Abstract—In this paper, we design a multimodal framework 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) that are exchanged using Dedicated Short Range Communication (DSRC). We merge the key features of rich texture descriptions of objects from 2D images using Convolutional Neural Networks (CNN), in addition, depth and distance between objects are provided by the 3D LIDAR point cloud and the awareness of hidden vehicles is achieved from BSMs’ beacons. We present a joint pixel to point cloud and pixel to V2V correspondence of objects in frames of driving sequences in the KITTI Vision Benchmark Suite. We achieve this by using a semi-supervised manifold alignment approach to achieve camera-LIDAR and camera-V2V mapping of their recognized persons and cars that have the same underlying manifold.

Index Terms—CNN, Velodyne, Point Cloud, V2V, LIDAR, DSRC, Manifold Alignment, BSMs, KITTI.

I. INTRODUCTION

For years, researchers on Vehicular Ad-hoc Networks (VANETs) and Autonomous vehicles presented various solutions to vehicle safety and automation, respectively. Yet, the developed research work 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 a tremendous need and importance to relate these two worlds together in several of their call for research proposal in 2017 [1]. They clearly emphasized the major importance of integration and fusion of data from various input modes in order to create a deeper understanding of object surroundings. Precisely, enriched 3D scene reconstruction by different input technologies and deep learning techniques are of a paramount importance to allow automated vehicle system to 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), incidents that could have been easily prevented if communications among vehicles and to the road infrastructure would have been considered [2], [3]. To respond to this need, we propose in this paper to enrich the learning of the 3D vehicle surroundings using multimodal inputs, namely LIDAR point cloud, RGB camera frames from the the Kitti Vision Benchmark Suite [4] and V2V-conveyed BSMs [5]. Learning the correspondence between the same objects in a driving sequence from different data inputs is a difficult but necessary task that self-driving vehicles have to handle especially with the curse of the data representation and dimensionality. Incorporating objects detected from these three sources is a mapping process that can be casted as a manifold alignment problem [6]. 3D point cloud is omnipresent in free space, obstacle detection and avoidance, path planning and in autonomous driving systems. 2D CNNs [7] for processing 3D LIDAR point cloud is widely used for point cloud data recognition. However, it is not considered an optimal solution since it requires a model to recover the original geometric relationships. Vote3Deep is developed in [8] for fast point cloud object detection from 3D CNN in order to keep the key power of LIDAR as distance and objects’ 3D shapes and depth.

The remainder of this paper is organised as follows. The related work is presented in Section II. Object recognition in scenes from driving sequences of the KITTI dataset is presented in Section III. Learning the LIDAR objects from LIDAR point cloud scans in KITTI dataset are studied in Section IV. We present the manifold alignment formulation and solution between the 3 Dimensional LIDAR space, D camera Space, and 3D V2V becons in Section V. BSM creation according to the LIDAR recognized objects from KITTI suite, number of recognized objects per input type and the performance of the alignment process are illustrated in Section VI. Section VII provides conclusion and future work.

II. RELATED WORK

One approach to understand the scene around vehicles is semantic segmentation that labels each pixel in an image with the category of the belonging objects. Labeling each pixel of the scene independently from its surrounding pixels is a very hard task to achieve. In order to know the category of a pixel, we have to rely on relatively short-range surrounding information and long-range information. In other words, to determine that a certain pixel belongs to a vehicle, a person or to any other class of objects, we need to have a contextual window that is wide enough to show the surroundings of the pixel and consequently to make an informed decision of the object class that contains the pixel. Techniques based on Markov Random Fields (MRF), Conditional Random Field (CRF) and many graphical models are presented in [9], [10], [11] to guarantee the consistency of labeling of the pixels in the context of the overall image. In addition, the authors in [12], [13] and [14] developed various methods for presegmentation into superpixels or segment candidates that are used to extract the categories and features characterizing
individual segments and from combinations of neighboring segments. Alternatively, the authors in [15] attempted to create 3D reconstruction of dynamic scenes by achieving a long-range spatio-temporal regularization in semantic video segmentation, since both the camera and the scene are in motion. The developed idea is to integrate deep CNN and CRF to perform sharp pixel-level boundaries of objects. To this end, deep learning has shown the best performance in inferring objects from not previously trained or seen scenes. Joseph et al. [16] developed a general purpose object detection system characterized by a resolution classifier and the usage of a 2 fully connected networks that are built on top of a 24 convolutional layers network. Additionally, a unified multi-scale deep CNN for real-time object detection is developed in [17] with many sub-network detectors with multiple output layers for multiple object class recognition. The goal of our 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 sensing inputs due to bad conditions (e.g., rainy or foggy weather). Our Framework requires additional prior knowledge about selection of labeled paired objects between the 3 types of data set that we want to correspond. Exploring the physical neighborhood correlation within these three datasets and their natural correspondence in the 3D physical space, we cast the merging problem of these three sets of data as a semi-supervised manifold alignment. Given some clear correspondence between data points from each pair of data sets, we align (i.e., pair) the rest of the points between the camera-LIDAR and camera-V2V data sets. The problem is casted as an eigenvalue problem over a graph-based compounded Laplacian matrix. Once the mapping of known points is done, the other points from each data sets can be easily added in aligned 3D environment, thus significantly enriching the vehicle knowledge of its surroundings.

III. ADAPTING RECOGNITION FROM KITTI CAMERA FEEDS USING DARKNET’S CNN

Inspired by CNN developed in [16], we propose to exploit the feature of Anchor Boxes that 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 CNN extractor. Fig.1 and Fig.3 represent the original frames of the two different driving sequences from KITTI.

Both of frame(a) and frame(b) present random object count per class either for image recognition or for labeled LIDAR objects. Fig.2 and Fig.4 represent the detected and recognized cars and persons in frame(a) and frame(b) respectively. Each detected object’s moment is the same as its surrounding box and is expressed in pixels. We notice that in Fig.2, some vehicles are not detected in the right side of the parked vehicles.

In Fig.4, our detection missed the vehicle next to the one in the background of the image. We will overcome the problem of undetected objects by the alignment and we will no longer have a missing knowledge of surrounding objects.

In Fig.5, we draw a pixel-wise adjacency between objects’ moments, which represents distances in terms of pixels. Unfortunately, this is not an accurate measure since objects might be overlapped and consequently the distance in pixels does not have any significance. For this purpose, we introduce that a paired labeled points between camera and LIDAR is the farthest object in the LIDAR scan and the farthest one being detected in the background of the image.
IV. 3D Objects from LIDAR Point Cloud Data

For simplification purposes, we are not considering every object from the LIDAR since a tremendous number of unknown objects is detected as a set of neighbored point cloud. Most of the recognized object classes from point cloud from 3D CNN or in the labeled data are unknown or do not represent major importance in the alignment. Velodyne LIDAR for both of the frames (a) and (b) are plotted in Fig.6 and in Fig.7, respectively.

![Figure 6. Corresponding Point Cloud for Frame (a).](image6)

The black centered area in both Fig.6 and Fig.7 is the car that is equipped with the 360° Velodyne spinning laser scanner. The circled laser beams represent the free space where no obstacles have been encountered. All cars and people in both of frame (a) and (b) are on the top part of the scans in front the of the vehicle that is equipped with the scanner. We note that the other black areas (at the corners) without the circled beams do not have objects that correspond to zones blocked by obstacles since LIDARs will not draw through objects.

![Figure 7. Corresponding Point Cloud for Frame (b).](image7)

![Figure 8. Adjacency of Detected Objects from LIDAR.](image8)

We note that the manifold representing the adjacent objects from LIDAR for the first frame as in Fig.8 contains larger number of objects comparing to the one from the Camera as in Fig.2. In Fig.8, the dots represent the objects detected from LIDAR with \((x,y,z)\) triplet that represents the relative position to the Velodyne LIDAR Scanner. For example, the vehicle next to the paired point is not detected in Fig.4, but is detected in Fig.8 in addition to other objects that are behind the camera and are captured by LIDAR scans.

V. Semi-supervised Alignment of Manifolds: Camera to LIDAR and Camera to BSMs

Our problem formulation of manifolds alignment is to be applied to find correspondence between source data containing recognized objects from 2D camera and objects from 3D LIDAR point cloud and from 3D V2V exchanged messages. The manifold alignment performs the mapping between the dataset by first successfully learning the low-dimensional embeddings by creating a weighted graph of the objects in the data. This is achieved by finding their correlation while preserving their neighborhood correlation to preserve the local structure of the data. Let \(X\), \(Y\) and \(Z\) be three separate data sets consisting respectively of \(x\), \(y\) and \(z\) recognized points from camera, LIDAR and V2V BSMs. We consider
creating three different Laplacian graphs for each data sets \( \mathcal{X}, \mathcal{Y} \) and \( \mathcal{Z} \). The neighborhood importance between a point \( t^{(i)} \) as a node in every Laplacian graph for each data set is found by solving the following optimization problem:

\[
\arg\min_{W_{ij}} \left\{ \left( t^{(i)} - \sum_{j \in N(i)} W_{ij} t^{(N(i,i))} \right)^2 \right\} \quad \text{s.t. } \sum_{j \in N(i)} W_{ij} = 1
\]

(1)

The distance matrix \( D_t \) of point \( t^{(i)} \) in one of those data sets to its nearest neighbor points in the same data set characterizes the value of the weight and can be represented as:

\[
D_t = \begin{bmatrix}
    t^{(i)} - t^{(N(i,1))} \\
    t^{(i)} - t^{(N(i,2))} \\
    \vdots \\
    t^{(i)} - t^{(N(i,N))}
\end{bmatrix}
\]

(2)

The neighborhood importance between a point \( t^{(i)} \) to another point \( t^{(N(i,i))} \) is proportional with the value on the edge between them, in the new low space representation.

\[
W_{ij} = \frac{\sum_{k=1}^{N} \{(D_i D_i^T)^{-1}\}_{jk} N_{i,N}}{\sum_{m=1}^{N} \sum_{n=1}^{N} \{(D_i D_i^T)^{-1}\}_{mn}}
\]

(3)

We note that \((D_i D_i^T)^{-1}\) is the element of the \( u \)th row and the \( v \)th column in the inverse of matrix \((D_i D_i^T)\).

The problem of the alignment between the camera source data \( \mathcal{X} \) to the destination LIDAR data \( \mathcal{Y} \) and separately to the \( \mathcal{Z} \) destination DSRC data set can be expressed as:

\[
\arg\min_{f,g} \left\{ \lambda^x \sum_{i,j} (f_i - f_j)^2 W_{ij}^x + \lambda^y \sum_{i,j} (g_i - g_j)^2 W_{ij}^y + \mu \sum_{i,j} |f_i - g_i|^2 \right\}
\]

(4)

where \( f = [f_1, \ldots, f_x]^T \) are vectors in \( R^x \) of the \( x \) camera points, while \( g = [g_1, \ldots, g_y]^T \) are vectors in \( R^y \) of the \( y \) LIDAR points and \( g = [g_1, \ldots, g_z]^T \) for \( z \) V2V points. For the different parts, we use the following weighting factors \( \lambda^x, \lambda^y \) and \( \mu \). We consider that \( \mathcal{P} \) is the set that contains the indices of paired points between camera and LIDAR for one alignment and the camera and the V2V BSMs for the other alignment, as shown in Fig 5. We minimize the first term in order to have the larger \( W_{ij}^x \) gets the smaller \( f_i - f_j \) is, which guarantees the preservation of the neighbor relations of \( \mathcal{X} \) camera data set within the elements of \( f \). We note that the same procedure applies to \( g \) for both \( \mathcal{Y} \) LIDAR dataset and \( \mathcal{Z} \) V2V BSMs dataset, while minimizing the second term. The final term in Eq. (4) has an effect to penalize discrepancies between the paired points in the \( f \) vector from \( \mathcal{X} \) and \( g \) vector from both \( \mathcal{Y} \) and \( \mathcal{Z} \). Eq. (4) can be reformulated as:

\[
\arg\min_{f,g} \left\{ \lambda^x f^T L^x f + \lambda^y g^T L^y g + \mu (f - g)^T (f - g) \right\}
\]

(5)

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

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

(6)

while \( L^y = [L^y_{ij}] \) used for camera-LIDAR \( \forall i, j \in \mathcal{Y} \) and \( L^z = [L^z_{ij}] \) used for camera-V2V \( \forall i, j \in \mathcal{Z} \). A hard constraint is imposed so that \( f_i = g_i \) (i.e., equal eigen vectors for the same index of paired points) \( \forall i \in \mathcal{P} \) (i.e., as \( \mu \rightarrow \infty \)). The problem in (5) is transformed into an eigenvalue problem as follows:

\[
\arg\min_{h} \left\{ h^T L^x h \right\}
\]

(7)

\[
s.t. h^T 1 = 0
\]

(8)

with \( Q^x = \mathcal{X}/\mathcal{P} \) and \( Q^y = \mathcal{Y}/\mathcal{P} \) and with taking the differences between the two separate alignments that are applied either in camera-LIDAR or camera-V2V BSMs alignment.

\[
L^x = \begin{bmatrix}
    \lambda^x L^x_{pp} + \lambda^y L^y_{pp} & \lambda^x L^x_{pq} & \lambda^x L^x_{pp} \\
    \lambda^y L^y_{qp} & \lambda^y L^y_{qq} & \lambda^y L^y_{pq} \\
    \lambda^y L^x_{pq} & \lambda^y L^y_{pq} & \lambda^y L^y_{qq}
\end{bmatrix}
\]

(9)

while having \( L^y_{pq} \) (\( L^y_{pq} \) or \( L^z_{pq} \)) is the sub-matrix of the matrix \( L^x(L^y \) or \( L^z \)) depending on both of types of the alignment. The solution of the alignment problem is the eigenvector \( h \) that corresponds to the smallest non-zero eigenvalue of \( L^x \). In addition, \( h \) is structured in a way that it begins with the \( \mathcal{P} \) paired points of \( f \) and \( g \), then it is followed by the remaining data points of \( f \) and ends with the rest of points of \( g \). Depending on the selected \( l \) dimensional embedding chosen for both of the alignment, we will end up with different \( l \) eigenvectors \( [h^{(1)}, \ldots, h^{(l)}] \) after each joint graph Laplacian for the two alignment process. The structure of the embedding of the points of camera-LIDAR points and camera-V2V BSMs points are contained in two different matrices that characterize the expected neighborhoods between the points of the datasets and is given by:

\[
E = \begin{bmatrix}
    f^{(l)}_{P} & \ldots & f^{(l)}_{P} \\
    f^{(l)}_{Q(s)} & \ldots & f^{(l)}_{Q(s)} \\
    f^{(l)}_{Q(s)} & \ldots & f^{(l)}_{Q(s)}
\end{bmatrix}
\]

(10)
VI. NUMERICAL ANALYSIS OF MAPPING ACCURACY AND ERRORS

The lack of timestamped 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 as shown in Fig.9. We consider that the only class of objects that generate BSM are cars. We note that the generated BSMs are simple and do not include the following fields, such as messages count, temporary ID, brake system status and acceleration set 4 way. For position accuracy, we consider using the position \((x,y,z)\) triplet of the object to the Velodyne LIDAR Scanner as a replacement to the real-world positioning from \((\text{Latitude}, \text{Longitude}, \text{Elevation})\).

\[ e = \sum_{i=1}^{x} \frac{c_{ij}}{x} \]

where \(c_{ij} = \begin{cases} 0 & x_i = y_j \text{ or } x_i = z_j \\ 1 & \text{Otherwise} \end{cases} \)

\(c_{ij}\) characterizes if the expected point \(y_j\) from LIDAR data set \(Y\) or \(z_j\) from V2V BSMs data set \(Z\) corresponds to point \(x_i\) from the camera data set \(X\). Fig.11 and Fig.12 present the results of the alignment between camera objects to the corresponding BSM objects respectively for frame\((a)\) and frame\((b)\).

The correspondance of every pair of points between both the camera to LIDAR and the camera to V2V BSMs are given by the lowest value in the final structure of the embedding \(\mathcal{E}\) matrix. We define the overall percentage error in mapping between all the points from \(X\) to \(Y\) and to \(Z\) by the number of erroneous correspondences over the number of points of \(X\) camera data set, as given by the equation:

The circled points in Fig.11 and Fig.12 represent the objects from camera that were not aligned with their correspondances in V2V BSMs data set. Enriched points in the V2V BSMs objects that do not have correspondences are enriching the 3D reconstruction of the scene. These objects are either out-of-sight of the camera or that were not recognized.

![Figure 9](image9.png)

**Figure 9.** Adjacency of Recognized Objects from DSRC.

![Figure 11](image11.png)

**Figure 11.** Camera-V2V Object Alignment Frame (a).

![Figure 12](image12.png)

**Figure 12.** Camera-V2V Object Alignment Frame (b).

![Figure 10](image10.png)

**Figure 10.** Number of Objects per Class and Frame.
Fig.13 and Fig.14 show the alignment between the objects retrieved from respectively from camera’s frame(a) and frame(b) with their corresponding objects from the point cloud.

Specifically, the points in circles were not mapped to any of those from the camera since they have low correlation with the neighboring points and are additive points from the LIDAR point cloud for both of the frames corresponding to both undetected or out-of-sight objects that were not detected by the frames. We notice that the mapping of the objects recognized from the camera to the ones from the LIDAR is more accurate and present more matching than to the V2V objects.

The enriched objects per technology after the mapping for both of the scene are given by the Table I. It represents the objects of the autonomous vehicle scene that are specific to the technology and that were not aligned between the data sets.

### VII. CONCLUSION

We developed a framework for corresponding between objects recognized from camera data set to LIDAR and to DSRC data sets that are characterized with the same underlying manifold. The mapping allows us to be more informed about one object that was paired between one or two different sensor inputs. Therefore, we gain in terms of textural details from RGB pictures, hidden information from V2V communication as well as 3D shape and accurate distance to surroundings through the 3D point cloud.

### REFERENCES

[1] “Cyber-physical systems (cps), program solicitation.” http://www.nsf.gov/pubs/2017/nsf17529/nsf17529.htm#toc. Published 2017.

[2] H. Somerville and G. Cherelus, “Uber resumes self-driving program three days after arizona crash.” http://mobile.reuters.com/article/daily-business-v2v-uber-arizona-crash. Published March 27, 2017.

[3] R. Grenoble, “Self-driving uber blows through red light on first day in san francisco.” http://www.huffingtonpost.com/2016/12/01/self-driving-uber-san-fran-red-light-video_us_58519c9c5e4b0732b82cedb01. Published December 14, 2016.

[4] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset.” Int. J. Rob. Res., vol. 32, pp. 1231–1237, Sept. 2013.

[5] Y. Maalej, A. Abderrahim, M. Guizani, B. Hamdaoui, and E. Balti, “Advanced activity-aware multi-channel operations1609.4 in vanets for vehicular clouds,” in 2016 IEEE Global Communications Conference (GLOBECOM), pp. 1–6, Dec 2016.

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

[7] V. Hegde and R. Zadeh, “Fusionnet: 3d object classification using multiple data representations,” CoRR, vol. abs/1607.05695, 2016.

[8] M. Engelcke, D. Rao, D. Zeng Wang, C. Hay Tong, and I. Posner, “Vote3Deep: Fast Object Detection in 3D Point Clouds Using Efficient Convolutional Neural Networks,” ArXiv e-prints, Sept. 2016.

[9] X. He and R. S. Zemel. “Learning hybrid models for image annotation with partially labeled data,” in Advances in Neural Information Processing Systems 21 (D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, eds.), pp. 625–632, Curran Associates, Inc., 2009.

[10] L. Ladicky, C. Russell, P. Kohli, and P. H. S. Torr, “Associative hierarchical crfs for object class image segmentation,” in 2009 IEEE 12th International Conference on Computer Vision, pp. 739–746, Sept 2009.

[11] D. Munoz, J. A. Bagnell, and M. Hebert, “Stacked hierarchical labeling,” in European Conference on Computer Vision, pp. 57–70, Springer, 2010.

[12] J. Tighe and S. Lazebnik, “Superparsing: Scalable nonparametric image parsing with superpixels,” in Proceedings of the 11th European Conference on Computer Vision: Part V, ECCV’10, (Berlin, Heidelberg), pp. 352–365, Springer-Verlag, 2010.

[13] M. P. Kumar and D. Koller, “Efficiently selecting regions for scene understanding,” in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3217–3224, June 2010.

[14] S. Gould, R. Fulton, and D. Koller, “Decomposing a scene into geometric and semantically consistent regions,” in 2009 IEEE 12th International Conference on Computer Vision, pp. 1–8, Sept 2009.

[15] A. Kundu, V. Vineet, and V. Koltun, “Feature space optimization for semantic video segmentation.” in CVPR, 2016.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[17] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, “A unified multi-scale deep convolutional neural network for fast object detection,” CoRR, vol. abs/1607.07155, 2016.

| Data set | Car | Person |
|----------|-----|--------|
| Camera   | 0   | 5      |
| LIDAR    | 6   | 0      |
| V2V      | 6   | 6      |
| BSMs     | 0   | 0      |

Table I. Enriched Objects from the 3 Different Modalities of Sensors.