Super-sensor for 360-degree Environment Perception: Point Cloud Segmentation Using Image Features

Robert Varga, Arthur Costea, Horaţiu Florea, Ion Giosan, Sergiu Nedevschi
Computer Science Department
Technical University of Cluj-Napoca, Romania
firstName.lastName@cs.utcluj.ro

***Abstract*—This paper describes a super-sensor that enables 360-degree environment perception for automated vehicles in urban traffic scenarios. We use four fisheye cameras, four 360-degree LIDARs and a GPS/IMU sensor mounted on an automated vehicle to build a super-sensor that offers an enhanced low-level representation of the environment by harmonizing all the available sensor measurements. Individual sensors cannot provide a robust 360-degree perception due to their limitations: field of view, range, orientation, number of scanning rays, etc. The novelty of this work consists of segmenting the 3D LIDAR point cloud by associating it with the 2D image semantic segmentation. Another contribution is the sensor configuration that enables 360-degree environment perception. The following steps are involved in the process: calibration, timestamp synchronization, fisheye image unwarping, motion correction of LIDAR points, point cloud projection onto the images and semantic segmentation of images. The enhanced low-level representation will improve the high-level perception environment tasks such as object detection, classification and tracking.**

Keywords—automated driving; environment perception; fisheye images; 3D LIDAR points; 360-degree perception; super-sensor

I. Introduction

The Society of Automobile Engineers (SAE) classifies driving into: driver only, assisted, partial automation, conditional automation, high automation and full automation corresponding to levels of automation from 0 (driver only) to 5 (fully automated) [1]. Level 3 of automation allows the human driver to do other activities while driving, whereas, levels 4 and 5 consider a complete adoption of the driving process by the vehicle while the driver is even able to fall sleep.

On a commercial level the automotive industry has already reached quite an advanced level, proved by the smart driving assistance technologies including adaptive cruise control, lane departure warning, and lane keeping assistance that come integrated with many new vehicles. Although the driver still has to interact with the vehicle, the previously mentioned technologies represent a significant step towards automated driving. It seems clear that a combination of adaptive cruise control with lane keeping assistance and an advanced environment perception will lead to an automated driving solution in the future.

On a research and pre-development level, vehicle manufacturers and research companies have organized automated driving demonstration events. For instance, Google presented a self-driving car. Daimler drove the route from Mannheim to Pforzheim with an automated driving prototype car. Renault demonstrated an automated valet parking technology on an electric vehicle, performing a drive along a controlled road.

Thus, while the levels of automation 0 to 2 are available on the market, intensive research is performed for levels 3 to 5, and in particular for the development of key base technologies that will enable them. One such technology is a sensor capable of robust 360-degree environment perception.

In this paper, we introduce the idea of building a super-sensor which is necessary for 360-degree environment perception for automated vehicles. In chapter II, we review the state of the art in single- and multi-modal perception and demonstrate the importance of the omnidirectional perception in developing automated and fully automated vehicles. In chapter III, all the available sensors are individually presented. In chapter IV, the building process and the data representation of the super-sensor is described. In chapter V, all the experimental results are shown. Finally, chapter VI draws the conclusions.

II. Related Work

Environment perception in automated and autonomous driving applications refers mainly to object detection, tracking and classification in the driving environment. The key elements to achieve these tasks are a redundant, robust, accurate and multimodal sensorial system providing a 360-degree coverage of the vehicle surrounding.

In this regard, the use of 3D passive sensors, represented by stereo systems, in combination with other modalities is currently in development. A single stereo sensor provides a significantly larger volume of information than other sensors and offers at least three different, but aligned modalities: depth, optical flow and grey level intensities. The fusion of these modalities increases the dimensionality of low level representation and, by consequence, the quality of detection, tracking and classification. In contrast, a 6D-vision approach [2] computes the 3D scene points and their associated 3D motion vectors. The exact position, moving direction and speed for each pixel is determined, offering the possibility of predicting the future positions. Rectangular elements like stixels that adapt very well to the objects in the traffic scene may be used for obtaining an optimized scene representation. Stixel motion estimation and

tracking across multiple frames is achieved by using the 6D-vision approach.

Concerning representation, in [3] the concept of classified elevation map was introduced allowing the navigable channel detection and accurate objects separation even in unstructured or crowded environments. The attributed polygonal lines are used for compact object representation suitable together with optical flow for tracking and classification tasks [4]. Superpixels on gray levels are used together with depth and optical flow for segmenting the obstacles in traffic scenarios [5]. 3D Voxel concepts [6] allow the detection even of hanging objects and can benefit from the optical flow or motion vectors for tracking purposes. Unfortunately, none of these approaches offers an exhaustive solution to the perception problem. There are uncertainties in the acquisition process, in the measurements of the sensors and in the models employed, a robust system must implement a higher level of redundancy of the sensors, a 360-degree coverage of the vehicle environment, an early low level fusion of the sensor data along with better detection, tracking and classification approaches.

There are several approaches presented in the literature regarding the problem of correcting the distortion of 3D LIDAR data when scanning the environment from a moving platform. The SLAM methods presented in [7] and [8] perform the correction in the process of integrating new scans in the global map. Both algorithms use features detected in the scans (edge lines, surfaces) to perform a matching between them and estimate the ego-motion of the vehicle. Another approach is presented in [9], where the authors also provide a solution for correcting the position of non-static objects by using a tracking algorithm.

Our approach stems from the work presented in [10] which employs the Iterative Closest Point algorithm applied on scan points to determine the ego-motion of the sensor platform. This approach is susceptible to errors when the scene contains moving elements such as pedestrians or other vehicles. We adapted the algorithm to use the data from other position sensors, and to correct the data to an arbitrarily chosen timestamp.

The papers [11] and [12] present different approaches for 3D point cloud segmentation. We will focus on methods that rely on multimodal information such as color and laser points. In [13] a graph-based method is proposed for color laser point segmentation. Split criterion is based on position, color and surface normal components. In [14] and [15] segmentation approaches are presented for UAV images enhanced with 3D point clouds. The paper [16] shows that including reflectance as an additional feature for segmentation significantly improves the results. In [17] Markov Random Fields are utilized for segmentation. The works [18] and [19] focus on surveying and reconstruction from RGB + 3D data. In [20] 3D reconstructed points from monocular image streams are enhanced with semantic classes. All methods relying on fusion between color and LIDAR perform segmentation in the 3D space. To the best of our knowledge, there are no approaches that transfer segmentation results from the image space onto 3D points.

III. Sensors, Calibration and Time Synchronization

This chapter describes the available sensors mounted on the vehicle. The calibration provides relative positions and orientations of LIDARs and fisheye cameras with respect to the vehicle reference coordinate system which coincides with the GPS/IMU reference frame. High accuracy calibration of each sensor is needed to achieve high quality of the super-sensor data. The available data coming from all sensors is harmonized using the calibration results to a common timestamp.

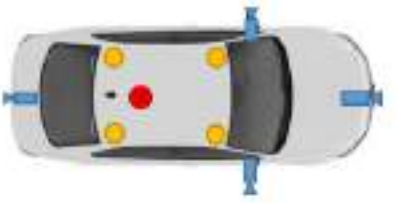

Fig. 1. Autonomous vehicle equipped with sensors: 4 fisheye cameras (blue); 4 LIDARs (yellow) and GPS/IMU (red)

A. Fisheye cameras

Four cameras are mounted on the vehicle, one in each separate direction: front, right, rear, left (see Fig. 1). This system of cameras delivers color images of 1280x800 pixel resolution, JPEG compressed, at maximum 30 fps. The front and rear cameras are mounted horizontally. The side cameras are tilted downwards. Due to the large horizontal field of view (190 degrees) of each camera, the system offers 360-degree coverage of vehicle surroundings with some overlap between the neighboring cameras. The disadvantage of the large field of view is that objects even at moderate distances appear small in the image.

The calibration process determines the intrinsic and extrinsic camera parameters. The extrinsic parameters are then used to register the cameras into a common coordinate system of a selected 360-degree LIDAR (master).

We use the unified projection model proposed by Geyer in [21] and adapted for fisheye lenses in [22]. The model uses only 9 parameters: ξ – mirror parameter; k_1, k_2, p_1, p_2 – radial and tangential distortion coefficients; f_u, f_v – horizontal and vertical focal distance in pixels; c_u, c_v – coordinates of the principal point in pixels. In the following, we define the forward projection function.

Let $\boldsymbol{X_c}$ be a 3D point in the camera reference frame, in non-homogeneous space:

$$\boldsymbol{X_c} = [X \quad Y \quad Z]^T$$

and let ρ be its distance from the camera center:

$$\rho = \sqrt{X^2 + Y^2 + Z^2}.$$

The first step is projecting the point onto the normalized image plane:

$$\boldsymbol{X} = \left[\frac{X}{Z + \xi\rho} \quad \frac{Y}{Z + \xi\rho} \quad 1\right]^T = [x \quad y \quad 1]^T$$

Next, radial and tangential distortion is applied onto the x and y components:

$$x_d = x(1 + k_1 r^2 + k_2 r^4) + 2p_1 xy + p_2(r^2 + 2x^2)$$

$$y_d = y(1 + k_1 r^2 + k_2 r^4) + 2p_2 xy + p_1(r^2 + 2y^2)$$

The final projection into pixel coordinates is obtained via multiplication with the internal matrix K in 2D homogeneous coordinates:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & c_u \\ 0 & f_v & c_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix}$$

The projection function P is then defined as the function applied on the 3D point and having as output the $[u\ v]$ pixel coordinates:

$$P(\boldsymbol{X}_c) = [u\ v\ 1]^T$$

B. 360-degree LIDARs

In order to increase the number of scanning planes and to achieve surround coverage of the environment, four 360 degree LIDARs are mounted on top of the vehicle (see Fig. 1). Each performs scans along 16 planes, covering 30 degrees in vertical direction and about 100m in depth. The update rate is set to 100ms (10 fps) and the expected accuracy is +/- 3cm. With this setup, a coverage of at least one scanning plane per degree is guaranteed for each side of the vehicle.

The calibration process determines the extrinsic parameters for all the LIDARs with respect to the chosen master. Then, the master sensor coordinate system is registered with respect to the standard vehicle coordinate system.

C. GPS/IMU

The vehicle also is equipped with a sensor capable of providing precise inertial and GPS data for measuring ego-motion position and orientation (see Fig. 1). The resolutions of the navigation parameters are: 0.01m in position, 0.05 km/h in speed, 0.03 deg. in pitch/roll angle, and 0.1 deg. in heading angle.

This sensor is also used for acquiring a universal timestamp for all sensor measurements. This is given by the GPS time and it is used for synchronizing data coming asynchronously from different sensors. Timestamp synchronization and ego-motion estimation is crucial for building the super-sensor.

IV. Building the Super-Sensor

The super-sensor perceives the environment like it has the capabilities of all available sensors mounted on the vehicle (cameras, LIDARs, GPS/IMU). Its measurements should be similar to the individual sensors measurements fused together at a low level. This low-level representation is generated by a four phase process:

- 3D points motion correction
- Image undistortion and unwarping
- Semantic segmentation
- Points projection and information fusion

The low-level representation consists of a multi-dimensional feature vector with the following features:

- Pixel position: (u, v) coordinates in the image
- Pixel color information: (R, G, B) values
- Pixel 2D optical flow vector: (du, dv) displacements computed on u and v axes
- Pixel *semantic segment* value
- Pixel *class* value
- 3D point position (X, Y, Z)
- 3D point velocity vector (Vx, Vy, Vz)
- 3D point class value (*3Dclass*)

In this work we only present the transfer of the pixel-level semantic segment value to its corresponding 3D point. The super-sensor and its associated low-level representation represents a scientific contribution that adds value and increases the accuracy of the high-level processing tasks like object detection, tracking and classification. This representation may be generated and used when all the available sensors are functioning. If there are some malfunctioning sensors, the high-level processing may be done by only using the information coming from either 3D sensors or image sensors that are operating normally.

In the following sections, all the necessary steps towards building the super-sensor and its low-level representation are presented.

A. 3D points motion correction

The movement of the ego vehicle introduces a distortion in the measured data, as the sensor's reference frame changes its position during the scan period. Thus, the coordinates of scanned points must be adjusted based on the movement, not only to ensure proper representation of the environment at a single moment in time, but also in order to synchronize the LIDAR data with data captured by the other sensors.

The synchronization timestamp is chosen to coincide with the timestamp of the most recent camera frame, which allows us to correctly project the points onto the images (we assume that all four fisheye cameras are synchronized). We denote the i^{th} point taken at time t_i from a generic laser as $\overline{\boldsymbol{X}}_{L,i} = [X\ Y\ Z\ 1]^T$, for which we define the parameter Δ_i as:

$$\Delta_i = \tau - t_i$$

where τ is the target timestamp taken from the most recent image frame.

The transformation that is used to correct individual points is computed from the ego-motion transform matrix of the vehicle, T_{Ego}. This encompasses the 6 degrees of freedom motion during the period Δ_0 and is represented as a 4x4 matrix.

Before applying the correction, the scanned point must first be represented in the vehicle's coordinate system, based on the sensor's extrinsic parameters, because this coincides with the coordinate system of the T_{Ego} transform. Thus, the ith point can be corrected using:

$$\widetilde{\boldsymbol{X}}_{L,i} = (T^L_{veh})^{-1} \cdot C_i \cdot T^L_{veh} \cdot \overline{\boldsymbol{X}}_{L,i}$$

where T_{veh}^{L} is the LIDAR-to-vehicle coordinate system transform and C_i is the correction transform for point i, which is computed from the T_{Ego} transform by raising it to a fractional power dependent on the Δ_i value:

$$C_i = T_{Ego}^{-\Delta_i/\Delta_0} = \exp(-\Delta_i/\Delta_0 \cdot \log(T_{Ego}))$$

Here, we apply the notions of matrix exponential and logarithmic functions as presented in [23] in order to compute this transform. For the implementation we use algorithms integrated in the Eigen linear algebra library. This correction scheme is valid irrespective of the temporal relation between the target timestamp and the start or end time of the scan.

The correction transform must be computed for each individual point of a scan for achieving accurate results. However, due to the high volume of data and the need for very fast processing, the process can be sped up by computing a lookup table of correction transforms for each scan.

B. Image undistortion and unwarping

The raw fisheye image is not used in practice for further processing since real world objects are highly distorted in the image. Once the forward projection model of the camera and lenses are known the image can be transformed to obtain a more suitable representation of the scene.

To obtain an undistorted and unwarped image we start from a target surface which represents a virtual imager. The surface is discretized and projected onto the distorted image. Color values can be obtained for each position by interpolating values from the original fisheye image. Either bilinear or bicubic interpolation can be performed. The described procedure clearly shows that there is no need for the inverse projection model.

The surface of projection can be: a single plane; multiple planes or a cylinder. The field of view, the resolution and the orientation of the surface needs to be determined to offer the optimal view of the scene. Knowing the camera orientation enables us to preserve the orientation of vertical lines. This is essential for higher level processing steps.

Using a single plane yields a single perspective image. Perspective images preserve straight lines, but for larger field of view it elongates objects at positions which are far from the image center. This kind of unwarping is convenient for the central region and should be applied with a reduced field view. A horizontal field of view close to 180 degrees is impossible to achieve with a single plane. Projecting onto multiple planes can resolve this issue.

We define the planar surface for projection as a planar grid placed at distance z=1 from the camera parameterized by *u* and *v* in the following way:

$$\begin{cases} x(u,v) = -t_h + 2t_h u/(w-1) \\ y(u,v) = -t_v + 2t_v v/(h-1) \\ z(u,v) = 1 \end{cases}$$

where $t_h = \tan(\alpha/2)$, $t_v = t_h w/h$, α is the horizontal field of view and *w*, *h* are the horizontal and vertical resolutions respectively and $u \in \overline{0:w-1}$, $v \in \overline{0:h-1}$. This implies that $x \in [-t_h, t_h]$ and $y \in [-t_v, t_v]$.

Another option is to use the side of a cylinder as the projection surface. The cylinder should have its axis aligned with the normal to the ground surface in order to preserve the orientation of vertical lines. Cylindrical unwarping can generate a single image with large horizontal field of view with smaller distortions than a single perspective image. Only vertical lines remain straight after this transformation.

We define the cylindrical surface as the face of a cylinder having its rotation axis aligned with the y axis and having a base radius of 1:

$$\begin{cases} x(u,v) = \sin(-\alpha + \alpha \cdot u/(w-1)) \\ y(u,v) = -\beta + \beta \cdot v/(h-1) \\ z(u,v) = \cos(-\alpha + \alpha \cdot u/(w-1)) \end{cases}$$

where α is the horizontal field of view and $\beta = \alpha h/w$. We define rotation matrices for each camera which perform the required rotation from the vehicle reference frame to the specific camera while preserving alignment with the vehicle reference frame (only 90-degree rotations):

$$\hat{R}_{camf}^{veh} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix} \quad \hat{R}_{caml}^{veh} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\hat{R}_{camb}^{veh} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix} \quad \hat{R}_{camr}^{veh} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

C. Semantic segmentation

We propose to segment the undistorted fisheye images in order to enable the semantic perception of the surrounding environment. The result of the segmentation is the pixel-wise labeling of images using representative semantic classes for traffic scenarios. Due to the unwarping of the fisheye images we can employ any segmentation solution designed for undistorted non-fisheye images and can use any available dataset for training purposes. To achieve robust segmentation results it is important to have a large and diverse training dataset with images that were labeled manually. The Cityscapes database [24] is an ideal training dataset, considering that it contains 25000 manually labeled images captured from traffic scenarios from 50 different cities.

The literature on semantic segmentation is extensive. There are several solutions providing different classification performances at different computational costs. We opt for a boosting-based solution based on our segmentation approach proposed in [25], which provides a good trade-off between segmentation performance and computational costs. Due to the different nature of depth from the Cityscapes training dataset (from stereo reconstruction) and depth from LIDAR, we employ only color image features and ignore depth features. Manual labelling of training images together with depth data (from LIDAR) will enable the learning of depth features for classification. To further reduce classification costs, we use only the 7 Cityscapes category classes as semantic labels (instead of the 19 classes at finer level): ground, sky, nature, construction, object, human and vehicle.

To generate image features the input images are decomposed into multiple channels, consisting of LUV color channels, gradient magnitude and six orientation channels. In order to

capture color and edge features at different scales and orientations a multiresolution filtering scheme is applied over the image channels resulting in multiresolution filtered channels. The filtering scheme is detailed in [25].

The multiresolution filtered channels are extended with deep convolutional neural network (CNN) channels, as described in [25]. These channels are obtained by applying deep CNN kernels over the input image. These kernels are learning-based and are able to capture different complex structures. The best results were achieved using LUV color and gradient channels together with the last 512 convolutional filters of the 4th layer (conv4_3) of the ImageNet pretrained VGG16 neural network [26].

The purpose of the multiresolution filtered channels and deep convolutional channels is to provide classification features for semantic pixel classifiers. A boosted decision forest is learned for each semantic class using multirange channel features as classification features (see details in [25]). The multirange features consist of channel values at different relative spatial positions and the boosting algorithm has the role of selecting the most relevant features for each class.

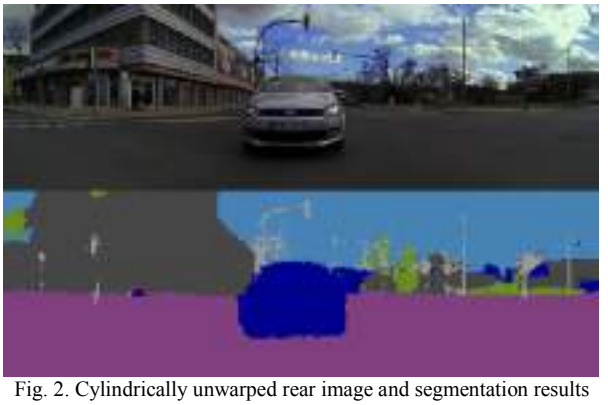

Fig. 2. Cylindrically unwarped rear image and segmentation results

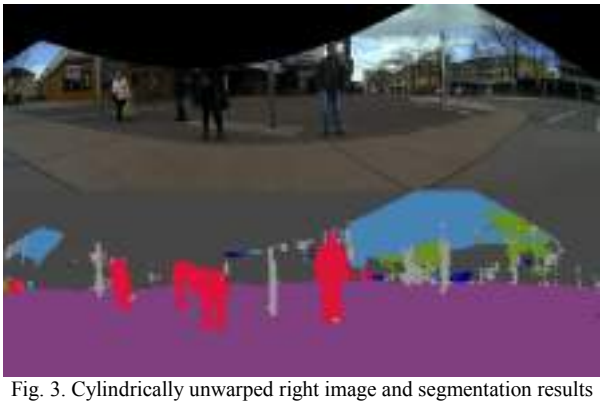

Fig. 3. Cylindrically unwarped right image and segmentation results

Semantic segmentation is achieved by computing the multiresolution filtered channels and deep CNN channels, and by applying the learned pixel classifiers. The input image is segmented into superpixels and only the center pixel of each superpixel is used for classification, resulting in a significant reduction of necessary classifications. The classification results are retained at superpixel level and are further refined using a dense conditional random field (CRF) [27]. Due to the low computational costs the multiclass labelling of the input image using 7 semantic classes is achieved at 100 ms using an NVidia GTX 980Ti GPU. In Fig. 2 and Fig. 3 we illustrate the semantic segmentation of the rear and front fisheye images.

D. Points projection and information fusion

Information fusion between the any laser and any camera can be obtained by projecting the 3D points from the LIDAR onto the imager of the camera. In the following we define the steps required to project onto the original fisheye images, the planar image and cylindrical image.

Let $\overline{\boldsymbol{X}}_L$ be a 3D point in the native coordinate system of a general laser sensor, in homogeneous coordinates:

$$\overline{\boldsymbol{X}}_L = [X \;\; Y \;\; Z \;\; 1]^T$$

Projection onto the fisheye image can be obtained by:

$$[u \; v \; 1]^T = P(T_{cam}^L \overline{\boldsymbol{X}}_L)$$

$$T_{cam}^L = \begin{bmatrix} R_{cam}^{veh} & T_{cam}^{veh} \\ \mathbf{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} R_{veh}^L & T_{veh}^L \\ \mathbf{0} & 1 \end{bmatrix},$$

i.e. we apply the projection function onto the point transformed to the camera coordinate system. The transformation from vehicle to camera uses the extrinsic rotation and translation found during calibration.

Projection onto the planar image aligned with the car reference frame can be obtained by:

$$[u \; v \; 1]^T = K \cdot nonh(\hat{T}_{cam}^L \overline{\boldsymbol{X}}_L)$$

$$K = \begin{bmatrix} w/(2\,t_h) & 0 & w/2 \\ 0 & h/(2\,t_v) & h/2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\hat{T}_{cam}^L = \begin{bmatrix} \hat{R}_{cam}^{veh} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \cdot \begin{bmatrix} R_{veh}^L & T_{veh}^L \\ \mathbf{0} & 1 \end{bmatrix}.$$

The function *nonh*() switches to non-homogeneous coordinates and afterwards we normalize the coordinates by the third component. Here, the transformation from vehicle to camera uses the rotation matrices defined for aligned views and no translation.

Projection onto the cylindrical image aligned with the car reference can be obtained by:

$$[u \; v \; 1]^T = K \cdot g(\hat{T}_{cam}^L \overline{\boldsymbol{X}}_L)$$

The internal matrix K has same form as defined previously but in this case t_h equals half the horizontal field of view. The function g finds the 3D point which is the intersection of a ray passing through the point $[X \; Y \; Z]$ and the face of the cylinder aligned with the vehicle reference frame. It can be shown the g has the following form:

$$g([X \; Y \; Z \; 1]^T) = [\text{asin}(X/r) \;\; Y/r \;\; 1]^T$$

$$r = \sqrt{X^2 + Z^2}\,.$$

Once the coordinates of the 3D point are known, information can be transferred from the image domain to the 3D points. We augment the point cloud with color information and with the class of the semantic segment.

It is important to note that cameras and lasers view the world from different viewpoints and so there can be cases where the laser measures distances to objects which are occluded in the camera view. Projecting such 3D points onto the image will result in erroneous associations with the occluding object. Resolving this issue can be achieved by considering consistency in color, making use of laser reflectivity or by reasoning in 3D space.

V. Experimental Results

A. 3D points motion correction

Fig. 4 and Fig. 5 illustrate 3D points projected onto the fisheye image. The points' color represents the distance to the camera. The effectiveness of the motion correction algorithm can be observed by comparing the two figures, with noticeable improvements visible especially on thin vertical objects (poles and trees). The uncorrected data appears shifted to the left. Also, points measured by different LIDARs at different timestamps do not overlap each other (see Fig. 4). These issues are solved by the motion correction (see Fig. 5).

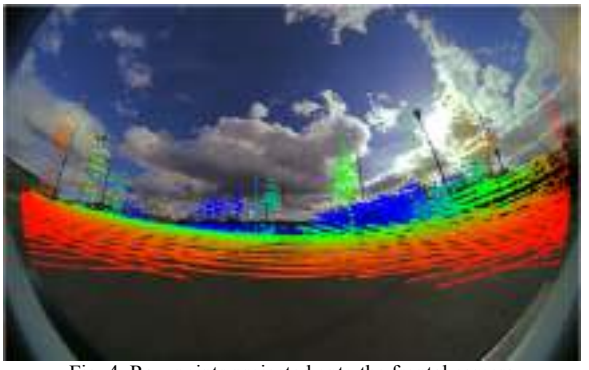

Fig. 4. Raw points projected onto the frontal camera

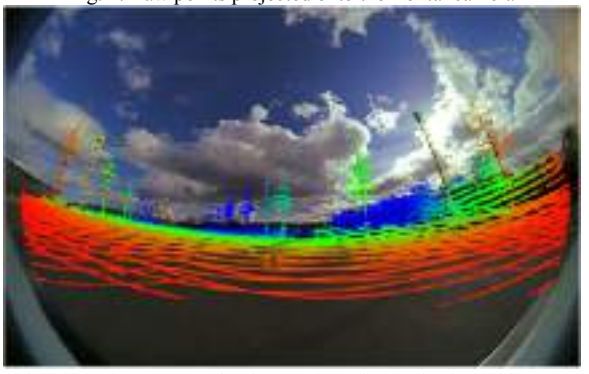

Fig. 5. Motion-corrected points projected onto the frontal camera

B. Image undistortion and unwarping

The following figures (Fig. 6 and Fig. 7) illustrate unwarping results for a single planar surface and a cylindrical surface. We also show the effect of alignment on vertical lines.

Fig. 6. Planar unwarping, 100-degree horizontal field of view; left: surface not aligned to vehicle; right: surface aligned to vehicle

Fig. 7. Cylindrical unwarping, 160-degree horizontal field of view; left: surface not aligned to vehicle; right: surface aligned to vehicle

C. Semantic segmentation

We evaluate the performance of the semantic segmentation using the Cityscapes [24] traffic scene validation set. We measure the classification performance for 7 semantic categories. As performance metric, we compute the intersection over union (IoU) for each individual class representing the number of true positive pixels divided by the sum of the number of true positive, false positive and false negative pixels. The employed segmentation approach achieves a mean IoU of 70.5%, mean accuracy (average class-level true positive rate) of 81.8% and global accuracy (pixel-level true positive rate) of 90.8% over the validation set. The class-level IoU is provided in Table I.

TABLE I. Intersection over Union score (%) for different semantic classes evaluated on the Cityscapes validation set

Flat	Nature	Object	Sky	Construction	Human	Vehicle	Mean IoU	Mean Acc.	Global Acc.
96	82	35	84	77	51	69	70.5	81.8	90.8

D. Points projections and information fusion

Fig. 8 and Fig. 9 show sample segmented point clouds along with the view offered by the frontal camera. The road plane is visible (magenta). The main traffic participants can also be distinguished: pedestrians (red), vehicles (blue), vegetation (green), buildings (gray).

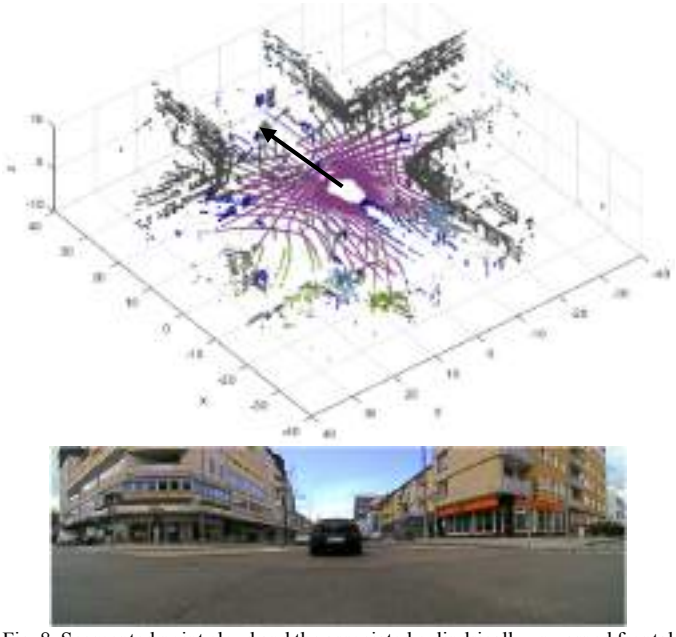

Fig. 8. Segmented point cloud and the associated cylindrically unwarped frontal view. The arrow indicates the orientation of the car.

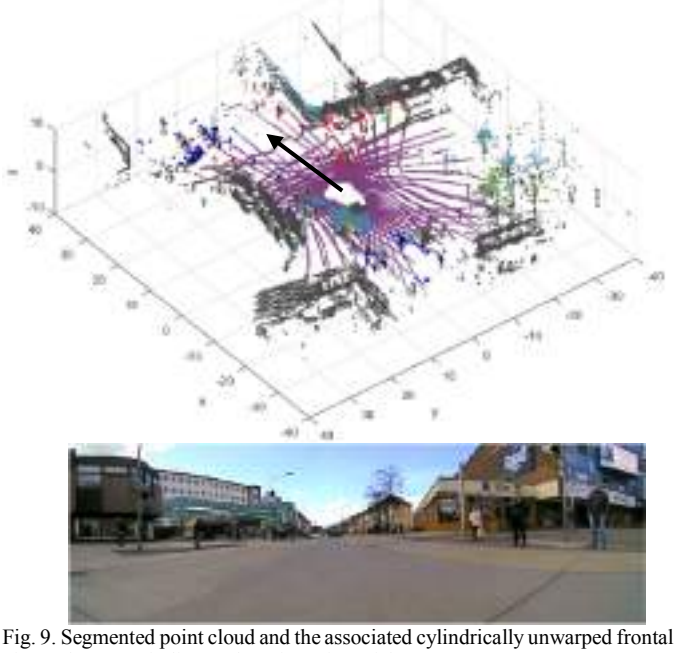

Fig. 9. Segmented point cloud and the associated cylindrically unwarped frontal view. The arrow indicates the orientation of the car.

VI. Conclusions

This work presented a solution for building a super-sensor from fisheye cameras and laser scanners. For a correct representation of the environment, all sensors must be aligned to a single moment in time. Motion correction is thus crucial and we have shown how to adjust the 3D LIDAR point cloud to the acquisition time of the cameras.

Afterwards, three different image unwarping methods were described. Cylindrically unwarped images offer the advantage of a large field of view at the cost of slight distortions. We have shown that these images can be used as input for powerful and fast semantic segmentation approaches that work in the image domain. The semantic classes were transferred onto the accurate 3D LIDAR point cloud. The transfer was also made in the other direction by providing depth information to image processing methods. The segmentation approach was evaluated on the popular Cityscapes dataset. The global pixel-level accuracy is over 90% and requires only 100ms for the full resolution image. Performing an annotation and training on cylindrical images should further increase the accuracy values.

The presented approach was integrated and tested on a real vehicle. The current paper establishes the required steps for obtaining the low-level fusion of LIDAR and camera data. More high-level processing steps are future work and will build upon this representation.

Acknowledgment

This work has been supported by the UP-Drive project (Automated Urban Parking and Driving), Horizon 2020 EU funded, Grant Agreement Number 688652. We would like to thank project partners Volkswagen for providing the vehicle infrastructure and Technical University of Prague for providing the calibration of the sensors.

References

[1] European Technology Platform on Smart Systems Integration (EPoSS). (2015). *European Roadmap „Smart Systems for Automated Driving"*. Available: http://www.smart-systems-integration.org/public/news-events/news/eposs-roadmap-smart-systems-for-automated-driving-now-published

[2] U. Franke, C. Rabe, H. Badino, and S. Gehrig, "6D-vision: fusion of stereo and motion for robust environment perception," presented at the Proceedings of the 27th DAGM conference on Pattern Recognition, Vienna, Austria, 2005.

[3] F. Oniga, S. Nedevschi, M. M. Meinecke, and T. Thanh-Binh, "Road Surface and Obstacle Detection Based on Elevation Maps from Dense Stereo," in *IEEE Intelligent Transportation Systems Conference*, 2007, pp. 859-865.

[4] S. Nedevschi, A. Vatavu, F. Oniga, and M. M. Meinecke, "Forward collision detection using a Stereo Vision System," in *2008 4th International Conference on Intelligent Computer Communication and Processing*, 2008, pp. 115-122.

[5] I. Giosan and S. Nedevschi, "Superpixel-based obstacle segmentation from dense stereo urban traffic scenarios using intensity, depth and optical flow information," in *IEEE 17th International Conference on Intelligent Transportation Systems (ITSC)*, 2014, pp. 1662-1668.

[6] A. Broggi, S. Cattani, M. Patander, M. Sabbatelli, and P. Zani, "A full-3D voxel-based dynamic obstacle detection for urban scenario using stereo vision," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2013, pp. 71-76.

[7] F. Moosmann and C. Stiller, "Velodyne SLAM," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 393-398.

[8] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Robotics: Science and Systems*, 2014.

[9] J. Rieken and M. Maurer, "Sensor scan timing compensation in environment models for automated road vehicles," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 635-642.

[10] S. Hong, H. Ko, and J. Kim, "VICP: Velocity updating iterative closest point algorithm," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 1893-1898.

[11] A. Nguyen and B. Le, "3D point cloud segmentation: A survey," in *2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, 2013, pp. 225-230.

[12] P. P. Sapkota. (2008, Segmentation of coloured point cloud data. *International Institute for Geo-Information Science and Earth Observation*.

[13] J. Strom, A. Richardson, and E. Olson, "Graph-based segmentation for colored 3D laser point clouds," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 2131-2136.

[14] A. Vetrivel, M. Gerke, N. Kerle, and G. Vosselman, "Segmentation of UAV-based images incorporating 3D point cloud information," in *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Munich, Germany, 2015, pp. 261-268.

[15] G. Vosselman, "Point cloud segmentation for urban scene classification," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Antalya, Turkey, 2013, pp. 257-262.

[16] A. Aijazi, P. Checchin, and L. Trassoudaine, "Segmentation Based Classification of 3D Urban Point Clouds: A Super-Voxel Based Approach with Evaluation," *Remote Sensing,* vol. 5, pp. 1624-1650, 2013.

[17] J. R. Schoenberg, A. Nathan, and M. Campbell, "Segmentation of dense range information in complex urban scenes," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 2033-2038.

[18] A. Abdelhafiz, B. Riedel, and W. Niemeier, "Towards a 3D true colored space by the fusion of laser scanner point cloud and digital photos," in *ISPRS Working Group V/4 Workshop 3D-ARCH*, 2005.

[19] T. Abmayr, F. Härtl, M. Mettenleiter, A. Heinz, B. Neumann, and C. Fröhlich, "ISPRS Internation Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences," 2004, pp. 198-203.

[20] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg, "Joint semantic segmentation and 3d reconstruction from monocular video," in *European Conference on Computer Vision*, 2014, pp. 703-718.

[21] C. Geyer and K. Daniilidis, "A Unifying Theory for Central Panoramic Systems and Practical Implications," in *Computer Vision — ECCV 2000: 6th European Conference on Computer Vision Dublin, Ireland, June 26–July 1, 2000 Proceedings, Part II*, D. Vernon, Ed., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 445-461.

[22] J. Courbon, Y. Mezouar, L. Eckt, and P. Martinet, "A generic fisheye camera model for robotic applications," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 1683-1688.

[23] V. Arsigny, O. Commowick, N. Ayache, and X. Pennec, "A Fast and Log-Euclidean Polyaffine Framework for Locally Linear Registration," *Journal of Mathematical Imaging and Vision,* vol. 33, pp. 222-238, 2009.

[24] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213-3223.

[25] A. D. Costea and S. Nedevschi, "Traffic Scene Segmentation based on Boosting over Multimodal Low, Intermediate and High Order Multi-range Channel Features," *Intelligent Vehicles,* 2017.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[27] V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," *Adv. Neural Inf. Process. Syst,* vol. 2, p. 4, 2011.