2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

Octree map based on sparse point cloud and heuristic probability distribution for labeled images.

Julie Stephany Berrio [1], Wei Zhou[1], James Ward [1], Stewart Worrall [1], Eduardo Nebot[1]

***Abstract*— To navigate through urban roads, an automated vehicle must be able to perceive and recognize objects in a three-dimensional environment. A high level contextual understanding of the surroundings is necessary to execute accurate driving maneuvers. This paper presents a novel approach to build three dimensional semantic octree maps from lidar scans and the output of a convolutional neural network (CNN) to obtain the labels of the environment. We present a heuristic method to associate uncertainties to the labels from the images based on a combination of the labels themselves, score maps retrieved by the CNN and the raw images. These uncertainties and the camera-lidar calibration parameters for multiple cameras are considered in the projection of the labels and their uncertainties into the point cloud. Every labeled lidar scan works as an input to an octree map building algorithm that calculates and updates the label probabilities of the voxels in the map. This paper also presents a qualitative and quantitative evaluation of accuracy, analyzing projection in single lidar scans and complete maps built with our probabilistic octree framework.**

I. Introduction

The development of intelligent autonomous systems such as self-driving vehicles is currently an active area of research. To enable essential capabilities such as optimal path planning, trustworthy driving decision making and accurate vehicle control, it is necessary for the vehicle to comprehend the geometrical shape, location and the type of all objects in proximity. The capability to classify the objects of the surrounding location depends on the perception potential of the vehicle sensor system. Cameras have been extensively used for object classification and scene understanding because they are low-cost and contain dense information [1].

A common approach for autonomous vehicles is to capture 3D information based on Light Detection and Ranging (lidar) systems. Lidar is widely used as a cost-effective and reliable sensor for representing the urban environment [2]. However they lack the resolution and capability of texture and colour detection available from vision. This becomes an issue when detecting boundaries of objects.

Sensor fusion algorithms make it possible to overcome the implicit limitations of each separate sensing modality and simultaneously take advantage of their best capabilities. The integration of laser range-finding technologies with existing vision systems enable a more comprehensive understanding of the 3D structure of the environment [3]. The benefit of lidar-camera fusion relies on an accurate calibration of intrinsic parameters of the camera and geometrical extrinsic parameters describing the relative location of the sensors.

[1]J. Berrio, W. Zhou, J. Ward, S. Worrall, E. Nebot are with the Australian Centre for Field Robotics (ACFR) at the University of Sydney (NSW, Australia). E-mails: {j.berrio, w.zhou, j.ward, s.worrall, e.nebot}@acfr.usyd.edu.au

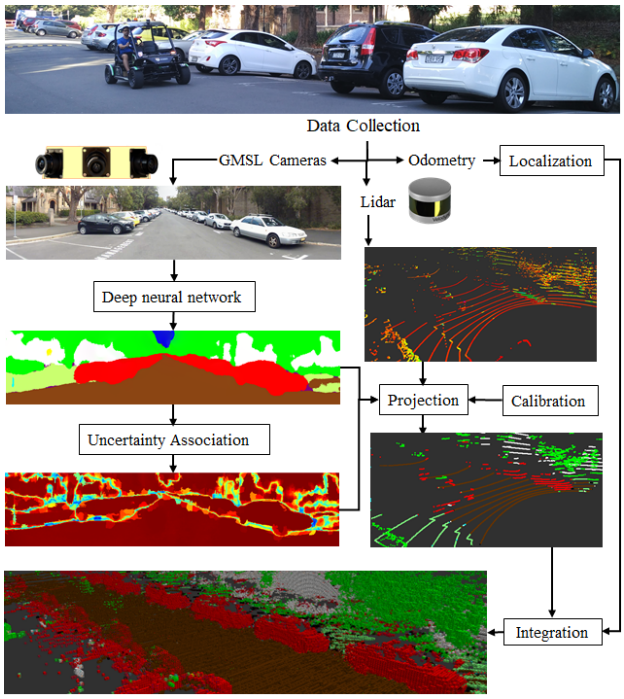


Fig. 1: Flowchart of the processing pipeline. An electric vehicle equipped with a sensing system is used for data collection. Information from camera and lidar feed a modified Octomap algorithm to build a semantic map.

Our work uses labeled images to provide a semantic representation of the environment. This information is obtained with an ENet CNN model [4] trained with the public dataset CityScapes [5] and fine-tuned with locally annotated images to boost the performance in our local environment [6].

In this paper, we present algorithms to construct a semantic octree map by probabilistically merging information obtained with semantically labeled images, 16-beam lidar [7] point cloud and vehicle localisation. The data used for this paper was collected with an electric vehicle (EV) equipped with a Velodyne laser sensor, three fixed lens Nvidia GMSL cameras, wheel encoders, and an IMU with that provides gyro, accelerometers and magnetometer information.

For this work, images from three different cameras were collected covering a 160° field of view. The images are processed by a trained ENet CNN model [4] to produce output images which contain the semantic information with a class index for each pixel. The CNN also provides score maps for each label that represent the probability distribution of the labels based on its own classification. We added a post-processing stage to associate uncertainty to the classification

978-1-5386-8094-0/18/$31.00 ©2018 IEEE

process. This takes the CNN probability distribution and also the characteristics of the original image into account. The characteristics of a particular image is obtained by forming superpixels that describe sections of the image with similarities.

The labeled images and their corresponding uncertainties are then associated with the points in the point cloud field based on the extrinsic calibration between the lidar and cameras. The inputs to the map building process are the labeled point cloud and vehicle's localization in order to perform the registration and probabilistic labeling of every 3D point into a 3D voxel grid called Octomap [8]. The result is a hierarchical 8-ary tree (octree) [9] structure map which includes 3D units (voxel) and their class probabilities. This work extends the results of [10] by incorporating uncertainties of different sources of information and presenting qualitative and quantitative operation results of the new algorithms. A flowchart of this technique is displayed in Fig. 1.

In the next section, we present a survey of the related work for point cloud classification and map representation. In Section III, we detail the components of the algorithm pipeline and its operation. The experiments, evaluation, and results are shown in Section IV.

II. Related Work

The current approaches to achieve real-time 3D reconstruction are mainly using stereo vision [11], [12], [13], [14], [15], structure from motion (SfM) [16], [17], [18] or scanning lidar sensors [19], [20]. Limitations such as the high computational cost required by stereo vision, the need for accurate calibration between consecutive images by SfM or the fact that lidars cannot provide color/texture information require new approaches to combine different sensor modalities [21], [22].

There are two main stages that are required in order to build a 3D semantic map. The first is selecting the proper classification method, which is dependent on the quality of information acquired from the sensors. The second is to adopt a suitable data structure to represent the map, taking into account that the structure must allow the integration of semantic labels [19].

Semantic segmentation divides the sensed environment into semantically meaningful parts, and classifies each object into one of the pre-determined categories. Different semantic segmentation techniques have been explored by researchers in the last decade, with approaches like Markov Random Fields (MRF) [14], [12], Conditional Random Fields (CRF) [13], [19] and 3D CNN [23] currently the most popular. There are applications that still make use of Bayesian classifiers and Support Vector Machines (SVM) but these approaches are no longer widely used.

The work of [24] used a Max-Margin Markov Network (M^3Ns) based on MRF where the authors adapted a functional gradient approach for learning high-dimensional parameters of random fields in order to perform discrete, multi-label classification. [25] describes a system that recognizes small objects in city scans by locating them using hierarchical clustering. The objects are then segmented with a graph-cut algorithm to finally describe and classify them with a Bayesian classifier.

The authors of [20] addressed the issue of the classification of 3D point clouds in urban areas using a full-waveform of airborne lidar data. A supervised SVM classifier operates on a 27-component feature vector extracted from the point cloud based and waveform data. [19] presents a 3D semantic outdoor mapping system with multi-label and multi-resolution octree map using conditional random fields (CRF) to classify point clouds. A similar approach is presented in [14], [11]. [15] introduced semantic scene completion network (SSCNet), an end-to-end 3D convolutional network for the semantic scene completion task of jointly predicting volumetric occupancy and semantic labels for full 3D scenes. In [23] the authors propose a 3D point cloud labeling system based on fully 3D CNN. This approach does not need prior knowledge for segmentation.

Previous methods required a density of points that can only be provided by laser sensors with very high resolution or working at very short distances. Our work aims to combine different sources of information with proper uncertainties to be able to obtain consistent results with low resolution lasers such as a Velodyne VLP-16 or high resolution lasers working at long ranges.

Different approaches exist to represent 3D maps, such as pure point clouds [11], [13], voxel grids, octrees [14], [12], surface elements (surfels) [26] and Gaussian Processes [27]. The desirable requirements for a data structure include memory-efficiency, allowing multi-resolution, being able to integrate semantic labels [19] and uncertainty. We can find in literature techniques such as [28] where the point cloud is decomposed into a concise, hybrid structure of inherent shapes and a set of remaining points. Lafarge and Mallet introduced in [29] anlgorithm that uses geometric 3D primitives to reconstruct buildings, trees and topologically complex grounds from a point cloud.

Our approach uses the semantic information extracted by a CNN model and projects the labels into the point cloud. This information is fed to an efficient probabilistic 3D mapping framework based on octrees [8] to generate the volumetric 3D environment model. This approach is suitable for a platform with lidars and cameras with an overlapped field of view (FOV) within an environment containing dynamic objects.

A similar approach was introduced in [21]. The authors demonstrated that although the algorithm used sparse lidar data, the 64 beam point cloud was sufficient to perform euclidean clustering of a single scan with relatively high accuracy for objects due to a 0.4° angular vertical resolution of the lidar. Our method uses data from a VLP-16 with a 2° angular vertical resolution meaning it is only possible to cluster points near to the vehicle (max 10 meters), discarding important information of the surroundings.

The authors of [21] transfer the labels to the point cloud using a Gauss Kernel, which depends on the label of nearest pixel taking in consideration adjacent pixels of the labeled

image. Our approach projects the label and associated uncertainty of the corresponding pixel to a 3D point (based on the extrinsic calibration) to build a semantic map within a probabilistic framework.

This paper uses the output of an ENet [4] CNN, which is a deep neural network architecture for real-time semantic segmentation. The first step consists of a downsampling process of the input images to improve processing speed in later stages. The drawback of downsampling is the loss of spatial information like accurate edge representation. This loss of information will affect the final labeled image with low-accuracy labels near to the boundaries of the classes.

In [30], the authors identified the segmentation difficulty of pixels for deep models by training and testing CNNs on a validation set with every pixel divided into three separate sets, named "easy", "moderate" and "extremely hard" sets. The easy sets include pixels that are correctly classified with larger than 95% confidence, while moderate sets consist of pixels which classification scores are smaller than 0.95. The extremely hard set encloses pixels that are misclassified with larger than 95% confidence. The authors showed that at least 68% of pixels in this last set are located on the boundaries between objects, demonstrating those boundary pixels are extremely hard to classify correctly because of large ambiguity.

In this paper we propose a heuristic method to associate probabilities to a CNN labeled image based on merging information from the original image and the CNN distribution. This process also considers the image similarity obtained with the superpixel algorithms. We finally evaluate the most likely label, associating lower confidence to sections with scattered label content.

III. Semantic 3D Map Building

This section explains the method used to build the semantic 3D map under a probabilistic framework. First, we describe the labeled image probability association process, then the method for labeling of the lidar point cloud. Once a labeled point cloud with associated probabilities is obtained, the next step is to use the Octomap map building approach [8] to update the class and occupancy uncertainty of volumetric units.

A. Labeled image probability association

Some semantic segmentation label results presented in [6] were used in this paper. The CNN was first trained with the public Cityscapes [5] dataset and then fine-tuned by annotating local images from the University of Sydney and surroundings. The labels used in Cityscapes dataset were remapped in 12 categories forming the set of classes $\mathbf{C}$, which includes 'sky', 'building', 'pole', 'road', 'undrivable road', 'vegetation', 'sign symbol', 'fence', 'vehicle', 'pedestrian', 'rider' and 'unlabeled' in order to train the model. Fig. 2 shows the model's output for one single image taken at the University of Sydney campus, which demonstrates the classification with common features.

To speed up the CNN, the ENet model down-samples the input image at an early stage and uses only a small

Original image Labeled image

Fig. 2: Semantic segmentation result from a fine-tuned CNN model on University of Sydney campus. Red • for vehicles, white is for buildings, brown • is for roads, green • is for vegetation, blue • is for sky, neon green • is for undrivable roads, yellow • for pedestrians and riders, cyan • for poles, gray • is for fence and purple • for unlabeled pixels.

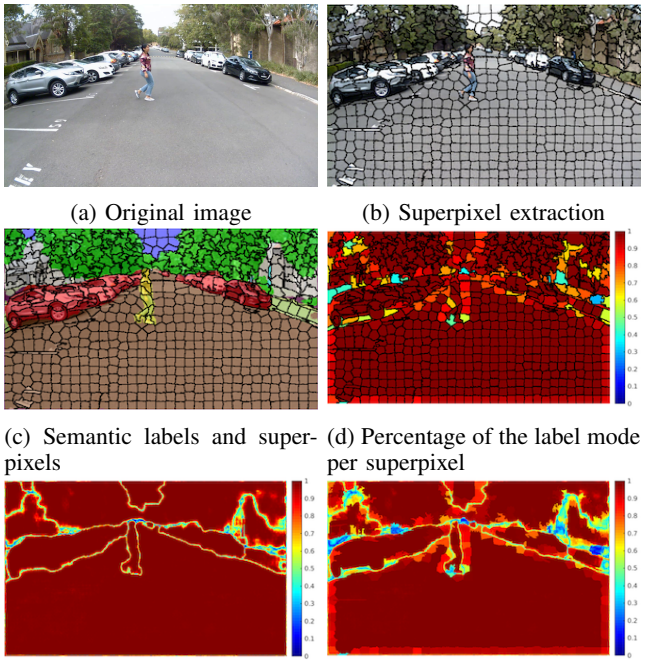
number of feature maps because many of the maps are redundant. By down-sampling the input image, the size of the input parameters is reduced and model over- fitting is controlled. However, this comes at the expenses of losing information which cannot be restored by resizing the output to the original input dimensions. The final resized image shows the effect of information loss especially in areas near to object edges which contain mainly sharpness information. Therefore, an increase of speed entails loss of accuracy in the model, which may result in noisy predictions especially in areas near to class boundaries.

Once the labeling process is done by the CNN, we proceed to associate the uncertainty to the labels based on the integration of two pieces of information, namely the overlap between the original image superpixels boundaries and the labeled image, and CNN score maps.

The first piece of information to be used to associate uncertainty is an evaluation of the labels inside the superpixels that form the image. This procedure is shown in Algorithm 1 and Fig. 3. Superpixels [31] is a method for region segmentation where every element contains pixels which have been grouped based on similarities in inter- and intra-region characteristics such as color, texture, brightness, contour, curvilinear continuity, etc. Each superpixel is a perceptually consistent unit since all pixels inside are most likely uniform. It is reasonable to assume that all pixels in a superpixel should belong only to one single class or label. However, this assumption depends on the adherence to the boundaries of the used segmentation method, size of superpixels and accuracy of the mapped labels.

The inputs to the algorithm 1 are the sub-sampled image I in Fig. 3a, the CNN labeled image $Labels$ and the number of desired superpixels n_{sp}. The algorithm starts by creating the output variable SSp_{map} of the same size of the labeled original image. The algorithm then segments the original image using the simple linear iterative clustering (SLIC) [32] method. This method was chosen due to its characteristics of speed, memory efficiency and adherence to the boundaries.

Fig. 3b shows the result of the SLIC algorithm, which processes the original image based on pixel characteristics and the input parameter n_{sp}. It will retrieve the superpixel labels L and the final number of superpixels contained in the



(a) Original image (b) Superpixel extraction

(c) Semantic labels and superpixels (d) Percentage of the label mode per superpixel

(e) CNN score map from semantic segmentation (f) Associated probabilistic distribution

Fig. 3: Uncertainty association process. The color bar represents the confidence level of the associated uncertainty. Dark red is the highest confidence and dark blue is the lowest confidence.

image N. Indexes of the superpixel labels are stored in an auxiliary variable idx. The next step evaluates the percentage of the most common category based on the CNN semantic segmentation for each every superpixel Fig. 3c. This is computed and stored in the corresponding coordinates of the output variable. The output variable Spp_{map} is represented in Fig. 3d. The superpixels with pixels belonging to one single class will have the maximum value of 1. The superpixels with multiple labels will have smaller values depending on the area occupied by the predominant class.

Algorithm 1 Algorithm for label uncertainty association based on superpixels

Input: I, $Labels$, n_{sp}
Output: Spp_{map}
Initialization :
$Spp_{map} \leftarrow zeros(size(I))$
$[idx, N] \leftarrow SLIC(I, n_{sp})$
for $LabelVal = 1$ to N **do**
  $idxs \leftarrow idx(LabelVal)$
  $lpsp \leftarrow \frac{histc(labels(idxs), mode(labels(idxs)))}{lenght(labels(idxs))}$
  $Spp_{map}(idxs) \leftarrow lpsp$
end for
return Spp_{map}

The second element involved in the uncertainty association is the CNN score maps. The network generates one score $s^k_{i,j}$ for each pixel location (i, j) from the sub-sampled image I and for each class $k \epsilon C$ [33]. We retrieve 12 class score maps from the CNN network output, one for each category. We illustrate the image label scores as class probabilities by applying a the softmax function [34] as a normalizer.

$$p(k|I, \theta) = \frac{e^{s^k}}{\sum_{c \epsilon C} e^{s^k}} \quad (1)$$

Where θ represents all the trained parameters of the CNN ENet architecture. In order to obtain a single variable map P_{map} that contains the probability values of the most likely label per pixel we extract the largest component of all $p(k|I, \theta)$ per pixel to obtain a single variable map P_{map} that contains the probability values of the most likely label per pixel.

$$P_{map} = \max[p(1|I, \theta), p(2|I, \theta), ..., p(12|I, \theta))] \quad (2)$$

Fig. 3e shows the result of this process. A feature to note is that even though the CNN allocates labels to the pixels near and belonging to the class boundaries, the confidence of these specific regions is lower than the rest of the pixels far from the boundaries. The final step is to integrate the information obtained from the superpixels and CNN scores map to create the final label probability distribution. Assuming the Spp_{map} and P_{map} as independent events, the compound probability can be formulated as the probability of the first factor multiplied by the probability of the second element.

$$S_{map} = P_{map} * Spp_{map} \quad (3)$$

Fig. 3f shows the final associated probability distribution proposed in this paper. The algorithm aggregates the CNN label probabilities (which assign low probabilities to label boundaries) with the ratio of the label mode within superpixels, exploiting SLIC method capability of adherence to the boundaries.

B. 3D projection

The three cameras used on the experimental vehicle provide partially overlapped images. The projection in these zones is done by prioritizing the label with less uncertainty. The point cloud is initially projected to the *GMSL_center_camera* image. The points within the camera's field of view receive the information of the corresponding label and its associated uncertainty. The remaining points are projected to the *GMSL_right_camera* and *GMSL_left_camera* images. The output of the projection process is a labeled point cloud with 160° FOV. All points outside the FOV of the three cameras are discarded.

Another source of error is due to the presence of misalignments between the image and the point cloud which may result in allocation of the wrong label. This paper proposes a voting correction method based on ring-clustering of individual scans to mitigate this problem. The Velodyne VLP-16 has a vertical resolution of 2 degrees between beams (rings), which makes it difficult to appropriately cluster the lidar scan for distant targets due to the vertically separation of the returned points.

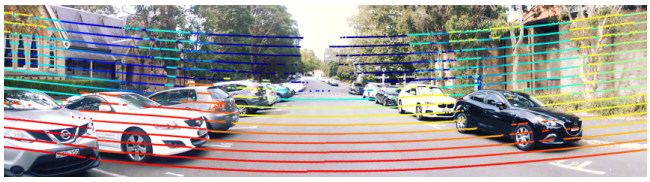


Fig. 4: Lidar point cloud projected into the images. The points in this image are coloured based on range.

Algorithm 2 is the proposed method to mitigate the point cloud mislabeling issue. First, the point cloud lpc is divided into two parts. The first part is the ground plane which is extracted by SAmple Consensus (SAC) Ransac model-fitting method, and the second part is all other obstacles. The $Ground\_Plane_{pc}$ is initially assigned labels *road* and *undrivable road*, labels that due to their position in the environment (same horizontal plane) could belong to the same cluster. Each point in the $Obstacles_{pc}$ point cloud that belongs to same $Ring$ number in the scan is clustered by the Euclidean clustering method. Every point p inside a cluster C feeds a label voting vector L_{vote}. The algorithm will then set each vote with a weight given by its label probability $p.prob$, prioritizing most likely labels inside the cluster. The label with the highest vote is assigned to all points inside the cluster. Each processed cluster is concatenated with the initial ground plane point cloud $Labeled\_pc$.

Algorithm 2 Label correction

Input: lpc, $labels$
Output: $Labeled\_pc$
Initialization :
$Ground\_Plane_{pc} \leftarrow SAC\_Ransac(lpc, inliers)$
$Obstacles_{pc} \leftarrow SAC\_Ransac(lpc, outliers)$
$Labeled\_pc \leftarrow Ground\_Plane_{pc}$
for $Ring = 1$ to 16 **do**
  $clusters \leftarrow EuclideanCluster(Obstacles_{pc}(Ring))$
  for each $C \in clusters$ **do**
    for each $p \in C$ **do**
      $L_{vote}(p.label) \leftarrow L_{vote}(p.label) + p.prob$
    end for
    $C.label \leftarrow labels(\text{argmax}(L_{vote}))$
  end for
  $Labeled\_pc \leftarrow concat(C, Labeled\_pc)$
end for
return $Labeled\_pc$

C. Voxel occupancy and label probability

The OctoMap approach of [8] builds a 3D map of voxels in an octree framework for a set of point clouds. Here, the occupancy grid mapping introduced by [35] is used for integrating the sensor readings. The probability $P(v|z_{1:t})$ of the voxel v to be occupied is updated depending on the current measurement z_t, a prior probability $P(v)$, and the previous estimate $P(v|z_{1:t-1})$, being $P(v|z_{1:t})$ specific to the sensor that generated z_t. A uniform prior probability is assumed and the initial value of $P(v)$ is set as 0.5.

The algorithms described in previous sections classify every 3D point of the point cloud into different semantic labels. We can then build a multi-label octree map taking into account the label c and probability $P(c|z_t)$ of each incoming sensor reading z and prior voxel class probability $P(c_n|z_{1:t-1})$ (where $n = 1 : 11$) to update both the label and the probabilities within the occupied voxel.

We calculate the posterior distribution $P(c|z_{1:t})$ from the corresponding posterior on time step earlier $P(c|z_{1:t-1})$ [37], by applying the Bayes rule, obtaining:

$$P(c|z_{1:t}) = \frac{P(c|z_t)P(z_t)P(c|z_{1:t-1})}{P(c)P(z_t|z_{1:t-1})} \tag{4}$$

We calculate the posterior distribution for the opposite event $\neg c$, assuming the remaining probability will be equally distributed among the remaining labels:

$$P(\neg c|z_{1:t}) = \frac{(1 - P(c|z_t))P(z_t)(1 - P(c|z_{1:t-1})}{(1 - P(c))P(z_t|z_{1:t-1})} \tag{5}$$

Computing the ratio of 4 and 5, we reduce the update problem to a Binary Bayes Filter:

$$\frac{P(c|z_{1:t})}{P(\neg c|z_{1:t})} = \frac{P(c|z_t)P(c|z_{1:t-1})(1 - P(c))}{(1 - P(c|z_t))(1 - P(c|z_{1:t-1}))P(c)} \tag{6}$$

By using log odds $l(x)$ as an alternate way to express probabilities, the product turns into a sum, simplifying the process of updating them with new evidence:

$$l_t(c) = l(P(c|z_t)) + l(P(c|z_{1:t-1})) - l(P(c)) \tag{7}$$

Eq. 7 requires that in order to change the voxel's label we need to integrate as many sensor readings belonging the same label as have been integrated before to determine its current state. To retrieve the most likely label c_{max} and its probability $P(n, c_{max})$ of a voxel v, the arguments of the maximum is computed for the probability vector $\text{argmax}_c[P(v, c_1), ..., P(v, c_{11})]$.

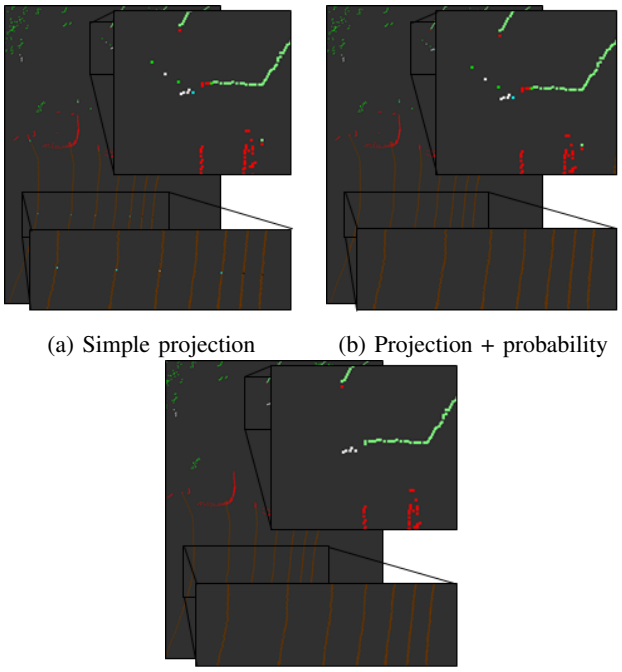
IV. Results

Our algorithm was tested on different datasets collected on the surroundings of The University of Sydney. All data were collected using an electric vehicle (EV) platform. The collected images are initially classified and the algorithms presented are used to evaluate the class uncertainty. This information is then projected into the point cloud. The VLP-16 laser sensor provides $360°$ FOV with 16 beams. The initial point cloud is reduced to approximately 45% of the original points after the projection process is performed. This section of the point cloud used corresponds to the overlapped zone between the lidar and the three cameras. The size of point cloud projected can change based on the scenario.

For evaluation purposes the algorithm was tested with two experiments. The first one evaluates the camera-lidar projection using a single scan. Three separate point cloud outputs were calculated in different stages of our algorithm: *1. simple projection* corresponds to the direct projection from

TABLE I: Single scan evaluation

Stage	Probability [<0.5]		Probability [0.5-0.65]		Probability [0.65-0.85]		Probability [>0.85]		Overall	
	T-P	F-P	T-P	F-P	T-P	F-P	T-P	F-P	T-P	F-P
Simple projection	0	0	0	0	0	0	0.848	0.152	0.848	0.152
Projection + probability	0.451	0.549	0.483	0.517	0.624	0.376	0.936	0.064	0.851	0.149
Projection + prob. + clust.	0.568	0.432	0.575	0.425	0.705	0.295	0.958	0.042	0.893	0.107



(a) Simple projection (b) Projection + probability

(c) Projection + probability + clustering

Fig. 5: Stages of the camera - lidar projection a) projection from camera to lidar, b) projection taking into account label probabilities, c) projection taking into account probabilities plus clustering for label correction

the left to right images to the point cloud (projects initially all the information of the left camera image into the point cloud, the remaining point cloud is labeled with center and right camera images) and sets a label probability of 0.8 to every point. *2. projection + probability* takes into account the associated label probability to decide which label and label probability should be projected into the point cloud in zones where the images are overlapped. *3. projection + probability + clustering* performs the same process of *projection + probability* besides beam clustering label fitting.

Qualitative results are shown in Fig. 5. This figure shows the same single lidar scan at the 3 different stages of projection in the algorithm. In Fig. 5a (bottom zoomed box) *simple projection* we can notice spurious labels, especially in those zones belonging or near to the projected image boundaries. This is due to the presence of "unlabeled" pixels especially in this part of the image and mislabels due to partial object occlusion. In this case the CNN cannot properly identify the semantic label resulting in a noisy classification. These issues can be overcome to some degree by evaluating the redundant information in the area given by the camera field of view overlap.

Based on the highest associated probability, the label is chosen and projected, removing the spurious labels caused by the previously exposed problems, the result of the process is shown in Fig. 5c (upper zoomed box).

Label correction is then performed based on beam clustering. Every point inside the beam is clustered and the algorithm evaluates the label of a point within a cluster through a voting process. It also eliminates outlier points belonging to distant locations and for which label accuracy is low. Fig. 5c depicts the output of the last projection stage.

To assess the accuracy of the projection in one single scan, we manually labeled 20 single scan scenarios as ground truth, and compared them with the different point cloud projections. The point cloud projections were first divided into four groups based on the assigned label probability. These probability groups are: *high* with probability range from 0.85 to 1.0, *high-moderate* with probability within 0.65 to 0.85, *moderate* with probability range withing 0.5 to 0.65, and the *moderate-low* with probability lower than 0.5.

Table I presents the quantitative results for the comparison of the ground truth with the point clouds. For all projections the same calibration parameters were used. For the *simple projection* point cloud, its overall accuracy is 0.848 and the CNN's global accuracy over the test data was 0.967 [6]. This reduction of global accuracy is caused by two factors - small misalignments in calibration parameters and rigorousness of the evaluation. The latter reason is the main cause of the drop in accuracy. The point cloud evaluation is more granular since the lidar can detect points belonging to different labels behind other objects - for example buildings occluded behind trees that can be detected through tree branches. Although the evaluation of the image may show an accurate segmentation, the point cloud evaluation will show a lower level of accuracy due to this effect.

Table I shows the percentages of true positives (T-P) for every group in all point clouds. The overall true-positive percentage of *projection + probability* is 0.851 while for *projection + probability + clustering* is 0.893. Considering the point cloud *simple projection* as a reference, the algorithm presented provides a consistent more accurate labeling performance with all probability ranges with *projection + probability + clustering* the best performer.

The accuracy of the groups is consistent with probability ranges, those groups with a high label probability contain a major percentage of correct points, with 0.936 and 0.958 the proportion of T-P for the high label probability group in *projection + probability* and *projection + probability + clustering* respectively.

The second experiments evaluate the performance of the algorithm with multiple scan maps. The maps are composed

TABLE II: Octomap evaluation

Stage	Probability [<0.5]		Probability [0.5-0.65]		Probability [0.65-0.80]		Probability [>0.80]		Overall	
	T-P	% P	T-P	% P	T-P	% P	T-P	% P	T-P	% P
Simple projection	0.480	31.2	0.342	0.4	0.798	10.5	0.871	57.9	0.740	100
Projection + probability	0.572	31.3	0.837	9.4	0.657	2.3	0.885	57.0	0.778	100
Projection + prob. + clust.	0.581	32.7	0.871	8.4	0.737	2.4	0.930	56.5	0.832	100

of 280 consecutive scans and were manually labeled to be used as a ground truth reference. The labeled point cloud synchronized with the current position of the EV feeds the adapted Octomap algorithm which evaluates the occupancy, updates probabilities and labels in every voxel. The voxel resolution was set to 10 cm for all experiments [10]. Fig. 6 shows the resulting octree for one scenario, using the same color code of Fig. 2 and the opacity proportional to the voxel label probability.

The voxels within the maps were first divided into four chunks based on their voxel label probability. The probability ranges were the same as the single scan, that is *high* voxel probability within 0.80 to 1.0, *high-moderate* voxel probability within 0.65 to 0.80, *moderate* voxel probability within 0.5 to 0.65, and *moderate-low* voxel probability with voxel label probability points lower than 0.5.

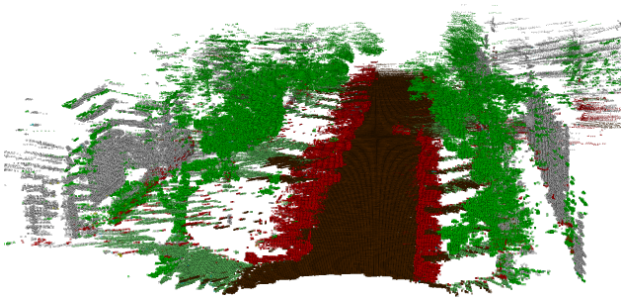


Fig. 6: Final map reconstructed with 1880 lidar scans

Table II shows the quantitative evaluation of the comparison between the ground truth with the octree maps. The overall correctness of the octree maps built with the point cloud *simple projection* is 0.740 while for *projection + probability* is 0.778 and for *projection + probability + clustering* 0.832. The fraction of T-P within the map differs from the single scan evaluation, with the former slightly lower than the latter. This happens because closer and/or bigger objects in the image are will generally contain more high probability points. Due to the proximity to the sensors, closer voxels will be represented by a higher density of points and be updated more frequently compared with more distant voxels.

Table II shows the overall octree maps T-P fraction and the percentage of points $\%P$ for different probability groups. A larger number of true positives are found in the *high* voxel probability group as it was expected. This number is reduced in the *high-moderate* voxel probability group and increased again for the *moderate* voxel probability group. This phenomenon is caused by the lack of compactness of some detected objects, such as tree branches, fences and even bikes. While they constitute a single label for the image they may not be a single solid element to the lidar, which is able to detect other obstacles behind these objects. Since the associated uncertainty is calculated based on the images (original and labeled) consistent mislabeling results in high associated probabilities for wrong labels.

The percentage of the map's total voxels within the each group is also shown in Table II. Since all points in *Simple projection* have an associated label probability of 0.8 (*high* probability), it is evident that a large amount of voxels are placed in *high* and *high-moderate* probability groups. A slightly reduced proportion of voxels are placed in the same groups for *Projection + probability* and *Projection + probability + clustering* with the difference being more accurate labeled voxels. Around 30% of the total voxels are within the *low* probability group. These voxels are generally located far from the sensor and have been just initialized or only updated few times with different labeled points.

V. Conclusions and Future Work

This paper presents a novel approach for building an octree semantic map from CNN labeled images and point cloud integration. The algorithm modelled the uncertainty of the labeling process based on two pieces of information: the original image superpixel segmentation overlaid on the the labeled images and CNN's score maps. The image superpixels automatically segment sections of the image that have some level of similarity based on texture, color, etc. These segments are allocated a label according to the predominant class. The associated uncertainty was used to determine the label to be projected into the synchronized point cloud. Experimental results were presenting showing that the accuracy of the projection can be improved from 0.848 to 0.893 overall. These results were consistent with a single scan and a full map form with 280 scans.

The projected single scans were input to an octomap mapping algorithm. It probabilistically and semantically reconstructed the 3D environment with volumetric units. This representation is very efficient since it allowed the reduction of 10 cm resolution volumetric units (voxel) to 3% of the total point cloud (composed by 3d points) representation for urban datasets taken at 1.3 m/s and processed at 10Hz.

The quality of the map was evaluated by comparing the final results with the manually labeled map used as ground truth. The results showed that the approach presented in this paper has a 9.2% improvement when compared to the direct label projection. We consider this a very important outcome since it allows the representation of 3D occupancy with semantic properties, and considers uncertainty present in process. It is also important to note that the integration process was developed and tested in a multi-camera laser system. The additional complexity of such systems need to be addressed

as they are essential to complete sensor coverage of the area surrounding a vehicle. Although extrinsic parameter errors are considered in this process at the laser camera integration level, current research is also looking at incorporating more detailed models of uncertainty to represent sensor calibration errors.

ACKNOWLEDGMENTS

This work has been funded by the ACFR, the University of Sydney through the Dean of Engineering and Information Technologies PhD Scholarship (South America) and the Australian Research Council Discovery Grant DP160104081 and University of Michigan / Ford Motors Company Contract "Next geration Vehicles".

REFERENCES

[1] G. Ros, S. Ramos, M. Granados, A. Bakhtiary, D. Vazquez, and A. M. Lopez, "Vision-based off-line perception paradigm for autonomous driving", in 2015 IEEE Winter Conference on Applications of Computer Vision. IEEE, 2015. pp. 231-238.

[2] V. Vo, L. Truong-Hong, D. F. Laefer and M. Bertolotto, "Octree-based region growing for point cloud segmentation", in ISPRS Journal of Photogrammetry and Remote Sensing, Volume 104, 2015, pp. 88-100.

[3] H. J. Chien, R. Klette, N. Schneider and U. Franke, "Visual odometry driven online calibration for monocular lidar-camera systems", in 23rd Inter. Conf. on Pattern Recognition, Cancun, 2016, pp. 2848-2853.

[4] A. Paszke, A. Chaurasia, S. Kim, E. Culurciello, "ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation", in arXiv:1606.02147, Jun 2016.

[5] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[6] W. Zhou, R. Arroyo, A. Zyner, J. Ward, S. Worrall, E. Nebot and L. M. Bergasa, "Transferring visual knowledge for a robust road environment perception in intelligent vehicles", in IEEE 20th International Conference on Intelligent Transportation Systems, 2017.

[7] C.L. Glennie, A. Kusari, A. Facchin, "Calibration and stability analysis of the VLP-16 laser scanner" in International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, 40 (3W4), pp. 55-60, 2016.

[8] A. Hornung, K.M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees" in Autonomous Robots, 2013; DOI: 10.1007/s10514-012-9321-0. Software available at http://octomap.github.com.

[9] . Meagher, "Geometric modeling using octree encoding" in Computer Graphics and Image Processing, 19 (2), pp. 129-147, 1982.

[10] J. S. Berrio P., J. Ward, S. Worrall, W. Zhou, E. Nebot "Fusing Lidar and Semantic Image Information in Octree Maps", in ACRA Australasian Conference on Robotics and Automation 2017, Sydney, Australia, 2017.

[11] S. Sengupta, E. Greveson, A. Shahrokni and P. H. S. Torr, "Urban 3D semantic modelling using stereo vision," 2013 IEEE International Conf. on Robotics and Automation, Karlsruhe, 2013, pp. 580-585.

[12] H. He and B. Upcroft, "Nonparametric semantic segmentation for 3D street scenes," 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, 2013, pp. 3697-3703.

[13] Vineet et al., "Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction," 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, 2015, pp. 75-82.

[14] S. Sengupta and P. Sturgess, "Semantic octree: Unifying recognition, reconstruction and representation via an octree constrained higher order MRF," 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, 2015, pp. 1874-1879.

[15] S. Song, F. Yu, A. Zeng, A. Chang, M. Savva, T. Funkhouser, "Semantic Scene Completion from a Single Depth Image," CoRR, abs/1611.08974, 2016.

[16] Y. Kang, K. Yamaguchi, T. Naito and Y. Ninomiya, "Road scene labeling using SfM module and 3D bag of textons," 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, Kyoto, 2009, pp. 657-664.

[17] A. Martinovi, J. Knopp, H. Riemenschneider and L. Van Gool, "3D all the way: Semantic segmentation of urban scenes from start to end in 3D," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 4456-4465.

[18] R. Daz, M. Lee, J. Schubert and C. C. Fowlkes, "Lifting GIS maps into strong geometric context for scene understanding," 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY, 2016, pp. 1-9.

[19] D. Lang, S. Friedmann and D Paulus, "Semantic 3D Octree Maps based on Conditional Random Fields", in International Conference on Machine Vision Applications, Kyoto, May 2013, pp. 185 -188.

[20] C. Mallet, F. Bretar, M. Roux, U. Soergel and C. Heipke, "Relevance assessment of full-waveform lidar data for urban area classification", in ISPRS Journal of Photogrammetry and Remote Sensing, Volume 66, Issue 6, 2011, pp. S71-S84.

[21] Y. Zhong, S. Wang, S. Xie, Z. Cao, et al., in "3D Scene Reconstruction with Sparse LiDAR Data and Monocular Image in Single Frame," SAE Int. J. Passeng. Cars Electron. Electr. Syst. 11(1):2018.

[22] C. Premebida, L. Garrote, A. Asvadi, A. P. Ribeiro and U. Nunes, "High-resolution LIDAR-based depth mapping using bilateral filter," 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, 2016, pp. 2469-2474.

[23] J. Huang and S. You, "Point cloud labeling using 3D Convolutional Neural Network," in 23rd International Conference on Pattern Recognition (ICPR), Cancun, 2016, pp. 2670-2675.

[24] D. Munoz, J. A. Bagnell, N. Vandapel and M. Hebert, "Contextual classification with functional Max-Margin Markov Networks," in 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, 2009, pp. 975-982.

[25] A. Golovinskiy, V. G. Kim and T. Funkhouser, "Shape-based recognition of 3D point clouds in urban environments", in 2009 IEEE 12th International Conf. on Computer Vision, Kyoto, 2009, pp. 2154-2161.

[26] P. Puri, D. Jia and M. Kaess, "GravityFusion: Real-time dense mapping without pose graph using deformation and orientation," 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, 2017, pp. 6506-6513.

[27] N. Akai and K. Ozaki, "3D magnetic field mapping in large-scale indoor environment using measurement robot and Gaussian processes," 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), Sapporo, 2017, pp. 1-7.

[28] R. Schnabel, R. Wahl and R. Klein, "Efficient RANSAC for Point-Cloud Shape Detection", in Computer Graphics Forum, vol. 26, no. 2, 2007, pp. 214-226.

[29] F. Lafarge and C. Mallet, "Creating Large-Scale City Models from 3D-Point" in Computer Vision and Pattern Recognition, Boston, MA, 2015, pp. 1713-1721.

[30] X Li, Z Liu, P Luo, CC Loy, X Tang, "Not All Pixels Are Equal: Difficulty-Aware Semantic Segmentation via Deep Layer Cascade", in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Hawaii, USA, 2017

[31] . Ren and J. Malik, "Learning a classification model for segmentation," Proceedings Ninth IEEE International Conference on Computer Vision, Nice, France, 2003, pp. 10-17 vol.1.

[32] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua and S. Ssstrunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, pp. 2274-2282, Nov. 2012.

[33] P. O. Pinheiro and R. Collobert, "From image-level to pixel-level labeling with Convolutional Networks," 2015 IEEE Conf. on Computer Vision and Pattern Recognition, Boston, MA, 2015, pp. 1713-1721.

[34] J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition", in F. Fogelman-Souli and J. Hrault, Eds., Neuro-Computing: Algorithms, Architectures, Springer-Verlag, New York, 1989.

[35] H. Moravec, A. Elfes, "High resolution maps from wide angle sonar", in Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 1985, St. Louis, MO, USA, pp 116121.

[36] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada. "An Open Approach to Autonomous Vehicles", IEEE Micro, Vol. 35, 2015, No. 6, pp. 60-69.

[37] S. Thrun, W. Burgard, D. Fox, "Probabilitistic robotics". MIT press, 2005.