ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

A HYBRID METHOD FOR POTHOLE DEPTH ESTIMATION: COMBINING LIDAR WITH POINT CLOUD

Sparsh Verma

Department of Computer Science and Engineering, Sharda School of Engineering and , Technology, Sharda University , Greater Noida, U.P.

Shashank Kumar

Department of Computer Science and Engineering, Sharda School of Engineering and Technology, Sharda University, Greater Noida, U.P.

Ayush Jindal

Department of Computer Science and Engineering, Sharda School of Engineering and Technology, Sharda University, Greater Noida, U.P.

Sandeep Kumar

Department of Computer Science and Engineering, Sharda School of Engineering and Technology, Sharda University ,Greater Noida, UP

ABSTRACT

This study presents an automated approach for detecting potholes and estimating their depth using LiDAR point cloud data. The methodology involves pre-processing raw LiDAR data, segmenting the road surface, clustering pothole regions, and calculating their dimensions. The proposed system utilizes DBSCAN clustering and convex hull techniques to accurately identify and measure potholes. The methodology involves collecting 3D point cloud data from LiDAR sensors, specifically from .las files, to ensure accurate surface reconstruction. The pre-processing stage applies voxel down sampling, statistical noise removal, and plane segmentation using RANSAC to differentiate road surfaces from potholes. Henceforth, an algorithm called DBSCAN is then employed that identifies the regions in which the potholes are present. Then an unambiguous extraction of pothole dimensions is done using the convex hull estimation. This methodology not only provides an accurate measurement of the pothole depth but also aids in real-time view of the potholes, which is important for the maintenance of roads and ensuring road safety.

General Terms

Algorithms, Measurement, Computer Vision, Data Processing,

Remote Sensing

Keywords

Computer Vision, DBSCAN clustering, LiDAR, Point Cloud, Potholes, RANSAC

1. INTRODUCTION

Detecting and estimating the dimensions of potholes poses a key challenge to road maintenance and ensuring road safety. There have been studies which show that if the repairment of deteriorating roads is delayed for a long time then it can result in a huge spike of approximately seven times in the maintenance prices within six years. As a result, the financial costs increase and there is an increase in the risks involved with the automobiles and commuters. Potholes not only cause damage to the vehicle's suspension but can also lead to accidents. Potholes also play a major role in the degradation of road surfaces. To counter this issue, the slow and ineffective traditional manual investigations are replaced by contemporary technologies like LiDAR based study of the point clouds, computer vision, and deep learning. These modern methods plays a crucial role in making the pothole detection error-free and automated, thus ensuring more effective maintenance of roads. This paper introduces a systematic approach to estimating pothole depth using LiDAR point cloud data. The method combines multiple techniques, including statistical filtering, DBSCAN clustering, and convex hull analysis, to improve accuracy. The process starts with collecting 3D point cloud data from LiDAR sensors, specifically using .las files, to ensure a highly detailed reconstruction of the road surface. The data then undergoes pre-processing, which includes reducing noise through statistical filtering, down sampling data points to improve efficiency, and using RANSAC to separate the road surface from potholes. The DBSCAN clustering algorithm then helps identify pothole regions, while convex hull estimation extracts their exact shape and size. After identifying pothole regions, the system calculates the depth by measuring the difference between the plane of road surface and lowest point within the pothole. This step ensures precise depth estimation, which is crucial for assessing road damage severity. Additionally, the extracted features from convex hull analysis help classify potholes based on their depth and size, allowing road maintenance teams to prioritize repairs effectively.

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

Convex hull analysis accurately defines the boundaries of the detected potholes. The area and perimeter of each and every pothole can be estimated by creating the convex hull around the clustered points of pothole. Additionally, this method helps in removing the noise and also the anomalies from the point cloud data which helps to assess or study the potholes accurately. The convex hull offers a reliable foundation for figuring out the pothole's volume, which is crucial for estimating and planning the quantity of the material needed for repair. This approach also provides the real-time view of pothole metrics. The size, volume and depth of potholes can be analysed interactively by the road authorities by making use of the system's ability to display a 3-D view of the potholes detected using Open3D. This visualization mainly helps in the planning of repair and maintenance work of roads. The given approach is not only applicable to roads but also in underwater The method is not just limited to regular roads but can also be applied in underwater domain.

This approach offers an accurate and automated way to estimate the depth of potholes by integrating the point cloud processing based on LiDAR, plane segmentation using Random Sample Consensus(RANSAC), DBSCAN clustering, and convex hull estimation. The proposed framework provides the base for further study and advancements in automated detection of potholes which will eventually improve road safety and repair plans. Using this technology, the road authorities can get assisted in quickly finding and addressing the damages caused on roads, thus ensuring smooth and safe travel for each and everybody. Furthermore, the accuracy of pothole detection can be levelled up by incorporating machine learning techniques. The software learns to differ between the potholes, cracks on roads and another irregularities by getting the models trained on large datasets of LiDAR point clouds. This system can be advanced more in future by incorporating the integration of AI-driven anomaly detection, real-time edge computing for faster processing, and LiDAR scanning with the help of drones for wider area coverage in road assessment.

The major contributions of the project work are as indicated :

Estimation of Depth based on LiDAR: To improve the depth estimation of potholes, this methodology uses high resolution LiDAR point cloud data. LiDAR provides accurate three dimensional spatial information in contrast to the conventional monocular depth estimation methods.

Surface Fitting and Noise Reduction: This is achieved by using RANSAC algorithm. This algorithm represents the road surface within the point cloud data thus differentiating the road plane from potholes. This technique helps in eliminating the eliminate noise and outliers, thus providing stable and reliable depth calculations.

DBSCAN Clustering and Convex Hull Analysis: The DBSCAN clustering is used for grouping the relevant points of data while on the other hand, convex hull analysis gives the exact shape and size of the pothole, thus the potholes are segmented accurately. This results in providing the precise estimation of volume and classification based on the level of severity.

Automated Pothole Severity Assessment: Apart from simply detecting the potholes, this method also helps in classifying the potholes on the basis of their area, depth and volume. The involvement of convex hull estimation helps in providing a more detailed examination of road conditions, thus helping in improved planning and decision making of the infrastructure.

Real-Time Visualization and Practical Implementation: By using Open3D, an interactive 3D visual of the potholes is obtained. This allows road authorities and engineers to visually study the dimensions and severity of potholes, thus making this approach a real world problem solving approach.

2. LITERATURE REVIEW

The algorithms used earlier in the past for pothole detection utilized two dimensional image segmentation[6]. Usually, there does not exist much reliablity in the two-dimensional methods when there are issues regarding bad light and reduced visibility. This is also due to the reason that the two dimensional methods lack the perception of depth, so they can't compute the spatial dimensions and also the volume of potholes. The 3D techniques are thus very much needed. There are many advanced methods and algorithms developed for the segmentation of point cloud which rely on 3D CNNs[1]. Others try to inculcate the classical 2D segmentation algorithms directly by projecting the 3D-point clouds onto the 2D-images following the classical methods of 2D-semantic segmentation. This type of method leads to the loss of specific or key information. There are some more different approaches related to this field of work[9],[10], but many of them undergo critical or minor limitations. For example, Super Point Graph or (SPG) and PCT capture almost every characteristic of a point cloud but leads to a huge overhead that diminishes the efficiency. Whereas, in PCT, this ends up with low efficiency due to the complexity and bulkiness as inherent in its transformers, which further incurs high computing power plus the long processing times, and that too for such simple point clouds, whereas in SPG, this is attributed to a kind of preprocessing through super point graphs along with a strong reliance on geometrically homogeneous partitioning, sometimes turning wrong segmentation [11].

The RandLA-Net is designed in such a way that it takes less amount of time in extracting out the properties and features from the point clouds. It employs a module called Local Feature Aggregation module that learns and retains only geometric patterns but for those point clouds which are small and simple[12]. While on the other hand, in SCF-Net, it's cost is low. However, extracting complex strucures is a difficult task for the isolated SCF local feature extractor and it gives low results[5]. There are other methods which are beneficial for for segmentation such as Deep Learning methods. Most of the image segmentation for potholes are relying on two dimensional data and spatial features for computing the level of severity. Pothole detection and feature extraction can be enhanced by segmenting 3D point cloud[2].

Methods which are based on discretisation change the point clouds into discrete forms [7]; for example, it's a 3D grid known as voxels. These voxels are then gemented through fully 3D CNNs. They further helps in dealing with big and vast point clouds. This led to the development of the PointNet[8]. When MLP is applied to every point and feature aggregation to global features is done for every point , PointNet gets failed in determining the local features efficiently, which is very much important for the analysis of point clouds[9]. DBSCAN is an algorithm which implements density based clustering to find clusters of any shape or size and also to identify the

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

outliers efficiently[17] . SegDecNet++ is a method proposed in this work which segments each pixel of road and the cracks by using integrated training of the per-image and the per-pixel classification[18].

Hu et al advanced a more profound point cloud segmentation model of the head, body, limbs, ears and other parts of the pig with Point net++[14].The deep learning techniques are not subject to direct application to irregularly distrubuted 3D point clouds for the recognition of object. There exists three ways in which the point clouds can be transformed into normal data; however, the obtained data must be sent to the 3D target detection network. In the first technique, the 3D point clouds are projected onto the 2-D planes for the purpose of creating a 2D image. Then the 3D point clouds are transformed to three-dimensional voxel grids The third approach involves the encoding of 3D point cloud data into RGB images for the detection of interest 3D objects[14]. The LiDAR point clouds are filtered to identify and provide the location of potholes and their severity by detecting the 3D points below the ground plane, related to the pothole[15]. Deep learning methods work well in the segmentation since they can learn features by default, manage nonlinearity, adapt to various differences in point density, capture hierarchical structures, and provide robustness over noisy data. The width of each pothole is determined by taking ultimate value of the widths at every place in a specific subset of that pothole[19].

Potholes which are present outside the driver's line of sight in real time are identified using LiDAR sensor thus giving the driver ample time to respond. Once the pothole is identified, the GPS position of the pothole is acquired using a GPS sensor. This information about the pothole is stored in a cloud database[20]. Just like the segmentation of 2D-image , the segmentation of 3D point cloud data consists of its own set of problems. The first issue is irregular nature of point cloud data and it is difficult to interpret like an ordinary photo. Features are quite tough to be extracted from it because of a lack of texture in two- dimensional images. Until now, there is very limited research study on segmentation of the 3D point cloud data obtained from the potholes present on the roads. Most significantly, as seen in figure 1 below, false potholes, which are created between the asphalt aggregate particles or gaps, are mistaken with real potholes. False' potholes are created because debris falls off the asphalt aggregate while driving [3]. There was an asphalt aggregate lying in the gap of the asphalt aggregate. It was left intact. Interaggregate space where a small pocket had developed and caused the development of 'false' pothole.

3. METHODLOGY

This work presents a comprehensive hybrid approach for estimating pothole depth by integrating monocular depth estimation, RANSAC-based surface detection, and Lidar generated point clouds**.** The methodology follows a structured pipeline to ensure accurate and reliable depth estimation.

3.1 Data Collection

The dataset consists of LiDAR-generated .las files containing 3D point clouds of road surfaces. These files are collected using LiDAR-equipped vehicles or the iPhone 13 Pro's built-in LiDAR sensor, which provides high resolution depth mapping. The iPhone's LiDAR scanner enables efficient, portable, and cost-effective data collection, capturing highly detailed 3D reconstructions of road conditions. This approach eliminates the need for expensive UAV-based LiDAR mapping, making road inspection more accessible and scalable. The combination of vehicle-mounted LiDAR sensors and handheld iPhone LiDAR scanning ensures comprehensive point cloud generation for accurate pothole detection and depth estimation.

3.2 Load Point Cloud Data

Point cloud data is an important representation format in 3D spatial analysis, commonly used in LiDAR-based applications. The. las (LASer) file format is a standard binary file format used to keep LiDAR based point cloud data, preserving essential attributes such as spatial coordinates (x, y, z), intensity, classification, and other metadata. Efficient processing and visualization of this data require conversion into a structured format, such as Open3D's Point Cloud object.

3.2.1 Reading. LAS Files Utilising the laspy library

The laspy library refers to a Python-based tool specifically designed for reading, writing, and manipulating. las and .las files. It allows access to individual point attributes, enabling the extraction of three-dimensional coordinates essential for further processing. When a .las file is read, it provides raw point cloud data, typically structured as an array which contains x, y & z co-ordinates along with other characteristics like intensity, return number, and classification labels.

3.2.2 Extracting (x, y, z) Coordinates

Once the .las file is loaded using laspy, the primary interest for generating a 3D representation is the spatial positioning of each point. The x, y, and z coordinates are withdrawn from the point records, forming a structured dataset that represents the geometric shape of the scanned environment. These coordinates collectively define the topology of objects captured in the LiDAR scan.

3.2.3 Converting to an Open3D Point Cloud Object

Open3D is a powerful open-source library used for 3D data processing, visualization, and analysis. After extracting the x, y, and z coordinates from the .las file, these values are formatted into a structured NumPy array and mapped to an Open3D.PointCloud object. This conversion enables efficient visualization, manipulation, and further processing such as filtering, segmentation, and depth estimation. The Open3D Point Cloud object acts as a container that efficiently stores and processes 3D spatial information.

3.3 Pre-processing of the Point Cloud

Point cloud data consists of numerous 3D points captured by LiDAR sensors, each with coordinates (x, y, z). The primary goal of pre-processing is to make the data more manageable for processing and to improve the quality of subsequent analysis. Below, I'll explain the key steps involved in pre-processing point cloud data for pothole detection.

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

3.3.1 Voxel Down sampling

It is used to reduce the point cloud density and make the dataset more manageable for faster processing without significantly losing information. Voxel down sampling decreases the number of points by grouping nearby points into a smaller number of representative points. This is done by dividing the point cloud into a 3D grid of cubic cells (voxels), and replacing all points within each voxel with a single representative point, often the centroid of the points within the voxel. This results in fewer points to process, which speeds up computations, especially for large datasets. For a voxel size V, the point cloud is divided into the voxels, and each voxel can be represented by a centroid point $Pvoxel$. The centroid of a set of points $\{P_i = (x_i, y_i, z_i)\}$ in a voxel is calculated as:

$$\boldsymbol{P_{voxel} = \frac{1}{n}\sum_{i=1}^{n} X_i, \frac{1}{n}\sum_{i=1}^{n} Y_i, \frac{1}{n}\sum_{i=1}^{n} Z}$$

where n refers to the number of points in the voxel. This reduces the point cloud to fewer points, preserving the overall structure.

3.3.2 Noise Removal

To remove outliers or points that do not represent meaningful information in the scene, such as points being far from the main point cloud or are erroneous measurements. Noise removal techniques focus on detecting and removing points that are unlikely to be part of the desired object (e.g., the road surface or potholes). These outliers can be caused by errors in sensor measurements or environmental factors. A common method to detect outliers is statistical analysis, such as the statistical outlier removal method.

3.3.3 Normal Estimation

To compute the surface normal for each point present in the point cloud, which is crucial for segmenting different parts of the scene and detecting potholes. Surface normal are vectors perpendicular to the surface at each point. They provide information about the orientation of the surface and are essential for understanding the geometry of the environment. Normal estimation is typically done by analysing the local neighbourhood of each point. One common method is to fit a plane to the neighbouring points and calculate the normal of that plane.

For a given point $\boldsymbol{Pi}$ = ($\boldsymbol{xi}$, $\boldsymbol{yi}$, $\boldsymbol{zi}$), we consider a local neighbourhood consisting of the nearest neighbours, denoted as $\boldsymbol{Pi1}$, $\boldsymbol{Pi2}$, …, $\boldsymbol{Pin}$. Then a plane is fitted to these points, often utilizing the **principal component analysis (PCA)**. Then the normal vector n can be obtained by finding the eigen-vector that corresponds to the least eigenvalue of the covariance matrix of the neighbour points.

3.4 Ground Plane Estimation

To detect the road surface (ground plane) in the point cloud, which is essential for accurate pothole depth measurement. In a typical road scene, the road surface is assumed to be relatively flat and horizontal, which can be modelled as a plane. Estimating this ground plane is important to correctly align the point cloud for depth measurements, as potholes will have negative depths relative to this plane.

3.4.1 RANSAC (Random Sample Consensus) Method

It is a robust method to fit a plane to the point cloud by iteratively selecting random subsets of points and fitting a plane to those points. The plane equation is given by:

$$ax + by + cz + d = 0$$

where (a, b, c) represents the normal vector of the plane, and d refers to the offset. RANSAC estimates the parameters of this plane by randomly selecting a set of 3 points, fitting the plane, and then determining how many points lie close to the plane (within a threshold distance). This process is then recurred for a fixed number of iterations, and the model having the most inliers (points that fit the plane) is selected as the ground plane.

3.4.2 Aligning the Plane with the XY-Plane

Once the ground plane has been detected, it's aligned with the XY-plane (horizontal plane) to ensure the correct orientation of point cloud. The plane normal n = (a, b, c) is used to compute the rotation matrix that will align the ground plane with the XY plane. This alignment ensures that the Z-axis of the point cloud corresponds to vertical direction, making depth measurements of potholes (in terms of Z-coordinates) accurate. The depth D of a pothole at a point (x, y, z) is calculated as:

$$D = z - z_{ground}$$

where $zground$ is the Z-coordinate of the ground plane at the corresponding x, y position.

3.4.3 Identifying Potholes Using Clustering

To separate the road surface (ground) from the points that represent potential potholes. The road surface is generally flat and horizontal, while the potholes represent deviations (depressions) in the road. To detect potholes, we first need to remove the road surface from the point cloud, leaving only those points which are above or below the surface, which could potentially represent potholes or other features (such as manholes or debris). Plane segmentation can be achieved using methods like RANSAC (Random Sample Consensus) to fit a plane model to the points that represent the ground. After this, the points that lie close to the estimated plane (within a specified distance

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

threshold) are considered part of the road surface. Points that deviate from the plane (i.e., points with a Z-value significantly different from the plane's Z-value) are considered to be part of other objects, including potential potholes. To segment the points from the road surface, a simple filter based on the Z-coordinate can be applied, where points with a Z-value difference greater than a certain threshold from the ground plane are retained as potential potholes.

3.5 DBSCAN Clustering

To group the remaining points (potential potholes) into clusters, with the assumption that potholes form distinct groups in the point cloud. Once the road surface is segmented out, we are left with the points that might represent potholes. These points are typically clustered into distinct groups in the 3D space. The DBSCAN, also known as the ‘Density-Based Spatial Clustering of Applications with Noise’ is an algorithm commonly used for this purpose because it can identify clusters having arbitrary shapes and is robust to noise (outliers).

3.5.1 How DBSCAN Works:

- **Core Points**: These points have less number of neighbours within a specified distance.
- **Border Points**: Points that do not have enough neighbours to be considered core points, but are within the neighbourhood of a core point.
- **Noise Points**: These Points are not core and also not the border points and are considered outliers.

3.5.2 DBSCAN Algorithm Steps:

1. For each point present in the point cloud, DBSCAN checks the number of neighbour points present within a predefined radius ϵ\epsilonϵ. If the number of points exceeds a specified threshold $minPts$, it becomes a core point.
2. The algorithm then expands the cluster by iteratively including points that are within the neighbourhood of the core points.
3. The result is a set of clusters, wherein each cluster represents a group of points that are close together spatially, possibly representing a pothole.

Formula: The Euclidean distance d between two $P1 =$

(x_1, y_1, z_1) and $P_2 = (x_2, y_2, z_2)$ is calculated as:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_1)^2)}$$

DBSCAN uses this distance measure to determine whether points belong to the same cluster based on the ϵ\epsilonϵ-radius and $minPts$.

3.5.3 Largest Cluster Selection

To choose the largest cluster from the DBSCAN results, which is assumed to represent the main pothole. After performing DBSCAN clustering, multiple clusters may be identified, including smaller clusters that may not represent actual potholes. The assumption is that the primary pothole will be the largest cluster (in terms of the number of points), as larger potholes will generally contain more points than smaller ones or noise. Once clustering is done, we can examine the size of each cluster and select the one with the most points. This cluster is then considered as the main pothole. The size of a cluster is simply the number of points it consists of. If $\boldsymbol{Ci}$ represents $\boldsymbol{i} - th$ cluster, then the size | $\boldsymbol{Ci}$ | is the number of points in that cluster:

$$|\, Ci \,| = Number\ of\ points\ in\ cluster$$

The largest cluster $\boldsymbol{C_{max}}$ is the one with the maximum size:

$$C_{max} = arg\ max_i \,|\, C\ i \,|$$

the primary pothole, which is the target for further analysis.

3.6 Depth Calculation

Reference Road Height: The road surface is assumed to be flat or have a known plane, typically the ground plane identified during the preprocessing step (using RANSAC). This reference height allows us to calculate the depth of any point present beneath the road surface.

Max Depth (Deepest Point): The deepest point in the pothole will give us an idea of how severe the pothole is. It is the point with the maximum negative difference in height from the reference road surface.

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

Average Depth (Mean Depth): This refers to the average height difference between the points in the pothole and the road surface. It gives a more general estimate of the overall depth of the pothole.

$$\boldsymbol{Depth_i = z_{road} - z_i}$$

Where:

- z_{road} is the Z-coordinate of the ground plane (the reference height),
- z_i is the Z-coordinate of a point in the pothole.
- Max Depth (Deepest Point):
  - $Max\ Depth = max(z_{road} - z_i)$

$$\forall\ Pi\ in\ the\ pothole$$

- Average Depth:

$$\boldsymbol{Average\ Depth = \frac{1}{n}\sum_{i=1}^{n}(z_{road} - z_i)}$$

Here n is the total number of points in the pothole.

3.7 Pothole Boundary Estimation

To estimate the outer boundary of the pothole in the point cloud. This helps in defining the area and size of the pothole.

- The **convex hull** is a geometric method that can be

  used to find the smallest convex polygon (or polyhedron in 3D) that encloses a set of points. For the pothole, this will represent the outer boundary of the points that are within the pothole.
- The convex hull algorithm works by finding the points present at the boundary of the pothole and enclosing them within the smallest convex shape. This gives us an accurate boundary for further measurements.

Formula: In 3D space, the convex hull is the smallest convex set that contains all the points in the dataset. If P is the set of points representing the pothole, the convex hull $ConvexHul(P)$ can be calculated as the smallest convex set that encloses all the points in P. This hull can be calculated using algorithms such as **Quick Hull** or **Gift Wrapping**. Once the convex hull is computed, the boundary of the pothole is given by the vertices and edges of the hull.

3.8 Size Estimation

To compute the size of the pothole, including its **length**, **width**, **area**, and **volume**.

- **Length and Width using Axis-Aligned Bounding Boxes (AABB):** The pothole's size is commonly represented using an **axis-aligned bounding box (AABB)**. An AABB is the smallest rectangle (or cuboid in 3D) that contains the entire set of points, with sides parallel to the coordinate axes. This allows us to compute the length and also the width of pothole.
- **Length:** It represents the distance present between the maximum and minimum x-coordinates of the bounding box.
- **Width:** The width of the pothole is the distance present between the maximum and minimum y-coordinates of the bounding box.

Formula: Let x_{mix}, x_{max} be the minimum and maximum x-coordinates of the bounding box, and y_{min}, y_{max} be the minimum and maximum y-coordinates.

- **Length (along x-axis):**

$$Length = x_{max} - x_{min}$$

- **Width (along y-axis):**

$$Width = ymax - ymin$$

- **Area using Convex Hull:**

  The **area** of the pothole can be calculated as the area of the convex hull. The convex hull gives a boundary around the pothole, and area of the shape formed by the hull can be computed using geometric formulas for polygons. For a 2D convex hull (projecting the points onto the $xy - plane$), the area A is calculated using:

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

$$\frac{1}{2}\sum_{i=1}^{n}(x_i y_{i+1} - x_i + 1\, y_i) + (x_n y_1 - x_1\, y_n)$$

where $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ are the coordinates of the vertices of the convex hull in counter clockwise order.

3.9 Volume Calculation

The pothole's volume can be computed as the volume between the convex hull and the road surface. This volume represents how much material is displaced due to the pothole. The volume is calculated by integrating the depth of the pothole over the area covered by the convex hull. Mathematically:

$$\boldsymbol{V} = \int_{convexhull} (z_{road} - z(x, y))\, \boldsymbol{dxdy}$$

where $z\,(x, y)$ refers to the height of pothole surface at a given point (x, y), and $zroad$ is the reference height of the road surface.

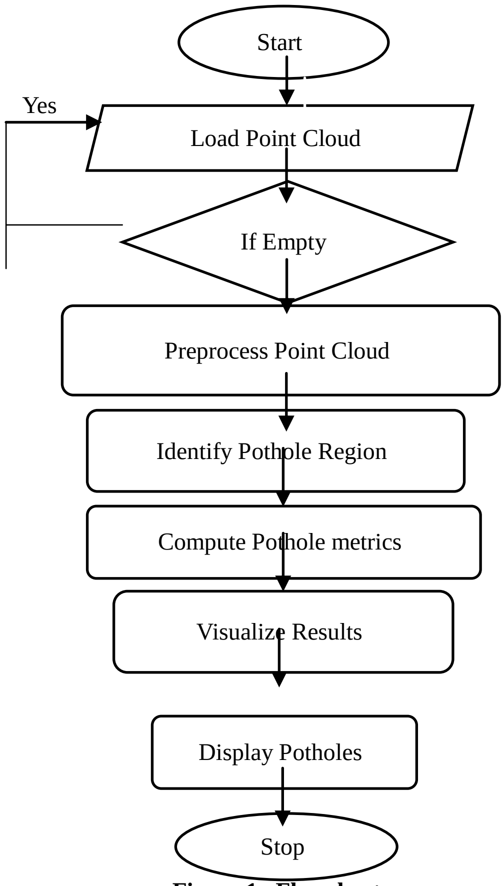

Figure 1. Flowchart

4. RESULTS AND DISCUSSIONS

Once the pothole dimensions have been calculated (depth, size, area, and volume), the next step is to visualize the results to aid in analysis and interpretation. Visualization helps in understanding the pothole's 3D structure, as well as its size, depth, and other metrics. The following methods describe how to visualize the point cloud and calculated results :

4.1 3-D Visualization using Open3D

To display the point cloud data along with the detected pothole and road surface in 3D, allowing an intuitive view of the pothole's shape and position relative to the road.

Point Cloud Representation: We can visualize the road and pothole by combining the points representing the road surface with those representing the pothole. Different colours can be used to differentiate between the two.

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

Figure 2. RANSAC Model of 3d Point Cloud of Pothole Lidar

Figure 3. RANSAC Model of 3d Point Cloud of Pothole Lidar

Pothole and Road Surface Display: After preprocessing the point cloud, we can render the entire scene (road + pothole) in 3D. This helps to understand the geometry of the pothole and its relationship to the road surface.

2D Visualization using Matplotlib

To display the pothole and road surface from a top-down and XYZ Plane view, along with information about the pothole's depth, size, area, and volume.

- **Top-Down view :** By projecting the 3D point cloud on to the XY-plane, a 2-D visualisation gets created showing a bird's-eye view of pothole and road surface, allowing a clear view of boundary and size.
- **Depth, Size, Area and Volume Display :** These key metrics can be displayed on the 2D-plot, either wihtin the plot or as annotations.

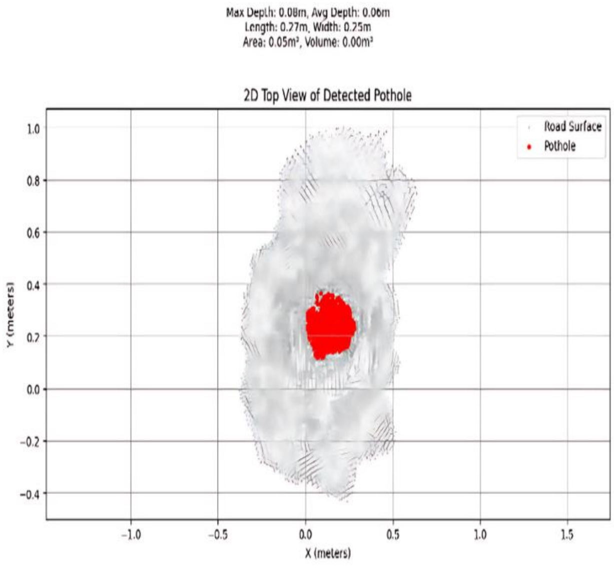

Figure 4. 2d Matplotlib Representation

ISSN: 2350-1278
Peer Reviewed & Refereed Journal (IF: 7.9)
Journal Website www.nrjitis.in

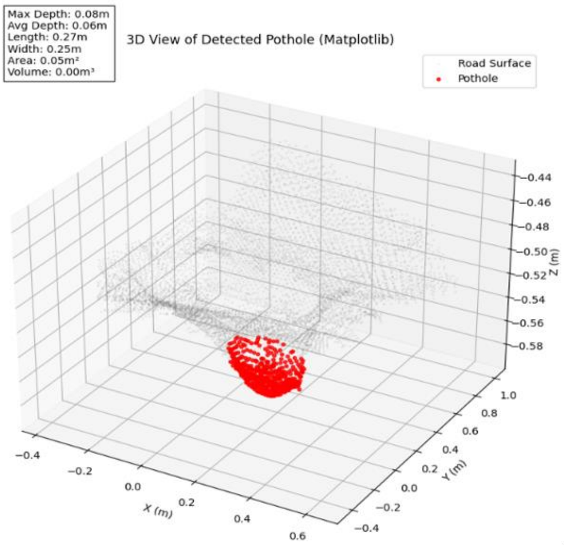

Figure 5. 3-D Matplotlib Representation

Three metrics are used in this project to compute the performance and accuracy of the pothole detection and depth estimation model across different files: . The following metrics provide an understanding of the model's capability to accurately identify and retrieve appropriate instances :

- **Precision (%) :** It is defined as the ratio of corrected predicted positive instances (True Positives) to the total predicted positive instances (True Positives + False Positives). It shows the capability of the model to ignore false positives.
- **Recall (%) :** It is known as the ratio of correctly predicted positive instances to all actual positive instances (True Positives + False Negatives). It shows the ability of model to to retrieve all appropriate illustrations.
- **F1 Score (%) :** It is defined as the harmonic mean of Precision

and Recall. It provides a balanced evaluation, especially in the case of uneven class distributions or when there is a desire to make balance between precision and recall.

The calculated values of **Precision**, **Recall**, and **F1 Score** for each test file is summarized below :

Table 1. Depth Estimation Result Table in %

File	Precision (%)	Recall (%)	F1 Score (%)
1.las	74	61	78
2.las	68	54	64
3.las	82	30	57
4.las	63	49	70
5.las	70	59	68
6.las	58	75	72
7.las	64	68	60
8.las	88	35	60
9.las	77	58	73
10.las	66	45	85

V. ACKNOWLEDGMENTS

Our thanks to the experts who have played an important role in contributing to the development of the template. Their efforts have aided in facilitating the process of submission and ensuring an uniform standard for all the authors. Also, our grateful thanks to Dr. Sandeep Kumar, Associate Professor, Sharda University for his continuous support and guidance which was pivotal in the successful completion of this research.

REFERENCES

1. N. Wang, J. Dong, H. Fang, B. Li, K. Zhai, D. Ma, Y. Shen, and H. Hu, "3d reconstruction and segmentation system for pavement potholes based on improved structure-from-motion (sfm) and deep learning," *Construction and Building Materials*, vol. 398, p. 132499, 2023.

2. M. Cheng, L. Hui, J. Xie, and J. Yang, "Sspc-net: Semi- supervised semantic 3d point cloud segmentation network," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 2, 2021, pp. 1140– 1147.

3. R. Fan, U. Ozgunalp, B. Hosking, M. Liu, and I. Pitas, "Pothole detection based on disparity transformation and road surface modeling," *IEEE Transactions on Image Processing*, vol. 29, pp. 897–908, 2019.

4. J. Xu, R. Zhang, J. Dou, Y. Zhu, J. Sun, and S. Pu, "Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 024–16 033.

5. S. Fan, Q. Dong, F. Zhu, Y. Lv, P. Ye, and F.-Y. Wang, "SCF-net: Learning spatial contextual features for large-scale point cloud segmentation," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 14499–14508.

6. Akagic, E. Buza, and S. Omanovic, "Pothole detection: An efficient vision based method using rgb color space image segmentation," in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2017, pp. 1104–1109.

7. Choy, J. Park, and V. Koltun, "Fully Convolutional Geometric Features," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 8957–8965.

8. R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 77–85.

9. M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," *Computational Visual Media*, vol. 7, pp. 187 199, 2021.

10. X. Lai, J. Liu, L. Jiang, L. Wang, H. Zhao, S. Liu, X. Qi, and J. Jia, "Stratified transformer for 3d point cloud segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8500–8509.

11. P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

12. Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large- scale point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 108–11 117.

13. Thibaud Ehret, *Monocular Depth Estimation: a Review of the 2022 State of the Art*, Image Processing On Line, 13 (2023),pp.38–56. https://doi.org/10.5201/ipol.2023.459.

14. Jiaxiu Dong, Niannian Wang, Hongyuan Fang, Hongfan g Lu, Duo Ma, Haobang Hu "Automatic augmentation and segmentation system for three-dimensional point cloud of pavement potholes by fusion convolution and transformer" , published in IEEE Advanced Engineering Informatics, Volume 60, April 2024, 102378. https://doi.org/10.1016/j.aei.2024.102378.

15. R. Ravi, A. Habib, and D. Bullock, "Pothole Mapping and Patching Quantity Estimates using LiDAR-Based Mobile Mapping Systems," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2674, no. 9, pp. 124–134, Jul. 2020, doi: https://doi.org/10.1177/0361198120927006.

16. LIAO Fei-Qin, MA Rong-Gui, WANG Duo, CHEN Xin- Long, "Highway Road Pothole Extraction Method Based on RANSAC", in IEEE Computer Systems and Applications,Volume 31, Issue 5, 2022 >230-237. DOI:10.15888/j.cnki.csa.008494.

17. L. Meng'Ao, M. Dongxue, G. Songyuan, and L. Shufen, "Research and improvement of DBSCAN cluster algorithm," in *2015 7th International Conference on Information Technology in Medicine and Education (ITME)*, 2015, pp. 537–540.

18. D. Tabernik, M. Šuc, and D. Skočaj, "Automated detection and segmentation of cracks in concrete surfaces using joined segmentation and classification deep neural network," *Constr. Build. Mater.*, vol. 408, no. 133582, p. 133582, 2023.

19. P. Thanh An, N. Minh Phuong, D. Le Quang, and L. Hong Trang, "A modified pothole detection approach to capture their width," *Phys. Scr.*, vol. 99, no. 4, p. 045007, 2024.

20. V. Prasad, S. Kumari, and P. VeereshCMandiSuryatejI, "Pothole detection using LiDAR," 2021.