IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 29, 2018, accepted April 17, 2018, date of publication May 21, 2018, date of current version June 20, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2839561

Vanets Meet Autonomous Vehicles: Multimodal Surrounding Recognition Using Manifold Alignment

YASSINE MAALEJ[1], (Student Member, IEEE), SAMEH SOROUR[1], (Senior Member, IEEE), AHMED ABDEL-RAHIM[2], AND MOHSEN GUIZANI[1], (Fellow, IEEE)

[1]Department of Electrical and Computer Engineering, University of Idaho, Moscow, ID 83844, USA
[2]Department of Civil Engineering and the National Institute for Advanced Transportation Technologies, University of Idaho, Moscow, ID 83844, USA

Corresponding author: Sameh Sorour (samehsorour@uidaho.edu)

ABSTRACT In the past two years, calls for developing synergistic links between the two worlds of vehicular ad-hoc networks (VANETs) and autonomous vehicles have significantly gone up to achieve further on-road safety and benefits for end-users. In this paper, we present our vision to create such a beneficial link by designing a multimodal scheme for object detection, recognition, and mapping based on the fusion of stereo camera frames, point cloud Velodyne LIDAR scans, and vehicle-to-vehicle (V2V) basic safety messages (BSMs) exchanges using VANET protocols. Exploiting the high similarities in the underlying manifold properties of the three data sets, and their high neighborhood correlation, the proposed scheme employs semi-supervised manifold alignment to merge the key features of rich texture descriptions of objects from 2-D images, depth and distance between objects provided by 3-D point cloud, and the awareness of self-declared vehicles from BSMs' 3-D information including the ones not seen by camera and LIDAR. The proposed scheme is applied to create joint pixel-to-point-cloud and pixel-to-V2V correspondences of objects in frames from the KITTI Vision Benchmark Suite, using a semi-supervised manifold alignment, to achieve camera-LIDAR and camera-V2V mapping of their recognized objects. We present the alignment accuracy results over two different driving sequences and show the additional acquired knowledge of objects from the various input modalities. We also study the effect of the number of neighbors employed in the alignment process on the alignment accuracy. With proper choice of parameters, the testing of our proposed scheme over two entire driving sequences exhibits 100% accuracy in the majority of cases, 74%–92% and 50%–72% average alignment accuracy for vehicles and pedestrians and up to 150% additional object recognition of the testing vehicle's surrounding.

INDEX TERMS VANETs, autonomous vehicles, LIDAR, manifold alignment, KITTI.

I. INTRODUCTION

A. MOTIVATION

For years, researchers on Vehicular Ad-hoc Networks (VANETs) and Autonomous vehicles presented various solutions for vehicle safety and automation, respectively. Yet, the developed works in these two areas have been mostly conducted in their own separate worlds, and barely affected one-another despite the obvious relationships. The US National Science Foundation and US Department of Transportation have expressed tremendous need and importance to relate these two worlds together in several of their calls for research proposal in 2017 [2]. They clearly emphasized on the major importance of integration and fusion of data from various input modes in order to create a deeper understanding of vehicle object surroundings. Precisely, enriched 3D scene reconstruction by different input technologies and deep learning techniques are of a paramount importance to develop autonomous vehicle systems that can perform effectively and safely on roads. These directions are strongly supported by the multiple accidents and traffic light violations made by autonomous vehicle prototypes from top players in the market (e.g., Tesla, Uber) [3]–[5], incidents that could have easily been mitigated if communications among vehicles and with the road infrastructure (e.g. Traffic signals, traffic signs, etc) would have been considered. Tesla's Autopilot fatal crash [5] was caused by the failure to recognize the white side of the

tractor trailer against a brightly lit sky. The autopilot's solely reliance on camera and lack of V2V communication information and LIDAR is the main cause of the crash. The crash encountered by UBER's autonomous car [3] resulted from the human-driven vehicle's failure to yield. The integration of V2V messages in UBER's pilot would allow the autonomous vehicle to know the intention of surrounding cars, along with the information offered from LIDAR and cameras. To respond to this crucial need, we propose in this paper to enrich the learning of the 3D vehicle surroundings using multi-modal inputs, namely LIDAR scans, camera frames, V2V-conveyed basic safety messages (BSMs). Despite the curses of the data representation and dimensionality, learning the correspondence between the same objects from different data inputs is a necessary task to both improve (i.e., get more accurate knowledge about detected items in each data set) and enrich (i.e., combine undetected items from all data set into one global picture) the understanding of autonomous vehicles about their surroundings, thus allowing them to make safer and more accurate driving decisions. Incorporating objects detected from these three sources into one scheme requires a mapping process between objects that possess similar underlying structure (i.e. they all represent many common features of a same environment) and neighborhood correlations (i.e., neighboring items in one data sets should be still neighbors in the others). Given this properties, we propose to cast this mapping problem as a manifold alignment problem [6]. Indeed, manifold alignment is a dimensionality reduction based mapping tool between data sets exhibiting similar underlying structure and neighborhood correlations, which makes it a perfect fit for the data sets of interest.

B. RELATED WORK

Analyzing camera frames using various techniques has been one of the mostly used techniques for autonomous vehicle surrounding recognition. One suggested technique is semantic segmentation, which labels each pixel in an image with the category of the belonging objects. To determine that a certain pixel belongs to a vehicle, a person or to any other class of objects, a contextual window that is wide enough is defined to show the surrounding of the pixel and consequently make an informed decision of the pixel's object class. Techniques based on Markov Random Fields (MRF), Conditional Random Fields (CRF) and many graphical models are presented in [7]–[9] to guarantee the consistency of pixel labeling in the context of the overall image. In addition, the authors in [10]–[12] developed various methods for image pre-segmentation into super-pixels, which are used to extract the categories and features from both individual segments and combinations of neighboring segments.

Alternatively, the authors in [13] attempted to create 3D reconstruction of dynamic scenes by achieving a long-range spatio-temporal regularization in semantic video segmentation, due to the fact that both the camera and the scene are in motion. The developed idea is to integrate deep convolutional neural networks (CNNs) and CRF to perform sharp pixel-level boundaries of objects. The proposed solution minimized the distances between the features associated with corresponding points in the scene, and consequently optimized the feature space that is used by the dense CRF. To this end, deep learning has shown the best performance in inferring objects from previously untrained scenes. In [14], the segmentation of the input images was achieved by representing the dynamic scene as a collection of rigidly moving planes and jointly recovering the geometry/3D motion when oversegmenting the scene. The developed piece-wise rigid scene is intended to represent real world scenes with independent object motions rather than pixel-based representations like partially used in [15].

Joseph *et al.* [16] developed a general purpose object detection system characterized by a resolution classifier and the usage of a two fully connected networks. These two networks are built on top of a 24-layer convolutional network, followed by two fully connected layers. Additionally, a unified mutiscale deep CNN for real-time object detection is developed in [17] with many sub-network detectors and multiple output layers for multiple object class recognition.

Another widely-used sensing technique for 3D environment reconstruction is LIDAR scanners. 3D LIDAR-generated point clouds were already used in distance ranging, obstacle detection and avoidance, path planning, and were thus imported to autonomous driving systems. 2D convolutional neural networks (CNNs) [18] have been designed for processing and recognizing objects from 3D LIDAR point cloud. However, this solution is not considered optimal since it requires a model to recover the original geometric relationships. Vote3Deep is developed in [19] for fast point cloud object detection using 3D CNN, in order to keep the key power of LIDAR as distance and objects 3D shapes and depth detection. The KITTI Vision Benchmark Suite [20] offers raw LIDAR and labeled objects from point cloud.

As clarified above, most autonomous driving systems rely on LIDAR, stereo cameras or radar sensors to achieve object detection, and scene flow estimation of objects on roads. Despite the great advancement in both technologies, they still are incapable of detecting hidden elements, such as hidden vehicles or pedestrians. Camera based systems may also fail in detecting geometrically line-of-sight entities due to limited visibility conditions, such as bad weather conditions (e.g., very bright sun, fog, heavy rain/snow) and close colours to surrounding nature (e.g., the cause of the prototype accident in [5]). Finally, the camera and LIDAR techniques fail in detecting road/traffic conditions, (e.g., red traffic lights, change in speed limit, etc), which may cause traffic violation (e.g., [4]) and even fatal problems. As aforementioned, all such problems may be resolved if these sensing technologies are complimented with actual in-flow information from both close vehicles and traffic infrastructure through VANETs. Google's self driving car project called WAYMO [21] is collaborating with Intel in order to create powerful chips responsible for the processing and fusion of data retrieved from radars, cameras, and LIDARs. Researchers in [22]

presented an approach to geometrically align points from LIDAR scans to points captured from a 360 degree camera.

In addition, VANETs offers various types of V2V and vehicle-to-infrastructure (V2I) safety messages on 7 control channels operating over a dedicated 75 MHz spectrum band around 5.9 GHz [23], [24]. Our aim in this paper is to present an augmented scene flow understanding and object mapping by considering not only LIDAR and cameras, but also DSRC-based V2V beacons exchanged between vehicles.

C. OUR CONTRIBUTION

The goal of this work is to merge the key features of LIDAR in giving accurate distances, camera with object textural details, and V2V beacons for the awareness of both hidden out-of-sight vehicles or vehicles not observed by the two other means. We first adapt the camera and LIDAR learning and object recognition schemes to prepare the resulting data for alignment. We also generate BSMs for vehicles detected in the KITTI Vision Benchmark Suite for alignment purposes. Exploiting the physical neighborhood correlation within the three data sets, and their natural correspondences in the 3D physical space, we then cast the merging problem of these three sets of data as a semi-supervised manifold alignment. Our proposed approach is to first identify few clear correspondences between data points from each pair of data sets, and employ them to align (i.e., pair) the rest of the points between the camera-LIDAR and camera-V2V data sets, thus establishing a full object correspondence among the three sets. To perform this alignment, we compute the neighborhood correlations and Laplacian matrix for in each data set using local linear embedding. The alignment problem is then formulated as an eigenvalue problem over a compounded Laplacian matrix. Once the mapping of paired points is done, the other points from each data set can be easily paired and the non-paired points can be added in the aligned 3D environment, thus significantly enriching the vehicle knowledge of its surroundings. We test this work using the scene flow, 3D LIDAR point clouds, and generated BSMs of the KITTI Vision Benchmark Suite, and perform the camera-LIDAR and camera-V2V alignment.

The remainder of this paper is organized as follows. Camera object recognition and data-set preparation from the KITTI suite scene flow and 3D LIDAR point clouds are presented in Sections II-A and II-B, respectively. We present the manifold alignment formulation and solution between the 3 Dimensional LIDAR space, camera Space, and V2V beacons in Section III. BSM creation according to the LIDAR recognized objects from the KITTI suite, number of recognized objects per input type and the performance of the alignment process are illustrated in Section IV. Section V summarizes the findings and conclusion of this paper and illustrate directions for future work.

II. PREPARING DATA SETS FOR ALIGNMENT

In this section, we will illustrate the steps taken in order to prepare the manifolds of the different data sets from the multi-modes of vehicle sensing of its surroundings. Luckily, the raw V2V data usually comes with the required information for alignment, since each BSM clearly indicate the GPS position of its transmitting vehicle. Thus, the adjacencies between vehicle objects from the V2V data can be easily computed and does not require any further recognition nor preparation. Thus, the next two subsections will focus on adapting the object recognition schemes from camera feeds and LIDAR scans to both extract the objects of interest to the alignment process, and determine their relative locations to one-another. This information will be then used to create manifolds for each of these data sets, representing the adjacency relations of their detected objects of interest, which will then enable us to feed them to the alignment process.

A. ADAPTING RECOGNITION FROM CAMERA FEEDS

In this section, we describe our approach of adapting the learning procedure from camera feeds, in order to prepare the camera data set that is suitable for our proposed alignment process. This adaptation consists of tailoring the darknet's CNN, developed in [16] to test the KITTI stereo images using the architecture illustrated in Fig. 1, to render both the two important object classes for the alignment process, namely ''Cars'' and ''Persons'', and the center of gravity of the object bounding box (or object pixels) in the 2D space. Inspired by this CNN design, we propose to exploit the feature of objects' anchor boxes, which predict the coordinates of the bounding boxes around recognized objects, to find their pixel adjacency directly from the fully connected layers that are developed on top of the convolutional network extractor (both illustrated in Fig. 1). We use a single CNN that processes the entire image pixels during both the training and test time, and predicts all the bounding boxes and their corresponding classes probabilities. We then apply a hard threshold on the probabilities of the object suspected to be either a ''Car'' or a ''Person'', beyond which these objects are deem to be so.

While parsing through the frames of any driving sequence, our tailored CNN stores the characteristics of each object in an organized way, including its class name, anchor box dimensions in pixels, and its calculated center of gravity in the 2D space. Moreover, counters are set to keep track of the total number of objects from each class per frame. The extracted details of objects are stored separately from one frame to another both within the same driving sequence and across different sequences. To test the efficiency of our tailored CNN, we applied it to recognize the objects in Fig. 2 and Fig. 4, representing two original frames (that we will denote in the rest of the paper as Frame (a) and Frame (b), respectively) from two different driving sequences from the KITTI Suite. Frame (a) and Frame (b) include different object counts per class either from image recognition or from labeled LIDAR object recognition. The model is being changed to support reading all the frames contained in both of the drive sequences 2011_09_28_*drive*_00016 and 2011_09_26_*drive*_0005 from the KITTI Suite. We opt to select from sequences recorded with RGB cameras instead

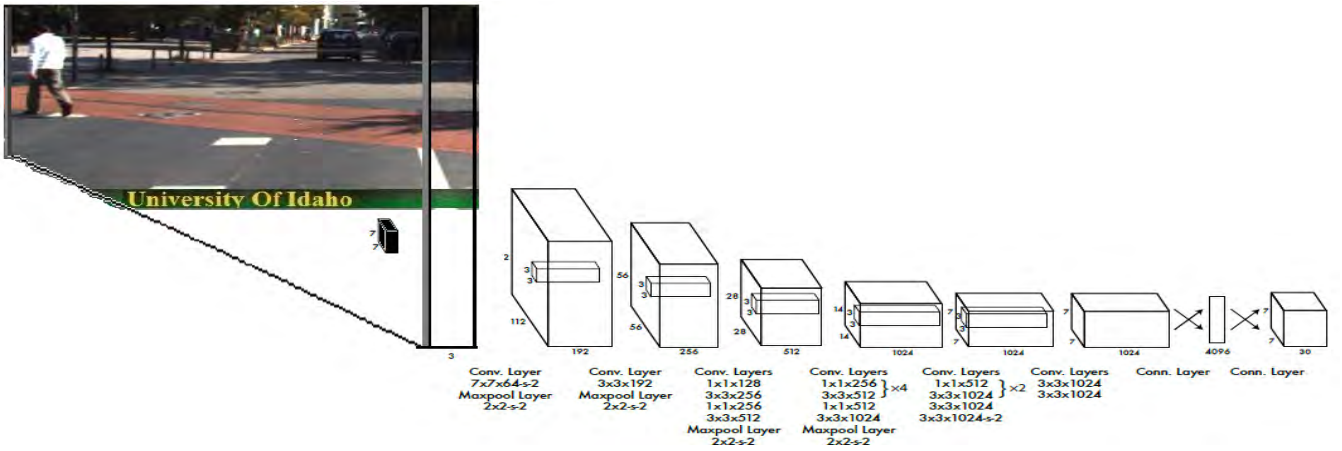

FIGURE 1. Darknet convolutional neural network architecture [16].

FIGURE 2. Original frame (a) from RGB sequence drive 2011_09_26 drive_0005 #117.

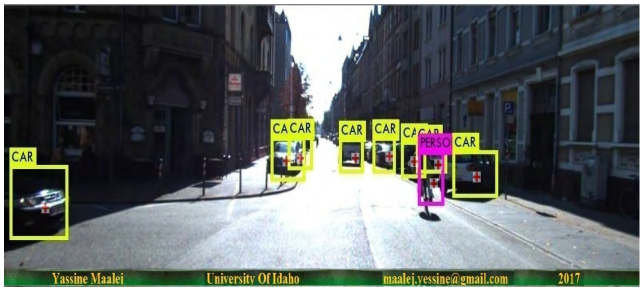

FIGURE 3. Detection of frame (a) objects and their centers of gravity.

FIGURE 4. Original frame (b) from RGB sequence drive 2011_09_28 drive_0016 #32.

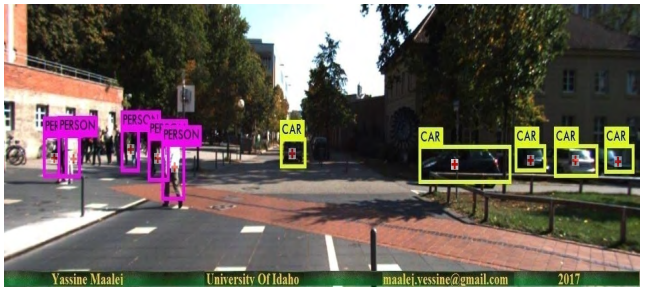

FIGURE 5. Detection of frame (b) objects and their center of gravity.

of gray-scale, for clarity of work purposes. The results of applying our tailored CNN on Frame (a) and Frame (b) are illustrated in Fig. 3 and Fig. 5, respectively. We can clearly see from both figures that the objects are perfectly recognized and labeled with their proper classes. The center of gravity of each object is depicted by a red cross within each object's anchor box. The centers of gravity are computed according to the pixels belonging to each object and are expressed in pixels in the 2D space.

It is worth noting that some vehicles are not detected in both Fig. 3 and Fig. 5. The failure in detecting these vehicles occurred due to the either the overlap between vehicles (as in the parked undetected vehicle on the right of street in Fig. 3) or the shaded zone they lie into (as in the case of the parked vehicle to the right of the leftmost recognized vehicle in Fig. 5) and thus the low variance, as perceived by the camera, between their pixel colours and those of the background. As aforementioned, the latter phenomenon was the cause of the fatal accident by the Tesla autonomous vehicle prototype, which calls for our proposed multi-modal sensing of the vehicle's 3D surrounding environment by adding LIDAR, and most importantly V2V (and also V2I and vehicle-to-pedestrian V2P) information to each vehicle's object recognition system. Indeed, the problem of these undetected objects will be resolved if the camera information is aligned with the received V2V information from the other vehicles, thus

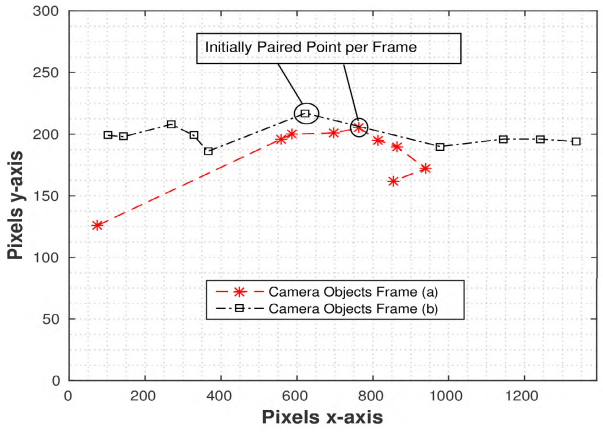

FIGURE 6. Pixel-wise manifolds of the recognized objects for both Frame (a) and Frame (b).

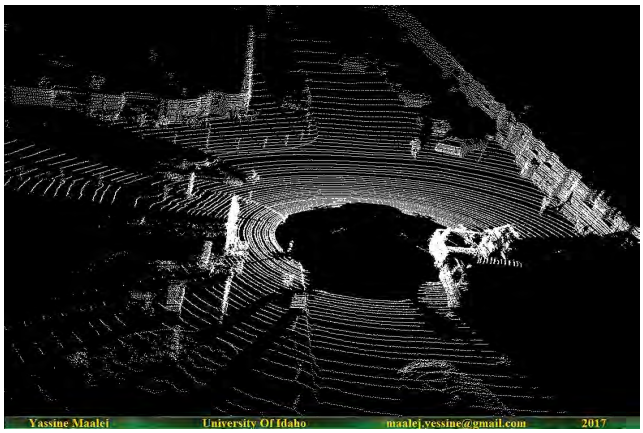

FIGURE 7. Point cloud scan corresponding to frame (a).

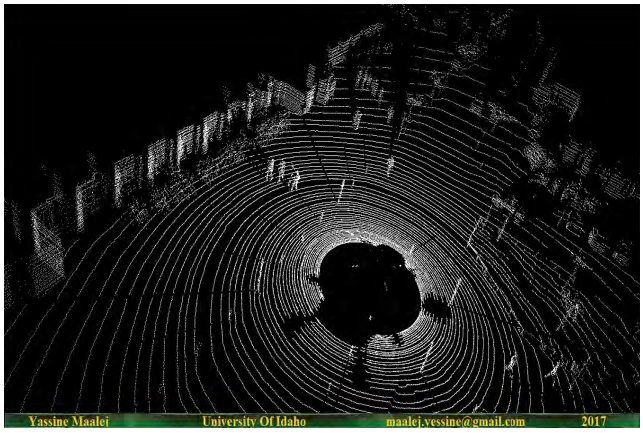

FIGURE 8. Point cloud scan corresponding to frame (b).

reducing chances for of having missing knowledge of the surrounding vehicles.

In Fig. 6, we plot the 2D pixel-wise manifolds of the car and people objects detected by camera for both Frame (a) and Frame (b). These manifolds illustrate the adjacency relations between the centers of gravity of the detected objects in each of the frames, which represents approximate distances in terms of pixels between them in the 2D space. The neighborhood relations between the objects from the camera frames will be thus expressed in terms of pixels, and will be inserted with that format to the *neighborhood weight calculations* that will be introduced in Section III-C. The obtained curves for both Frame (a) and Frame (b) in Fig. 6 thus represent camera data set manifold in the pixel domain.

It is important to note here that the representation of distant objects in the manner illustrated in Fig. 6 can give a meaningful value of distance between them, and can be easily correlated with distances from both LIDAR scans and V2V BSMs positions. However, due to the 2D nature of camera images, the distance representation between very close or overlapping objects in the image may be highly inaccurate and will not provide relevant information to the alignment process.

B. ADAPTING RECOGNITION FROM LIDAR POINT CLOUDS

This section illustrates the adaptation of object recognition from LIDAR point clouds to prepare a data set that is suitable for our proposed alignment process. To simplify this process, we are not considering the recognition of every object from the LIDAR point clouds, since a tremendous number of unknown, and most importantly un-mappable, objects can be detected as a set of neighbored point clouds. As in the previous section, we will thus restrict the recognition of the Vote3Deep 3D CNN, developed in [19], to identify only ''Cars'' and ''Persons'', since the remaining items do not represent major importance in the alignment process. The Vote3Deep 3D CNN uses feature-centric voting to detect persons and cars that are spatially sparse along many unoccupied regions, without the need to transform the 3D point cloud to a lower dimensional space.

The Velodyne LIDAR scans corresponding to both of Frame (a) and Frame (b) are plotted in Fig.7 and in Fig.8, respectively. The black centered area in both figures is the car that is equipped with the 360° Velodyne spinning laser scanner. The circled points in both figures represent the free space contour lines where no obstacles have been encountered (i.e., each circle represent points of equal distance from the vehicle with no objects in them). All cars and persons in both of Frame (a) and Frame (b) are represented by more dense dots (almost creating a 3D surface) due to the reflections of the laser beams from these objects. We note that the other black areas without circled points nor objects correspond to zones in which the LIDAR beams were blocked by obstacles. Consequently, the LIDAR scans cannot provide any knowledge of what is in these zones.

Fig. 9 depicts the 3D manifolds of the detected car and people objects from the LIDAR scans of both Frame (a) and Frame (b). The relative position of each each object with respect to the Velodyne LIDAR Scanner position is represented by an (x,y,z) triplet in the 3D space. As in Fig. 6, the manifolds represent the adjacency relations between the centers of gravity of the detected objects in each of the frames,

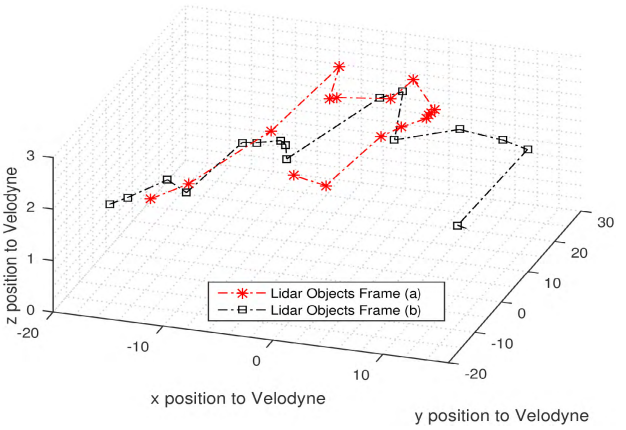

FIGURE 9. Manifolds of the detected objects from LIDAR point clouds corresponding to both frame (a) and frame (b).

which represents the approximate distances between them in the 3D space.

We note that the manifolds representing the detected objects from the LIDAR point clouds contain a larger number of persons and cars compared to those detected from the camera in Fig. 6. For example, the undetected shaded vehicle next to the leftmost detected vehicle in Fig. 5 by the camera was detected by the LIDAR point clouds as shown in Fig.9. Moreover, the objects that are behind of the camera were captured by the LIDAR spinning scans.

III. PROPOSED ALIGNMENT APPROACH

Having the manifolds of all vehicle sensing modes created, this section will illustrate the proposed approach to align these three manifolds, thus linking each object in each data set to its corresponding ones in the two others. Objects with no correspondences (e.g., objects detected in one data set but not the other) can then be added to the global knowledge of the vehicle, thus creating one enriched scene for the vehicle about its surroundings.

Our proposed approach to align these three data sets is founded on aligning both the LIDAR and V2V data sets individually with the camera data set. This choice is driven by the fact that the camera component is the most widely used component in autonomous vehicles, both at the research and prototyping level. Basically, the main two types of input data that the system takes are raw RGB frames from camera, LIDAR points cloud recorded in the driving sequences and V2V generated BSMs. Consequently, the alignment process is applied between camera, LIDAR and V2V data are principally the anchor boxes surrounding objects extracted from 2D Convnet, 3D position of objects retrieved after detection from 3D LIDAR and the V2V positions of cars. Once both data sets are aligned with the camera data sets, the correspondences between each two objects in them will be set naturally through their common corresponding object in the camera data set. Consequently, the rest of this section will focus on aligning only two data sets, one from the LIDAR or V2V data set, and the other being the camera.

The manifold alignment procedure between any two correlated data sets (i.e., having some correlation to one another in some space) is based on jointly embedding the objects of these data set in a lower dimensional space while both:

- Aligning points of initial correspondence between these data sets (i.e., points that are initially known to correspond to one another in the two data sets) in this lower dimensional space.
- Preserving the neighborhood correlation (i.e., the local structure) in each of them.

Once this embedding is done, each pair of points from the two data sets that have the closest proximity in this lower dimensional embedding are declared as corresponding to one-another. Clearly, our data sets of interest have clear correlation in the 3D space as a subset of each of them represent the exact same objects perceived by the vehicle through three different modes. They thus qualify to be aligned using manifold alignment. Given the above methodology, the proposed alignment process between our camera and LIDAR/V2V data sets will follow three main steps:

1) Neighborhood weight computation within each data sets (to preserve local structures).
2) Initial correspondance determination
3) Semi-supervised alignment

Each of these steps will be explained in details in the next three subsections.

A. NEIGHBORHOOD WEIGHT COMPUTATION

The neighborhood weights are a one dimensional representation of the distance/adjacency relations between each pair of objects in any one data set. A higher/lower weight value between two data points symbolizes their higher/lower proximity in their original space. Computing these weights are usually done through dimensionality reduction techniques, such as locally linear embedding, heat kernels, ... etc. In our work, we select the locally linear embedding (LLE) technique [25] because it gives more importance in preserving neighborhood correlation in the one dimensional space.

To compute the neighborhood weights using LLE for any data set, the N data points having the closest Euclidian distance to each higher dimensional data point $\mathbf{t}^{(i)}$ are first identified as its neighbor set $\mathcal{N}(i)$. Let $\left[\mathbf{t}^{(\mathcal{N}(i,1))}, \ldots, \mathbf{t}^{(\mathcal{N}(i,N))}\right]$ be such set of points for data point $\mathbf{t}^{(i)}$. Given this set, LLE thus computes the neighborhood weights of $\mathbf{t}^{(i)}$ using the following optimization problem:

$$\arg\min_{W_{ij}} = \left\{ \left| \mathbf{t}^{(i)} - \sum_{j \in \mathcal{N}(i)} W_{ij} \mathbf{t}^{(\mathcal{N}(i,j))} \right|^2 \right\} \quad \mathbf{s.t} \sum_{j \in \mathcal{N}(i)} W_{ij} = 1 \tag{1}$$

Clearly, a closer the point $\mathbf{t}^{(\mathcal{N}(i,j))}$ to $\mathbf{t}^{(i)}$ will have a higher weight W_{ij}. Points $\mathbf{t}^{(j)}$ that are not in $\mathcal{N}(i)$ will have $W_{ij} = 0$. The above optimization problem can be solved for each point $\mathbf{t}^{(i)}$ using a closed form solution as follows [25]. Defining the

distance matrix $\mathbf{D}_i$ of point $\mathbf{t}^{(i)}$ as

$$\mathbf{D}_i = \begin{bmatrix} \mathbf{t}^{(i)} - \mathbf{t}^{(\mathcal{N}(i,1))} \\ \mathbf{t}^{(i)} - \mathbf{t}^{(\mathcal{N}(i,2))} \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{t}^{(i)} - \mathbf{t}^{(\mathcal{N}(i,N))} \end{bmatrix} \tag{2}$$

W_{ij} for all $\mathbf{t}^{(j)} \in \mathcal{N}(i)$ can be computed as:

$$W_{ij} = \frac{\sum_{k=1}^{N} \left\{ (\mathbf{D}_i \mathbf{D}_i^T)^{-1} \right\}_{jk}}{\sum_{m=1}^{N} \sum_{n=1}^{N} \left\{ (\mathbf{D}_i \mathbf{D}_i^T)^{-1} \right\}_{mn}} \tag{3}$$

Note that $\left\{ \left(\mathbf{D}_i \mathbf{D}_i^T\right)^{-1} \right\}_{uv}$ is the element of the uth row and the vth column in the inverse of matrix $\mathbf{D}_i \mathbf{D}_i^T$. By repeating this procedure for all data points in each of the data sets, we obtained all required neighborhood weights for the alignment process.

B. INITIAL CORRESPONDENCES DETERMINATION

The second step of the alignment process consists of determining at least one point of correspondence between the camera and LIDAR/V2V data set. Some techniques were employed in the literature to estimate these initial correspondences based on the geometry of the data sets [26], [27]. The main idea behind these techniques is to match some local geometries within the data sets, and declare points having a few highly similar local geometries as corresponding. In most cases, this requires going through a long search among all combinations of points to find high similarities and thus close to correct correspondences, which sometimes increases the complexity of the process.

Luckily, the data sets considered in this paper already possess some geometric references that can somewhat related them to one-another. For instances, the front camera and LIDAR are usually aligned in the vehicle, and it is thus easy to roughy determine LIDAR reflected beams (and thus its corresponding detected objects if any) from a certain direction/range corresponding to a certain direction/range of the camera picture (and thus its detected object). We can thus use one of these objects from the same exact direction and distance from the vehicle to tag them as initial correspondences. For example, we can pick the initial point of correspondence between these two data sets to be the ones representing the one or few furthest point(s) captured by both the camera and LIDAR within the same direction(s) from the vehicle. If we apply this approach for Frame (b), we can align, we can estimate that the leftmost identified vehicle in Fig.5 with the vehicle identified by the LIDAR beam reflected from that direction. This approach is also suitable for the alignment with V2V as we can estimate the distance from the camera and/or LIDAR to this farthest recognized object in a certain direction and match it to the most likely received GPS that matches it in distance and direction from the vehicle. Consequently, we will employ this approach to identify one or few point of correspondence across the three data sets, and employ them in the alignment process.

It is important to mention here that, although some errors may occur in this initial alignment process, but so could happen in the proposed techniques in the literature [26], [27] given the exact same three data sets. Most importantly, the error in our case will be in slight range within the entire geometry of the data sets, and we can still get good alignment results for the other objects.

C. SEMI-SUPERVISED ALIGNMENT

As aforementioned, the semi-supervised alignment of two manifolds is done by jointly embedding of their data points in a lower dimensional space while both aligning the initial points of correspondence and preserving the neighborhood correlation in each of them. The problem of aligning between the camera data set (that we will denote by $\mathcal{X}$) to the LIDAR/V2V data set (that we will tab by $\mathcal{Y}$)

can be expressed as:

$$\arg\min_{\mathrm{f,g}} \left\{ \lambda^x \sum_{i,j} [f_i - f_j]^2 \, W_{ij}^x + \lambda^y \sum_{i,j} [g_i - g_j]^2 \, W_{ij}^y + \mu \sum_{i \in \mathcal{P}} |f_i - g_i|^2 \right\} \tag{4}$$

where $f = [f_1, .., f_X]^{(T)}$ and $g = [g_1, .., g_Y]^{(T)}$ are vectors in $\mathbb{R}^X$ and $\mathbb{R}^Y$ of the X camera points and Y LIDAR/V2V points, respectively, $\mathcal{P}$ is the set of paired points between $\mathcal{X}$ and $\mathcal{Y}$, whereas W_{ij}^x and W_{ij}^y are the neighborhood weights between the ij pair of points within data sets $\mathcal{X}$ and $\mathcal{Y}$, respectively.

Clearly, this problem is a three-objective function optimization, with weighting factors λ^x, λ^y and μ. The first two terms aim to separately find the vectors $\mathbf{f}$ and $\mathbf{g}$ that are separately preserving the neighborhood structures of the points in $\mathcal{X}$ and $\mathcal{Y}$, respectively. Indeed, minimizing the first term will occur by attributing a smaller $f_i - f_j$ term (i.e., close values of f_i and f_j) for any $i - j$ pair of points in $\mathcal{X}$ with larger W_{ij}^x. Minimizing the second term plays the exact same role with the elements of vector $\mathbf{g}$ to preserve in it the neighborhood structure of $\mathcal{Y}$. The third term aims to equalize (i.e., align) the elements in $\mathbf{f}$ and $\mathbf{g}$ that correspond to paired points in $\mathcal{X}$ and $\mathcal{Y}$. Indeed, it penalizes discrepancies between f_i and g_i corresponding to each point $i \in \mathcal{P}$.

With some simple manipulation, Eq. (4) can be re-written as:

$$\arg\min_{\mathrm{f,g}} \left\{ \lambda^x \mathbf{f}^T L^x \mathbf{f} + \lambda^y \mathbf{g}^T L^y \mathbf{g} + \mu (\mathbf{f} - \mathbf{g})^T (\mathbf{f} - \mathbf{g}) \right\} \tag{5}$$

where $L^x = [L^x_{ij}]\ \forall\ i,\ j \in \mathcal{X}$, such that:

$$L^x_{ij} = \begin{cases} \sum_j W^x_{ij}, & i = j \\ -W^x_{ij} & j \in \mathcal{N}_i \\ 0 & \text{Otherwise} \end{cases} \tag{6}$$

while $L^y = [L^y_{ij}]\ \forall\ i,j \in \mathcal{Y}$ such that L^y_{ij} is defined by replacing each W^x_{ij} by W^y_{ij} in (6). In general, the problem in (5) is ill-defined. Nonetheless, imposing a hard constraint to make $f_i = g_i\ \forall\ i \in \mathcal{P}$ (i.e. as $\mu \to \infty$), the problem in (5) is be re-casted as an eigenvalue problem. Define vector $\mathbf{h}$ as:

$$\mathbf{h} = \begin{bmatrix} \mathbf{f}_{\mathcal{P}} = \mathbf{g}_{\mathcal{P}} \\ \mathbf{f}_{\mathcal{Q}^x} \\ \mathbf{g}_{\mathcal{Q}^y} \end{bmatrix} \tag{7}$$

where $\mathcal{Q}^x = \mathcal{X}/\mathcal{P}$ and $\mathcal{Q}^y = \mathcal{Y}/\mathcal{P}$. In other words, $\mathbf{h}$ is vector structured in a way that it begins with elements of $\mathbf{f}$ and $\mathbf{g}$ corresponding to the paired points in $\mathcal{P}$, followed by the remaining (i.e., unpaired) elements of $\mathbf{f}$, and ends with the remaining (i.e., unpaired) elements of $\mathbf{g}$.

Having this vector defined, and setting $\mu \to \infty$ in (5), we can re-write the problem as:

$$\arg\min_{\mathrm{h}} \left\{ \frac{\mathbf{h}^T L^z \mathbf{h}}{\mathbf{h}^T \mathbf{h}} \right\} \qquad \text{s.t.} \quad \mathbf{h}^T 1 = 0 \tag{8}$$

where:

$$L^z = \begin{bmatrix} \lambda^x L^x_{\mathcal{P}\mathcal{P}} + \lambda^y L^y_{\mathcal{P}\mathcal{P}} & \lambda^x L^x_{\mathcal{P}\mathcal{Q}^x} & \lambda^y L^y_{\mathcal{P}\mathcal{Q}^y} \\ \lambda^x L^x_{\mathcal{Q}^x\mathcal{P}} & \lambda^x L^x_{\mathcal{Q}^x\mathcal{Q}^x} & \mathbf{0} \\ \lambda^y L^y_{\mathcal{Q}^y\mathcal{P}} & \mathbf{0} & \lambda^y L^y_{\mathcal{Q}^y\mathcal{Q}^y} \end{bmatrix} \tag{9}$$

with $L^x_{\mathcal{I}\mathcal{J}}$ ($L^y_{\mathcal{I}\mathcal{J}}$) defined as the sub-matrix of $L^x(L^y)$ corresponding to the rows indexed by the elements in $\mathcal{I}$ and the columns indexed by the elements in $\mathcal{J}$. λ^x and λ^y are computed following the work in [27] as: $\lambda^x = X/(X+Y)$ $\lambda^y = Y/(X+Y)$. The solution of the problem in (8) is known to be the eigenvector $\mathbf{h}$ that corresponds to the smallest non-zero eigenvalue of L^z. Finding the optimal vector $\mathbf{h}^*$ naturally defines the optimal vectors $\mathbf{f}^*_{\mathcal{Q}^x}$ and $\mathbf{g}^*_{\mathcal{Q}^y}$. By finding the elements of closest distance from these two latter vectors, the corresponding unpaired points in $\mathcal{X}$ and $\mathcal{Y}$ are paired together. This concludes the alignment process.

Note that the vectors $\mathbf{f}^*_{\mathcal{Q}^x}$ and $\mathbf{g}^*_{\mathcal{Q}^y}$ may not (and most probably won't) be of the same dimension, and thus the remaining points in the longer vector should correspond to objects that were detected by one mode but not the other. These points can thus be added to global vehicle understanding to its surrounding while respecting its neighborhood correlations (i.e., weights) within its corresponding data set. When this alignment and object addition operation is repeated for between the camera-LIDAR and camera-V2V data set pairs, this results in an enriched construction of the vehicle surroundings that no single mode of information (camera, LIDAR, or V2V) can attain individually.

IV. ALIGNMENT ACCURACY AND GAIN TESTING

In this section, we aim to test both the accuracy of our proposed alignment scheme and its gain in enriching vehicles' knowledge about its surroundings using multi-modal inputs. In these tests, we will use the KITTI benchmark as our source of camera feeds and LIDAR scans of vehicle surroundings, due to its wide approval and adoption in the testing of various autonomous vehicle recognition and driving systems. The raw data recording of both of the driving sequences contain color stereo sequences recorded with a 0.5 Megapixels camera stored in png format, 3D Velodyne LIDAR point clouds stored as binary float matrix, 3D GPS/IMU data for location and timestamps information stored in text files.

The only problem with this benchmark is that it does not include a library of BSMs as it did not assume any V2V communications. We will thus first generate a V2V BSM streams for the vehicles in the entire sequences and perform the tests using these generated BSMs. The driving sequences 2011_09_28_*drive*_0016 and 2011_09_26_*drive*_0005 are recorded during daytime in non-rush hour around the campus of Karlsruhe Institute of Technology and metropolitan area of Karlsruhe in Germany, respectively. The former sequence has a total size of 0.7 GB, 192 frames of 1392*512 pixels each and contains 11 distinct cars and 9 distinct persons. while the latter sequence has a total size of 0.6 GB, 160 frames of 1392*512 pixels each and contains 12 distinct cars and 3 distinct persons. We will then illustrate the detailed alignment accuracy and gain results for Frame (a) and Frame (b). Finally, we present the aggregate alignment accuracy on the entire two aforementioned sequences.

In the remaining of this section, we define the alignment accuracy (in percentage) as the accuracy in mapping all the points from $\mathcal{X}$ (i.e., camera data set) to $\mathcal{Y}$ (LIDAR/V2V data sets). More formally, it is defined as the percentage of points in $\mathcal{X}$ mapped to wrong (i.e., non-actually corresponding) points in $\mathcal{Y}$ normalized by the total number of points of $\mathcal{X}$ (i.e., the camera data set) We also define the alignment gain (in percentage) as the percentage of added objects from the alignment process (i.e., from mapping LIDAR/V2V detected objects to the camera detected objects), normalized to the original number of detected objects in the camera data set.

A. V2V BSMS GENERATION FROM KITTI BENCHMARK OBJECTS

The lack of time-stamped V2V BSMs data that corresponds to camera frames and LIDAR scans have encouraged us to dynamically generate the V2V related messages of the recognized objects from the LIDAR scans. We chose the LIDAR scans over the camera feeds to generate BSMs as usually BSMs can be received from larger distances that the range of cameras and thus should include a larger number of vehicles. Moreover, since we are using the camera as our reference alignment set (i.e., our methods aligns both the LIDAR and V2V data sets with camera data set), it is better generate the V2V BSMs from another set to have stronger discrepancies

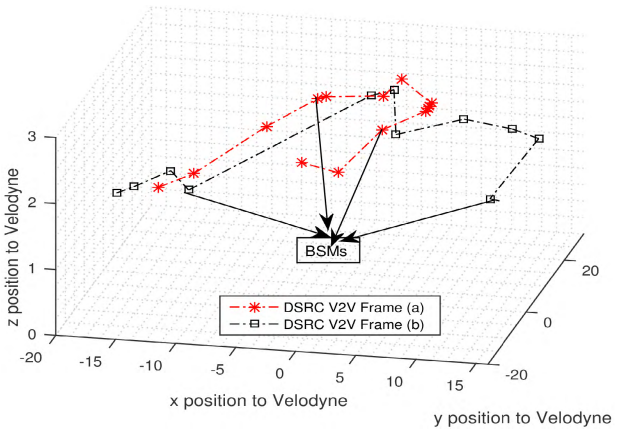

FIGURE 10. Adjacency of recognized objects from DSRC.

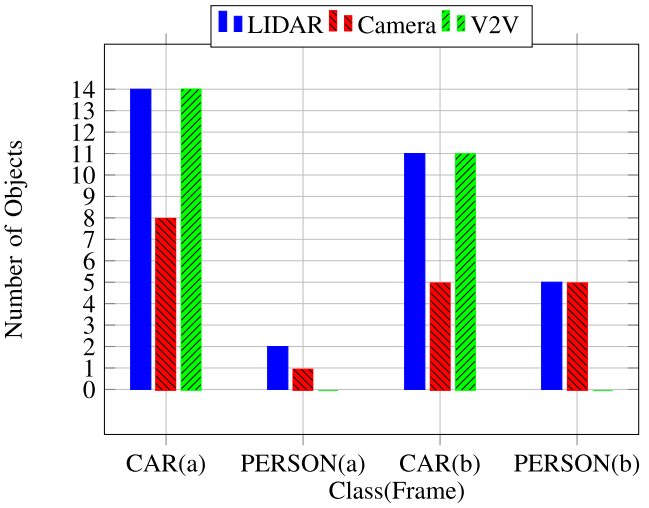

FIGURE 11. Number of objects per class for both frame (a) and frame (b).

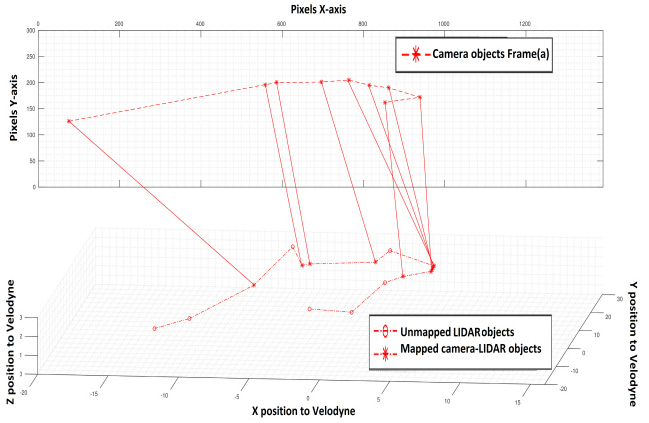

FIGURE 12. Camera-LIDAR object alignment of frame (a).

in the data sets and make the alignment more challenging. Note that, although they are generated from LIDAR data in our testing, the V2V BSM data sets and their corresponding manifolds will be significantly different from those of the LIDAR data sets, since the former ones include objects identified as "Car" only. Indeed, pedestrians do not typically generate BSMs, and thus objects identified as people in the LIDAR data set will not show up in the V2V BSM data set.

For each identified vehicle by the LIDAR in every frame of the two aforementioned sequences, we generate a simple BSM only stating its position. The vehicles detected per-scan, whether moving or parked, are assumed to generate the V2V BSM beacons. It is important to note that, in practice, parked vehicles may or may not send beacons depending on the employed V2V standard. However, this fact will not affect the alignment process proposed in this paper, but will rather only increase or decrease the number of objects to be aligned. We employed the position *(x,y,z)* triplet of the object of the Velodyne LIDAR Scanner to emulate the GPS coordinate system *(Latitude, Longitude, Elevation)* that will be used in practical scenarios. Note that we did not include to our generate data set other typical BSM fields, such as messages count, temporary ID, brake system status, and acceleration, as they will not be used in the alignment process. In practical scenarios, the vehicle can easily extract the position information from the BSM for alignment purposes.

Fig. 10 depicts the manifolds of the generated V2V BSM data sets of Frame (a) and Frame (b) respectively. As expected, the shape of and the number of points in the manifolds in Fig. 10 are different from the ones formed by LIDAR objects in Fig. 9, due to the lack of objects identified as "Persons" from the former.

B. ALIGNMENT RESULTS FOR FRAME (A) AND FRAME (B)

To understand the accuracy and gain results of our alignment process on Frames (a) and (b), we first illustrate in Fig. 11 the number of detected objects per class in these two frames. We note that, for both frames, the LIDAR scans have detected more cars and persons than the camera feeds. Due to the method with which they are obtained, the V2V data set have the same number of cars as the LIDAR data set, but no persons as expected.

Fig. 12 and Fig. 13 represent the detailed camera-to-LIDAR and camera-to-V2V alignment results, respectively, for Frame (a).

Similarly, Fig. 14 and Fig. 15 represent the detailed camera-to-LIDAR and camera-to-V2V alignment results, respectively, for Frame (b).

The top portion of all four figures illustrate the camera dat set points in 2D, where as the bottom figure illustrate the LIDAR or V2V data set points in 3D.

The first result to report from these figure is that the alignment accuracy was 100% for all the four alignment tasks. In other words, all data points from the camera were mapped to the points of the exact same objects in the LIDAR and V2V data sets. The second observation is the presence of objects from the camera data set that were not aligned with points from the V2V data set for both Frame (a) and Frame (b), which are represented by the circled points in the top portions of Fig. 13 and Fig. 15, respectively. By comparing

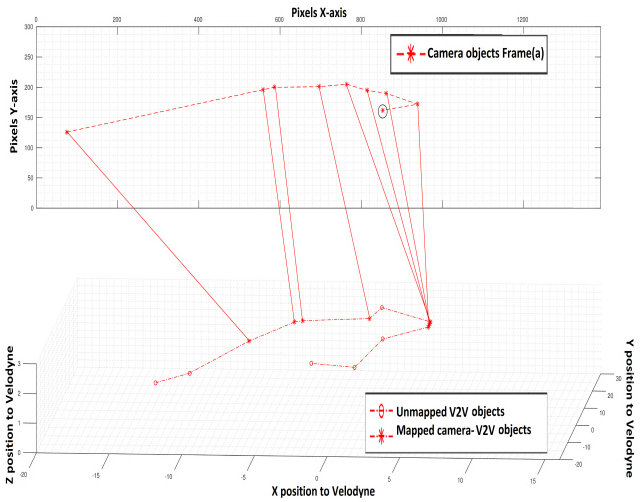

FIGURE 13. Camera-V2V Object alignment of frame (a).

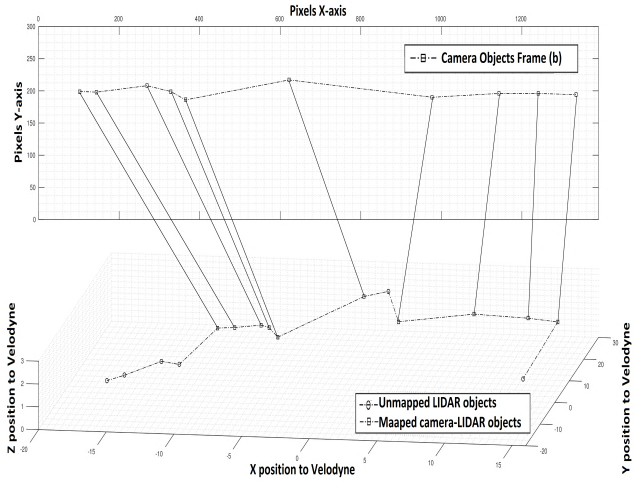

FIGURE 14. Camera-LIDAR object alignment of frame (b).

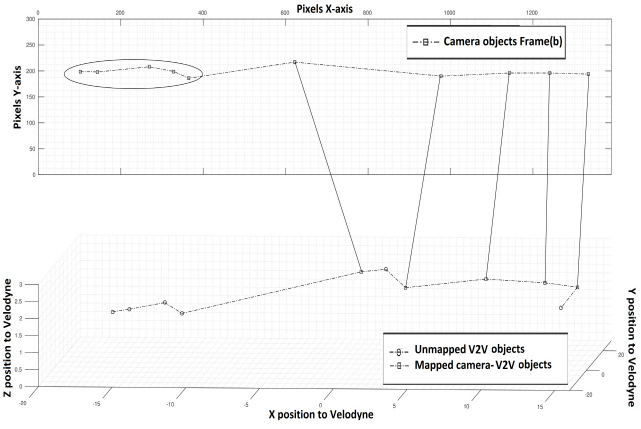

FIGURE 15. Camera-V2V object alignment of frame (b).

the positions of these points with the recognized objects of Frame (a) and Frame (b) in Fig. 3 and Fig. 5, respectively, it is easy to notice that all such unmapped points from the camera data set represent only pedestrians. This behavior is

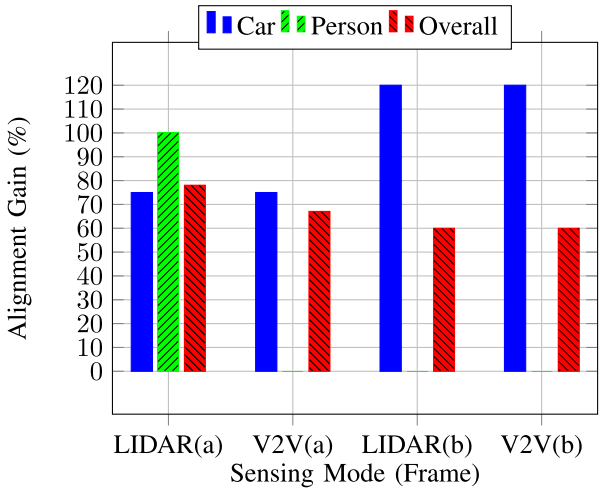

FIGURE 16. Alignment gains for frame (a) and frame (b).

indeed expected as it is known that V2V data set does not include points representing pedestrians. We can note from the top portions of Fig. 12 and Fig. 14 that this case did not occur when aligning with the LIDAR data set, due to the ability of the LIDAR to detect "Person" objects, and their proper mapping to those detected by the camera. These facts also clarify the number of mapped points between camera and LIDAR exceed those mapped between the camera and V2V objects.

Another important observation is the presence of unmapped data points in the bottom portions of all four figures, represented by different markers. These points consists of the objects detected by the LIDAR scans and V2V BSMs but were not detected by analyzing the camera feeds for both frames. Consequently, these points constitute the components contributing to the alignment gain as they enrich the vehicle recognition of its surrounding compared to the sole use of camera feeds (widely used in autonomous vehicle experiments and prototypes). The "Car", "Person", and overall alignment gains, obtained for both LIDAR and V2V alignment process with the camera information, are illustrated in Fig. 16. We can see that up to 120%, 50%, and 78% alignment gains can be obtained by in the car, person, and overall objects, respectively. This clearly show the merits of our multi-model surrounding recognition system. We can also notice the overall LIDAR gains are always equal or larger than the V2V gains in this test, because the V2V data sets represent a fraction of the LIDAR data set. However, in practical scenarios, V2V gains can jump significantly as V2V BSMs cover larger distances (around 250 meters) than those scannable by cameras and LIDARs, and thus can significantly enrich the knowledge about a much wider range of its surroundings.

We finally studied the sensitivity of the alignment accuracy in response to the change in the percentage of points selected for neighborhood weight calculations. Indeed, the used number of neighbors (N) used to compute these weights, and its percentage (i.e., ratio) with the respect to the entire data set

TABLE 1. Number of objects per class for both of the driving sequences.

ID	Sequence 09_26_0005						Sequence 09_28_0016					
	Camera		LIDAR		V2V		Camera		LIDAR		V2V	
	CAR	PERSON	CAR	PERSON	CAR	PERSON	CAR	PERSON	CAR	PERSON	CAR	PERSON
0	1	2	4	6	4	-	4	6	9	7	9	-
5	1	2	4	5	4	-	4	6	9	7	9	-
10	1	1	4	5	4	-	4	5	11	5	11	-
15	2	1	5	4	5	-	5	6	11	5	11	-
20	2	1	4	4	4	-	5	5	11	6	11	-
25	2	2	4	3	4	-	5	5	11	5	11	-
30	2	2	3	3	3	-	5	5	10	6	10	-
32	x	x	x	x	x	x	5	5	10	6	10	-
35	2	2	4	3	4	-	5	6	10	8	10	-
40	2	1	4	3	4	-	4	6	10	9	10	-
45	2	2	4	3	4	-	5	5	11	8	11	-
50	2	1	5	3	5	-	6	5	11	8	11	-
55	3	1	5	2	5	-	5	5	10	7	10	-
60	2	1	5	2	5	-	4	7	10	6	10	-
65	2	1	6	2	6	-	4	5	10	6	10	-
70	2	2	6	2	6	-	4	4	10	6	10	-
75	2	2	6	2	6	-	4	5	8	5	8	-
80	1	2	6	2	6	-	4	5	8	5	8	-
85	1	2	6	2	6	-	3	6	8	5	8	-
90	3	2	7	2	7	-	4	7	8	6	8	-
95	2	2	11	2	11	-	3	7	5	6	5	-
100	4	2	10	2	10	-	4	9	4	6	4	-
105	6	2	12	2	12	-	4	6	4	5	4	-
110	7	1	13	2	13	-	4	6	4	5	4	-
115	8	1	14	2	14	-	3	6	4	5	4	-
117	8	1	14	2	14	-	x	x	x	x	x	x
120	7	1	14	2	14	-	3	5	3	5	3	-
125	8	1	14	2	14	-	3	4	3	6	3	-
130	10	1	14	2	14	-	3	5	3	6	3	-
135	9	1	13	2	13	-	3	5	3	6	3	-
140	10	1	12	2	12	-	3	5	4	6	4	-
145	10	1	13	2	13	-	2	4	4	5	4	-
150	10	3	11	2	11	-	3	5	3	5	3	-
153	10	3	12	2	12	-	3	5	3	6	3	-

size, can be crucial factors in the accuracy of the alignment. For Frame (a) and Frame (b) respectilvey, Fig.17 and Fig.18 depict the accuracy performance of both camera-LIDAR and camera-V2V alignments against the percentage of the number of neighbors in the neighborhood weight computation process, normalized by the total size of each of the data sets. Both figures clearly justify our initial intuition by showing the significant effect of this parameter on the alignment accuracy.

We can also observe that having a mid-range percentage of the points in the neighborhood weight computation (from 30 to 40 %) results in much better accuracy compared to both lower (10 to 25 %) and higher ranges (50% and above). The degradation in the lower range can be explained by the effect of outliers (i.e., points seeming to be close, especially in the camera and LIDAR scans, while not being truly so) in weight computations when the chosen percentage of neighbors is small. When the percentage increases to the mid range, the number of considered neighbors in the computation becomes quite larger, thus diminishing the effect of the smaller percentage of outliers in misrepresenting neighborhood correlation. However, as this percentage grows beyond a certain limit, the algorithm will relate each point to a lot of points, some of which definitely not being in its true vicinity. Thus, the concept of neighbourhood dilutes, which results in the exhibited performance degradation.

Another final notice about Fig.17 and Fig.18 is the the camera-V2V alignment process always results in a better performance. This can be interpreted by the fact that the V2V data set object are known to be all vehicles. Consequently, it is easy in this case to reject any alignment between a point in the V2V data set with a point representing a person from the camera data set. In other words, we can easily restrict the alignment between the V2V data set and only the objects identified as cars in the camera data set, which reduces the number of points to be aligned, removes all potential errors of aligning cars to persons (which can be the case in with LIDAR), and thus results in better overall alignment performance.

C. ALIGNMENT RESULTS OVER ENTIRE SEQUENCES

To extend our testing beyond two limited frames (i.e., only Frame (a) and Frame (b)), this section illustrates the alignment results for two driving sequences from the KITTI data set in terms of both accuracy and gain. In such more practical setting of an autonomous vehicle driving, we cannot know in advance the number of objects in each frame. Consequently,

TABLE 2. Alignment accuracy per sequence drive.

ID	Sequence 09_26_0005				Sequence 09_28_0016			
	CAR		PERSON		CAR		PERSON	
	LIDAR	V2V	LIDAR	V2V	LIDAR	V2V	LIDAR	V2V
0	100	100	100	-	75	100	83.33	-
5	100	100	100	-	75	100	83.33	-
10	100	100	100	-	75	100	80	-
15	50	100	0	-	80	80	83.33	-
20	50	100	0	-	60	80	80	-
25	50	100	50	-	80	100	80	-
30	100	100	100	-	100	100	100	-
32	x	x	x	-	100	100	100	-
35	50	100	50	-	80	100	83.33	-
40	50	50	0	-	100	100	100	-
45	50	50	50	-	80	80	80	-
50	50	100	0	-	66.66	83.33	60	-
55	66.66	100	0	-	80	100	60	-
60	50	100	0	-	100	100	57	-
65	50	50	0	-	100	100	100	-
70	50	50	50	-	100	100	100	-
75	50	100	50	-	75	75	60	-
80	100	100	100	-	75	100	80	-
85	100	100	100	-	100	100	83.33	-
90	66.66	66.66	50	-	100	100	71.42	-
95	50	50	50	-	100	100	71.42	-
100	75	75	50	-	75	100	55.55	-
105	83.33	100	50	-	100	100	83.33	-
110	100	100	100	-	75	100	66.66	-
115	100	100	100	-	66.66	66.66	66.66	-
117	100	100	100	-	x	x	x	x
120	100	100	100	-	66.66	100	80	-
125	87.5	87.5	0	-	66.66	66.66	75	-
130	100	100	100	-	66.66	66.66	80	-
135	100	100	100	-	33.33	33.33	80	-
140	90	100	0	-	100	100	40	-
145	80	100	100	-	100	100	50	-
150	100	100	100	-	66.66	100	80	-
153	90	90	66.66	-	66.66	100	80	-

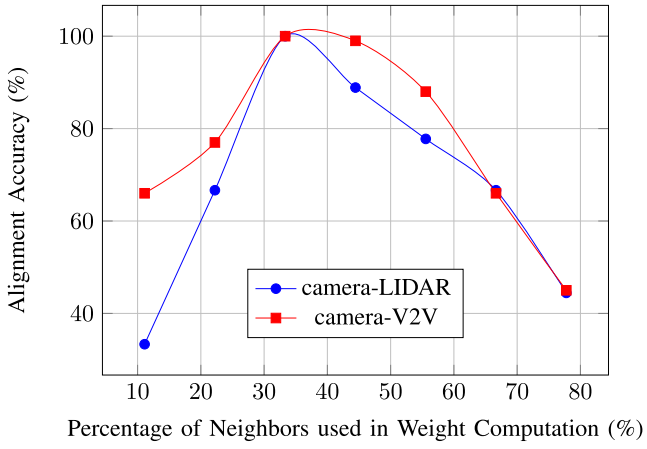

FIGURE 17. Effect of number of neighbors on the alignment accuracy for Frame (a).

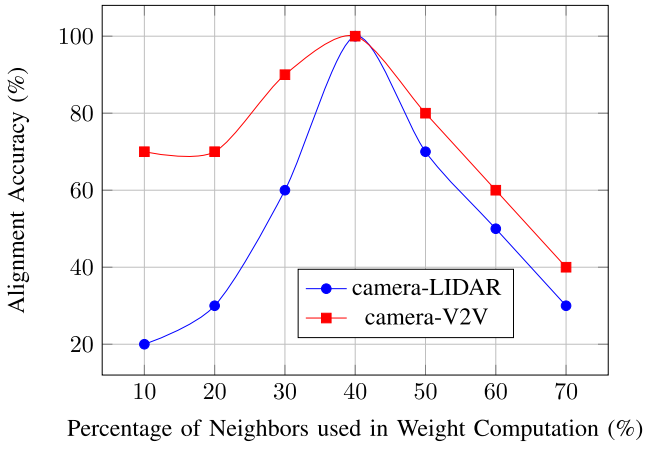

FIGURE 18. Effect of number of neighbors on the alignment accuracy for Frame (b).

the selection of the percentage of neighbors used to create the weights of the graph are performed using the best values obtained from both of the two frames (a) and (b) as well as previous studies on manifold alignment (usually in the range between 30% to 40%).

For both driving sequences 2011_09_26_*drive*_0005 (consisting of 160 frames, the 117-th being Frame (a)) and 2011_09_28_*drive*_0016 (consisting of 192 frames, the 32-nd being Frame (b)), we tested the alignment accuracy performance and gains. Since immediate subsequent frames do not

TABLE 3. Alignment gains for the driving sequences.

ID	Sequence 09_26_0005						Sequence 09_28_0016					
	CAR		PERSON		OVERALL		CAR		PERSON		OVERALL	
	LIDAR	V2V	LIDAR	V2V	LIDAR	V2V	LIDAR	V2V	LIDAR	V2V	LIDAR	V2V
0	300	300	200	-	233.33	100	125	125	16.66	-	60	50
5	300	300	150	-	200	100	125	125	16.66	-	60	50
10	300	300	500	-	350	150	175	175	0	-	77.77	77.77
15	150	150	300	-	200	100	120	120	0	-	45.45	54.54
20	100	100	300	-	166.66	66.66	120	120	20	-	60	60
25	100	100	50	-	75	50	120	120	0	-	70	60
30	50	50	50	-	50	25	100	100	20	-	60	50
32	x	x	x	x	x	x	100	100	20		60	50
35	100	100	50	-	75	50	100	100	33.33	-	63.63	45.45
40	100	100	200	-	133	33.33	150	150	50	-	90	60
45	100	100	50	-	75	50	120	120	60	-	90	60
50	150	150	200	-	166.66	100	83.33	83.33	60	-	72.72	45.45
55	66.66	66.66	100	-	75	50	100	100	40	-	70	50
60	150	150	100	-	133.3	100	150	150	0	-	45.45	54.54
65	200	200	100	-	166.6	133.33	150	150	20	-	77.77	66.66
70	200	200	0	-	100	100	150	150	50	-	100	75
75	200	200	0	-	100	100	100	100	0	-	44.44	44.44
80	500	500	0	-	166.66	166.66	100	100	0	-	44.44	44.44
85	500	500	0	-	166.66	166.66	166.66	166.66	0	-	44.44	55.55
90	133.33	133.33	0	-	80	80	100	100	0	-	27.27	36.36
95	450	450	0	-	225	225	66.66	66.66	0	-	10	20
100	150	150	0	-	66.66	100	0	0	0	-	0	0
105	100	100	0	-	75	75	0	0	0	-	0	0
110	85.71	85.71	100	-	87.5	75	0	0	0	-	0	0
115	75	75	100	-	77.77	66.66	33.33	33.33	0	-	0	11.11
117	75	75	100	-	78	66.66	x	x	x	x	x	x
120	100	100	100	-	100	87.5	0	0	0	-	0	0
125	75	75	100	-	77.77	66.66	0	0	50	-	28.57	0
130	40	40	100	-	45.45	36.36	0	0	20	-	12.5	0
135	44.44	44.44	100	-	50	40	0	0	20	-	12.5	0
140	20	20	100	-	27.27	18.18	33.33	33.33	20	-	25	12.5
145	30	30	100	-	36.36	27.27	100	100	25	-	50	33.33
150	10	10	0	-	15.38	7.69	0	0	0	-	0	0
153	20	20	0	-	23.07	15.38	0	0	20	-	12.5	0

significantly change in nature, performing alignment of such frames will not yield to any tangible new information, thus adding non-justified computational burden on the system. Consequently, we consider a practical scenario by aligning the received multimodal data every 5 frames of each sequence drive in order to capture tangible dynamism of the objects without enduring excessive unnecessary computational loads. It is important to note that the number of identified objects as cars and vehicles vary from one frame to another, and from one sequence to the other, as presented in Table. 1.

For each of these frames, we first prepared the camera and LIDAR data sets as explained in Section II, and generated BSMs for all vehicles in each of the considered scenes as highlighted in Section IV-A. We then preformed the camera-LIDAR and camera-V2V alignment processes for each of these frames, using 33% (as identified from Frame (a)) and 40% (as identified from Frame (b)) of each point's neighbors in its weights computations for sequences 2011_09_26_*drive*_0005 and 2011_09_28_*drive*_0016, respectively. We recorded the alignment accuracy and gains for each of the aligned frames and finally averaged these results over the entire number of used frames in each of the sequences.

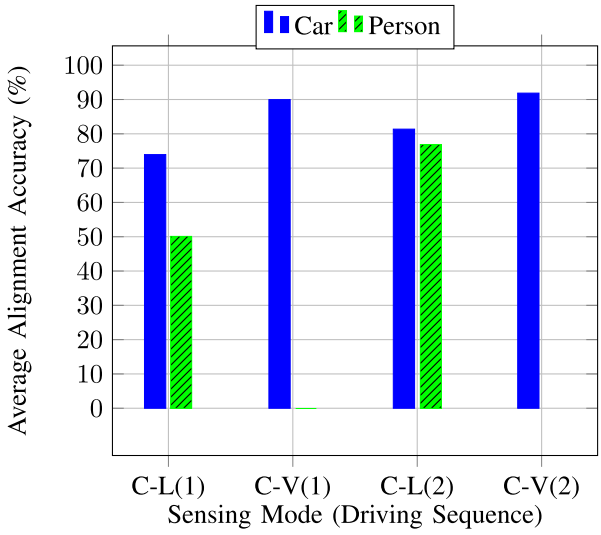

FIGURE 19. Average alignment accuracy of the driving sequences (1): Sequence 09_26_0005 and (2): Sequence 09_28_0016.

The per-frame alignment accuracy for both sequences are presented in Table. 2, and the average alignment accuracy results are illustrated in Fig. 19. We can first notice that our

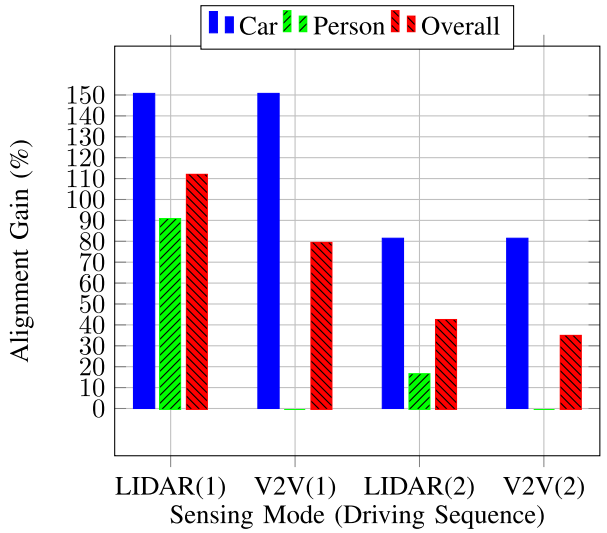

FIGURE 20. Average alignment gains of the driving sequences (1): Sequence 09_26_0005 and (2): Sequence 09_28_0016.

proposed scheme achieved 100% of alignment accuracy per-frame in many frames of both of driving sequence. This result demonstrates the fact the neighboring distance in pixels is very meaningful and describes the objects' neighborhood as in the LIDAR and V2V data sets, especially when objects are sparse. Some alignment results were less accurate (50%, 66.66% and 75%) due to some instances of overlapping and very close objects (especially for persons). The resulting variations in accuracy levels is also strongly related to the number of objects, the selected percentage of neighbors as well as how the objects are dispatched in the 2d camera space, 3D LIDAR representation and V2V environment. Moreover, we can notice a better achieved alignment accuracy achieved in the camera-V2V alignment compared to the camera-LIDAR one, which can be explained both as in the previous section and the fact of not having human alignment in the V2V cases (which is one of the causes of lower alignment performance due to object overlaps and higher proximity). Looking at Fig. 19, we can observe between 74% and 92% of average overall alignment accuracy for vehicles and 50% to 78% average accuracy for persons. These results are quite encouraging, especially on the vehicle side, as they exhibit high accuracy of alignment, thus yielding to a better understanding of the observed objects (especially vehicles) by each autonomous vehicle. Even for the cases of mis-alignment, they occur due to same object overlap or close proximity, which does not represent a sever drawback to the overall vehicle recongnition of its surroundings. Indeed, all mis-aligned objects are all from the same type and are occupying the same neighborhood within the vehicle surroundings, thus not significantly affecting its decisions.

Table. 3 illustrates the alignment gains obtained in each frame in both considered sequences and the average gains across all frames per are plotted in Fig. 20. The depicted results show significant gains (from 17% and up to 150%) in the knowledge of the vehicle using the proposed multimodal surrounding recognition approach. Again, the alignment gains of the objects of type persons in the V2V data set are null because pedestrian do not send BSMs.

V. CONCLUSION

In this paper, we developed a multimodal surrounding recognition scheme for autonomous vehicles, capable of mapping corresponding recognized objects from camera, LIDAR, and V2V data sets, The proposed scheme exploits the fact that the spacial relationships between objects in all three data sets can be characterized by manifolds, both representing intersecting sets of these objects and exhibiting high neighborhood correlations. We thus employ a manifold alignment approach to learn the correspondences between similar objects within the different data sets and enrich them with the other objects not detected by each mode, thus accomplishing a more robust perception of its surroundings. We first tested the proposed approach for two specific scenes to gain preliminary insights and more rigorous analysis of the main factor affecting the alignment accuracy, such as the percentage of neighborhood correlation, overlapping objects, etc. We then applied the learned information in testing our proposed scheme in entire driving sequences. For many cases, 100% alignment accuracy was achieved, and alignment accuracy averages of 74%-92% and 50%-78% were obtained for cars and persons, respectively. Cases of mis-matches were further shown to occur only within the same type of objects and for objects that overlap or are in very close proximity to one-another, thus minimally effecting the possible decisions of the recognizing vehicle.

ACKNOWLEDGMENT

This paper was presented at the Proceedings of the IEEE Global Communications Conference 2017 [1].

REFERENCES

[1] Y. Maalej, S. Sorour, A. Abdel-Rahim, and M. Guizani, "VANETs meet autonomous vehicles: A multimodal 3D environment learning approach," in *Proc. IEEE Global Commun. Conf., Commun. Softw., Services Multimedia Apps (Globecom CSSMA)*, Singapore, Dec. 2017, pp. 1–6.

[2] (2017). *Cyber-Physical Systems (CPS), Program Solicitation*. [Online]. Available: http://www.nsf.gov/pubs/2017/nsf17529/nsf17529.htm#toc

[3] H. Somerville and G. Cherelus. (Mar. 27, 2017). *Uber Resumes Self-Driving Program Three Days After Arizona Crash*. [Online]. Available: http://mobile.reuters.com/article/idUSKBN16Y1WB

[4] R. Grenoble. (Dec. 14, 2016). *Self-Driving Uber Blows Through Red Light on First Day in San Francisco*. [Online]. Available: http://www.huffingtonpost.com/entry/self-driving-uber-san-fran-red-light-video_us_5851c9c5e4b0732b82fedb01

[5] T. T. Team. (Jun. 30, 2016). *A Tragic Loss*. [Online]. Available: https://www.tesla.com/blog/tragic-loss

[6] J. Ham, D. D. Lee, and L. K. Saul, "Semisupervised alignment of manifolds," in *Proc. AISTATS*, 2005, pp. 120–127.

[7] X. He and R. S. Zemel, "Learning hybrid models for image annotation with partially labeled data," in *Advances in Neural Information Processing Systems*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. New York, NY, USA: Curran Associates, Inc., 2009, pp. 625–632. [Online]. Available: http://papers.nips.cc/paper/3620-learning-hybrid-models-for-image-annotation-with-partially-labeled-data.pdf

[8] L. Ladický, C. Russell, P. Kohli, and P. H. S. Torr, "Associative hierarchical CRFs for object class image segmentation," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 739–746.

[9] D. Munoz, J. A. Bagnell, and M. Hebert, "Stacked hierarchical labeling," in *Proc. Eur. Conf. Comput. Vis.* New York, NY USA: Springer, 2010, pp. 57–70.
[10] J. Tighe and S. Lazebnik, "SuperParsing: Scalable nonparametric image parsing with superpixels," in *Proc. 11th Eur. Conf. Comput. Vis. (ECCV)*. Berlin, Germany: Springer-Verlag, 2010, pp. 352–365. [Online]. Available: http://dl.acm.org/citation.cfm?id=1888150.1888179
[11] M. P. Kumar and D. Koller, "Efficiently selecting regions for scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3217–3224.
[12] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 1–8.
[13] A. Kundu, V. Vineet, and V. Koltun, "Feature space optimization for semantic video segmentation," in *Proc. CVPR*, 2016, pp. 1–8.
[14] C. Vogel, K. Schindler, and S. Roth, "3D scene flow estimation with a piecewise rigid scene model," *Int. J. Comput. Vis.*, vol. 115, no. 1, pp. 1–28, 2015. [Online]. Available: http://dx.doi.org/10.1007/s11263-015-0806-0
[15] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3061–3070.
[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[17] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. (Jul. 2016). "A unified multi-scale deep convolutional neural network for fast object detection." [Online]. Available: https://arxiv.org/abs/1607.07155
[18] V. Hegde and R. Zadeh. (Nov. 2016). "FusionNet: 3D object classification using multiple data representations." [Online]. Available: https://arxiv.org/abs/1607.05695
[19] M. Engelcke, D. Rao, D. Z. Wang, C. H. Tong, and I. Posner. (Sep. 2016). "Vote3Deep: Fast object detection in 3D point clouds using efficient convolutional neural networks." [Online]. Available: https://arxiv.org/abs/1609.06666
[20] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013. [Online]. Available: http://dx.doi.org/10.1177/0278364913491297
[21] A. MARSHALL. (Sep. 19, 2017). *With Intel's Chips, Google Could at Last Deliver Self-Driving Cars*. [Online]. Available: https://www.wired.com/story/waymo-and-intel-self-driving/ published
[22] V. De Silva, J. Roche, and A. M. Kondoz. (Oct. 2017). "Fusion of LiDAR and camera sensor data for environment sensing in driverless vehicles." [Online]. Available: https://arxiv.org/abs/1710.06230
[23] D. Jiang, V. Taliwal, A. Meier, W. Holfelder, and R. Herrtwich, "Design of 5.9 GHz DSRC-based vehicular safety communication," *IEEE Wireless Commun.*, vol. 13, no. 5, pp. 36–43, Oct. 2006.
[24] Y. Maalej, A. Abderrahim, M. Guizani, B. Hamdaoui, and E. Balti, "Advanced activity-aware multi-channel operations1609.4 in VANETs for vehicular clouds," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.
[25] L. K. Saul and S. T. Roweis. (2000). *An Introduction to Locally Linear Embedding*. [Online]. Available: http://www.cs.toronto.edu/~roweis/lle/publications.html
[26] Y. Pei, F. Huang, F. Shi, and H. Zha, "Unsupervised image matching based on manifold alignment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 8, pp. 1658–1664, Aug. 2012.
[27] K. Majeed, S. Sorour, T. Y. Al-Naffouri, and S. Valaee, "Indoor localization and radio map estimation using unsupervised manifold alignment with geometry perturbation," *IEEE Trans. Mobile Comput.*, vol. 15, no. 11, pp. 2794–2808, Nov. 2016.

YASSINE MAALEJ (S'15) received the Baccalaureate Diploma degree (Hons.) in mathematics from the Hedi Chaker School, Tunisia, and the M.S. degree (Hons.) in computer science engineering from ENSI, University of Manouba, Tunisia, in 2011 and 2015, respectively. He is currently pursuing the Ph.D. degree with the ECE Department, University of Idaho, USA. He served as a Research Scholar within the Chip Laboratory, Pennsylvania State University, State College, USA. His research interests include machine learning, object detection, LIDAR processing, artificial intelligence, sensor fusion, self-driving cars, and vehicular communication.

SAMEH SOROUR (S'98–M'11–SM'16) received the B.Sc. and M.Sc. degrees in electrical engineering from Alexandria University, Egypt, in 2002 and 2006, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Canada, in 2011. He held post-doctoral fellowship position at the University of Toronto and the King Abdullah University of Science and Technology. In 2013, he joined the King Fahd University of Petroleum and Minerals. In 2016, he joined the University of Idaho, where he is currently an Assistant Professor with the Department of Electrical and Computer Engineering. His research interests lie in the broad area of advanced communications/networking/computing/learning technologies for smart cities applications, including cyber physical systems, Internet of Things (IoT) and IoT-enabled systems, cloud and fog networking, network coding, device-to-device networking, coordinated autonomous driving, autonomous systems, intelligent transportation systems, and mathematical modeling and optimization for smart systems.

AHMED ABDEL-RAHIM received the B.Sc. degree in civil engineering from Assuit University, Egypt, in 1984, the M.Sc. degree from Minia University, Egypt, in 1990, and the Ph.D. degree in civil engineering from Michigan State University, USA, in 1998. He is currently a Professor with the Department of Civil and Environmental Engineering, University of Idaho, where he is also the Director of the National Institute for Advanced Transportation Technology. His research interests include intelligent transportation systems, connected-vehicle operations, traffic operation and control, transportation network modeling and optimizations, hardware-in-the-loop simulation, and highway design and traffic safety. He is a licensed Professional Engineer in the state of Idaho and a member of the American Society of Civil Engineers.

MOHSEN GUIZANI (S'85–M'89–SM'99–F'09) received the B.S. (Hons.) and M.S. degrees in electrical engineering and the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He served as the Associate Vice President of Graduate Studies with Qatar University, the Chair for the Computer Science Department, Western Michigan University, and the Chair of the Computer Science Department, University of West Florida. He also served in academic positions at the University of Missouri-Kansas City, the University of Colorado at Boulder, Syracuse University, and Kuwait University. He is currently a Professor and the ECE Department Chair, University of Idaho, USA. He has authored nine books and over 450 publications in refereed journals and conferences. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid. He is a Senior Member of ACM. He also served as a member, the chair, and the general chair of a number of international conferences. He received the teaching award multiple times from different institutions and the best research award from three institutions. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker from 2003 to 2005. He guest edited a number of special issues in IEEE journals and magazines. He currently serves on the editorial boards of several international technical journals and the Founder and the Editor-in-Chief of *Wireless Communications and Mobile Computing* (Wiley) journal.