SEMANTIC ENRICHMENT OF 3D POINT CLOUDS USING 2D IMAGE SEGMENTATION

A. Rai[1] *, N. Srivastava[1], K. Khoshelham[2], K. Jain[1]

[1]Department of Civil Engineering, Indian Institute of Technology Roorkee, Roorkee, Uttarakhand, India – abhsihek_r@ce.iitr.ac.in
[2] Department of Infrastructure Engineering, The University of Melbourne, Parkville, Victoria, Australia

KEY WORDS: 3D Point Cloud, Semantics Segmentation, Indoor Mapping, 2D-3D, Deep Learning.

ABSTRACT:

3D point cloud segmentation is computationally intensive due to the lack of inherent structural information and the unstructured nature of the point cloud data, which hinders the identification and connection of neighboring points. Understanding the structure of the point cloud data plays a crucial role in obtaining a meaningful and accurate representation of the underlying 3D environment. In this paper, we propose an algorithm that builds on existing state-of-the-art techniques of 2D image segmentation and point cloud registration to enrich point clouds with semantic information. DeepLab2 with ResNet50 as backbone architecture trained on the COCO dataset is used for indoor scene semantic segmentation into several classes like wall, floor, ceiling, doors, and windows. Semantic information from 2D images is propagated along with other input data, i.e., RGB images, depth images, and sensor information to generate 3D point clouds with semantic information. Iterative Closest Point (ICP) algorithm is used for the pair-wise registration of consecutive point clouds and finally, optimization is applied using the pose graph optimization on the whole set of point clouds to generate the combined point cloud of the whole scene. 3D point cloud of the whole scene contains pseudo-color information which denotes the semantic class to which each point belongs. The proposed methodology use an off-the-shelf 2D semantic segmentation deep learning model to semantically segment 3D point clouds collected using handheld mobile LiDAR sensor. We demonstrate a comparison of the accuracy achieved compared to a manually segmented point cloud on an in-house dataset as well as a 2D3DS benchmark dataset.

1. INTRODUCTION

The use of LiDAR sensors on hand-held devices is becoming increasingly popular for applications like 3D mapping, augmented reality, and spatial planning. LiDAR sensor onboard a handheld mobile device provides raw data in the form of RGB color images and depth maps. 3D point cloud can be generated from RGB-D images which is a more nuanced form of 3D data representation. 3D point cloud data has information about the coordinates of feature points in a 3D coordinate system along X, Y, and Z directions and color information in the RGB channel. Semantic segmentation, also known as per-point classification, is a crucial task in point cloud scene understanding. It involves assigning an object label to each individual point in the point cloud, enabling comprehensive classification and analysis of the scene. The RGB image can be used to infer each pixel into an object class using a 2D Deep learning model for semantic segmentation. This approach can leverage the power of 2D deep learning techniques for semantic segmentation to enhance the richness of 3D point cloud data. By employing advanced algorithms and models, we are able to accurately assign semantic labels to individual points in the point cloud, thereby augmenting the information and facilitating a more detailed analysis and interpretation of the scene.

In recent years, extensive research has been dedicated to the development of deep learning approaches for point cloud semantic segmentation. However, this research problem remains challenging due to several factors. Firstly, point clouds are characterized by their large, unordered, and sparse data nature, which renders traditional convolutional operations ineffective. Due to its large size and irregular point sampling density, segmentation losses consistency and accuracy (Lyu et al., 2020). Secondly, RGB-D dataset is highly susceptible to noise and outliers due to lower data quality generated using handheld mobile LiDAR. Thirdly, the data collected suffers from problems of occlusion, thus producing incomplete data. Fourthly, deep learning algorithms necessitate a significant amount of labeled training data to achieve optimal performance in real-world scenarios (Qi et al., 2017a; Qi et al., 2017b). Unfortunately, obtaining a sufficiently large annotated 3D dataset is often difficult. In contrast, 2D deep learning for semantic segmentation has seen significant advancements and benefits from the availability of abundant annotated 2D datasets.

In this research paper, we propose a new algorithm that semantically enriches 3D point cloud data using 2D image semantic segmentation. The proposed algorithm incorporates a 2D semantically segmented image, generated using Deep Learning for each color image, classified into several classes, such as walls, floors, ceilings, doors, windows, etc., before generating the point cloud. This step ensures that each point in the resulting point cloud contains information about the class to which it belongs. By including semantic information at an early stage, our algorithm circumvents the computationally intensive task of 3D segmentation, resulting in faster and more efficient processing. 2D image segmentation uses the state-of-the-art image segmentation algorithm on RGB images to produce a segmentation map. After we have collected and pre-processed all the required inputs, i.e., RGB image, depth image, confidence image, semantically segmented image, and sensor information, the 3D point cloud is generated using depth images for the whole dataset. Point clouds thus generated are in the camera coordinate system and lack correlation with the other point clouds of the dataset. These point clouds are then registered to produce a combined point cloud of the mapped environment. Completed point clouds have X, Y and Z coordinates, and attributes including R, G, B - true color and R', G', B' – pseudo color. Thus, the main contribution of this work can be summarized as follows:

- We propose a process to utilize the RGB-D dataset for adding 3D semantic class information to the 3D point cloud.

- The proposed methodology can freely utilize any off-the-shelf 2D segmentation and classification algorithm for 3D point cloud segmentation.
- Proposed methodology segments and classifies the data before creating a 3D point cloud, thereby enabling more accurate analysis, interpretation, and understanding of the underlying structure and features within the point cloud data.

The algorithm also has the potential to reduce the time and effort required to create accurate semantically enriched 3D models of mapped environments. The proposed algorithm builds on existing techniques in image segmentation and point cloud registration, providing a novel and streamlined approach to indoor mapping and localization. By adding this additional information to the point cloud, the resulting data becomes more useful for applications that require object recognition, scene understanding, and localization.

2. RELATED WORKS

In this section, we discuss related works in the field of semantic segmentation of 2D and 3D datasets. Other related works which employ 2D semantics for 3D segmentation are also discussed which is crucial to understand our work.

2.1 2D Semantic Segmentation

Image segmentation has been a persistent problem in computer vision. It plays a crucial role in various visual understanding systems and applications, including medical image analysis, autonomous vehicles, video surveillance, and augmented reality (Minaee et al., 2021). Image segmentation can be categorized into semantic segmentation, instance segmentation, and panoptic segmentation. Semantic segmentation involves labeling pixels with object categories, while instance segmentation detects and delineates individual objects of interest. Several segmentation algorithms have been proposed, ranging from traditional methods like thresholding, region-growing, and clustering, to more advanced techniques such as active contours, graph cuts, and deep learning models (Otsu, 1979; Nock and Nielsen, 2004; Dhanachandra et al., 2015; Najman and Schmitt, 1994; Kass et al., 1988; Boykov et al., 2001; Plath et al., 2009). Deep learning models have revolutionized image segmentation with significant performance improvements, consistently achieving top accuracy rates on benchmark datasets (Armeni et al., 2017). This has led to a paradigm shift in the field of image segmentation. There has been a significant improvement in the performance of deep segmentation models over the past 7-8 years. Image segmentation has greatly benefited from deep learning with scope for improvement.

2.2 3D Semantic Segmentation of Point Cloud

3D semantic segmentation and labeling is a fundamental task for several use cases like scene understanding, autonomous driving, SLAM, etc. Annotating raw 3D point clouds obtained from sensors like LiDAR and Time of Flight (ToF) sensors provide fine details of semantics. There are two parts to this task 3D segmentation and classification of points to assign them an object class. 3D segmentation methods can be classified as edge-based, region-growing, model fitting, hybrid, and machine learning approaches (Grilli et al., 2017). This research area has benefited by harnessing the abilities of Deep learning also, but it has limitations due to coarse voxel predictions and a lack of global consistency in point clouds (Tchapmi et al., 2017). 3D segmentation remains challenging due to the order-less structure of the point cloud. Classification of 3D point clouds assigns each point with semantic information about the class. 3D point cloud classification has gained lots of interest among researchers in past years to classify LiDAR data using contextual information. (Weinmann et al., 2013; Guo et al., 2014; Niemeyer et al., 2014; Schmidt et al., 2015; Weinmann et al., 2014; Xu et al., 2014; Hackel et al., 2016).

2.3 2D-3D Semantic Information Enrichment

3D point cloud data segmentation and classification using 2D data can be broadly classified into two types: projection-based methods and fusion-based methods. Projection-based methods typically project 3D data onto a 2D image or set of images and then perform subsequent segmentation using image processing methods to enrich the semantic information. This approach benefits from the mature development of 2D semantic segmentation. Colored 3D point clouds have been projected onto RGB images in spherical projection and a convolutional neural network (CNN) is used for semantic segmentation of 3D point clouds (Castillo et al., 2021; Tabkha et al., 2019). Fusion-based techniques seek to combine original 3D point clouds with 2D semantic data to provide an enhanced semantic representation. These techniques often incorporate geometric processing methods with deep learning models. One method is to use deep learning models, such as PointNet (Qi et al., 2017a) or PointNet++ (Qi et al., 2017b), to extract features from the 3D point cloud and merge them with the semantic data collected from 2D photos. In order to project individual perspective views, Eder et al., (2020) segmented a spherical panoramic image into tangential icosahedral planes. These projections were then input into a pre-trained 2D semantic segmentation CNN for fusion.

3. METHODOLOGY

In this section, we will discuss the methodology employed in this research paper. We start with a discussion on the data that is required for this approach. In this paper, we have semantically enriched a 3D point cloud prepared using RGB-D images employing Deep learning for semantic segmentation of 2D images to a new image with each class of object represented as a unique color. Thus, a segmentation map or image is generated from RGB image of the scene. This segmentation map is used along with the RGB, depth and confidence images to generate a 3D point cloud for each image of the data collected.

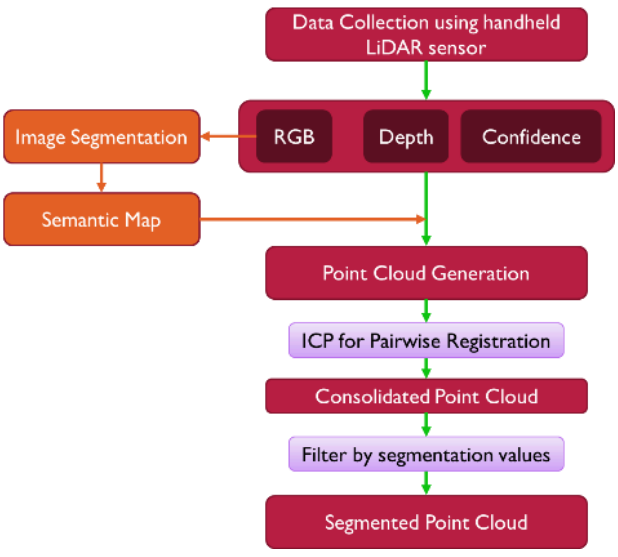


Figure 1. Schematic of the overall methodology.

Therefore, for the data part, we require an acquisition system that can acquire 2D RGB images with depth data. There are a lot of sensor systems available which can be used to obtain RGB-D images such as Time of Flight (ToF) sensor, Stereo vision sensors, Structured Light sensors or LiDAR sensors. These sensors provide colored images having brightness information in three channels i.e. red, blue and green and a corresponding depth image which is an array of pixels with brightness values in the gray channel equivalent to the distance of each pixel from the sensor.

3.1 Dataset Acquisition

RGB-D data is collected using an Apple iPad pro 3rd Generation device. RGB color information is collected using the 12 mega-pixel camera sensor onboard the iPad device with $f/1.8$ aperture. In addition to a camera sensor iPad also has a Time-of-Flight (ToF) 3D LiDAR sensor which collects depth information about the environment. Figure 1 shows data collection and the sensor setup of the iPad pro which has a main camera, ultra-wide camera, and LiDAR sensor. Each colored image is supplied with a corresponding depth image with distance from the sensor as the value assigned to that pixel on the image. This information is crucial in generating a 3D point cloud of the scene captured as we can translate each image pixel to the corresponding depth to prepare a 3D point cloud. But this point cloud does not have proper transformation applied to it as each 3D point cloud is in its own camera coordinate system. Thus, it lacks positional and rotational information which can accurately place it in the world coordinate system.

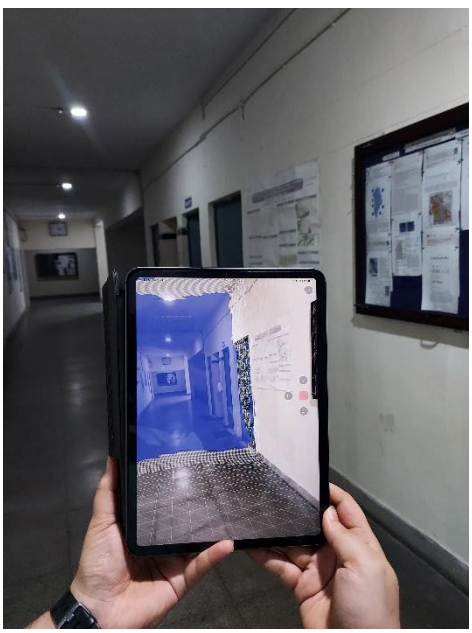


Figure 2. Data collection using a handheld LiDAR sensor on iPad.

The Polycam application for 3D scanning with LiDAR was used to collect data by utilizing its capabilities to record and store RGB-D data. It also provides an additional confidence map that contains a score for each pixel in the corresponding depth map. This score denotes the accuracy of the depth calculated for the pixel. The confidence map is extremely useful and makes the task of removing ambiguous depth values from the recorded data effortless from the start. The confidence map has pixel values in the gray channel ranging from black to white. Black corresponds to zero or no confidence and white corresponds to 1 or maximum confidence. Thus, while preparing a point cloud this confidence map can be used to filter out the points with low confidence scores and generate a better initial point cloud for each image.

Data collected using Polycam with Apple iPad Pro for the reconstructed scene contained 600 color images, 600 depth maps, and 600 confidence maps. This data is accompanied by ancillary text data which provides us with valuable information about the sensors' intrinsic parameters, i.e., focal lengths in the x-direction and y-direction, x & y position of the principal point, width & height of the image, values of the elements of camera view matrix. The size of the RGB images is 1024 by 768, whereas the size of depth and confidence images is 256 by 192 which is 1/4th the size of colored images. The maximum depth that is recorded by the ToF LiDAR sensor is 5 meters. The data was recorded for a portion of the ground floor corridor of the Geomatics Engineering Building with a floor area of 120 meter2 and length of the corridor equal to 28 meters. This area was selected for the purpose of testing and analysis of the algorithm as it is a relatively uncomplicated indoor environment that does not contain cluttered or unorganized objects. S-pattern was utilized while capturing images by ensuring that at least 60% overlap between subsequent images. The data was checked for the presence of any blurry images. This ascertains the data to have enough distinct features even in a small set of images.

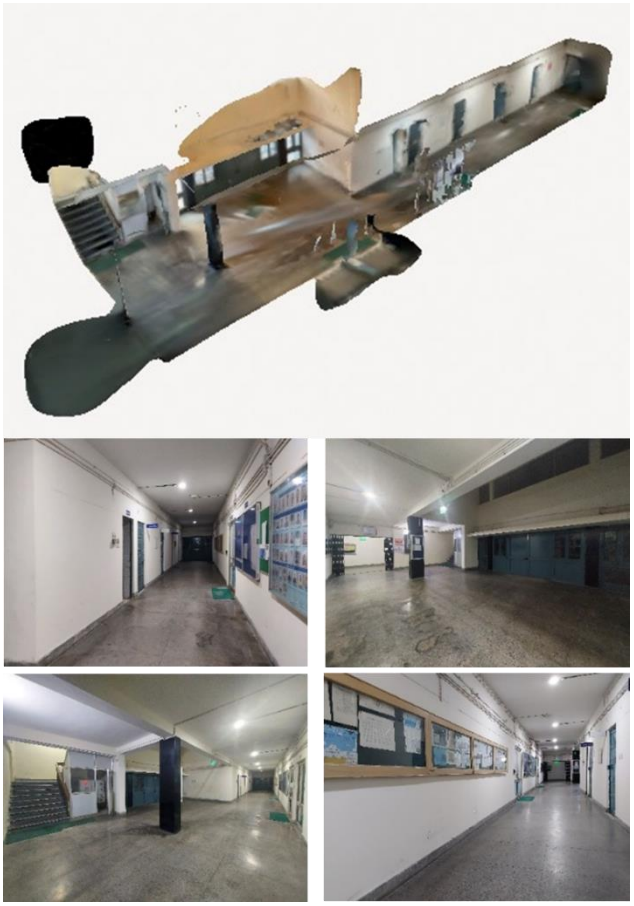


Figure 3. Overview of the reconstructed indoor scene and the raw images.

3.2 2D Semantic Segmentation

To generate a segmentation map of the indoor environment we use deep learning for semantic segmentation. We employed

DeepLabv2 with ResNet for indoor scene segmentation. DeepLabv2 with ResNet is a state-of-the-art deep learning model designed for image semantic segmentation. Leveraging the backbone architecture of ResNet, DeepLabv2 shows a staggering improvement in performance for semantic segmentation tasks. DeepLabv2 employs atrous spatial pyramid pooling (ASPP) module, enabling it to amass multi-scale contextual information for accurate prediction at different scales and accurately delineating object boundaries. In addition to ASPP, DeepLabv2 also incorporates a fully connected conditional random field (CRF) module as a post-processing step which assists in refining the segmentation map and optimizing the boundaries of segmented regions. Thus, DeepLabv2 with ResNet demonstrates its competency in undertaking complex semantic segmentation tasks.

The model used was pre-trained on the large-scale COCO dataset for panoptic segmentation on more than 330,000 annotated images across 80 things and 91 stuff classes. This extensively trained model effectively generalizes and performs robustly in various real-world scenarios. Pre-trained DeepLabv2 model could classify each pixel into different classes but for our use case, we have used an already-trained model for inference in indoor scene semantic segmentation. A pre-trained deep learning model is used to demonstrate that the developed methodology can utilize any semantic segmentation deep learning model from an already available model garden, thus enhancing the utility of this approach. A segmentation map is generated using DeepLabv2 with ResNet which will act as an input while generating a 3D point cloud for each image. Figure 3 shows the segmented maps inferred using Deeplabv2 with ResNet50 from RGB color images with classification labels.

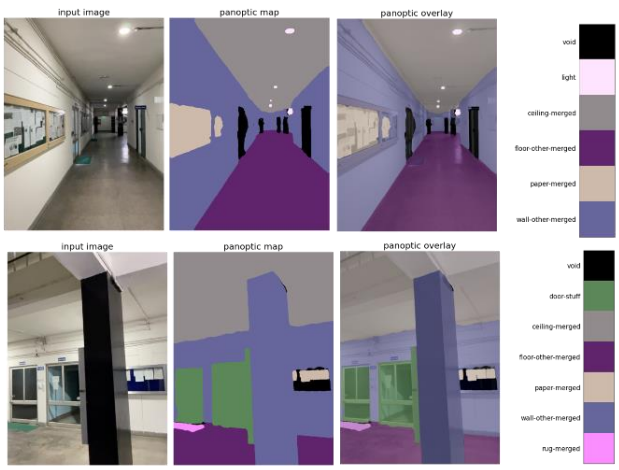


Figure 4. Segmented maps from RGB images with label information.

3.3 Point Cloud generation with Semantic Information for each Image

Once we have gathered all the input data for an indoor scene, which are RGB images, Depth maps, confidence maps, segmentation maps, and ancillary data for each image, we can begin the process of 3d point cloud generation. The first step is to check for data consistency i.e., ensuring that each RGB image has its respective depth, confidence, and segmentation maps available and has the exact dimensions as the RGB image per the information derived from the ancillary data file. We will also remove blurry or out-of-focus images to get an adequate 3D point cloud. Once preliminary checks are over, we proceed to generate a 3D point cloud from 2D images. Data collected using a camera exists in a 2D pixel coordinate system *I(x, y)* with its origin at the top left corner and positive x and y directions towards right and down, respectively. To generate a 3D point cloud from the 2D image, first, the origin of the pixel coordinate system is transferred from the top left corner to the principal point (c_x, c_y) which is the intersection of the optical axis and image plane. This is done by subtracting c_x from I_x and c_y from I_y, where c_x and c_y are coordinates of the principal point and I_x and I_y are coordinates of pixels in the pixel coordinate system. The second step is to use the principle of similar triangles and from the geometry explained in figure 5, use the focal length of the camera f_x (focal length in the x-direction) & f_y (focal length in the y-direction), and depth value 'z' from the depth map to compute the X_c, Y_c, Z_c position in the camera coordinate system $C(X_c, Y_c, Z_c)$ of the 3D point for each pixel of the image which has a value on confidence map above a threshold value.

$$X_c = (I_x - c_x) * {}^{z}/_{f_x} \quad (1)$$

$$X_y = (I_y - c_y) * {}^{z}/_{f_y} \quad (2)$$

Above mentioned equation 1 & 2 are used to calculate X and Y coordinates of points in camera coordinate system respectively. The 3D coordinates for all the points generated for each image are stored in the memory. Class information is appended pointwise by deriving it from the semantic segmentation map generated using deep learning. This process is imitated for the whole set of images and 3D point clouds are sequentially stored.

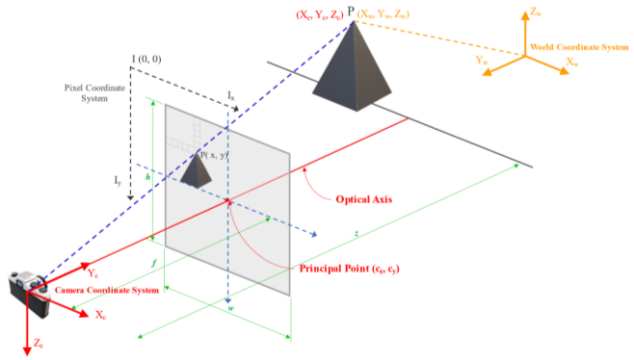


Figure 5. Geometry of image transformation from Pixel Coordinate System to Camera Coordinate System.

Each image from the dataset generates a 3D point cloud with X, Y, and Z positions of each point along with attributes about the actual color carried forward from RGB image and semantic class information as a pseudo color representing the class to which that point belongs. Each point has data stored as $p_i = X_c, Y_c, Z_c, R, G, B, R', G', B'$, where p_i represents the i^{th} 3D point, X_c represents the position in the x-direction, Y_c represents the position in the y-direction, Z_c represents the position in the z-direction, R represents the value in Red channel, G represents the value in Green channel, B represents the value in Blue channel, R' represents the value in Red channel for pseudo color, G' represents the value in Green channel for pseudo color, B' represents the value in Blue channel for pseudo color.

3.4 Co-registration of Point Clouds

3D point clouds generated from all the images available are in the camera coordinate system of that image and hence are unaligned with each other to replicate the actual geometry in the

world coordinate system. Correspondence estimation is used to approximate the transformation which is applied to each point cloud such that the 3D indoor scene can be recreated. We exploited the Open3D Python library to perform the correspondence estimation and pairwise registration of the point clouds. Open3D is an open-source library that is fast, easy to use and supports 3D data processing workflows helpful in performing various operations on point cloud data (Zhou et al., 2018).

Algorithm 1 Co-registration of Point Clouds

1. Import Point Clouds
   - Downsample point clouds using desired voxel size for efficient computation.
   - Estimate Normals.
2. Create Pose Graph.
   - Full registration for all the fragment point clouds.
     - Pairwise Registration of fragments using ICP (Point to Plane).
       - Coarse registration with a larger value of maximum correspondence distance.
       - Fine registration with a smaller value of maximum correspondence distance.
     - If the target cloud is a consecutive cloud to the source cloud which is the odometry case:
       - Try:
         - Color ICP for finer registration and use the transformation achieved.
       - Else:
         - Use the transformation achieved using full registration.
         - Add the new transformation as a node and calculate the pose graph edge.
     - Else:
       - Calculate the pose graph edge.
3. Pose Graph Optimization.
   - Global Optimization of the generated pose graph in the previous step.
   - Get the pose of each node and save it as a 4 x 4 matrix.
4. Alignment of fragment point clouds.
   - Multiply the transformations with the coordinates of points to align all the fragment point clouds in the world coordinate system.

First step is to estimate normals for the point cloud, this is carried out using Open3D built-in function. This step is essential for pairwise geometric registration using point-to-plane metric in the subsequent step. In the second step, pairwise geometric registration is done which comprises of two steps. Iterative Closest Point (ICP) algorithm is used to register the pair of point clouds twice, which require the estimation of surface normal in point clouds to aid in correspondence estimation, distance calculation and alignment convergence. First geometric registration is done by taking the maximum correspondence distance as a larger value for coarse registration and an identity matrix as the starting transformation matrix. The transformation matrix generated from this step is passed to the next reiteration of ICP for fine registration by reducing the maximum correspondence distance and using the transformation matrix from coarse registration as the starting transformation matrix to fit iteratively. Since the ICP algorithm always converges gradually to the nearest local minimum of mean square distance metric by using two different maximum correspondence distances we reduce the prospect of two point clouds being aligned imperfectly. This step provides us with a 4 by 4 transformation matrix and ICP information matrix to be used for further optimization.

A pose graph optimization is set up which has two key elements: nodes and edges. Each node is connected to the graph by edge constraint which defines the relative pose between nodes that aligns the nodes in the world coordinate system. It is usually seen that pairwise alignments are error-prone (Choi et al., 2015), therefore, pose graph edges are divided into two classes. Odometry edges connect neighboring nodes which can be aligned using a variant of the ICP algorithm. Color ICP algorithm (Park et al., 2017) is employed to register the point clouds accurately for the odometry case. Loop closure edges establish a connection between non-neighboring nodes which is aligned by global registration although it is less reliable. Pose graph optimization is performed using global optimization Levenberg Marquardt method which is the recommended method since it gives better convergence. Global optimization is executed which optimizes poses by considering all the nodes and edges and seeks to achieve a tight global alignment. Pose or Transformation matrix for each point cloud which is a 4 x 4 matrix containing rotational and translational elements is exported for aligning all the fragment point clouds to world coordinate system.

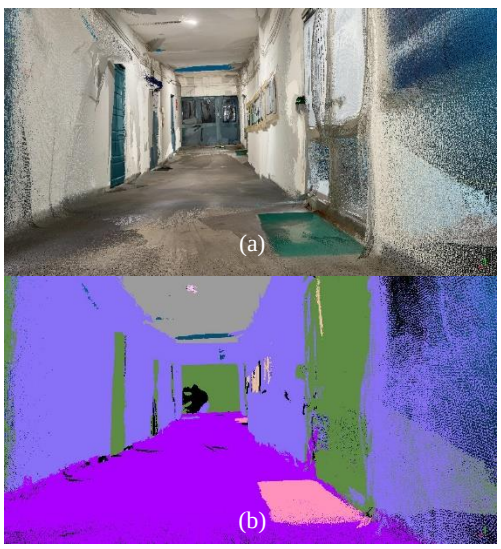
3.5 Global Alignment of Point Clouds and Classification

Transformation matrix for each point cloud obtained by global optimization yields the best alignment for fragmented point clouds in the world coordinate system. Using equation 1, the point cloud can be converted from a camera coordinate system to the world coordinate system. The transformation matrix obtained in the previous stage is a 3 by 4 matrix comprising of 9 rotational elements and 3 translation elements multiplied with coordinates of points in the homogeneous camera coordinate system. This yields the transformed coordinates for points in each fragment point cloud. Multiplying each fragment point cloud with its corresponding transformation matrix aligns all the point clouds to best represent the indoor scene.

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = [R \mid t]_{3x4} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (3)$$

where, $R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$, $t = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$

Each point in transformed fragment point cloud has information about its position in the world coordinate system, actual color in RGB and pseudo color which represents the class to which object belongs. To combine all the point clouds, cloud compare software is used. All the point clouds are imported in the cloud compare software with default settings. Once imported, all the point clouds can be visualized already aligned as per the transformation applied. Combine function in the cloud compare can be used to convert fragment point clouds into a single combined point cloud.



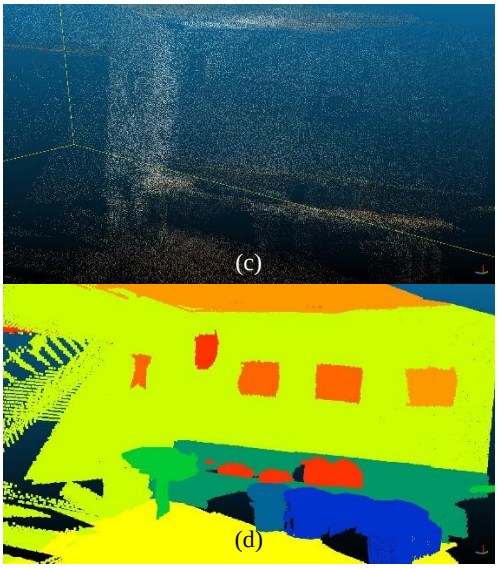


Figure 6. a) Color Point Cloud (Internal Dataset), b) Segmented point cloud (Internal Dataset), c) Color Point Cloud (2D3DS Dataset), d) Segmented point cloud (2D3DS Dataset).

The pseudo color information is now used to segment the combined point cloud as per class. Pseudo color information is represented as scalar value in cloud compare software. Scalar values can be split into using values. Since pseudo colors are distinct color groups, colors representing each class are differentiated on the histogram. These gaps and value ranges are identified using histogram and split function is used to split the combined point cloud into segmented point clouds as per semantic classification.

4. RESULTS

We conducted tests on various configurations to evaluate the effectiveness of our methodology as described in Section 3. In addition, we performed evaluations on both an internal dataset and a subset from the widely used 2D3DS benchmark dataset (Armeni et al., 2017). To assess the performance of our proposed pipelines, we need to establish appropriate performance metrics. While the 2D3DS dataset provides labels, our internal dataset does not. Therefore, we manually labeled the ground truth data using image processing software with a graphical user interface, as our focus is on testing rather than training a convolutional neural network (CNN). Since we are working with test data only and do not require a large dataset, manual annotation suffices. A subset of 2D3DS dataset was used for comparison. Lounge 1 of area 3 was used for the assessment using our proposed methodology. For evaluating the performance of each scenario, we employ the Intersection over Union (IoU) metric, which is widely used in the field. The IoU is computed over the confusion matrix C of size N × N, where N represents the number of classes (8 in our dataset i.e. wall, ceiling, floor, door, rug, light, paper, uncategorized). Each entry cij in the confusion matrix denotes the number of samples belonging to the ground truth class i that are predicted as class j. The per-class IoU is calculated using the formula:

$$IoU_i = \frac{c_{ij}}{c_{ij} + \sum_{j \neq i} c_{ij} + \sum_{k \neq i} c_{ki}} \quad (4)$$

The mean *IoU* (*mIoU*) is then computed as the average of the per-class *IoU* values:

$$mIoU = \frac{1}{N} \sum_{i=1}^{N} IoU_i \quad (5)$$

By using these performance metrics, we can quantitatively evaluate the effectiveness of our methodology across different scenarios.

Class	Unclassified	wall	Ceiling	rug	door	paper	light	floor	Row Total	Producer Accuracy	IoU
Unclassified	3201	433	23	195	847	478	5	265	5447	58.77	0.51
wall	0	39256	1647	0	433	23	0	1169	42528	92.31	0.87
Ceiling	0	511	10684	0	0	0	14	0	11209	95.32	0.83
rug	13	0	0	776	0	0	0	61	850	91.29	0.74
door	486	239	0	0	11895	0	0	33	12653	94.01	0.85
paper	266	165	0	0	0	689	0	0	1120	61.52	0.43
light	12	0	6	0	0	0	11	0	29	37.93	0.23
floor	0	1372	0	6	54	0	0	24732	26164	94.53	0.89
Column Total	3978	41976	12360	977	13229	1190	30	26260			
User Accuracy	80.46757164	93.52	86.4401	79.427	89.92	57.899	36.67	94.18			

Figure 7. Confusion Matrix of Internal Dataset with User Accuracy, Producer Accuracy and *IoU* for each class.

Figure 7 above shows the confusion matrix for our internal dataset. We can use this to calculate user and producer accuracy for each class of object in 3D point cloud. An overall accuracy of 91.24% was achieved. The accuracy of labels is dependent on the CNN model used and the accurate registration of the fragmented point clouds. Using the discussed metrics, the segmentation results on our internal dataset and 2D3DS dataset are as given in the table below:

Method	Overall Accuracy	*mIoU*
Ours (Internal dataset)	91.24%	0.67
Ours (2D3DS dataset)	89.63%	0.49
Castillo et al. 2021 (2D3DS dataset)	89.6%	0.47

Table 1. Comparison of performance of our proposed approach on the internal dataset & 2D3DS benchmark dataset and Castillo et al. 2021 method on 2D3DS benchmark dataset.

5. DISCUSSIONS

We conducted experiments on real-world datasets collected using a LiDAR sensor onboard a handheld mobile device in different scenes to evaluate the performance of our algorithm. The results demonstrate the algorithm's effectiveness in generating semantically enriched point clouds and are highly computationally efficient. The semantically enriched 3D point cloud generated using our proposed algorithm has the potential to revolutionize the fields of indoor localization, mapping, and navigation. The resultant semantically labeled point cloud yielded using the proposed algorithm becomes more informative, allowing for easier and more accurate analysis and interpretation. The accuracy of the data from LiDAR sensors in mobile devices is not comparable to Terrestrial Laser Scanners (TLS) or Total Stations but is acceptable in most application cases (Díaz Vilariño et al., 2022). This is mainly because faster data collection speeds up model generation while incorporating 2D semantically segmented maps reduces the computational complexity of 3D segmentation. This approach significantly reduces the time and resources required for indoor scene analysis and modeling by eliminating the need for a separate 3D segmentation step, which can be computationally expensive and time-consuming. The resulting point cloud is also more informative and provides a better understanding of the indoor environment. Additionally, the enriched point cloud can also be used for 3D modeling of indoor environments, which is essential for various applications, such as architectural design, indoor navigation, and emergency response planning.

RGB-D datasets obtained from handheld mobile devices offer a promising future for fast and easy data collection. They provide a wealth of information that combines color and depth data, enabling more comprehensive scene understanding. The convenience and accessibility of handheld mobile devices make

RGB-D data collection accessible to a wider audience, while the real-time nature of data acquisition allows for dynamic and iterative processes. Handheld mobile devices, such as smartphones and tablets, are becoming increasingly equipped with depth sensors, such as time-of-flight (ToF) or structured light sensors, alongside RGB cameras. These sensors capture depth information by measuring the distance between the sensor and the objects in the scene, allowing for the creation of accurate depth maps. The convenience and accessibility of handheld mobile devices make data collection more accessible to a larger user base. The ubiquity of smartphones and tablets means that anyone can capture RGB-D data with relative ease, eliminating the need for expensive and specialized equipment. In addition to this, utilizing off-the-shelf 2D CNN models for segmentation is relatively easier than the 3D segmentation of point cloud data. Also, unlike TLS&RGB & pano, which uses 3D point cloud data to synthesize color images for inferencing semantic segmentation, our approach uses the raw RGB-D data itself for segmentation and creates a unidirectional pipeline. This also eliminate the need for training CNN model on panoramic or any other projected images. The proposed algorithm can be improved by employing better 2D semantic segmentation models like DeepLabv3++ to improve the accuracy of segmentation. An improved co-registration algorithm can also be used to maintain the geometrical consistency of the recorded dataset. We have seen in our experimentation that the accuracy of 3D representation can also be improved by employing a filtering algorithm to remove noise and redundant points.

The semantically enriched point cloud can be used to generate 3D models with more accurate spatial information and semantic meaning, which can aid in these applications. The proposed algorithm offers a promising solution to the challenge of indoor mapping, localization and path planning, providing a new avenue for research in the field. Our work contributes to the growing body of research focused on developing advanced techniques for indoor mapping and localization. The proposed algorithm has the potential to benefit a variety of applications, such as robot navigation, augmented reality, and building inspection. Overall, the proposed method provides a cost-effective and efficient way to semantically enrich 3D point cloud data, which can be utilized in various applications.

6. CONCLUSION

We have introduced a pipeline for semantic segmentation of point clouds in indoor scenes, utilizing the RGB-D dataset. Our approach performs semantic segmentation using off-the-shelf 2D convolutional neural networks (CNNs) on RGB color images. The segmented image is supplied along with RGB and depth image to generate fragments of semantically enriched point clouds. By employing a pre-trained 2D CNN for semantic segmentation, we achieve satisfactory results without the need for manually annotated training data or specialized 3D point cloud networks. This allows us to capitalize on large 2D labeled datasets for 3D point cloud semantic segmentation. Additionally, our findings demonstrate that we achieved reasonable class labels using a network trained on more commonly available rectilinear images. Unlike other studies which perform 2D segmentation on reprojected panoramic images, our proposed methodology incorporates semantic information from the initial stages of RGB-D dataset conversion to 3D point cloud. This will enable researchers to directly utilize already available state-of-the-art 2D semantic segmentation Neural Network models rather than training on a custom dataset. This significantly reduces the workload and expedites the integration of new deep learning frameworks for 3D point clouds. Our algorithm can benefit further by using a more robust algorithm for fragmented point cloud alignment. The modular nature of our pipeline enables us to rapidly test and deploy newer segmentation and alignment algorithms.

ACKNOWLEDGEMENT

The author acknowledges the financial support provided by the Dean of Resources and Alumni Affairs (DORA), Indian Institute of Technology to present this research paper at the ISPRS Geospatial Week 2023 conference.

REFERENCES

Armeni, I., Sax, S., Zamir, A.R., Savarese, S., 2017. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint* arXiv:1702.01105, https://doi.org/10.48550/arXiv.1702.01105.

Besl, P. J., McKay, N. D.,1992: A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14(2), 239-256.

Boykov, Y., Veksler, O., Zabih, R., 2001. Fast Approximate Energy Minimization Via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(11), 1222-1239.

Sanchez Castillo, E., Griffiths, D., Boehm, J., 2021. Semantic Segmentation of Terrestrial Lidar Data Using Co-Registered RGB Data, *International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences*, XLIII-B2-2021, 223–229. https://doi.org/10.5194/isprs-archives-XLIII-B2-2021-223-2021.

CloudCompare (version 2.10.2) [GPL software], (2019). Retrieved from http://www.cloudcompare.org/ (24 February 2019).

Dhanachandra, N., Manglem, K., Chanu, Y.J., 2015. Image Segmentation using K-means Clustering Algorithm and Subtractive Clustering Algorithm. *Procedia Computer Science* 54, 764-771.

Díaz-Vilariño, L., Tran, H., Frías, E., Balado, J., Khoshelham, K.,2022. 3d Mapping of Indoor and Outdoor Environments using Apple Smart Devices, *International Archives of the Photogrammetry Remote Sensing and Spatial Information Sciences*, XLIII-B4-2022, 303–308, https://doi.org/10.5194/isprs-archives-XLIII-B4-2022-303-2022.

Eder, M., Shvets, M., Lim, J., Frahm, J.M., 2020. Tangent Images for Mitigating Spherical Distortion. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12426-12434.

Grilli, E., Menna, F., Remondino, F., 2017. A Review of Point Clouds Segmentation and Classification Algorithms. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W3, 339–344. https://doi.org/10.5194/isprs-archives-XLII-2-W3-339-2017.

Guo, B., Huang, X., Zhang, F., Sohn, G., 2014. Classification of Airborne Laser Scanning Data using JointBoost. *ISPRS Journal of Photogrammetry and Remote Sensing* 92, 124-136.

Hackel, T., Wegner, J.D., Schindler, K., 2016. Fast Semantic Segmentation of 3D Point Clouds with Strongly Varying Density. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-3, 177-184. https://doi.org/10.5194/isprs-annals-III-3-177-2016

Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1(4), 321-331. https://doi.org/10.1007/BF00133570.
Low, K.L., 2004. Linear Least-squares Optimization for Point-to-plane ICP Surface Registration. Technical Report TR04-004, Department of Computer Science, University of North Carolina, Chapel Hill, NC, USA.

Lyu, Y., Huang, X., Zhang, Z., 2020. Learning to Segment 3D Point Clouds in 2D Image Space, *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12252-12261.

Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., Terzopoulos, D., 2021. Image Segmentation using Deep Learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44(7), 3523-3542.

Najman, L., Schmitt, M., 1994. Watershed of a Continuous Function. *Signal Processing* 38(1), 99-112.

Niemeyer, J., Rottensteiner, F., Soergel, U., 2014: Contextual Classi-fication of Lidar Data and Building Object Detection in Urban Areas. *ISPRS Journal of Photogrammetry and Remote Sensing* 87, 152-165.

Nock, R., Nielsen, F., 2004. Statistical Region Merging. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(11), 1452–1458.

Otsu, N., 1979. A Threshold Selection Method from Gray-level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics* 9(1), 62– 66.

Park, J., Zhou, Q.Y., Koltun, V., 2017. Colored Point Cloud Registration Revisited. *In Proceedings of the IEEE international conference on computer vision*, 143-152.

Plath, N., Toussaint, M., Nakajima, S., 2009. Multi-class Image Segmentation using Conditional Random Fields and Global Classification. *In Proceedings of the 26th Annual International Conference on Machine Learning*, 817-824.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. Pointnet: Deep Learning on Point Sets for 3D Classification and Segmentation. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 652-660.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Conference on Neural Information Processing Systems* 30.

Choi, S., Zhou, Q. Y., Koltun, V., 2015. Robust Reconstruction of Indoor Scenes. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5556-5565.

Tabkha, A., Hajji, R., Billen, R., Poux, F., 2019. Semantic Enrichment of Point Cloud by Automatic Extraction and Enhancement of 360 Panoramas. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W17, 355–362. https://doi.org/10.5194/isprs-archives-XLII-2-W17-355-2019, 2019.

Tchapmi, L., Choy, C., Armeni, I., Gwak, J., Savarese, S., 2017. Segcloud: Semantic Segmentation of 3D Point Clouds. *In 2017 International Conference on 3D Vision (3DV)*, 537-547.

Weber, M., Wang, H., Qiao, S., Xie, J., Collins, M.D., Zhu, Y., Yuan, L., Kim, D., Yu, Q., Cremers, D., Leal-Taixe, L., 2021. Deeplab2: A Tensorflow Library for Deep Labeling. *arXiv preprint*, arXiv:2106.09748, https://doi.org/10.48550/arXiv.2106.09748.

Weinmann, M., Jutzi, B., Mallet, C., 2013. Feature Elevance Assessment for the Semantic Interpretation of 3d Point Cloud Data. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-5/W2, 313–318. https://doi.org/10.5194/isprsannals-II-5-W2-313-2013, 2013.

Weinmann, M., Jutzi, B., Mallet, C., 2014. Semantic 3D Scene Interpretation: A Framework Combining Optimal Neighborhood Size Selection with Relevant Features. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3, 181–188, https://doi.org/10.5194/isprsannals-II-3-181-2014.

Weinmann, M., Schmidt A., Mallet C., Hinz S., Rottensteiner F., Jutzi B., 2015. Contextual Classification of Point Cloud Data by Exploiting Individual 3D Neighbourhoods. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3/W4, 271–278. https://doi.org/10.5194/isprsannals-II-3-W4-271-2015.

Xu, S., Vosselman, G., Oude Elberink, S., 2014. Multiple Entity based Classification of Airborne Laser Scanning Data in Urban Areas. *ISPRS Journal of Photogrammetry and Remote Sensing* 88, 1-15.

Zhou, Q.Y., Park, J., Koltun, V., 2018. Open3D: A Modern Library for 3D Data Processing. *arXiv preprint*, arXiv:1801.09847, https://doi.org/10.48550/arXiv.1801.09847.