applied system innovation

MDPI

Supplementary Materials: Determination of Point-to-Point 3D Routing Algorithm Using LiDAR Data for Noise Prediction

Shruti Bharadwaj [1,†], Rakesh Dubey [1,†], MD Iltaf Zafar [1,†], Rashid Faridi [2], Debashish Jena [3] and Susham Biswas [1,*]

[1] Department of Computer Science and Engineering, Rajiv Gandhi Institute of Petroleum Technology, Amethi 229304, India; pgi17001@rgipt.ac.in (S.B.); pgi19001@rgipt.ac.in (R.D.); pgi15001@rgipt.ac.in (M.I.Z.)
[2] Department of Geography, Aligarh Muslim University, Aligarh-202001, India; rashid.faridi@gmail.com
[3] Department of Management Studies, Rajiv Gandhi Institute of Petroleum Technology, Amethi 229304, India; djena@rgipt.ac.in
* Correspondence: susham@rgipt.ac.in; Tel.: +91-9153-5270-4626
† These authors contributed equally to this work.

Supplementary Section S1

Literature Review

Route determination in the outdoor environment is used to perform a major function in various applications. Viewshed analysis, solar shadow determination, the transmission of resources from one point to another, noise propagation, etc. are some examples of these applications [1]. Apart from these examples, it has also been found that 3D route determination also plays an important role in modeling noise propagation.

GIS (Geographical Information system) is a sophisticated collection of tools for storing and retrieving spatial data from the actual world, as well as modifying and displaying it [1]. To trace the data alteration at each stage of the process, a GIS data management system might be employed. Alteration in input data, data simplification, interpolation, calculation, etc. that could affect the accuracy of outputs are all examples of data manipulation. There are a variety of noise prediction methods now available. They require a platform on which to construct a model from several components. Noise Estimation is a combination of Noise sources, terrain data, and Noise model is shown in Figure S1.

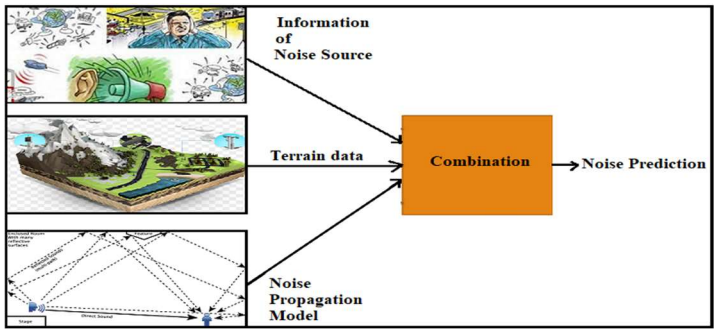

Figure S1. Noise prediction.

Roadways, railways, etc. can be examples of linear noise sources, while industry sites, airports, etc. can be examples of area noise sources. Screens, buildings, embankments (with or without screens), bridges, roadways, gullies, open spaces, different forms of grounds, vegetation, atmospheric conditions, etc. must all be included in the noise models. Various noise prediction techniques employed various concepts to define these noise values, one of which is Lima, which used the Line segmentation principle (see Figure S2) to define these parameters. It is accomplished by using the projection approach;

various portions are created from the source to the destination solely to examine the route variation for obstacle screening using "rubber-band logic" [2].

In the case of Lima modeling, worst-case analysis determines the side deviation from the barrier. This analysis is based on rotating the plane of propagation which means the route from which maximum noise is reached at the destination's point. Another widely used noise prediction model is Nordic prediction [3]. The prime consideration of this model is to select the building parameters. The model calculates the noise level with different corrections such as Distance correction, Angle of view correction, Screen and screen ground correction, and Thick Barrier correction. In this barrier correction principle, the authors first tried to select the next road segment with source and destination point information. Following this, they constructed the angle of view triangle as shown in the landmark and the road stretch is depicted in Figure S3.

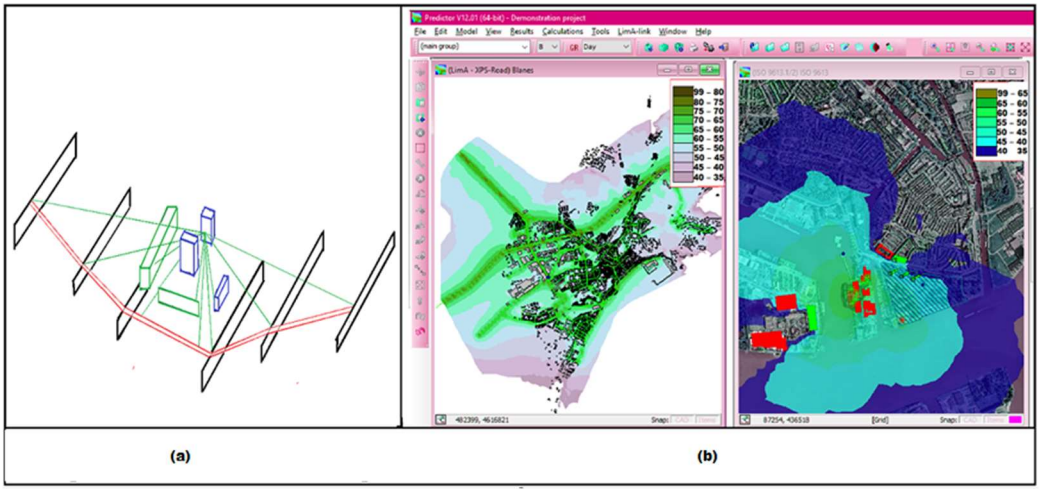

Figure S2. The two figures in the Lima Modeling: (**a**) Principle of the line segmentation using projection; (**b**) Demo of Lima technique to create model and noise map (courtesy; [2]).

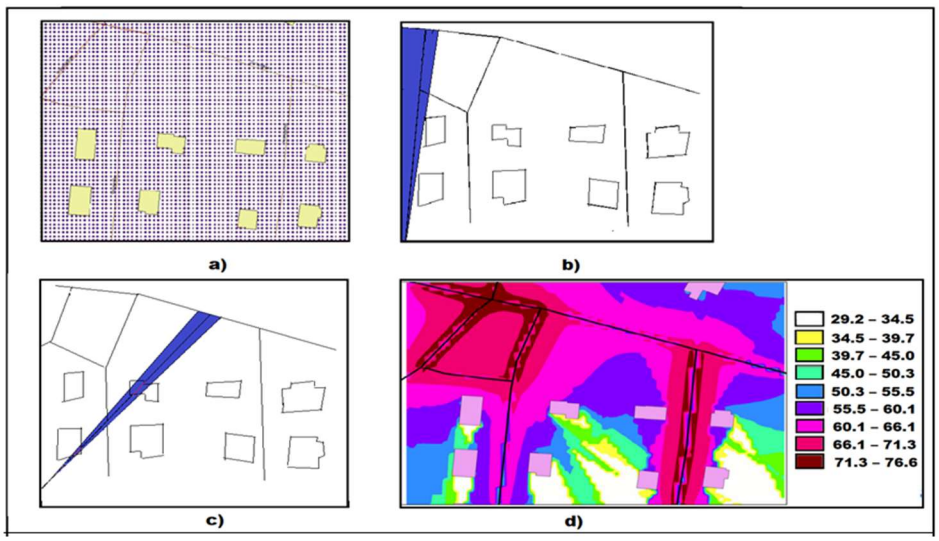

Figure S3. Three figures show the principle working of Nordic Noise propagation model to calculate: (**a**) Generated receiver for Noise prediction; (**b**) non-barrier angle and its bisector; (**c**) Barrier angle and its bisector; (**d**) Noise map of an area (courtesy; [3]).

Nowadays the most systematic analysis of noise levels is for the road traffic noise. The most used Noise propagation model is SoundPlan, which calculates the amount of noise generated by roads, trains, and industry. The model includes a separate toolbox with GIS-style data entry and manipulation devices. SoundPlan works on the idea that the user

must import the attributes by setting heights, distances, and other parameters. SoundPlan assumes that there are two ways to reach the destination, i.e., (a) direct with no obstruction that involves determining the distance attenuation, and (b) bypassing the obstruction [4] Here, the noise is calculated by dividing the calculated area into a grid with specific steps (s), such as 5 m, 10 m, or a maximum of 30 m. After determining the noise levels at the grid points, isolines are used to connect the sites with equal sound levels, leading to the equal-sound lines depicted in Figure S4. The fundamental disadvantage of this noise prediction model is that it only works with a small area and only uses rough terrain data, resulting in a huge inaccuracy. To eliminate all of these flaws, a model that incorporates high-resolution 3D digital terrain information generated by laser survey must be defined.

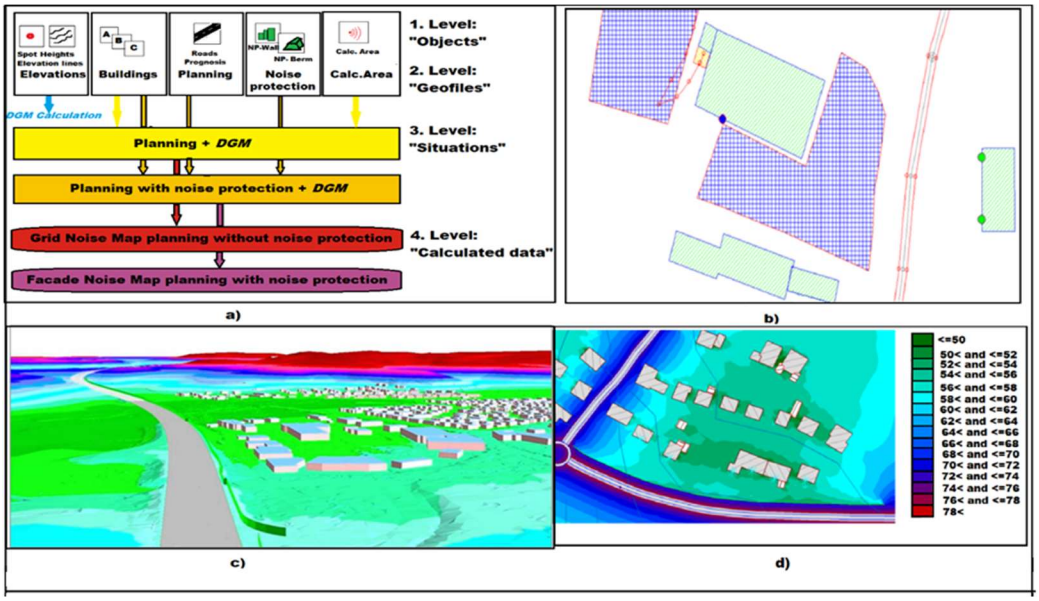

Figure S4. (**a**) Step to proceed SoundPlan Techniques; (**b**) Graphic screen (Geo-database); (**c**) Model creation in SoundPlan Modeling; (**d**) Isolines creation by means of direct and indirect routes (courtesy; [4]).

Determining the shortest route of noise propagation is challenging, the following techniques [5–7] are employed to discover the shortest routes. These algorithms' function using graph theory, with the root node as the source and the goal node as the destination. In this sense, a graph is made up of vertices (also known as nodes) connected by edges (also called links). The Greedy algorithm, which is a type of algorithm that finds the maximum value at every node in the graph to find the shortest route, is important in this area. Even if it finds a node with a very large value throughout the search, the greedy algorithm will take that route because it can't return. As a result, it takes some time to find an appropriate solution.

The Dijkstra algorithm, on the other hand, seeks to determine the quickest route from one location to another based on the least weight. This can only be used for ages that aren't negative. The Bellman ford algorithm is similar to Dijkstra's, with the exception that it can also be used for negative weight. However, it takes longer to search. There is another algorithm known as the A* algorithm, which performs the search for all existing nodes. This algorithm is the combination of the greedy and Dijkstra algorithms [8]. It takes more search time due to the complex network. Another algorithm is the Ant colony algorithm that finds the shortest route between the nest and source of food. For example, from the below-mentioned diagram A is the nest where ants reside and B is the source of food. Ants start wandering from A in the given 4 directions (C, D, E, F) and laying down pheromone [9]. Theoretical analysis of ant algorithm is more difficult because it necessitates a bigger number of repetitions. The Genetic algorithm is the most frequently used algorithm. It can

be used to find the quickest route based on the unique DNA of living creatures, which determines everyone's property. In comparison to other algorithms, its coding is quite simple. But it does not always provide the best solution. One most used algorithm for shortest route calculation is a Dijkstra algorithm, using A as the source node and B as the destination node. The route (A-D-I-B) is depicted in Figure S5 to reach Node B from several links examined by the algorithm. The other algorithm produces the same result, however, the A* algorithm differs in terms of space, time complexity, and heuristic values. These algorithms are not capable of finding routes in 3D complex scenarios.

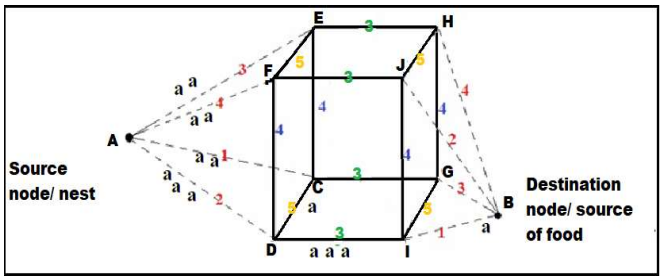

Figure S5. Diagram showing results of Dijkstra and other existing algorithms.

The problem of finding an optimal route has always been challenging in computer science, mathematics, and geoinformatics. Sometimes optimal route problem is considered as an optimization problem [10] attempted to provide solutions for both efficiency and accuracy. As we discussed earlier, many algorithms are designed to solve optimization problems [11]. A Genetic algorithm is an evolutionary optimization algorithm that has been used to fit the shortest route problem. Canali and Mittal proposed a solution for the optimal route problem using genetic algorithms [12,13]. The author adopts the concept of triangular Fuzzy numbers and then ranks fuzzy numbers to check the optimal solution. The proposed work of Win et al. discussed an application that works on finding the nearest location in case of fire with the help of a drone map [14]. Apart from these optimization algorithms, some algorithms [15] that are dependent on spatial analysis are used to bring meaning out of spatially referenced data, and the set of methods that provides the optimal route determination. The results from Spatial analysis techniques are majorly dependent on data quality [5,16] as well as the understanding that the GIS users have concerns regarding the method used [17]. A GIS user must have a great understanding of the methods used for preparing results. They must always prepare spatial data to make it suitable for use in the analysis. Otherwise, the results of a spatial analysis may be distorted. If the GIS user has a desired outcome from the analysis, they even use manipulations towards the desired goal. In Joshi' research, an algorithm is proposed for detecting fire alarm systems and an optimal solution to reach faster at the site [18].

Many road traffic noise analyses [19–22] also discussed the need of finding an optimal route for calculating the noise effect on people who reside near the road. Before extracting features from LiDAR 3D point cloud data, there is a need to understand which feature is helpful and which is not [23–25]. Thus, a proposal of a dedicated algorithm for accurately extracting road boundaries is necessary [26]. The proposed algorithm has been tested on "Xinda", an autonomous driving platform [27] and the authors have presented a survey that is based on Deep learning. Another discussed method is for 3D data understanding, object detection, shape classifications, and object segmentation [28] that presented a comparison of existing algorithms and approaches to turn LiDAR 3D point cloud into 2.5D urban scenes. [29] It discusses the detection of tree features and enhances the possibility of mapping [30,31] developed a neural network method for feature extraction and classification system based on the UGVs technique. Chen et al. discussed that the research based on LiDAR-based DTM generation is increasing exponentially [32].

Guoping et al. discussed a method that is based on LiDAR intensity information [33]. This method further represents the statistical methods for the classification of ground and non-ground points. After feature extraction from LiDAR point cloud data, DTM & DEM are required [34,35]. Many researchers discussed different ways to handle it. Sharam and Sulaiman proposed an algorithm for DTM generation based on filtering methods [36,37]. In [38], Meng et al. discussed the use of LiDAR data in DEM generation for coastal modeling, forest handling, etc. Bello et al. discussed the application of deep learning for classification, feature extraction, and DEM generation from 3D LiDAR data [39]. Priestnall et al. discussed several methods for surface feature extraction from a Digital Surface Model [40].

LiDAR-Light Detection and Ranging technique offer 3D point data of very high accuracy [41]. Nevertheless, because the data lacks precise topographical information, it is difficult to extract terrain parameters and then determine noise routes from the source to various destination sites. After determining the shortest or main routes, topographical parameters are calculated and included in a noise model with noise data for 3D noise level prediction.

Supplementary Section S2 (Methodology)

Section S2.1 (LiDAR Data acquisition)

1. Academic Area of the RGIPT campus (project area) is taken from google maps (Figure S6(a)) that consists of academic block 1 (AB1), academic block 2 (AB2), and administrative area. (Part of the RGIPT campus near the academic area consisting of buildings and grounds is used for LiDAR data generation and determination of various routes

2. For the area taken in step 1, LiDAR data is collected from the Terrestrial Laser scanner in the form of a 3D point cloud and consists of x, y, and z information as shown in Figure S6(b). FARO Terrestrial Laser Scanner makes the data acquisition. It is a high-speed 3D laser scanner for detailed point cloud measurement. The LASER scanner produces a 3D image of a complex environment using the laser technique. These images are gathering millions of 3D points. It uses the phase shift technology; constant waves of IR light of varying n lengths are projected outward. These waves have reflected in the scanner when in contact with an external object. The distance from the scanner to the object is determined by calculating phase shifts. X, Y, Z of each point are calculated by using encoders. The scan covers 360° x 300° Field of view. Need to set some parameters before the scan can begin.

3. First, choose a scan profile that specifies the location where the scan will be performed (indoor or outdoor). We have chosen to go outdoors for more than 20m. Second, choose the scan resolutions. Scan resolutions are 1/1, 1/2, 1/4, 1/5, 1/8, 1/10, 1/20, 1/32. It depends on the speed of the scan and the quality of the scan. If the speed of the scan is important, choose a lower value. If scan data quality is more important, choose a higher value. Here, data quality is important, so we chose 1/1 (4x), 4x means the laser hits 4 times at a single point and gives the average value. We start the scan by switching on the GPS to get georeferenced data. Then, take a small area of RGIPT with the help of placing Targets (spherical in shape). These targets will help in stitching the data of the next scan to the previous scan. So, by choosing areas one by one and placing targets in each area, the scanning for campus RGIPT is done. After the almost 60 scans on campus, we collected millions of 3D point cloud data of the project area of RGIPT. This data has a point accuracy of up to ±3cm.

Section S2.2 (Building Edges and corner extraction)

1. From step 2 we got the LiDAR plot of an area, for that plot digital elevation and digital surface model are created as shown in Figure S6(c). Basically, after the creation of two, this step gives the output of the subtraction of the BEM from the DSM to get elevated data of the RGIPT area. Figure S6(d) shows elevated points of the building of the Academic area. This is done only for extracting the edges and corners of the building in that area.

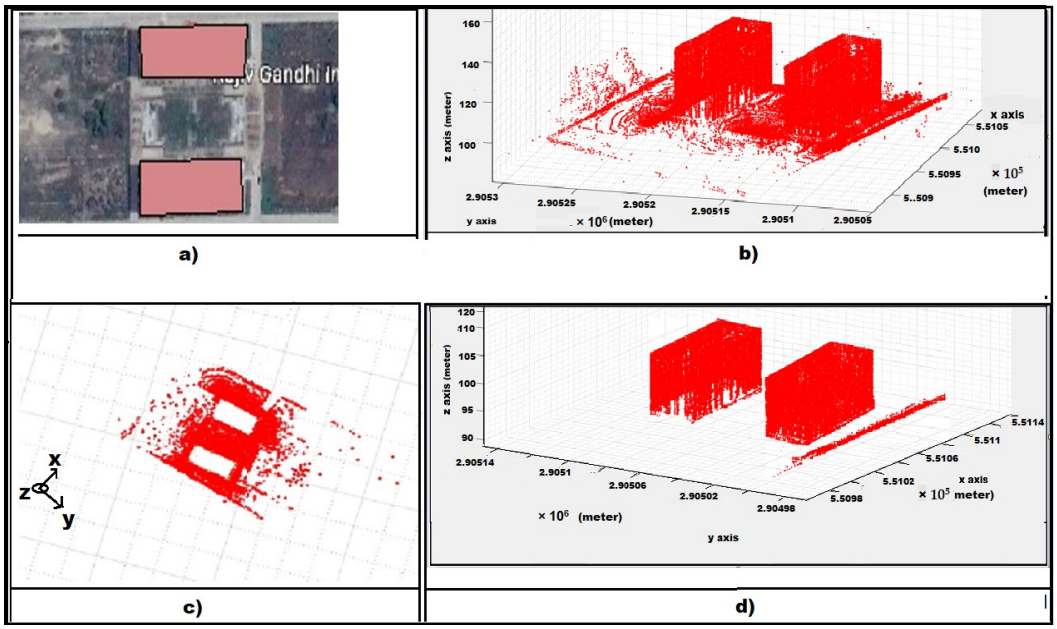

Figure S6. (**a**) Shows the area of RGIPT from google map; (**b**) The plot shows LiDAR point data for the area in step 1; (**c**) Extracted Ground points from 3D point cloud data. (**d**) Elevation data after Subtraction of BEM from DSM where building of RGIPT campus.

2. Partition of an area shown in (Figure S7 (a)), in an equal number of rows (i) and column (j). Then start moving in cell (i, j) with incrementing (i) row and (j) column. After each incrementation, check whether it contains a point or not. If the cell contains any point, assign that point to the building naming buiding1, as shown in Figure S7(b). A threshold is set in (m) for differentiating one building point from another building. This differentiation is done by checking the neighboring cells of that cell containing the point. If a neighboring cell contains the same elevated value, assign that point to the same building. Otherwise, assign it to the new building. This procedure will repeat till the end of (i, j). An array is formed for different buildings and which contains the respective point [36].

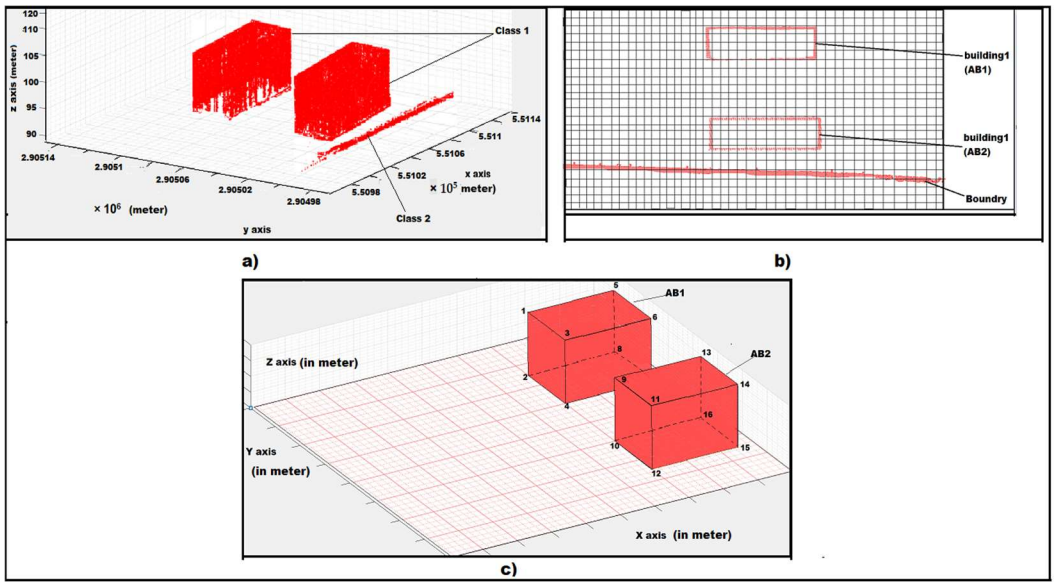

Figure S7. (**a**) Non-Ground points are now classified for extraction of building coordinates, for this class 1 is mentioned for Regular shaped and class 2 is for non-regular shape; (**b**) Planimetric view of non-ground points after classification in different classes, the partition is done in order to find out the building edges and corners; (**c**) 1,2,3,4,5,6,7,8 and 9,10,11,12,13,14,15,16 are the building corners of AB1 & AB2 extracted from the algorithm.

3. From the array of different buildings, Corner points for each building are determined (Figure S7(c)) and stored separately in an array by using the minimum and maximum criteria of x and y. For building 1, x is a group of *(x1, x2, x3, x4, x5, …., x10)*, and y is a group of *(y1, y2, y3, y4, y5, …., y10)* which comes under Building 1. Similarly for building 2, x is a group of *(x11, x12, x13, …., x20)* and y is a group of *(y11, y12, y13, …., y20)* which comes under Building 2. Corner Points of each building individually are calculated by *(min x, min y), (min x, max y), (max x, min y),* and *(max x, max y)* as shown in Figure S7(c).

$$Building\ 1 = [(x1, y1), (x2, y2), (x3, y3) \ldots\ldots (x10, y10)]$$
$$Building\ 2 = [(x11, y11), (x12, y12), (x13, y13) \ldots\ldots (x20, y20)]$$

For N, building corners are -0Pb=((N-1) ×10) +1

$$Building\ N = [(xb, yb), (xb + 1, yb + 1), (xb + 2, yb + 2) \ldots\ldots (xb + 9, yb + 9)]$$

Section S2.3 (Indirect Route determination)

2.3.1. Route over the Top

The results for the complex buildings scenario are shown in Figures S8(a) and S8(b). The route over the top for the project area of RGIPT is shown in Figure S8(c) for an example purpose.

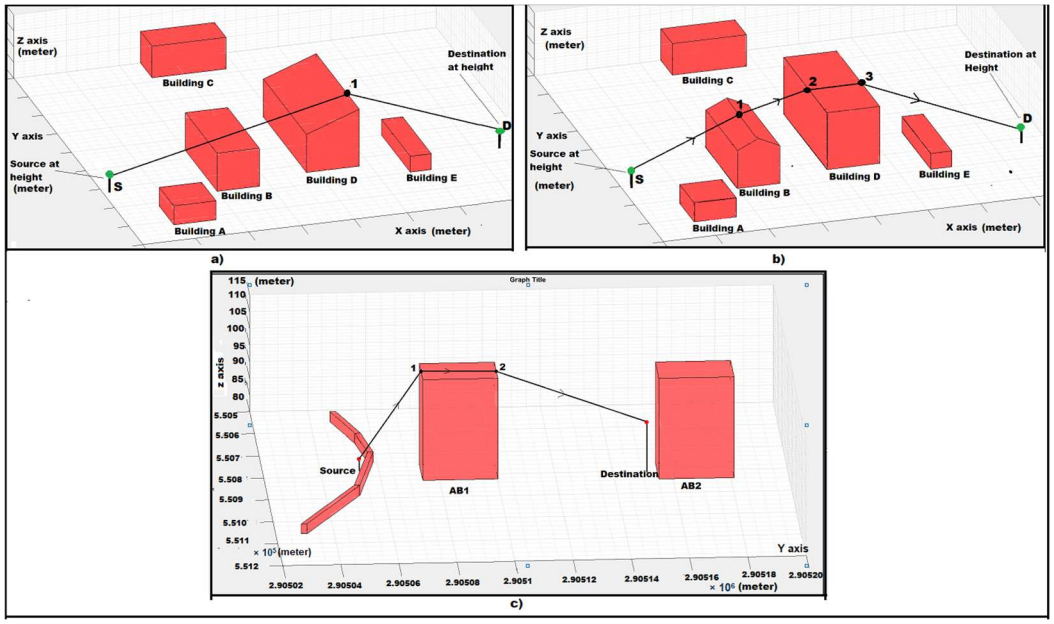

Figure S8. (**a**) The route over the top of the building for setup contains slanted building is S-1-D. (Values in x and y-axis are in meter); (**b**) The route over the top of the building for the hut-shaped building is S-1-2-3-D (Values in x and y-axis are in meter); (**c**) Source-1-2-Destination is the route over the top for the academic area of RGIPT between the pair of Source-Destination.

2.3.2. Route around the sides

Routes for the project area of an RGIPT campus for both right and left sides are shown in Figures S9(a) and S9(b).

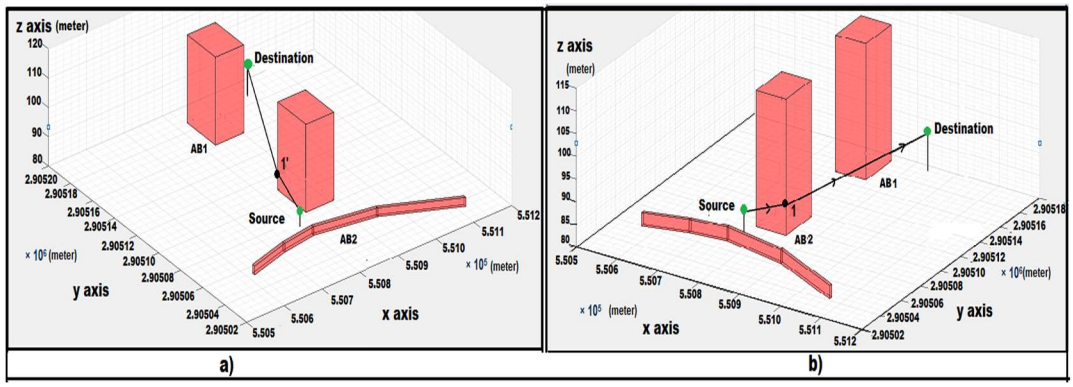

Figure S9. (**a**) Source-1-Destination is the left-side route around the sides of the building for an academic area of the RGIPT campus. It consists of AB1 & AB2. Source point and destination point are taken manually to discuss the case for the project area; (**b**) Source-1-Destination is the right-side route around the sides of the building for an academic area of the RGIPT campus. It consists of AB1 & AB2. Source point and destination point are taken manually to discuss the case for the project area.

2.3.3. Reflection Route

Initially, for the Reflection route, first consider the source and destination points over the 3D environment and the area. Create a line between source S1 (xs, ys, zs) and the destination point D1(xd, yd, zd) where zs=0 and zd=0. Now calculate the distance between the two-point S and D by using. According to Snell's law for reflection, the angle made by the incident ray is equal to the reflected ray that is Sin α (i) = Sin β (r). As shown in Figure S10, the incident and reflected angle that is α and β [39].

$$Distance = \sqrt[2]{((x_d - x_s)^2 + (y_d - y_s)^2 + (z_d - z_s)^2)}$$

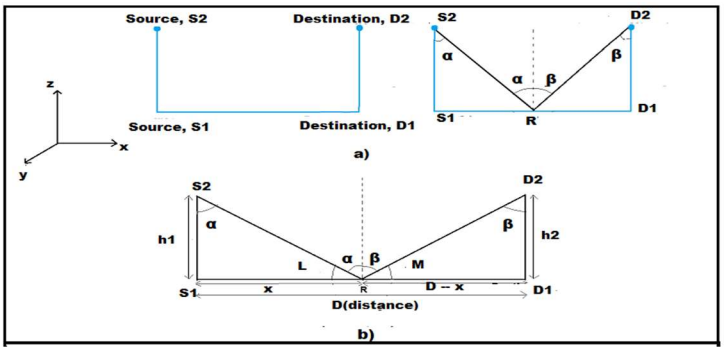

Figure S10. Source(S1) is the source point at ground and Source(S2) at some height (which can vary) and Destination (D1) is the destination point at ground and Destination(D2) at some height (which can vary). Finding the reflection point "R" for a general case where source at "h1" height and destination at "h2" height is taken with incident angle α and reflected angle β.

a. Ground Reflection

It is dependent on the type of ground present between a pair of source and destination.

Case 1 (Uniform Ground): When there is a uniform plane between source and destination. As shown in Figure 14, the angle α and β will be equal, when $\alpha = \beta$ then angle $L=M$

$$L = \frac{\pi}{2} - \alpha$$

$$M = \frac{\pi}{2} - \beta$$

$$\tan L = \left(\frac{h1}{x}\right)$$

$$\tan M = \left(\frac{h2}{D - x}\right)$$

From the above relationship, "x" that is a distance of a point of Reflection from the source, is calculated, which in addition to the source point gives the coordinate of the reflection point. To determine the classification of a Project area is done to determine the type of ground for Reflection. In Reflection, ground type plays a significant role. Classification is shown in Figure 15.

Figure S11. Image classification is done to determine the ground type of an area for reflection.

Case 2 (Non-Uniform Ground): When there is a non-uniform plane between source and destination. As shown in Figure S11(a) non-uniform plane exists between source and

destination. A plane *S1-S2-D2-D1* is taken, the YZ plane perpendicular to the X. From *S1* to *D1* number of points are taken from point cloud LiDAR data. Normal is drawn to the plane that exists between *S1* and the point for each point. For every point difference, the angle of incidence and Reflection is calculated along with route length. The point with a minimum difference in angle and minimum route difference is mentioned as reflection point "R", where *S1(1,12,0), S2(1,12,2), D1(15,12,0), D2(15,12,4)* are coordinate (x, y, z).

Firstly, point 1 is taken, make a line between the *S1* and point 1. It is required to extend the *Line S1-1* to intersect the *D1-D2 line* at "T". Draw a normal with point 1 perpendicular to line S1-T, as shown in Figure S12(b). Join S2-1 and check whether this line intersects any points taken between *S1* and *D1*. If it intersects, skip this point else calculate the angle of incidence. Similarly, for the reflected one, if the *1-D2* line intersects any point that is taken between *S1* and *D1*, then skip that point else calculate the angle of Reflection. Further, it is required to Calculate the difference between both angles and route length.

1. Repeating the above (1) for point 12 as shown in Figure S12(c). Calculate the angle difference along with route length. One after one, calculate these two parameters for all points from 1 to 29. And check whether, at which point, a difference of both angle and route length is minimum.
2. Reflection point (R) is calculated for the project area of RGIPT for a pair of source-destination as shown in Figure S12(d).

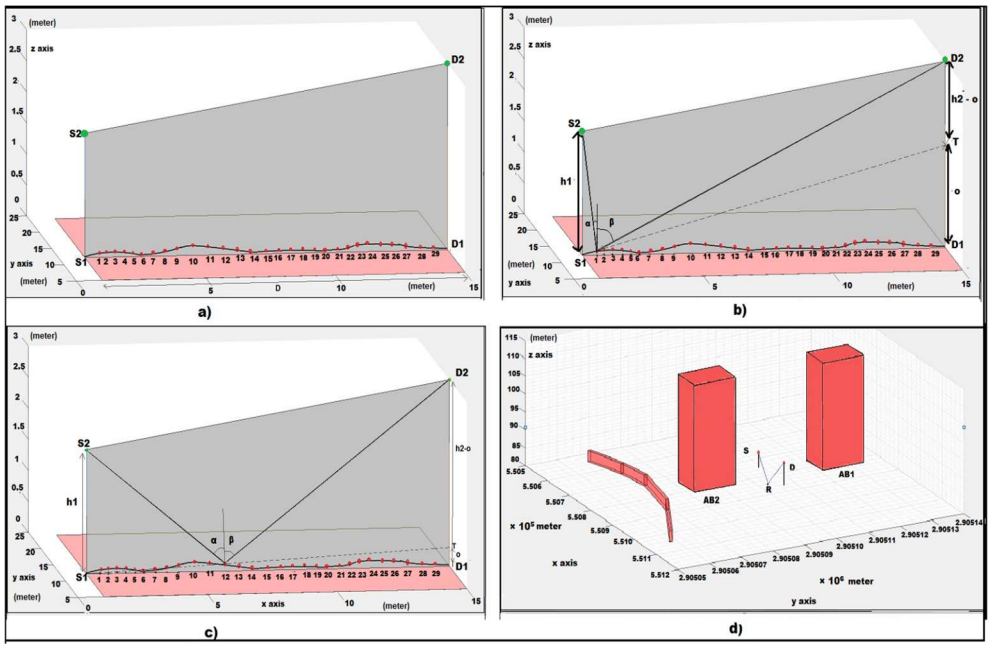

Figure S12. (**a**) Here, the plane between the source and destination is non-uniform. There exist number of point data defines the non-uniformly. Point 1 to Point 29 defines random points on the source–destination plane. "h1" is height of source from the plane at S1 and "h2" is height of destination from the plane at D1. (**b**) Point number "1" is taken to check the difference between angle incidence α and angle of reflection β. Where "T" defines the intersection point of line between the S1 and the point "1" with Line D1-D2. (**c**) Here, point number "12" is taken and calculate the "T" point which is an intersection of line S1-point 12 and Line D1-D2. After that a normal is drawn at point number 12 per perpendicular to S1-T line to calculate difference between α and β. (**d**) "S" is source and "D" is destination R is reflection point found by applying law of reflection between S and D.

3. After finding out the most appropriate point of Reflection (R), a buffer region is taken around the point of Reflection between source and destination, as shown in Figure S13(a). No point exists in buffer at different planes for that buffer. A

triangulation is formed for points in the buffer, and the centroid of each triangle is calculated. Every centroid pair of incidents and reflected angle is calculated using previous steps in case 2. Now whichever point contains a small angle difference between reflected and incident angle and the shortest route length will consider as the Final reflection point.

4. New Reflection point (R′) is calculated as shown in Figure S13(d).
5. New Reflection point (R′) is calculated for the project area of RGIPT for a pair of source-destination as shown in Figure S13(e).

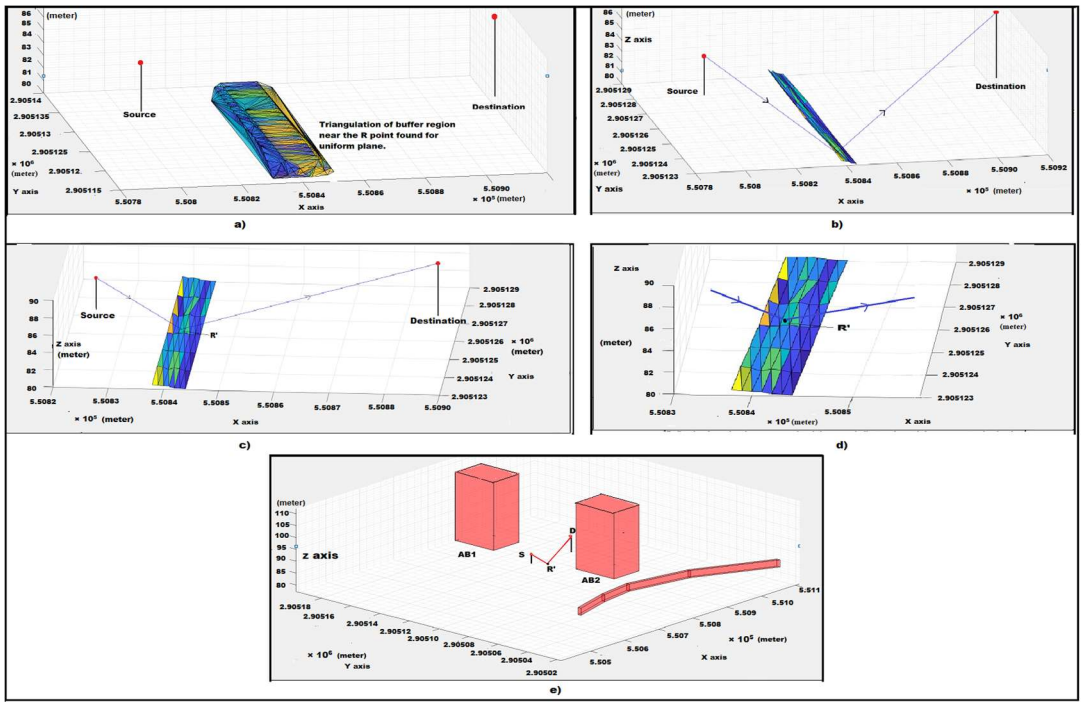

Figure S13. (**a**) Triangulation of buffer region near the R point found for uniform plane. This process is done to find the accurate reflection on ground which is affected due to non-uniform plane; (**b**) Checking every point in buffer to check at which point angle difference between incident angle and reflected angle is least and the route difference is least; (**c**) In this figure Reflection point (R′) is found after checking all points in buffer region near the reflection point first find between source and destination; (**d**) Reflection from the plane on ground and the arrow indicates the path from source to destination; (**e**) "S" is source and "D" is destination, R′ is new reflection point received.

b. Wall Reflection

Calculation of reflection point on the building wall, here taking a building with a source at height *h1* and destination at height *h2* as shown in Figure S14(a). A Cutting plane XY (ijkl) is made perpendicular to the Z-axis to check the intersection of a plane with the building.

1. Length of a k-l line intersecting the building at m and n is equal to the distance between *S2-D2*, as shown in Figure S14(b). Several points are taken on the line k-l. There are two angles: the incident angle and the reflected. For each point between *k* and *l*, the difference between the two angles and route length is calculated. The point where both difference and route length are minima will be the point of reflection on the wall.
2. For each Point between *k* and *l* (from point 1 to point 15), a normal is drawn perpendicular to the *k-l*, and one by one difference in angle and route length for each point is calculated. As shown in Figure S14(c), (point 10) is concluded as the point of reflection on the wall.
3. The reflection point on the wall is calculated for the Project area of the RGIPT campus, as shown in Figure S14(d).

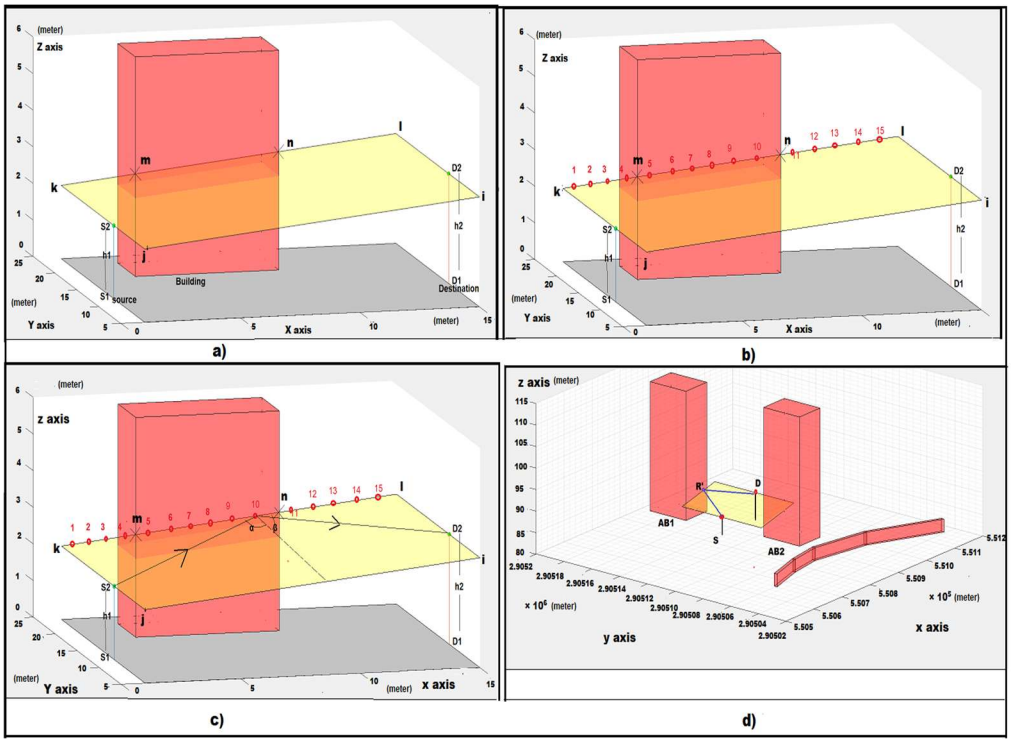

Figure S14. (**a**) Source and destination at height h1 and h2 is given. "ijkl" is the cutting plane (XY) which is perpendicular to Z (between source and destination). Plane "ijkl" cuts the building at "m" & "n"; (**b**) Length of k to l is equal to the line length between S2 and D2. Number of points from 1 to 15 are taken between k-l and check the difference if incidence and reflected angle for every point. Difference value least with minimum route length will be the point of reflection on wall; (**c**) Point of reflection where difference between incident angle "α" and reflected angle "β" is minimum. Point that has the minimum route length is the point number 10; (d) "S" is source and "D" is destination and R' is the new reflection point found on wall of building.

Section S3 (Results and Discussions)

Section S3.1 (Error Determination for instantaneous noise)

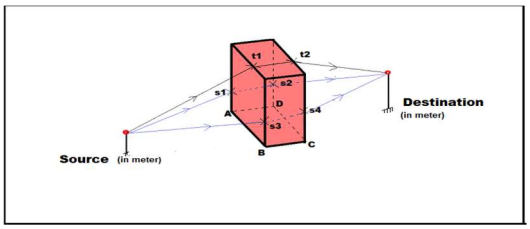

Figure S15. Source at S (4,4,5,4) and destination at D (11,4,5,8), building has A, B, C, D ground corner points of building and height of building is 15m. Here t1, t2 are point of intersection on building which forms top way path. Similarly, s1, s2, s3, s4 are points on building forming path around the sides.

The building corner is estimated by estimating the least square for LiDAR data of the building as shown in Figure S16(a). There can be many least square estimations to find intersection points, as shown in Figure S16(b). Due to the accuracy in LiDAR data during the acquisition of 3D points being ±3cm, now there is a maximum error of ±6cm between any two-point when both points are wrongly taken, as shown in Figure S16(c). Now here for the top way route (Source-t1-t2-Destination), all the point has an error of ±3cm. The maximum error between Source and t1 is ±7.5cm considering building edge error. Similarly, ±9cm between t1 and t2, ±7.5 cm t2 and Destination. The following is needed to calculate error propagation to calculate the total error E.

$$E = \sqrt[2]{E1^2 + E2^2 + E^3}$$

Where E1, E2, E3 are maximum error between Source-t1, t1-t2, and t2-Destination.

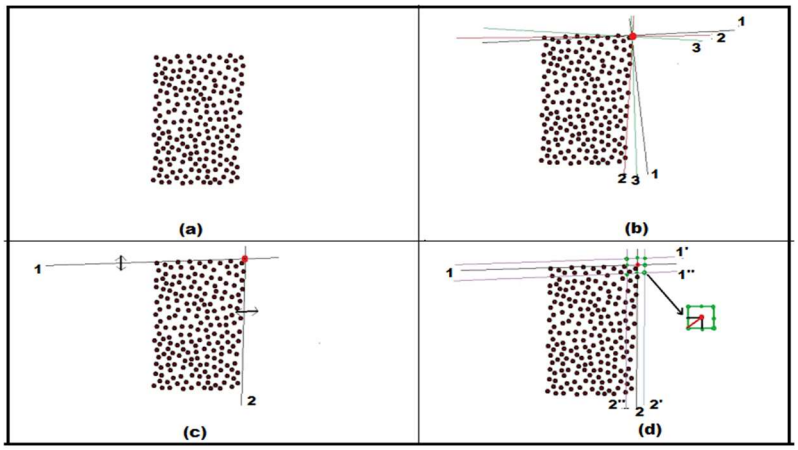

Figure S16. (**a**) Building LiDAR points with Accuracy (±3 cm); (**b**) Point accuracy is (±3 cm) means in all directions the maximum error is 6 cm, From the least square method 1, 2, 3 pairs of lines intersect to find a building corner; (**c**) With (±3 cm) Accuracy at each point, there may be a shift of 3 cm in both directions while calculating the building corner; (**d**) For the Building corner points due to LiDAR point accuracy is (±4.5 cm).

Due to this error propagates, the author will define the effect on the noise level at the destination at 3 different frequencies of 50Hz, 250Hz, and 1000Hz. Error propagates in between each pair of points. These errors can change the route length, direct route, indirect route, and noise level at the destination point. In this case table, 1 is shown below for a low frequency that is 50 Hz for that noise level without error, and the noise level with maximum error is calculated. Where D.A is distance attenuation and B. A is barrier attenuation in Table S1.

Table S1. Noise level calculation with error or without error at destination location.

Frequency	50 Hz									Total (dB)
	DR	IDR (1)	IDR (2)	IDR (3)	IDR (4)	IDR (5)	IDR (6)	IDR (7)	IDR (8)	
Route length	19.42	58.87	24.60	23.15	66.94	74.70	82.77	38.87	38.37	
Error	0.06	4.14	0.13	0.13	4.14	4.14	4.15	0.15	0.15	
Route Length (Error)	19.47	63.00	24.73	23.28	71.08	78.84	86.92	39.02	38.51	
Route difference (Error)	0	43.52	5.25	3.80	51.61	59.37	67.44	19.54	19.09	
Route difference	0	39.45	5.18	3.73	47.52	55.29	63.36	19.45	18.95	
D. A (Error)	36.80	46.99	38.86	38.34	48.03	48.96	49.78	42.83	42.71	
D. A	36.76	46.41	38.82	38.29	47.51	48.47	49.36	42.79	42.68	
B. A (Error)	0	18.68	18.85	18.97	19.06	18.10	18.80	18.06	18.81	
B. A	0	18.56	18.78	18.92	19.02	17.35	18.72	17.12	18.73	
Source	110	110	110	110	110	110	110	110	110	
Source (Error)	105	105	105	105	105	105	105	105	105	
dB Value at receiver	73.24	45.03	52.40	52.80	43.46	44.18	41.92	50.08	48.59	58.71
dB Value at the receiver (Error)	73.21	39.34	47.29	47.70	37.90	37.97	36.42	44.12	43.48	52.69
Error										6.023

References

1. Choi, J.; Zhang, B.; Oh, K. The shortest path from shortest distance on a polygon mesh. *J. Theor. Appl. Inf. Technol.* **2017**, *95*, 4446–4454.
2. Tandel, B.; Sonaviya, D. A Quick Review on Noise Propagation Models and Software. In Proceedings of the ICSBE-2016—7th International Conference On Sustainable Built Environment, Kandy, Sri Lanka, 16–18 December 2016; p. 6.
3. Bendtsen, H. The Nordic prediction method for road traffic noise. *Sci. Total Environ.* **1999**, *235*, 331–338. https://doi.org/10.1016/S0048-9697(99)00216-8.
4. Hadzi-Nikolova, M.; Mirakovski, D.; Ristova, E.; Ceravolo, S. Modeling and Mapping of Urban Noise Pollution with SoundPLAN Software. *Univ. Goce Delcev.* **2012**, *185*, 38–41.
5. Kogut, J.P.; Pilecka, E. Application of the terrestrial laser scanner in the monitoring of earth structures. *Open Geosci.* **2020**, *12*, 503–517. https://doi.org/10.1515/geo-2020-0033.
6. Veronese, L.D.P.; Ismail, A.; Narayan, V.; Schulze, M. An Accurate and Computational Efficient System for Detecting and Classifying Ego and Sides Lanes Using LiDAR. *IEEE Intell. Veh. Symp. Proc.* **2018**, *2018*, 1476–1483. https://doi.org/10.1109/IVS.2018.8500434.
7. Asal, F.F.F. Comparative Analysis of the Digital Terrain Models Extracted from Airborne LiDAR Point Clouds Using Different Filtering Approaches in Residential Landscapes. *Adv. Remote Sens.* **2019**, *08*, 51–75. https://doi.org/10.4236/ars.2019.82004.
8. Madkour, A.; Aref, W.G.; Rehman, F.U.; Rahman, M.A.; Basalamah, S. A Survey of Shortest-Path Algorithms. *arXiv* **2017**, arXiv:1705.02044.
9. Duan, P.; AI, Y. Research on an Improved Ant Colony Optimization Algorithm and its Application. *Int. J. Hybrid Inf. Technol.* **2016**, *9*, 223–234. https://doi.org/10.14257/ijhit.2016.9.4.20.
10. Zarrinpanjeh, N.; Dadrass Javan, F.; Naji, A.; Azadi, H.; De Maeyer, P.; Witlox, F. Optimum path determination to facilitate fire station rescue missions using ant colony optimization algorithms (case study: City of Karaj). *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *43*, 1285–1291. https://doi.org/10.5194/isprs-archives-XLIII-B3-2020-1285-2020.
11. Alasadi, H.A.A.; Aziz, M.T.; Dhiya, M.; Abdulmajed, A. A Network Analysis for Finding the Shortest Path in Hospital Information System with GIS and GPS. *J. Netw. Comput. Appl.* **2020**, *5*, 10–23. https://doi.org/10.23977/jnca.2020.050103.
12. Canali, C.; Lancellotti, R. GASP : Genetic Algorithms for Service Placement in Fog Computing Systems. *Algorithms* **2019**, *12*, 201. https://doi.org/10.3390/a12100201.
13. Mittal, H.; Okorn, B.; Jangid, A.; Held, D. Self-Supervised Point Cloud Completion via Inpainting. *arXiv* **2021**, arXiv:2111.10701.
14. Win, L.L. Finding optimal route information and network analysis for emergency service. *Int. J. Adv. Res. Dev.* **2018**, *3*, 22–25.
15. Medrano, F.A. Effects of raster terrain representation on GIS shortest path analysis. *PLoS ONE* **2021**, *16*, e0250106. https://doi.org/10.1371/journal.pone.0250106.
16. Jyothi, S.; Padmavati, S.; Visvavidyalayam, M.; Varma, M.K.R.; Varma, S.A.K. Digitizing the Forest Resource Map Using ArcGIS. *Int. J. Comput. Sci. Issues* **2010**, *7*, 300.
17. Tiwari, S.K.; Kumaraswamidhas, L.A.; Garg, N. Accuracy of short-term noise monitoring strategy in comparison to long-term noise monitoring strategy. *Indian J. Pure Appl. Phys.* **2021**, *59*, 569–576.
18. Joshi, G.; Pal, B.; Zafar, I.; Bharadwaj, S.; Biswas, S. Developing intelligent fire alarm system and need of UAV. *Lect. Notes Civ. Eng.* **2020**, *51*, 403–414. https://doi.org/10.1007/978-3-030-37393-1_33.
19. Dubey, R.; Bharadwaj, S. Collaborative air quality mapping of different metropolitan collaborative air quality mapping of different metropolitan. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2021**, *XLIII-B4-2*, 87–94. https://doi.org/10.5194/isprs-archives-XLIII-B4-2021-87-2021.
20. Dubey, R.; Bharadwaj, S.; Zafar, M.I.; Mahajan, V.; Srivastava, A.; Biswas, S. GIS Mapping of Short-Term Noisy Event of Diwali Night in Lucknow City. *ISPRS Int. J. Geo-Inf.* **2021**, *11*, 25. https://doi.org/10.3390/ijgi11010025.
21. Dubey, R.; Bharadwaj, S.; Zafar, M.I.; Bhushan Sharma, V.; Biswas, S. Collaborative noise mapping using smartphone. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *43*, 253–260. https://doi.org/10.5194/isprs-archives-XLIII-B4-2020-253-2020.
22. Zafar, M.I.; Bharadwaj, S.; Dubey, R.; Biswas, S. Different scales of urban traffic noise prediction. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *43*, 1181–1188. https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-1181-2020.
23. Bharadwaj, S.; Dubey, R.; Biswas, S. Determination of the Best Location for Setting up a Transmission Tower in the City. In Proceedings of the 2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC), Aurangabad, India, 30–31 October 2020; pp. 63–68. https://doi.org/10.1109/ICSIDEMPC49020.2020.9299612.
24. Bharadwaj, S.; Dubey, R. Raster data based automated noise data integration for noise raster data based automated noise data integration for noise. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2021**, *43*, 159–166. https://doi.org/10.5194/isprs-archives-XLIII-B4-2021-159-2021.
25. Dubey, R.; Bharadwaj, S.; Biswas, D.S. Intelligent Noise Mapping Using Smart Phone on Web Platform. In Proceedings of the 2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC), Aurangabad, India, 30–31 October 2020; pp. 69–74. https://doi.org/10.1109/ICSIDEMPC49020.2020.9299597.
26. Sun, P.; Zhao, X.; Xu, Z.; Wang, R.; Min, H. A 3D LiDAR Data-Based Dedicated Road Boundary Detection Algorithm for Autonomous Vehicles. *IEEE Access* **2019**, *7*, 29623–29638. https://doi.org/10.1109/ACCESS.2019.2902170.

27. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep Learning for 3D Point Clouds: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 4338–4364, doi:10.1109/TPAMI.2020.3005434.
28. Nguyen, H.T.; Pearce, J.M.; Harrap, R.; Barber, G. The application of LiDAR to assessment of rooftop solar photovoltaic deployment potential in a municipal district unit. *Sensors* **2012**, *12*, 4534–4558, doi:10.3390/s120404534.
29. Masiero, A.; Tucci, G.; Vettore, A. Tree Detection with a Mobile Laser Scanner. *IOP Conf. Ser. Earth Environ. Sci.* **2021**, *76*, 012034. https://doi.org/10.1088/1755-1315/767/1/012034.
30. Song, W.; Zou, S.; Tian, Y.; Fong, S.; Cho, K. Classifying 3D objects in LiDAR point clouds with a back-propagation neural network. *Human-centric Comput. Inf. Sci.* **2018**, *8*, 1–12, doi:10.1186/s13673-018-0152-7.
31. Murcia, H.F.; Tilaguy, S.; Ouazaa, S. Development of a Low-Cost System for 3D Orchard Mapping Integrating UGV and LiDAR. *Plants* **2021**, *10*, 2804.
32. Chen, Z.; Gao, B.; Devereux, B. State-of-the-art: DTM generation using airborne LIDAR data. *Sensors* **2017**, *17*, 150, doi:10.3390/s17010150.
33. Guoping, L.; Chunxiang, C.; Xiaowen, L.; Hao, Z.; Qisheng, H.; Linyan, B.; Chaoyi, C. Classification of Lidar Point Cloud and Generation of DTM from Lidar Height and Intensity Data in Forested Area. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2008**, *XXXVII. Pa*, 313–318.
34. Bharadwaj, S.; Dubey, R.; Zafar, M.I.; Srivastava, A.; Bhushan Sharma, V.; Biswas, S. Determination of Optimal Location for Setting Up Cell Phone Tower in City Environment Using Lidar Data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *43*, 647–654. https://doi.org/10.5194/isprs-archives-xliii-b4-2020-647-2020.
35. Sharma, V.B.; Singh, K.; Gupta, R.; Joshi, A.; Dubey, R.; Gupta, V.; Bharadwaj, S.; Zafar, M.I.; Bajpai, S.; Khan, M.A.; et al. Review of structural health monitoring techniques in pipeline and wind turbine industries. *Appl. Syst. Innov.* **2021**, *4*, 59. https://doi.org/10.3390/asi4030059.
36. Sharma, M.; Paige, G.B.; Miller, S.N. DEM development from ground-based LiDAR data: A method to remove non-surface objects. *Remote Sens.* **2010**, *2*, 2629–2642, doi:10.3390/rs2112629.
37. Sulaiman, N.S.; Majid, Z.; Setan, H. DTM generation from LiDAR data by using different filters in open–source software. *Geoinf. Sci. J.* **2010**, *10*, 89–109.
38. Meng, X.; Currit, N.; Zhao, K. Ground filtering algorithms for airborne LiDAR data: A review of critical issues. *Remote Sens.* **2010**, *2*, 833–860, doi:10.3390/rs2030833.
39. Bello, S.A.; Yu, S.; Wang, C.; Adam, J.M.; Li, J. Review: Deep learning on 3D point clouds. *Remote Sens.* **2020**, *12*, 1–37, doi:10.3390/rs12111729.
40. Priestnall, G.; Jaafar, J.; Duncan, A. Extracting urban features from LiDAR digital surface models. *Comput. Environ. Urban Syst.* **2000**, *24*, 65–78, doi:10.1016/S0198-9715(99)00047-2.
41. Iordan, D.; Popescu, G. The accuracy of LiDAR measurements for the different land cover categories. *Earth Obs. Surv. Environ. Eng.* **2015**, *IV*, 158–164.