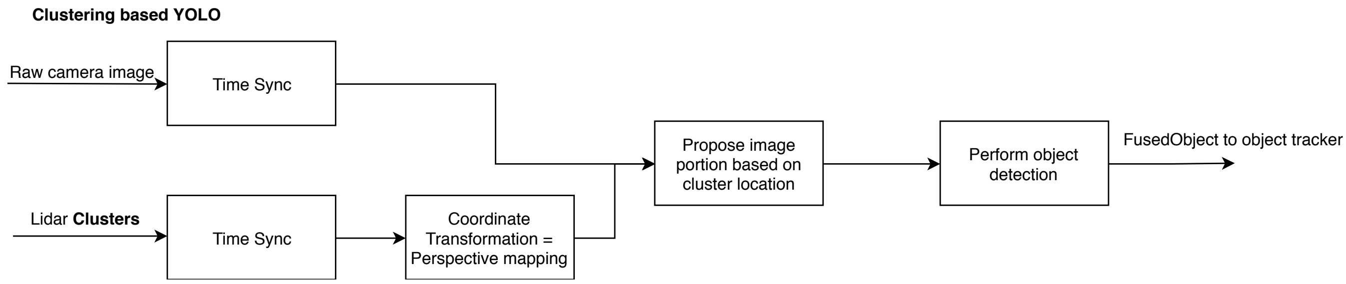

Fig. 3: Fusion 2: LiDAR based hypothesis

which will first detected objects in the image space. These detected objects will serve as a hypothesis for the fusion step. As seen in Fig. 2 the inputs of the fusion method are detected camera objects and a raw point cloud. In order to be able to process the point cloud data fast enough a down-sampling filter is used.

A. Preprocessing

The preprocessing of the first fusion system consists of an image object detection algorithm. We opted for a machine learning approach to detect objects. Therefore the famous You Only Look Once (YOLO) model is be used. YOLO is a real-time object detection model. The main advantage of the YOLO model is that the model only needs to be applied once to the image. This makes it significantly faster than other prior object detection methods [7]. There are pretrained versions of YOLO available. These pretrained models are trained on the COCO data set [8]. The version used in this research is trained on 80 objects from the COCO data set. The most important objects this network can detect are: persons, cars, traffic lights, trucks, trains, and buses. The output of the camera object detector are the classes of the objects alongside a bounding box representing the location of the object on the image.

B. Fusion

After preprocessing the actual fusion step takes place. In this step the detected camera objects are used as hypothesis to find associated LiDAR data. The detected objects bounding boxes are used to predict a region of interest (ROI) in the image space. Subsequently the relevant LiDAR points are transformed to the image space. This transformation consist of projecting the LiDAR points onto the image plane. For this projection to be correct it is crucial the position of the camera and LiDAR are precisely determined. The distance between the center of the LiDAR and the center of the camera determines the image plane coordinates. The Center of Projection (COP) is the location of the LiDAR center relative to the center of the camera. In Fig. 4 an example projection is drawn. Here the COP is in the origin and the image plane is placed at x = a. This means the camera is placed at a distance of a in front of the LiDAR. Furthermore the camera and LiDAR are placed in a perfect line behind each other on the same height. Hence the COP is at the origin.
To map a 3D point (x_1, y_1, z_1) to the image plane x = a:

$$k = \frac{a - x_1}{x_1 - copx}$$

$$x_1' = x_1 + k * (x_1 - copx) = a$$

$$y_1' = y_1 + k * (y_1 - copy)$$

$$z_1' = z_1 + k * (z_1 - copz)$$

Because the axis of the LiDAR and camera are different, the projected point' coordinates on the image space become:

$$ImageX = y_1'$$

$$ImageY = z_1'$$

With the points and the detected object in the same coordinate space a simple data association can be applied. It is important to note that one component of the LiDAR data is lost after the projection. This can introduce an error in the detection For example a car is being detected behind a street pole. The LiDAR points that falling onto the pole are seen a points of the car. Thus these points need to be

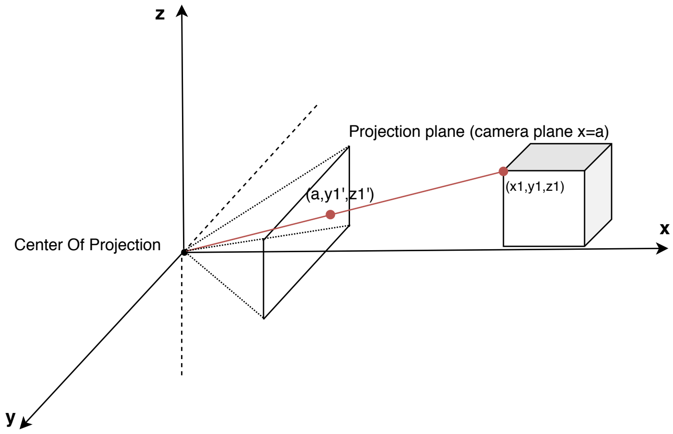

Fig. 4: LiDAR point projection: point (x1,y1,z1) projected onto the camera plane x = a

filtered. Therefore the final step of the fusion step is applying a clustering algorithm on the associated points to filter the error. This clustering will filter all points in front and behind the car as they will be left out of the object' cluster.
Finally we have fused objects of both LiDAR and camera data. These fused objects contain a class, a bounding box location on the 2D camera image, and a point cloud from the LiDAR data. Using the data of these fused object further properties of the objects can be inferred. For example in this method also a 3D bounding box of the object is being determined. This 3D bounding box is calculated based on principle component analysis (PCA) using the object' LiDAR point cloud [9]. First the eigenvectors are determined using PCA. Afterward the object' point cloud is transformed to the origin such that the eigenvectors correspond to the axes of the space. Next the minimum and maximum point are calculated, these points are used to determine the box width, height, and depth. Finally, the quaternion is calculated using the eigenvectors (which determines how the final box gets rotated), and the transform to put the box in correct location is calculated.

The output of the system are the fused objects with their properties. These objects can be made accessible for higher levels of autonomous driving software. In this research specifically the fused objects are outputted to Robot Operating System (ROS) topics.

V. Fusion 2: LiDAR hypothesis

In contrast to the first fusion system, this system uses the LiDAR input to propose the hypothesis. This method allows for more exact fusion and determining objects in 3D. In the following subsections the preprocessing and the fusion step are discussed in detail.

A. Preprocessing

In the preprocessing phase the LiDAR data will be used to detect objects. These objects will be extracted from the point cloud using an euclidean clustering algorithm. This is possible because only points of the same object are close enough to each other to form a cluster. The clustering algorithm will construct a Kd-search tree [10] of the point cloud and use a threshold to find all points close enough to each other. It must be noted that the raw point cloud from the LiDAR sensor is not directly ready to be used for the clustering algorithm. First the point cloud is down sampled using a leaf size of 0.5 and a voxelgrid [11]. Afterwards a ground plane removal algorithm will find and filter out the ground plane points. This is done using the RANSAC algorithm [12]. Lastly all LiDAR points that cannot be projected on the image are filtered out. This leaves a down sampled point cloud with only relevant possible object points which the clustering algorithm will use to propose objects. It is important to note that many of these used algorithms have critical parameters. For example the clustering algorithm uses a threshold to determine if points are close enough to each other. One can agree that this value determines the quality of the output. The optimal threshold value can vary from situation to situation. As described in the future work section these critical parameters should be monitored and controlled during the fusion process.
The output of the LiDAR clusterer are the clusters themselves.

B. Fusion

In the fusion step of system two the LiDAR clusters are first projected onto the image space. This is done using the same method used in fusion system one. Subsequently the projected clusters are used to propose a ROI of interest in the image space, in which an image object detection algorithm will validate the proposition by classifying the objects. This ROI is proposed by finding the minimum and maximum X and Y value of the projected cluster images plus an predetermined offset parameter to increase the detection changes. Again, it must be noted this offset parameter has a notable impact on the correct detection rate. If this parameter is to big, the ROI is to large and multiple objects may be detected as one. On the other hand if this parameters is to small, smaller objects may not be detected correctly.
After the the proposition is verified the bounding box get updated based on the classification. Now we have fused objects with both LiDAR point cloud data, and image data. As in system one the same properties (class, 2D and, 3D bounding box) are inferred using this data. The output of the system is the same as in fusion system one.

VI. Evaluation and benchmark

The test setup consists of the Velodyne VLP16 LiDAR and a StereoLabs Zed 3D camera. The LiDAR has a 360 degree field of view and 16 channels over 30 degrees of vertical field of view. This means the LiDAR outputs a 3D point cloud of the world around. The camera will be used to output a 1344x376 pixels image. Only one camera channel is used. These sensors will be connected to a system running a NVIDIA GTX 1070 GPU and Intel Core i5 4570 CPU which will process and fuse the data. As mentioned earlier the Robotic Operating System

Fig. 5: Example detection's: (upper image) example detection of fusion system 1 (lower image) example detection of fusion system 2. These detection where made on the test setup described in section VI. The data set used is the KITTI 0059 raw data set.

(ROS) will be used to orchestrate the communication between the nodes of the system. Because there are only two sensors used they are both placed on top of the vehicle facing forward. The position of these sensors is critical in the fusion process to estimate the position of detected objects.

A. *Carla Simulator*

In order to develop the systems a virtual environment is chosen. Therefore, the Carla [13] simulator is used. This simulation will simulate both sensors with equal properties as described above. Furthermore a ROS-bridge will be used to allow the execution of the simulation without the need to rewrite the actual fusion algorithms. As we have no direct way of gathering ground truth data the simulation is not suited for evaluation of the systems.

B. *KITTI benchmark*

To evaluate the performance of the systems the KITTI raw data set is used [14]. This data set is composed by the KITTI team and consists of camera, LiDAR, GPS, and IMU data. As Geiger, Lenz, and Urtasun [15] describe the setup in their paper. Although their equipment is different than our test setup, our system can easily cope with the different input. The data set was launched to be a real-world computer vision benchmark. The benchmark is divided into different categories. The benchmark used in this research is the 2D- and 3D-object detection benchmark. Because our system did not allow to execute the original KITTI benchmark, we use a self made benchmark that follows the general ideas of the KITTI benchmark.

The data used to benchmark our systems is the KITTI raw data sets. These data sets are small fragments of the KITTI's team recordings and the detected objects are labeled. For each frame of the data sets fragments a label file is available which can be used as ground truth to evaluate our systems. The labels where collected by multiple stages of human annotations and can therefore be considered very accurate. The label files contain one line of text for each objects visible in the associated frame. Each line describes the objects properties. The properties used in this research to evaluate are the class name, 2D bounding box, and 3D bounding box information.

In order to evaluate, our system outputs a similar label file for each frame with the same properties per object. An evaluation script is used that compares the label files frame by frame. The script will compare the detected objects with the ground truth objects, and determine whether or not a detection was successful. The criteria for successful detection are of course the same object class along with a minimum of fifty percent

overlapping bounding boxes. When these criteria are met a detected object in a frame is considered correct.
To effectively evaluate our systems the precision and recall are calculated for each raw benchmark data set. These values tell us how the systems performs on that specific benchmark. To calculate the values the evaluation script collects the correctly detected objects (true positives), the falsely detected objects (false positives), and the not detected objects (false negatives). The precision and recall are then given by:

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

$$recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

VII. Results

The evaluation of the systems is done using the method described in section VI-B. The parameters that will be compared are:

- **Execution time**: The time it takes to process one frame in milliseconds. Lower is better.
- **Precision**: How many of the detected objects are correctly detected objects. Higher is better.
- **Recall**: How many of the total detectable objects are detected. Higher is better.

In the next tables you can find the results of the benchmarks.
KITTI data set 0022:

Method	Execution time	Precision	Recall
Fusion 1 2D	0,0805958 s	72,46%	90,98%
Fusion 1 3D	0,0805958 s	3,87%	4,85%
Fusion 2 2D	0,120369 s	84,37%	37,43%
Fusion 2 3D	0,120369 s	13,77%	5,64%

KITTI data set 0059:

Method	Execution time	Precision	Recall
Fusion 1 2D	0,079720 s	94,08%	76,88%
Fusion 1 3D	0,079720 s	19,88%	15,25%
Fusion 2 2D	0,156222 s	89,28%	25,84%
Fusion 2 3D	0,156222 s	40,89%	9,73%

VIII. Conclusion

This research presents two fusion systems which predict a 2D image bounding box and 3D LiDAR point cloud bounding box. The main purpose is to compare the performance of a LiDAR based fusion and a camera based fusion. Based on the results above one can conclude that the first fusion system, based on image ROIs, performs better. It has consistent higher recall and precision for the 2D benchmark. The execution time of the first system is also considerably lower. The reason for this is because the first system only needs to process the image information once. This is in contrast with the seconds system where for each cluster a small part of the image information is processed. Processing these small image parts take less time, nevertheless the total processing time of all clusters together is longer. The current clustering algorithm also proposes irrelevant clusters. An example of this are parts of buildings. These parts will not count as correct detected objects, but do require processing time. The main reason the second fusion system performs worse in terms of precision and recall is because of the euclidean clustering algorithm. This algorithm uses parameters that need to be specifically tuned to each situation. Therefore the resulting clusters are not always optimal. This means when for example two cars are parked close to each other, they can be detected as one cluster because the threshold parameters was too high in that situation. Comparing the 3D detection results one can clearly notice the performance is remarkably lower than the 2D detection's. The reason for this is because the 3D bounding box calculation method is not optimal. However the 3D precision of the second fusion system is notably higher. This is because the method is based on clustering the point cloud, which lead to more accurate 3D detection's.

IX. Future work

The proposed systems in this research are situated on level zero and level one of the JDL model. This includes currently only the data alignment and object refinement stages. It is future work to further extend the level one outputs to also track the detected objects in time, and in doing so infer more properties about the objects. Using optimal state estimation algorithms, for example Kalman filters, it will be possible to predict an objects heading and velocity. Based on that information a threat level (level three output) can be determined for each object.
Furthermore, as described earlier, the fusion process can also be optimized during execution. This can be done by monitoring and updating system parameters like thresholds. This processing is the fourth level of the JDL model.
To further improve the results of the systems, especially the second proposed fusion system, other LiDAR feature extraction algorithms can be explored. Currently a euclidean clustering method is used which is difficult to tune to each situation and therefore the result are not outstanding. It would also be possible in the future to propose a third fusion system that consists of a mix of system one an system two. Certain situation allow the LiDAR to best propose the ROI. In other situations the camera may be better. It is future work to provide a solution for this.

References

[1] D. P. Mandic, D. Obradovic, A. Kuh, T. Adali, U. Trutschel, M. Golz, P. De Wilde, J. Barria, A. Constantinides, and J. Chambers, "Data fusion for modern engineering applications: An overview," *Artificial Neural Networks: Formal Models and Their Applications*, pp. 715–721, 2005.

[2] F. E. White, "JDL, Data Fusion Lexicon," *Technical Panel For C3*, vol. 15, no. 0704, p. 15, 1991.

[3] F. Zhang, D. Clarke, and A. Knoll, "Vehicle detection based on LiDAR and camera fusion," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1620–1625, IEEE, oct 2014.

[4] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep Continuous Fusion for Multi-sensor 3D Object Detection," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11220 LNCS, pp. 663–678, 2018.

[5] D. Xu, D. Anguelov, and A. Jain, "PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation," tech. rep.

[6] Z. Shuqing, "System and method for fusing radar/camera object data and LiDAR scan points," feb 2016. US Patent 9,476,983.

[7] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv*, 2018.

[8] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014.

[9] "Point Cloud Library find minimum oriented bounding box of point cloud (c++ and pcl)." http://codextechnicanum.blogspot.com/2015/04/find-minimum-oriented-bounding-box-of.html.

[10] "Point Cloud Library module kdtree." http://docs.pointclouds.org/trunk/group__kdtree.html.

[11] "Point Cloud Library pcl::voxelgrid¡ pointt ¿ class template reference." http://docs.pointclouds.org/1.8.1/classpcl_1_1_voxel_grid.html.

[12] M. a. Fischler and R. C. Bolles, "Random Sample Paradigm for Model Consensus: A Apphcatlons to Image Fitting with Analysis and Automated Cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[13] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16, 2017.

[14] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[15] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[16] D. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.

[17] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," 2013.

[18] A. Geiger, F. Moosmann, O. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *2012 IEEE International Conference on Robotics and Automation*, pp. 3936–3943, IEEE, may 2012.

[19] D. Gohring, M. Wang, M. Schnurmacher, and T. Ganjineh, "Radar/Lidar sensor fusion for car-following on highways," in *The 5th International Conference on Automation, Robotics and Applications*, pp. 407–412, IEEE, dec 2011.

[20] W. Wang, X.-H. Huang, and M. Wang, "Out-of-Sequence Measurement Algorithm Based on Gaussian Particle Filter," *Information Technology Journal*, vol. 9, pp. 942–948, may 2010.

[21] N. Kaempchen and K. C. J. Dietmayer, "Data Synchronization Strategies for Multi-sensor Fusion," 2003.

[22] J. P. Hwang, S. E. Cho, K. J. Ryu, S. Park, and E. Kim, "Multi-Classifier Based LIDAR and Camera Fusion," in *2007 IEEE Intelligent Transportation Systems Conference*, pp. 467–472, IEEE, sep 2007.

[23] J. Dong, D. Zhuang, Y. Huang, and J. Fu, "Advances in Multi-Sensor Data Fusion: Algorithms and Applications," *Sensors*, vol. 9, pp. 7771–7784, sep 2009.

[24] H. Cho, Y.-W. Seo, B. V. Kumar, and R. R. Rajkumar, "A multi-sensor fusion system for moving object detection and tracking in urban driving environments," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 9, pp. 1836–1843, IEEE, may 2014.

[25] D. Balemans, J. D. Hoog, P. Hellinckx, and S. Vanneste, "A Survey On Multisensor Data Fusion for Autonomous Applications," p. 3, 2018.

[26] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *2011 IEEE International Conference on Robotics and Automation*, pp. 1–4, May 2011.