

AI-Based Environment Segmentation Using a Context-Aware Channel Sounder

Anuraag Bodi^{1,2}, Samuel Berweger³, Raied Caromi¹, Jihoon Bang^{3,4}, Jelena Senic^{3,4}, Camillo Gentile¹

¹Communications Technology Laboratory, Gaithersburg, Maryland, USA

²Associate, Prometheus Computing LLC, Cullowhee, North Carolina, USA

³Communications Technology Laboratory, Boulder, Colorado, USA

⁴Associate, Department of Physics, University of Colorado, Boulder, Colorado, USA
camillo.gentile@nist.gov

Abstract—We describe how the data acquired from the camera and Lidar systems of our context-aware radio-frequency (RF) channel sounder is used to reconstruct a 3D mesh of the surrounding environment, segmented and classified into discrete objects. First, the images captured by the camera are segmented into objects through an AI-based algorithm. Then the segmented images are projected onto the point cloud captured by the Lidar. Since the receiver end of the channel sounder is mounted on a mobile robot, the data is acquired in the local coordinate system and so must be transformed to a global coordinate system to synthesize a single, holistic point cloud of the environment. Finally, the synthesized point cloud is tessellated into a 3D mesh. The segmented mesh can be used for the automated – *i.e.*, without human analysis – reduction of the data acquired by the RF system of the sounder into an object-specific channel model.

Index Terms—5G, 6G, artificial intelligence, channel model

I. INTRODUCTION

The evolution of wireless communications networks over the last 40 years has witnessed ever wider bandwidths and ever more antennas, to deliver data at higher throughput, lower latency, etc. Thanks to the associated advances in microelectronics, today's radio-frequency (RF) channel sounders have fine enough resolution in the respective delay and angle domains to resolve and characterize the properties of *individual* channel multipaths. This marks a significant departure from channel modeling in the past when key parameters, such as the pathloss exponent, delay spread, angle spread, Rician K-factor, etc., representing the collective behavior of the channel over *all* multipaths received, would be used to characterize simple stochastic models.

The ability to characterize individual multipaths has ushered in a class of channel models said *quasi-deterministic* [1]: the geometrical parameters of the multipaths are predicted through (simplified) *deterministic* raytracing, yet they still retain *stochastic* parameters for calibration against measurements, for reliability. The deterministic component renders the model generalizable to environments where no measurements were taken, provided that a geometrical representation of the new environment is available. Another advantage of quasi-deterministic models is their inherent spatial-temporal consistency for multiple input multiple output (MIMO) in 4G systems, beamtracking in 5G systems [2], and sensing in 6G systems [3]. Their disadvantage with



Fig. 1. RX end of our context-aware channel sounder, shown with the 60 GHz horn antenna array integrated as the RF system. The Lidar is mounted above the RF array and the panoramic camera is mounted above the Lidar.

respect to stochastic models is the amount of time required to reduce the measurements into the parameters of the more complex structure, necessitating the identification, classification, and characterization of distinct objects in the surrounding environment through human visual inspection, with support from raytracing.

Thanks to pre-distortion filtering and super-resolution multipath extraction, the RF systems of our channel sounders can resolve individual multipaths with resolution of at least order 1 ns in delay and 2° in angle [4], making them ideal for quasi-deterministic modeling. And because they integrate electronically switched antenna arrays and direct sampling at the intermediate frequency (IF), the channel can be captured in fractions of millisecond, enabling collection of hundreds to thousands of measurements in just minutes or hours. In fact, while in the past the measurement campaign comprised the lion's share of channel modeling, in our case it is the ensuing reduction procedure that often involves weeks or even months after a measurement campaign of a just few hours, due to the human analysis required.

To render the channel reduction procedure scalable to the thousands of measurements that our RF systems can acquire, we have supplemented it with camera and Lidar systems for context awareness [5], acting as the “eyes” of the channel

sounder in the place of human analysis. The data acquired by the camera and Lidar systems is used to reconstruct a 3D mesh of the environment, segmented into discrete objects. This allows that data acquired by the RF system to be automatically reduced into an object-specific channel model, by raytracing the mesh [5].

In Section II, we describe our context-aware channel sounder, followed by Section III that discusses how the images captured by the camera are segmented into discrete objects through an AI-based algorithm and then projected onto the point cloud captured by the Lidar. Since the receiver end of the channel sounder is mounted on a mobile robot, the data acquired in the local coordinate system of the robot must be transformed to global coordinate system to synthesize a single, holistic point cloud of the environment, as described in Section IV. Finally, the synthesized point cloud is tessellated into a 3D mesh, as explained in Section V. The last section is reserved for conclusions.

II. CONTEXT-AWARE CHANNEL SOUNDER

In this section, we provide an overview of our context-aware channel sounder, as well as of a measurement campaign conducted in a lobby environment as an example of the segmentation process.

A. RF System

The RF system used for this example has spherical 60 GHz horn arrays at the transmitter (TX) and receiver (RX) and so is said 3D double directional [4], meaning that it can estimate angle-of-departure of the multipaths (in azimuth and elevation) from the TX and estimate their angle-of-arrival to the RX. During an RF acquisition, the complex channel impulse response between all pairs of TX and RX antennas is recorded by transmitting a pseudorandom (PN) sequence with 2 GHz bandwidth, direct IF sampling at the RX, and match filtering in postprocessing. Predistortion filtering is used to calibrate for the nonidealities of the hardware, increasing the dynamic range of the system to 55 dB. The channel impulse responses are synthesized through the SAGE super-resolution algorithm to extract in addition to 3D double-directional angle, the path gain and delay of the multipaths; super-resolution algorithms nominally yield resolution in the angle and delay domains about five times the inherent beamwidth and bandwidth of the system, respectively. As a result, the estimation error when compared against the known properties of the ground-truth line-of-sight path is at least 2 dB in path loss, 1 ns in delay, 1° in azimuth, and 2° in elevation. Critically, SAGE also de-embeds the gain patterns of the antennas, ensuring that the estimated properties of the multipaths reflect the channel alone and not the properties of the hardware. Triggering between the TX and RX antennas is provided by way of untethered Rubidium clocks on each end.

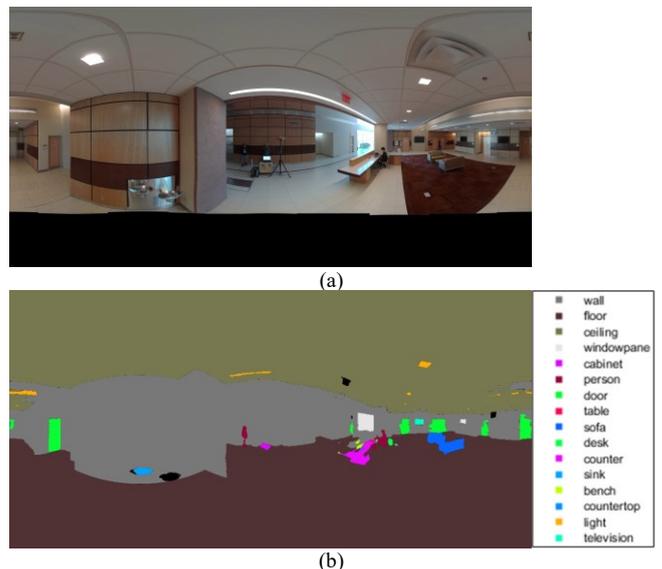


Fig. 2. (a) Panoramic image of the Lobby recorded by the camera. (b) Segmentation of the image into a mask of discrete objects categorized in the legend.

B. Lidar System

The Lidar system is mounted above the RF system at the RX, as shown in Fig. 1. The OS0-128 Lidar from Ouster¹ captures a point cloud by way of mechanical scanning 2047 angles within a 360° azimuth field-of-view and recording the intensity value at 128 points of a vertical laser array within the 90° elevation field-of-view, rendering an equivalent angular resolution of 0.17° in azimuth and 0.7° degrees in elevation.

C. Camera System

The panoramic camera is mounted above the Lidar at the RX, as shown in Fig. 1. The iSTAR Pulsar camera from NCTech¹ records the RGB value at each pixel of an 11000 × 5500 spherical image within the 360° azimuth and 145° elevation field-of-view synthesized by four fisheye lenses. The spherical image of the Lobby environment is shown in Fig. 2(a).

D. Measurement Campaign

The measurement campaign was conducted in the Lobby environment shown in Fig. 2(a). The data was collected by mounting the RX on a mobile robot, equipped with a laser-guided navigational system that reports its heading and the location of each measurement. A total of 4956 measurements were collected as the robot traversed a 2D grid throughout the environment, roughly with 10 cm spacing between the grid points.

¹Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by National Institute of Standards and Technology (NIST), nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

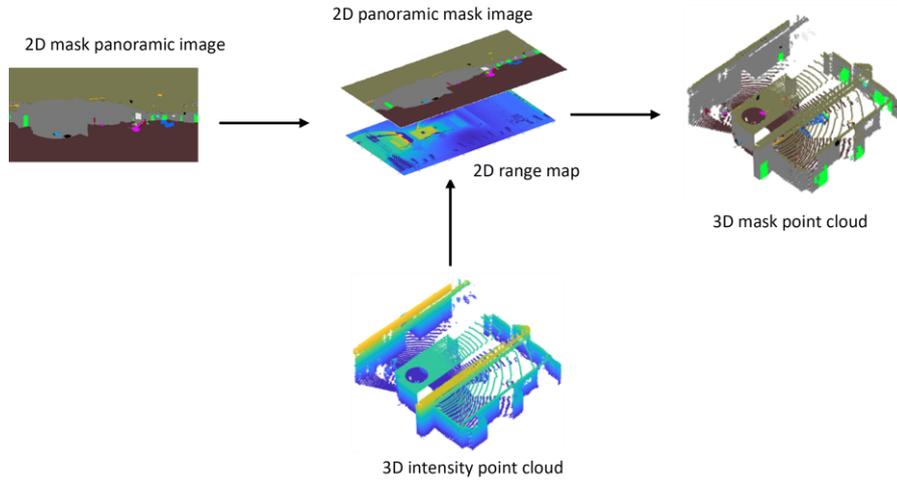


Fig. 3. Workflow for projection of the mask image onto the point cloud. The 2D mask panoramic image is projected onto the 2D range map which is unfolded from the 3D intensity point cloud. Then the 2D projected mask is folded back into a 3D mask point cloud.

III. MASK POINT CLOUD

The segmentation of the environment into discrete objects is actually realized by segmenting the point cloud captured by the Lidar into discrete objects. Firstly, a mask is generated by segmenting the images captured by the camera into different objects, as explained in the first subsection. Then the segmented mask is projected onto the point cloud, as explained in the second subsection.

A. Semantic Segmentation

The panoramic images captured by the camera were used to classify the objects in the environment via semantic segmentation. Semantic segmentation is a computer vision task that generates a mask, by assigning a label to each pixel in an image that indexes the object category to which it belongs. Our segmentation was accomplished via a state-of-the-art, AI-based algorithm trained on a vast database of objects. The backbone of the algorithm is the Swin Transformer [7], which is a novel vision transformer that uses shifted windows to capture local and global information in an efficient way. Unlike conventional convolutional neural networks, vision transformers employ self-attention mechanisms to process image patches as tokens. Swin Transformer consists of a hierarchical architecture that divides the input image into patches of different resolutions and applies self-attention within and across these patches. This allows the algorithm to learn rich features, to achieve high confidence in the segmentation process.

The algorithm was trained on the ADE20K dataset, which is a large-scale semantic segmentation dataset containing over 20000 images exhaustively annotated with pixel-level objects [8]. The dataset comprises a total of 150 semantic categories, including indiscrete objects like floors and ceilings and discrete objects like tables, chairs, and

televisions; of course, it includes other object categories not present in our environment, such as cars, beds, etc. In postprocessing, we removed the latter categories that resulted from detection error, yielding a total of 16 categories present in the lobby. The segmentation algorithm was trained on square images whereas our camera captured 360° panoramic images. Therefore, we “folded” the panoramic images onto a cube and fed the six faces to the segmentation algorithm instead. Upon segmentation, the segmented faces were then unfolded back into a single panoramic mask.

As an additional step, we filtered the noise in the mask by applying a morphological majority window with a 3×3 neighborhood. The filter sets a pixel to 1 if five or more pixels in its neighborhood are 1; otherwise, it sets the pixel to 0. The results are masks that contains integer data with each pixel value indexed to a distinct object category. Fig. 2(b) shows the mask segmented from the image in Fig. 2(a).

B. Mask Projection

The six mask face images were unfolded into a single 11000×5500 panoramic image that extends across the azimuth and elevation field-of-view of the camera. In kind, we unfolded the intensity point cloud captured by the Lidar into a 2048×128 range map that extends across the azimuth and elevation field-of-view of the Lidar. Then the unfolded mask was projected onto the unfolded range map through rasterization. Since the resolution of the camera is much higher than the Lidar’s, the mask values projected onto the range map were interpolated without loss in resolution. The final step was to fold the projected mask back into a 3D point cloud. The complete workflow is illustrated in Fig. 3.

The projection did consider the different elevation field-of-view of the Lidar (90°) and the camera (135°), so the unfolded mask was cropped to match the unfolded range map. The projection also did account for the off-axis displacement between the Lidar and the camera on the RX mount, as explained in [6].

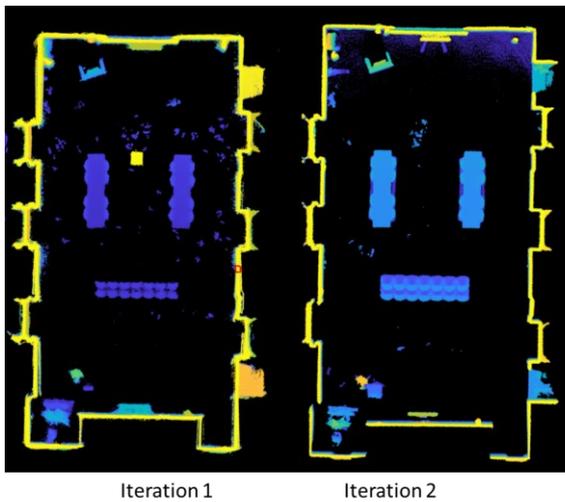


Fig. 4. A cross-section of the synthesized point clouds across all measurements, after one iteration and after two iterations of the ICP algorithm are performed. The ICP algorithm after two iterations has smaller error, as seen in the thickness of walls and level of details in the chairs in blue color in the middle of the room.

IV. POINT CLOUD SYNTHESIS

Each measurement in the campaign will have a different observation of the environment since it is acquired at a different RX position on the mobile robot; this is true even if the environment layout is concave, as the three systems have limited field-of-view in azimuth and elevation. To generate a single, holistic view of the environment across all point clouds, they must be synthesized; as well, this improves the resolution of the individual point clouds while reducing the noise in each. Synthesis involves transforming the point clouds captured in the local system of the robot to a global coordinate system. This means that a separate 3D affine transformation that accounts for both rotation and translation between the local system of each measurement to the global system must be found. While it is true that the robot reports its heading and location, from which the transformations could be estimated, precision synthesis requires tolerance beyond what the robot heading reports, whose error can be up to 15° . Instead, coarse transformations are first found, followed by fine transformations, as explained in the next two sections.

A. Coarse Transformations

The coarse transformations are found through the fast global registration (FGR) algorithm [9]. We choose the local coordinate system of the first point cloud as the global coordinate system for all. Then transformations to the first point cloud from all other points clouds were found, individually.

After denoising the point clouds by removing outliers – points that are isolated from other points in the same cloud and so cannot be clustered – the remaining points are clustered through the k-nearest neighbor algorithm. Then Fast Point Feature Histogram (FPFH) features, which are

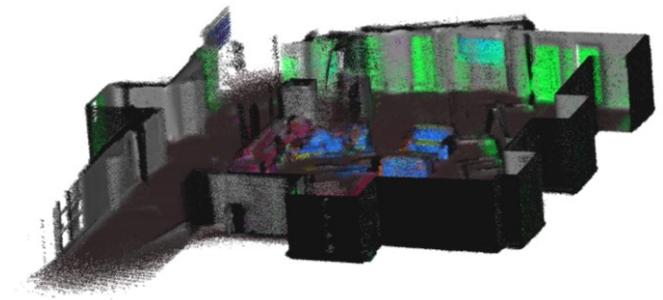


Fig. 5. After the fine transformations were applied to each of the point clouds, they were superimposed across the 4956 measurements.

orientation invariant in images *e.g.*, corners, are extracted from each cluster. Next, the algorithm searches for correspondence pairs between two FPFH features in different point clouds. A reciprocity test is used to determine if a pair of corresponding features are rightly matched. Reciprocity test is performed for a pair (p,q) of FPFH features in point cloud P1 and P2 such that p lies in q 's k-NN cluster *i.e.*, when k-NN is run on P2 with p added, p is within q 's nearest neighbor cluster. Similarly, q should in p 's k-NN cluster. If they are in each other's k-NN cluster, then the two pass the correspondence test. If they do not pass the test, then they are said to be outliers. Correspondence pairs which are outliers are discarded and reciprocity test is performed again till the outlier ratio is less than 5%. Finally, the corresponding features are mapped to each other through the affine transformation, which requires at least four pairs; otherwise least-squares is used.

The FGR algorithm also performs checks for consistency across all the transformations found, *e.g.*, the transformation from point cloud A plus the transformation from point cloud B to point cloud A must be equal to the transformation from point cloud B (within some tolerance).

B. Fine Transformations

Once the coarse transformations are applied, we implement the iterative closest point (ICP) algorithm [10] to obtain fine transformations. As the FGR algorithm, it finds transformations to the first point cloud from all other points clouds. The major difference between the two algorithms is that, instead of mapping between corresponding features, a mapping between all corresponding points in the two clouds is found. As such, the transformation is highly over constrained due to the sheer number of points. After the fine transformations are found, they are applied. The algorithm is then repeated once again to improve resolution. Fig. 4 shows the enhanced resolution between the two successive iterations.

Once all the fine transformations are found through the ICP algorithm, they are applied. Then the transformed point clouds are superimposed on each other. The result of the point clouds superimposed across all measurements is shown in Fig. 5. The combined point cloud has 65 million points superimposed across the 4956 measurements.

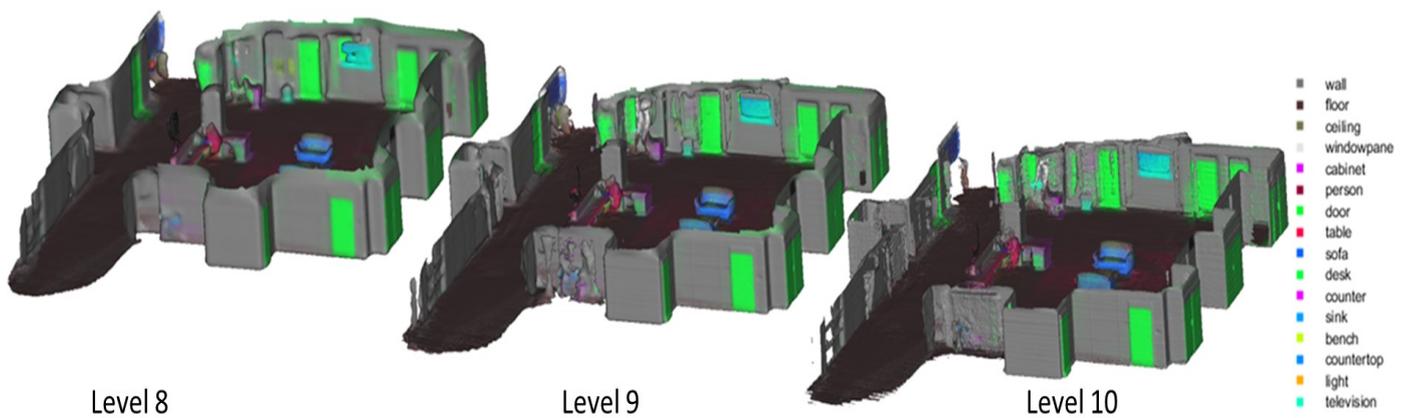


Fig. 6. 3D reconstruction for different octree depths.

V. POINT CLOUD TESSELATION

Raytracing is based on a structure of faces with edges, to compute reflections and diffractions respectively. Hence a point cloud structure is not suitable for raytracing. Instead, the synthesized point cloud is tessellated into a 3D mesh. First, an octree is generated from the point cloud. An octree is a tree data structure in which each node is divided into eight children [11]. It recursively partitions the point cloud into eight octants, where the objective is to obtain roughly the same number of points in each octant. The number of recursions is determined by the local density of the points in the cloud, which in turn determines the resolution of the final mesh. The number of recursions is referred to as the depth of the octree.

Finally, the Poisson reconstruction [12] is used to tessellate the points in an octant to a 3D mesh. Poisson reconstruction is performed by constructing an implicit function across the points in an octant. Then the implicit function is used to construct a surface with a given precision using Delaunay refinement. Once the Poisson reconstruction is completed, the mesh filtered to remove unnecessary features such as duplicate, huge, or tiny triangles. Fig. 6 shows the triangle density of the point cloud and Fig. 7 shows the reconstructed 3D mesh of the Lobby at different octree depths. If the octree depth is too low, fine details in smaller objects like chairs and tables will be missed thus reducing the resolution. Greater octree depth results in

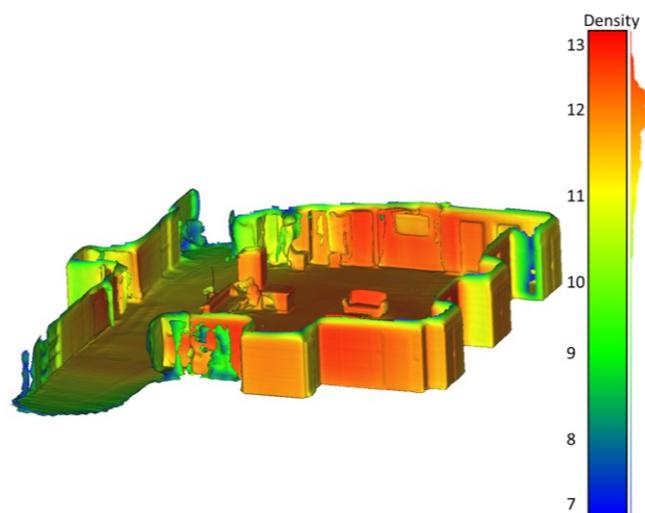


Fig. 7. Triangle density of the tessellated point cloud.

higher resolution, as seen level 9 and 10 in Fig. 6. However, higher resolution does increase the noise level, which is observable in depth 10. We found that a good balance between resolution and noise was struck using level 9. The octree level 8 mesh has 221,000 triangles and 110,000 vertices. Level 9 has 1 million triangles and 523,000 triangles. Level 10 has 4.8 million triangles and 2.4 million vertices.

REFERENCES

- [1] Gentile, Camillo, et al. "Quasi-deterministic channel model parameters for a data center at 60 GHz." *IEEE Antennas and Wireless Propagation Letters* 17.5 (2018): 808-812.
- [2] Blandino, Steve, et al. "Markov multi-beamtracking on 60 GHz mobile channel measurements." *IEEE Open Journal of Vehicular Technology* 3 (2021): 26-39.
- [3] Chuang, Jack, et al. "Quasi-deterministic channel propagation model for human sensing: Gesture recognition use case." *Submitted to IEEE Trans. on Wireless Communications*, Sept. 2023.
- [4] Sun, Ruoyu, et al. "Design and calibration of a double-directional 60 GHz channel sounder for multipath component tracking." *European Conf. on Antennas and Propagation*, March 2017.
- [5] Gentile, Camillo, et al. "Context-Aware Channel Sounder for AI-Based Radio-Frequency Channel Modeling." *Submitted to European Conf. on Antennas and Propagation*, March 2024.
- [6] Mi, Hang, et al. "Measurement-Based Prediction of MmWave Channel Parameters Using Deep Learning and Point Cloud." *Submitted to IEEE Trans. on Antennas and Propagation*, March 2024.
- [7] Ze Liu, et al. "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows." *IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada, 2021 pp. 9992-10002.
- [8] Bolei Zhou, et al. "Scene Parsing through ADE20K Dataset," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 5122-5130.
- [9] R. Benjemaa and F. Schmitt, "Fast global registration of 3D sampled surfaces using a multi-z-buffer technique," *Image and Vision Computing*, vol. 17, no. 2, pp. 113–123, Feb. 1999, doi: [https://doi.org/10.1016/s0262-8856\(98\)00115-2](https://doi.org/10.1016/s0262-8856(98)00115-2).
- [10] Rusinkiewicz, S. and Levoy, M. (no date) 'Efficient variants of the ICP algorithm', *Proceedings Third International Conference on 3-D Digital Imaging and Modeling [Preprint]*. doi:10.1109/im.2001.924423.
- [11] M. A. Lodhi, J. Pang and D. Tian, "Sparse Convolution Based Octree Feature Propagation for Lidar Point Cloud Compression," *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023, pp. 1-5, doi: 10.1109/ICASSP49357.2023.10096990.
- [12] M. Kazhdan, M. Chuang, S. Rusinkiewicz, and H. Hoppe, "Poisson Surface Reconstruction with Envelope Constraints," *Computer Graphics Forum*, vol. 39, no. 5, pp. 173–182, Aug. 2020, doi: <https://doi.org/10.1111/cgf.14077>.