mmSV: mmWave Vehicular Networking using Street View Imagery in Urban Environments

Ahmad Kamari, Yoon Chae, Parth Pathak
George Mason University
{akamari,ychae2,phpathak}@gmu.edu

Abstract

As we move towards a future of connected and autonomous vehicles, high-speed and low-latency connectivity between vehicles is becoming increasingly important. This paper investigates enabling high data rate mmWave links in vehicle-to-vehicle (V2V) scenarios using street view images. We find that mmWave V2V links in urban settings suffer from frequent and prolonged blockages, resulting in unreliable connection and high beamforming overhead. Our work proposes mmSV, a system that creates 3D reflection profiles from street view images to assist vehicles in finding mmWave reflections from the environment in real-time. mmSV consists of two key components: material identification which identifies materials from street view images to determine their reflectivity and create 3D reflection map, and environment-driven ray-tracing and beamsearching which finds a high-SNR beam using predicted 3D material maps. Our extensive experimental results on the mmWave testbed show that mmSV can provide highly reliable V2V mmWave connectivity with low beamforming overhead.

CCS Concepts

• **Networks** → *Network measurement*; *Network simulations*;

Keywords

mmWave Networks, V2V/5G Communication, Beamforming

ACM Reference Format:

Ahmad Kamari, Yoon Chae, Parth Pathak. 2023. mmSV: mmWave Vehicular Networking using Street View Imagery in Urban Environments . In *The 29th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '23), October 2–6, 2023, Madrid, Spain.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3570361.3613291

1 Introduction

The number of connected and autonomous vehicles is expected to surpass 300 million by 2035 [6] in the United States with over 5 million fully autonomous vehicles [5]. These vehicles are expected to be equipped with a range of sensors including RGB and infrared cameras, LiDARs, automotive radars, ultrasonic sensors, GPS, IMUs, and more. A limitation in using these sensing modalities is that they have a limited sensing range with largely line-of-sight operations in the presence of occlusions [59]. Various safety, fuel efficiency, and infotainment-related applications require cars to exchange their sensing information with other vehicles in their surrounding. For example, sharing raw or partially processed sensor data (e.g., LiDAR feed) with other vehicles can enable them to enhance their sensing range. It can help a vehicle create a "see-through" or "bird's eye" view of the environment. However, this necessitates high-capacity connectivity, surpassing current standards (e.g., 802.11p/DSRC [1], V2V LTE [15]). 3GPP 5GNR V2X [4] envisions applications like platooning, extended sensors, advanced driving, and remote driving, requiring data rates exceeding 1 Gbps. Although 802.11ax can provide high data rates, it requires wider bonded channels and/or more MIMO streams. However, achieving dynamic channel bonding and large spatial multiplexing gains in high mobility is extremely challenging [21]. Furthermore, the spatial reuse decreases with 802.11ax due to omnidirectional transmission and interference, resulting in higher latencies and packet collisions. Relaying traffic through nearby base stations can increase the latency to tens of milliseconds [53] which is not acceptable for real-time safety-critical vehicular applications. On the other hand, millimeter-wave (mmWave) V2V links offer multi-gigabit data rates, making it possible to realize a range of connected and autonomous vehicle applications.

A major challenge arises from V2V mmWave links' high fragility due to frequent blockages. Our characterization reveals that mmWave channels encounter repeated blockages from other vehicles, even when transceivers are rooftop-mounted. This is due to significant diversity in vehicle heights ranging from 3.5 ft to 6.5 ft for different types of passenger cars [55] and up to 13 ft for buses and trucks. Even when considering urban traffic with a majority of passenger vehicles and infrequent buses, mmWave links between vehicles are frequently blocked and the blockage durations can be much longer (in orders of seconds) depending on traffic density. This is different from V2I channels (characterized in [76] and [51]) where placing the base stations (BS) at a height on roadside units or a nearby building is shown to achieve much

ACM MobiCom '23, October 2–6, 2023, Madrid, Spain

ACM ISBN 978-1-4503-9990-6/23/10.
https://doi.org/10.1145/3570361.3613291

fewer blockages. We also find that V2V channels can benefit significantly from reflections available through nearby buildings and infrastructure, especially in urban settings. However, leveraging these reflections in the presence of constant fast mobility and prolonged blockages (i.e., long NLoS durations but short beam coherence durations) requires frequent beam searching to maintain connectivity and to ensure reliable and real-time sensor data transfer between vehicles.

While there is a plethora of prior work [16, 18, 42, 66] in using out-of-band information (such as camera and LiDAR) for improving mmWave beamforming, these works are limited to V2I scenarios and require cameras or LiDAR sensors to be deployed in the environment or on BSs. Another prior work [80] performs indoor LiDAR scanning to develop reflection profiles for zero-short beamforming. Extending this work to outdoor scenarios is not straightforward as it would require large-scale LiDAR profiling of the environment and publicly available LiDAR datasets [19, 22, 44, 69] are limited to terrain mapping and few urban locations. Sharing vehicle location information over an out-of-band channel such as DSRC as proposed in [26] is also not sufficient as they do not help mmWave transceivers to profile environmental reflectors such as buildings. Hence, there is clearly a need for a novel, practical solution that can enable reliable V2V mmWave connectivity with low beamforming overhead.

In this paper, we ask the following question: can we leverage widely available street view images for improving the mmWave connectivity between vehicles? We claim that street view (SV) images could be an important yet unexplored solution to performing fast recovery from V2V blockages. There are various benefits of using SV images. They are readily available from a range of different map providers including Google [8] and Microsoft Bing [11]. They are fairly up-to-date especially for urban areas with dense traffic [10]. And most importantly, these images are collected at reasonably high resolution (e.g., 16384×8192) so that they can be used for performing various image recognition tasks [74, 75, 81]. The main idea of our work is to design a system (referred to as mmSV) that is capable of leveraging existing SV images to create 3D reflection maps through material identification. These maps can then be utilized by vehicles to perform efficient raytracing and low-overhead beamsearching in real-time. As a result of this process, vehicles are able to identify high-SNR beams for communication and achieve uninterrupted high data rates and low latency. Developing mmSV requires us to address multiple challenges. We describe these challenges and our solution approach next.

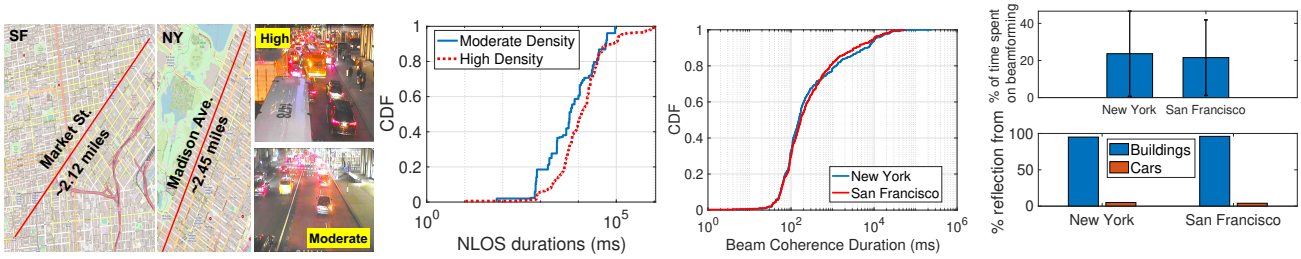
Challenge 1. Given the SV images, how can we create 3D reflection maps that can be useful to predict mmWave reflection? The SV images provide a detailed view of the surrounding buildings and other infrastructure that are made up of a diverse set of materials such as concrete, metal, glass, wood, stucco, vinyl, etc. As known from prior mmWave measurements [47, 48], these materials have different permittivity and reflect the mmWave signals differently.

To address this challenge, we develop a material segmentation and identification model that can classify the material patches in SV images. We find that there is no publicly available dataset for outdoor construction materials. Therefore, we develop our own large dataset with over 1200 SV images and 160 K material patches with careful manual labeling. The dataset is employed for material segmentation and subsequent dynamic object removal to extract 2D material maps. These maps are then projected on 3D building structures for creating a 3D material reflection map that can then be used by vehicles to identify potential mmWave reflections.

Challenge 2. How can a vehicle use the 3D material maps to communicate over mmWave efficiently and reliably with another vehicle in urban settings? A transmit (Tx) vehicle can fetch the 3D material map based on its location to identify and localize reflectors of different materials around it, but how can it know which mmWave paths are available, their SNR and angle of departure (AoD), and which transmit beams to use to realize those paths?

mmSV develops an environment-driven hierarchical raytracing method that operates in two stages. In the first stage, it utilizes "raytubes" (a few carefully chosen rays) aimed at object walls, pinpointing surfaces with high-probability reflection. In the second stage, uniform shoot and bounce raytracing is performed on the first stage's reflective surfaces. Incorporating material type and reflectivity data enables real-time mmWave path and AoD calculations. mmSV then uses restricted beamsearching to check the availability of these paths and their signal strength before identifying the beam that can provide high SNR to the receiver. mmSV also performs implicit localization of Rx vehicles from Tx using the channel measurements of restricted beamsearching, eliminating the need for out-of-band communication completely.

Summary of evaluation. We evaluate mmSV using our 60 GHz mmWave testbed with software radios and 802.11ad access points mounted over vehicles and carts. We perform mmWave experiments on two road segments with vehicles at over 260 separate locations to evaluate mmWave channel predictions based on 3D material maps. We summarize the evaluation results as follows: (1) Our material identification shows that mmSV can identify the material with a mean accuracy of 90.62% for patch material classification and 86.7% for full scene material segmentation. This includes 11 commonly found SV materials such as brick, stucco, vinyl, stone, and glass. Our 3D material map generation model is shown to be robust and generalizable as it is trained on a large number of urban SV images and tested on a completely different set of SV images in different urban areas (randomly chosen across the United States). (2) We find that mmWave paths predicted from 3D material maps created through SV



(a) (b) (c) (d)

Fig. 1: (a) Two road segments of NY and SF used in our mmWave channel characterization along with different traffic density, (b) NLoS durations between two vehicles at different traffic density, (c) beam coherence duration, and (d) beamforming overhead and percentage reflections from buildings for two road segments.

images can correctly predict 93.2% of mmWave paths with a mean distance (from Tx to Rx via reflector) error of 1.26 *m* and mean azimuth angle error of 3.53°. We also find that the mean predicted reflection loss error is 1.75 dB in all four segments. (3) Our environment-driven raytracing remarkable speed up (99× compared to the conventional shoot and bounce method) with the use of 3D material maps, making it possible to perform the process in real-time for the vehicles. (4) Our end-to-end evaluation shows that the SNR difference of the beam found using our 3D material map compared to the highest-SNR beam is less than 1 dB. mmSV achieves this SNR difference by probing only 6 beams on average out of 64 beams used in exhaustive beam scanning. (5) Lastly, mmSV achieves mean link-layer throughput of 810 Mbps and latency of 0.17 *ms* compared to 309 Mbps and latency of 1.32 *ms* using default 802.11ad protocol.

Contributions. The contributions of the paper can be summarized as follows:

- mmSV presents a first-of-its-kind material identification model for SV images with the application in wireless propagation modeling.
- mmSV designs an end-to-end solution for mmWave V2V communication including 3D material map design, low-cost environment-driven raytracing, and restricted beamsearching to maintain high-SNR mmWave connectivity.
- We extensively evaluate mmSV using real SV images and mmWave testbed to demonstrate the feasibility of the system and its efficacy over conventional mmWave protocols used in the V2V context.

2 Motivation

We first characterize LoS/NLoS durations along with beam coherence durations for typical urban V2V channels. We pick two road segments (2.45 miles of Madison Avenue in New York and 2.12 miles of Market Street in San Francisco as shown in Fig. 1a) for the purpose. To capture realistic traffic patterns, we use the traffic reports provided by the Department of Transportation for NY [9] and SF [7] on both road segments and calculate the traffic density. These resources offer the Annual Average Daily Traffic (AADT) for the road segments which is the total volume of vehicle traffic on a highway/road for a year divided by 365 days.

We categorize the traffic densities into two categories: high (900 vehicles per hour) and moderate (600 vehicles per hour). We then use these traffic densities to simulate realistic vehicle trajectories on the same road segments (number of lanes, intersections, etc.) in SUMO [52], a state-of-the-art traffic simulator. We then use these traffic density values and simulate realistic vehicle trajectories for the same road layouts using the SUMO traffic simulator [52]. We also use the OSM (Open Street Map [56]) data for the two road segments and export their 3D profile (building structures, maps, road layout, etc.). The 3D profile and the traffic vehicle trajectories are then input into the Remcom Wireless InSite [3] channel simulator for propagation estimation. We use CAD models of sedans (average height 4.6 ft), SUVs (average height 5.9 ft), and buses (average height 13 ft) in Remcom Wireless InSite. Based on a percentage of vehicles on the road reported to be SUVs (9% [23]) and trucks/buses (15% [7, 9]), we set the total number of SUVs and buses to 24% of cars in our simulation. We equip different pairs of vehicles with mmWave radios and 64-element phased array antennas. As these pairs act as transmitters (Tx) and receivers (Rx), their LoS paths can be blocked by other vehicles taller than them, and NLoS paths are enabled through buildings and other nearby reflectors (e.g., metal road signs). For each segment, we simulate over 2000 instances (snapshots) where each instance corresponds to different locations of vehicles on the road segment including the Tx and Rx vehicles. Depending on their location, the Tx and Rx vehicles are surrounded by different other vehicles and buildings on the road segment, changing the mmWave blockages and reflections.

NLoS durations. Fig. 1b shows the NLoS duration which is the duration between two consecutive instances where the LoS path is available between the two vehicles. We observe that the vehicles have to rely on NLoS paths for long durations due to blockages and vehicle mobility. Also, as the vehicle density increases, the probability of longer NLoS durations also increases. This is due to more blockages from other vehicles along with reduced vehicle mobility as the traffic increases within a constrained physical space of the road segment. Over 60% of NLoS durations are found to be longer than 1*s* for moderate density while the number increases to 85% for high density.

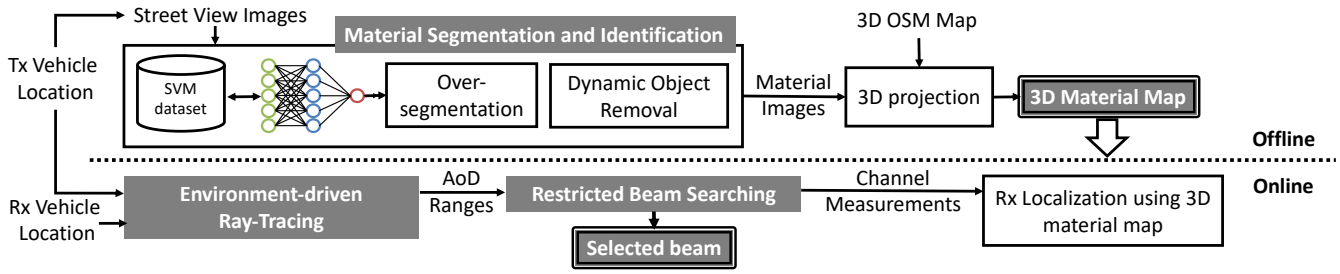


Fig. 2: Overview of mmSV

Beam coherence and BF overhead. Fig. 1c shows the beam coherence durations for two road segments where a beam coherence duration is defined as the duration for which the transmitter and receiver mmWave vehicles have the same best (highest SNR) beam pair. If the beam coherence durations are short, the two endpoints require frequent beamsearching, incurring high overhead. We observe from Fig. 1c that beam coherence durations are much shorter (less than 1 second 80% of the time) for both road segments. This shows that mmWave V2V channels in urban environments not only result in long NLoS durations (Fig. 1b), the beams have to be continuously adapted due to mobility, in turn significantly increasing the beamforming overhead. Fig 1d shows the beamsearching overhead defined as the percentage of time spent on beamsearching within every NLOS time window. In order to calculate this time, we consider the total time spent on Sector Level Sweep (SLS) and the SSW feedback duration. We find the beamforming overhead (based on 802.11ad) to be 24% on average (Fig. 1d) for the two road segments due to continuous beam adaptation in the presence of high mobility and blockages.

Type of reflectors. Further analyzing the beamsearching results, we find that stationary ambient reflectors such as the buildings and other objects around the road play a major role in providing NLoS paths compared to dynamic reflectors such as other cars. Although other vehicles (being metal body objects) provide reflections, such reflections are short-lived (less than 50 *ms* 95% of the time) due to their own mobility. On the other hand, buildings are available for reflection for longer durations in most urban environments. As shown in Fig. 1d, 94.7% of reflections on average in both road segments were from buildings while the remaining 5.3% were from other vehicles. This shows that identifying, localizing, and determining the reflectivity of stationary ambient reflectors such as buildings could be critically helpful in improving link connectivity while ensuring low beamforming overhead.

3 System Overview

Fig. 2 shows the overview of mmSV which consists of an offline material identification part and an online restricted beamsearching part. In the offline phase, material segmentation and identification are performed on SV images. This process includes over-segmentation and removal of dynamic objects (such as cars, people, etc.) from SV images that are not part of the stationary infrastructure (buildings, etc.). The material identification model is trained using our street view material (SVM) dataset. The output of the model is 2D material images translated from SV images which are then combined with 3D open street maps (OSM). The OSM maps include information such as the height and shape of the buildings that are used for projecting the 2D material images on 3D surfaces and creating 3D material maps. These 3D material maps can be prepared offline to be used by the vehicles while communicating over mmWave.

During the online phase, a Tx vehicle can leverage the 3D material maps (downloaded/preloaded) based on its location to perform raytracing. mmSV's environment-driven raytracing reduces the raytracing overhead by focusing the rays on surfaces of interest with reflecting materials based on our 3D material maps. The AoD ranges calculated from raytracing are then used to determine a small subset of beams that can be probed by the Tx vehicle to find a high-SNR beam to the Rx vehicle. The Tx vehicle also performs mmWave channel measurements and uses that to localize the Rx vehicle with the help of 3D material maps even when the LoS path is blocked, eliminating the need for any out-of-band communication. The estimated Rx location is then adapted in the next beamsearching time instance for performing environment-driven raytracing and restricted beamsearching. This way, mmSV creates an end-to-end V2V mmWave communication system that leverages the 3D material maps at every step.

4 3D Material Maps

4.1 Material Segmentation and Identification

mmSV's material segmentation and identification consists of three modules as shown in Fig. 2: (1) Material identification, (2) Over-segmentation, and (3) dynamic object removal.

4.1.1 Material Identification Given a pixel in an SV image, the main goal of this step is to determine the probability that the pixel belongs to a specific material category. At the core of this prediction module is a patch classifier that classifies the pixel at the center of the patch into a material category. The image classification models like GoogLeNet [72] take images as input and output a class and are not directly applicable to our purpose. Our material dataset requires a patch-based segmentation model and segmentation models such as [41] that rely on polygon annotation are not suitable for our dataset. Instead, our machine learning model

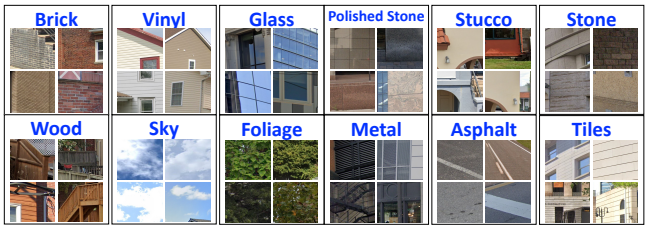


Fig. 3: Examples of patches in our SVM dataset.

is adapted from GoogLeNet [72] for the purpose of material identification. While other well-known models such as Resnet [37] and AlexNet [45] can be used for the purpose, the stride at each layer results in a prediction every 224 pixels in Resnet which is sparse for our material identification purpose. We modify the GoogLeNet model to get the predictions every 16 pixels. GoogLeNet introduces an inception module that uses a combination of filters with different sizes (1×1, 3×3, and 5×5) with less computational cost (10× lower cost) than previous architectures like AlexNet. This feature eases the ability to define a deeper structure with minimum accuracy sacrifice. The structure consists of multiple inception modules stacked together. To classify all pixels of an SV image, we convert the model to a sliding window model by replacing the last fully connected layer with a convolutional layer to densely classify a grid across the image. This results in a fully convolutional network that can classify images of any size. This convolutional structure results in a pixel probability map that can be used in the next steps. The weights of the network remain unchanged in our model and the stride of each layer controls the density of the output prediction. We use dense Conditional Random Field (CRF) [43] to predict labels for every pixel. CRF uses energy

$$E(x|I) = \sum_i \psi_i(x_i) + \sum_{i<j} \psi_{ij}(x_i, x_j) \quad (1)$$

where ψ_i is the unary energy (negative log of the aggregated softmax probabilities p_i) and $\psi_{ij}(x_i, x_j) = w_p \sigma(x_i \neq x_j) k(f_i - f_j)$ is the pairwise term that connects every pair of pixels (x_i, x_j) in the image. We use a single pairwise term with a Potts label compatibility term σ weighted by w_p and unit Gaussian kernel k. For a feature f_i, we convert the RGB image to $L^*a^*b^*$ color space and use color (I_i^L, I_i^a, I_i^b) and position (p^x, p^y) as pairwise features for each pixel, yielding $f_i = [\frac{p_i^x}{\theta_p d}, \frac{p_i^y}{\theta_p d}, \frac{I_i^L}{\theta_L}, \frac{I_i^a}{\theta_{ab}}, \frac{I_i^b}{\theta_{ab}}]$, where d is the smaller image dimension and θ values are the CRF model parameters.

4.1.2 Street View Material (SVM) Dataset Although the problem of material identification has been studied in the context of the indoor and outdoor environment in prior works such as [20, 67], the datasets used in these works are not directly useful in mmSV. This is because prior datasets such as the MINC dataset [20] include a range of materials such as food, carpet, hair, etc. that are not useful for SV images. Similarly, these datasets lack various materials such as vinyl and stucco that are commonly used construction materials [24] found in outdoor building images.

Material	# of patches	Material	# of patches
Asphalt	47013	Stone	5365
Sky	48214	Metal	4234
Foliage	15879	Tile	4152
Vinyl	6562	Wood	3356
Stucco	5894	Polished Stone	2920
Brick	5774		
Glass/mirror	8060	Ceramic	2500

Table 1: Number of patches in our SVM dataset

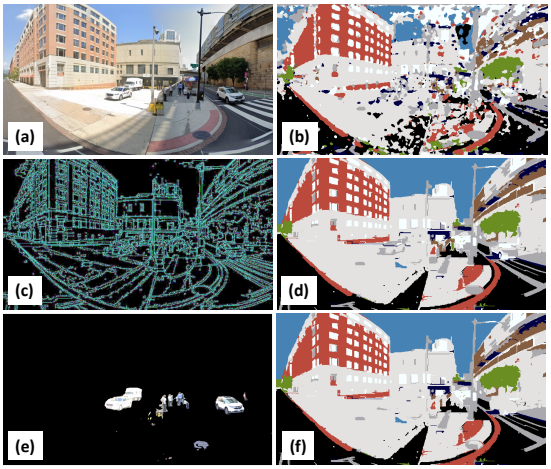


Fig. 4: Material identification process: (a) original SV image, (b) material classification with sloppy boundaries, (c) superpixels, (d) output of over-segmentation, (e) dynamic object masks, (f) final predicted material SV image with dynamic objects removed.

To address this challenge, we develop the SVM dataset using 1200 street view images (1000 for training, 200 for evaluation) collected from various online SV image platforms such as Google Street View [8] and Bing maps [11]. The street view images are selected randomly from urban areas such as New York, San Francisco, and Chicago. We manually label the patches of size 256×256 in these images to create a total of 132, 423 new patches in the SVM dataset. We annotate the street view images in two ways. First, we manually identify segments that are carefully drawn polygons containing the same material. Second, we also identify clicks which are single points in an image with their corresponding material label. Examples of new extracted patches are shown in Fig. 3. The dataset includes 13 material categories found in these images as listed in Table 1. For materials that are already included in the indoor material MINC dataset, we augment our dataset with its patches to further improve the amount of data. Overall, we include 17.3% patches for some of the materials already included in the MINC-2500 dataset. To the best of our knowledge, the SVM dataset is a first-of-its-kind material dataset for SV images and the dataset can be useful for a range of applications in wireless research including channel estimation, propagation modeling, etc.

SVM Material Classification. In order to reduce the processing time, our model uses SV images that are cropped from the top and bottom. We find that the top 25% of all SV

images used in our study do not have any objects (i.e., sky) and cannot be used for reflection. Similarly, the bottom 25% of the images are asphalt. While prior works [82] have shown the possibility of ground reflection, these reflections are not feasible in presence of a blocking vehicle. This means that mmSV limits the SV images using $-45°$ to $+45°$ elevation angles from the LoS ray connecting the Tx and Rx.

In order to further increase the density of the output predictions, we perform grid prediction at three different scales ($d/\sqrt{2}$, d and $d\sqrt{2}$ where d is the smaller dimension of the image). We then upsample the output maps using bilinear interpolation and average the three predictions. Since the dense CRF is computationally expensive, we upsample the output to a fixed size of 360×1440. Since the FoV of SV images covers $360°$ horizontally and $90°$ vertically, dividing the width and height of the CRF output by these angles results in 4×4 pixel prediction for each angle in the image.

4.1.3 Over-segmentation Although the above material identification does yield an approximate segmentation, it has two issues: (i) pixel misclassification causes noise in the output material map and (ii) as reported in previous works [20], the output shows sloppy boundaries when the model is trained on clicks which are single points in an image with their corresponding material label. Even though our model is trained on patches extracted from both segments and clicks, it encounters the same issue (Fig. 4b). The idea of over-segmentation is to partition the image into pixel groups such that the pixels in the same group share common characteristics, leading to a prediction of them having the same material as a group. The process removes outliers (a small number of pixels misclassified within otherwise correctly classified pixels). We leverage the idea of image segmentation from [31] to detect superpixels for the purpose of over-segmentation. Here, a superpixel is a group of pixels that belong to the same surface in an image and are assumed to correspond to the same material. We use a greedy algorithm presented in [31] to obtain the superpixels. We note that while the superpixel detection is performed using an existing method, its integration and application for pixel-by-pixel material classification is novel and unique to our work. An example of extracted superpixels is shown in Fig. 4c, and Fig. 4d shows the result of applying the over-segmentation.

4.1.4 Dynamic Objects Removal The street view images usually contain dynamic objects such as vehicles, humans, etc. We remove these objects from the images to only retain stationary objects such as buildings, etc. We use an existing model DeeplabV3 [25] to perform semantic segmentation on the SV images. The model is pre-trained using the cityscapes dataset [27]. The model detects the pixels associated with dynamic objects (shown in Fig. 4e). The detected pixels result in a mask that we use as the areas to be removed from the material identification results of the previous step. We use the method in [73] to remove the objects. Fig. 4f shows the results of this method with dynamic objects removed from the SV images.

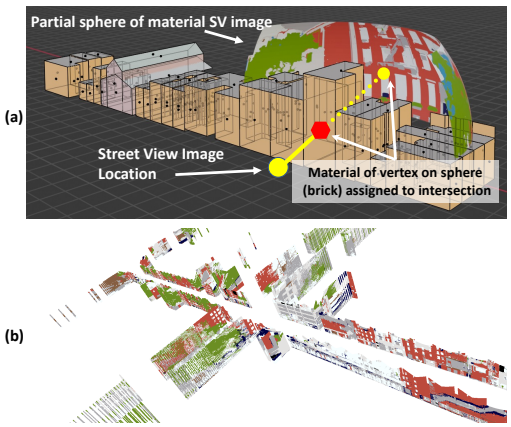


Fig. 5: (a) Projection of material map on 3D buildings, (b) Example of final generated 3D material map.

4.2 Creating 3D Material Map

To the best of our knowledge, no prior work has demonstrated creating a 3D material map using publicly available maps and SV images. To create 3D material maps, mmSV combines the results of material identification with 3D building geometries available from OpenStreetMap [56]. The process involves two steps.

(1) Material projection and point cloud creation. The first step is to combine the image data with 3D shapes of the buildings to project the material prediction pixels onto the 3D building surfaces and find the location of each pixel in 3D space to create a point cloud. In order to do this, given a material image, we form a partial sphere (partial because the material identification only covers $90°$ elevation) with a fixed radius as the initial position of material image pixels (shown in Fig 5a). Now, a ray emitting from the origin of the SV image to a pixel in the SV image sphere is calculated and the intersection of the ray with a surface of a building is determined. The intersection point is assigned the same material as the pixel in the material SV image sphere. The process is repeated for all pixels in the material SV image and for all SV images of the road segment to create a point cloud of the building surface with materials.

(2) Plane detection and final 3D material model generation. The final step is to convert the point cloud to its corresponding surfaces and create a unified 3D material model of the road segment that can be used for ray tracing. Given the material point cloud from the previous step, we run the RANSAC [32] algorithm to find the surfaces (planes) that fit each part of the point cloud. Each surface is then split into square faces of size $10\,cm \times 10\,cm$. Based on the material identification result for each point, we assign the material of the face to be the one that is most commonly predicted for points in that face. Finally, we merge the neighboring faces with the same material to create surfaces through an

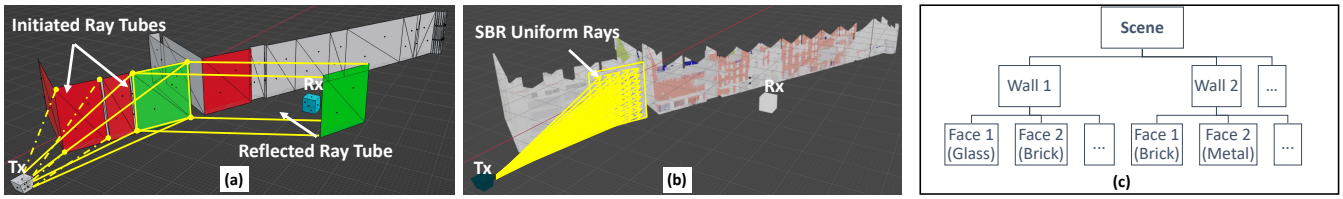


Fig. 6: Environment-driven raytracing: (a) initiated and reflected ray tubes, (b) SBR within ray tube of interest towards a wall with material faces, (c) tree data structure used for modeling our environment

iterative process. An example of the final 3D material map generated through the process is shown in Fig. 5b.

5 V2V Comm. using 3D Material Maps

Given the 3D material map of a given set of locations or a road segment, we now address two outstanding questions: (1) How can a Tx vehicle leverage the 3D material map to determine the available mmWave paths to the Rx vehicle? We develop an **environment-driven raytracing** process that focuses the rays based on the 3D material maps to efficiently and accurately determine the AoD of mmWave paths. (2) How can the Tx vehicle select a small subset of beams to probe such that a high-SNR beam to the Rx vehicle can be found? We devise a **restricted beamsearching** method that uses the raytracing output to maintain uninterrupted connectivity with the Rx vehicle by finding a high-SNR beam.

5.1 Environment-driven Raytracing

A commonly used method for determining signal propagation paths is shoot-and-bounce raytracing (SBR [50]). In SBR, rays are launched from the Tx in all directions with uniform ray density. The rays can then be traced in space and their behavior (reflection, diffraction, scattering, etc.) can be determined when they interact with different environment objects. A key drawback of SBR is that although it launches rays in all directions, it is possible that only a small subset of rays actually contribute to the received signal at the receiver. This is because SBR in its basic form is oblivious to the environment and presence of objects. This is the limitation that mmSV tries to remove as it has detailed information about the environment. We develop an environment-driven raytracing whose basic idea is to launch rays towards environment objects such that there is a higher probability of them contributing to the received signal strength at the Rx.

mmSV incorporates the 3D material map into the raytracing by developing a two-phase hierarchical raytracing process. In the first phase, it performs macro-scale path-tracing using frustum raytracing. The purpose of this phase is to find objects that can have a high contribution to the received signal if rays are reflected from them (i.e., objects and surfaces that can provide a reflected path with a high probability). In this phase, mmSV considers walls as primitives as shown in Fig. 6a (a plane with two triangles without considering the material) and shoots ray tubes towards the walls. Ray tubes are pyramids whose apexes correspond with the location of the Tx antenna and their base face coincides with one of the walls as shown in Fig 6a. The main benefit of this approach is the fact that mmSV only shoots a limited number of rays in specific directions (toward the walls) and only needs to trace the corners of the tubes which reduces the search space and the processing time. In the second phase, given the ray tubes from reflecting walls from the first phase, SBR is performed with uniform ray density only within those ray tubes. This phase considers the faces of different materials within the selected walls, uses them to determine the reflection loss, and ultimately identify the mmWave paths, their AoD, and RSS. Phase 1 of ray tracing follows the following steps.

Step 1. Launch the initial set of ray tubes from the Tx toward the four corners of planes (walls) as shown in Fig. 6a.

Step 2. Determine if the ray tube intersects with the corresponding wall. The intersection is calculated between each ray of the ray tube and the wall edges. To determine if a ray and edge intersect, we use the Plücker coordinate representation [58]. Plücker coordinates specify rays in three-dimensional space using six-dimensional vectors. Given a ray r with its origin point P and direction L, its Plücker coordinates are given by: $\pi_r = \{L : L \times P\} = \{U_r : V_r\}$ One feature of Plücker coordinates is that given two rays r and s, the permuted inner product represented by $\pi_r \odot \pi_s = U_r \cdot V_s + U_s \cdot V_r$ specifies their relative orientation. In general, if $\pi_r \odot \pi_s > 0$, s goes counterclockwise around r. If $\pi_r \odot \pi_s < 0$, s goes clockwise around r, and if $\pi_r \odot \pi_s = 0$, s intersects or is parallel to r. This process of using Plücker coordinates to determine intersection is repeated for all edges of the wall and all rays of a ray tube. This determines if the ray tube completely intersects, partially intersects, or misses the wall.

Step 3. If there is an intersection, generate a reflected ray tube based on the intersection state. If there is a complete intersection, the reflected ray tube can be calculated using the wall's normal and angle of incidence. For a ray tube with four corners, if r is a corner ray, this can be done by calculating $r^f = r - 2(r.n)n$ where r^f is the reflected corner ray of r, and n is the normal of the wall. Fig. 6a shows a reflected ray tube. If the ray tube partially intersects the wall, it is clipped and only the parts of the ray tube that impinge the wall will be used to calculate the reflected ray tube.

Step 4. If there is no intersection, check if the Rx is within the reflected ray tube. If yes, add the initial wall to set $\mathbb{W}$. If no, discard the ray tube.

Step 5. Repeat Steps 2-4 for each initial ray tube until the maximum number of bounces has been reached.

The result of Phase 1 is set $\mathbb{W}$ which is a set of walls that can provide reflections and will be used along with SBR in Phase 2. In Phase 2, the Tx initiates an SBR with a uniform

density only for walls in $\mathbb{W}$ (Fig. 6b). As shown in Fig. 6c, each wall in $\mathbb{W}$ is divided into faces where each face has a material. The SBR results in a set of paths $P = \{p_1, p_2, ...\}$ where each p_i includes the angle of departure (ϕ_i, θ_i), distance d_i and the faces $F_i = \{f_1, f_2, ...\}$ it goes through to reach the receiver. Each $f_j = \{AoI_j, m_j\}$ where AoI_j and m_j are the corresponding angles of incident and material for that face. The RSS of each path p_i can be calculated as $RSS_{p_i} = P_T - L_P(d_i) - \Sigma_{j=1}^{n} L_R(f_j)$ where P_T is the transmit power, L_P is the free space path loss, and L_R is the reflection loss for faces f_j. L_R depends on the complex material permittivity and signal incident angle. 60 GHz mmWave reflection loss for various materials has been extensively studied in [47, 48]. We validate the measurement results using our mmWave testbed as well for 50 different locations across a university campus (we do not include this result due to space limitation). These reflection loss values for material and angle are stored as a lookup table that we refer to calculate the RSS for each path.

5.2 Restricted Beamsearching

If the Tx vehicle knows the AoD of mmWave paths estimated from the environment-driven raytracing, it can use them to directly determine the corresponding Tx beams(s) to use. However, such as zero-shot beamforming is not feasible in our scenario due to two reasons. First, the LoS path and the reflected paths determined from 3D material maps can be blocked by dynamic objects in real-time. This includes blockages from objects like other vehicles or pedestrians. Second, the AoDs calculated from the 3D material maps can have errors. This can be due to the inaccuracy in OSM building information and street view images as well as Tx and Rx location errors. Due to these reasons, we propose to employ restricted beamsearching (instead of zero-shot beamforming) that searches a small subset of beams that have high gain in the direction of estimated paths.

5.2.1 Determining beams to probe Tx should account for three sources of errors: (i) its own GPS location error, (ii) material map error due to inaccuracies in OSM 3D maps, and (iii) error in calculated Rx location. We use prior empirically known values [36, 49] for the first two and our measured mean prediction error for Rx location to calculate the AoD ranges that can then be used for beam selection. Using the range of AoDs determined, mmSV selects a subset of beams that can cover those angles. Specifically, a beam that achieves the highest gain for any angle within a given AoD range is added to that path's subset of beams to be probed. This subset of beams is then probed by the Tx vehicle to determine any dynamic blockages and the best beam (highest SNR) is chosen for communication.

5.2.2 Integrated Restricted Beamsearching and Localization using 3D Maps Our environment-driven ray tracing described above assumed that the Tx vehicle knows the location of the Rx vehicle. Since we are primarily interested in Tx beamforming, the Tx needs to know the relative Rx location. While it is possible for the Rx vehicle to share its GPS location, such communication requires an out-of-band channel (e.g., sub-6 GHz) which is not always feasible in practice. Hence, mmSV relies on in-band mmWave-based localization that leverages our previously developed 3D maps while building on restricted ray tracing and beamsearching.

Given the inter-dependency between Rx localization and environment-driven restricted beamsearching (beamsearching and raytracing require Rx location while Rx localization itself is estimated using restricted beamsearching based channel measurements), mmSV develops an integrated process for both where the Rx localization is performed after each restricted beamforming. The restricted beamforming can be triggered every time instance whenever there is a change in SNR for the current beam. The SNR change can be a result of Rx mobility relative to Tx or blockage (e.g., by another vehicle) of their current mmWave path. A new beam needs to be determined in both cases through the use of restricted beamsearching.

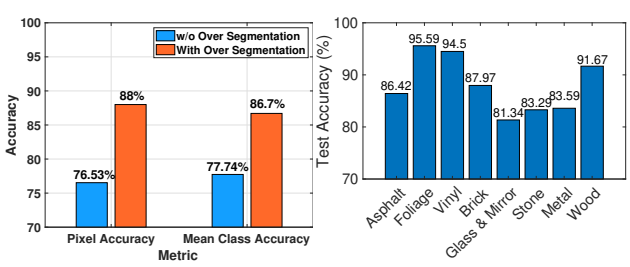
Step 1: AoD range estimation through environment-driven raytracing. At each time step t_n, Tx uses its own location (available through GPS) and Rx location known from the previous time step t_{n-1}. Here, the time steps refer to consecutive time instances when Tx performs beamsearching. Tx uses the environment-driven raytracing process discussed in Sec. 5.1 to determine the AoD ranges from Tx to Rx.

Step 2: Restricted beamsearching and CIR measurements. Based on the AoD ranges estimated in the previous step, Tx identifies the beams with the highest gains in those directions using the process discussed in Sec. 5.2. Tx probes this subset of beams and identifies the one with the highest SNR for communication with Rx at time t_n. It also collects the CIR measurements and calculates the power delay profile (PDP) for each probed beam for Rx localization. The PDP can be extracted from the Short Training Field (STF) and Channel Estimation Field (CEF) at the beginning of each 802.11ad frame. We use the method in [30] to extract the PDP for each beam. The CIR and PDP measurements are fed back to the Tx from Rx and the feedback is counted towards the beamforming overhead calculation.

Step 3: Rx localization. The Tx separately estimates the location of Rx using different path directions by correlating the corresponding path amplitude with the beam patterns. We follow the approach presented in [57]. The main idea here is to probe a set of beams and then correlate the beam gain pattern for each beam with the measured SNR response of that beam to identify the AoD. The AoD then combined with the ToF measurement can be used to localize the Rx. A key difference here is that [57] approach probed all beams while mmSV only probes a small subset of beams. However, these small subsets of beams are selected based on the estimated

Material	AlexNet	mmSV	Material	AlexNet	mmSV
Asphalt	99.9	99.9	Stone	74.8	83.6
Sky	99.8	99.8	Metal	69.6	85.1
Foliage	97.7	97.9	Tile	71.8	86.5
Vinyl	97.3	99.5	Wood	74.4	83.9
Stucco	96.9	98.3	Polished Stone	70.2	81.5
Brick	87.2	92.2			
Glass	83.1	87	Ceramic	76.8	82.8
Mean class accuracy		**AlexNet: 84.57**		**mmSV: 90.62**	

Table 2: Material classification accuracy (%)



(a) (b)

Fig. 7: Full scene material segmentation accuracy (a) with and w/o over-segmentation, (b) by material.

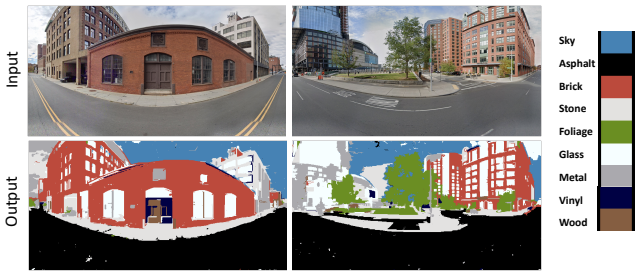


Fig. 8: Examples of original street view images and corresponding material images created by mmSV.

AoDs, yielding a comparable performance. Also, localization in [57] is limited to the LoS path, but with the help of 3D material maps, we are able to extend the same approach to NLoS paths as well. For the distance measurement, we use the Fine Time Measurement (FTM) ranging procedure described in IEEE 802.11mc [2] to get ToF measurement from FTM-capable 802.11ad devices in our testbed. We assume TX and RX are not aware of each other's location at the initial timestep and they perform exhaustive 802.11ad-type beamforming in the first step to determine it. This is only needed during the first timestep.

After independently estimating the Rx location using each available path (LoS or NLoS), mmSV calculates a geometric median of all estimated locations using Weiszfeld's algorithm [79] as the predicted Rx location. The Rx location estimated at time t_n will now be used at the next beamsearching at t_{n+1} starting from Step 1 above.

6 Evaluation

We evaluate mmSV in two parts. First, we understand the performance of our street view material identification module using our SVM dataset and a number of additional street view images. Second, we use our 60 GHz mmWave software radio testbed along with 802.11ad commercial devices to evaluate the reflections over generated 3D material maps, restricted raytracing and beamsearching, and end-to-end throughput and delay.

6.1 Street View Material Identification

6.1.1 Patch classification We compare our street view material identification model with an existing well-known model (AlexNet [45]). We use our SVM dataset with 160K patches with material labels for the purpose. Here, we use 85% of the patches for training, 5% for validation, and 10% for testing. For both models, we use stochastic gradient descent with a batch size of 64, momentum of 0.9, an initial learning rate of 10^{-2}, and weight decay of 10^{-4}. The learning rate (lr) updates as $lr = lr \times 0.1^{\frac{epoch\#-1}{40}}$ every epcoch.

As discussed in Section 4, the choice of patch scale has a significant impact on the accuracy. If a larger scale is used, the patch includes more context but the spatial resolution decreases. To determine the patch scale, we evaluate our model with different patch scale values (6.25%, 12.5%, and 25%) and we find that the highest accuracy is achieved at 6.25%. Our results show that a larger patch scale only works better in classifying materials such as glass, sky, and brick. For the glass category, the patch is misclassified due to visible light reflection, resulting in a mirror image of the object/material opposite to the glass surface. The same is true when the glass is see-through and the material behind the glass is detected as the material instead of glass. When a larger patch scale is used, the inclusion of the frame around the glass windows, walls, or doors helps in better classifying them as glass as opposed to material in reflection or behind the glass. On the other hand, a smaller patch scale better detects materials that have a grainy texture (e.g., concrete).

Table 2 shows the patch classification accuracy for different materials for our material identification model and AlexNet. The material categories such as asphalt, sky, foliage, vinyl, stucco, etc. have high accuracy since they are more commonly found in typical street view images in urban settings, resulting in more data in terms of their number of patches. In contrast, materials such as wood, polished stone, and ceramic are relatively less common. Even with such an imbalance of the data, our model can provide a mean accuracy of 90.62%, significantly outperforming state-of-the-art models such as AlexNet which achieves 84.57% accuracy.

6.1.2 Full scene material segmentation We now evaluate pixel-by-pixel material classification performance on street-view images. In order to do so, we collect additional 200 street-view images to create a test and validation image set. These images are randomly chosen from urban areas throughout the United States. We manually annotate the 200 images using carefully drawn polygon and their material labels. We split the image set into test (80%) and validation set (20%). We evaluate 2430 combinations of CRF

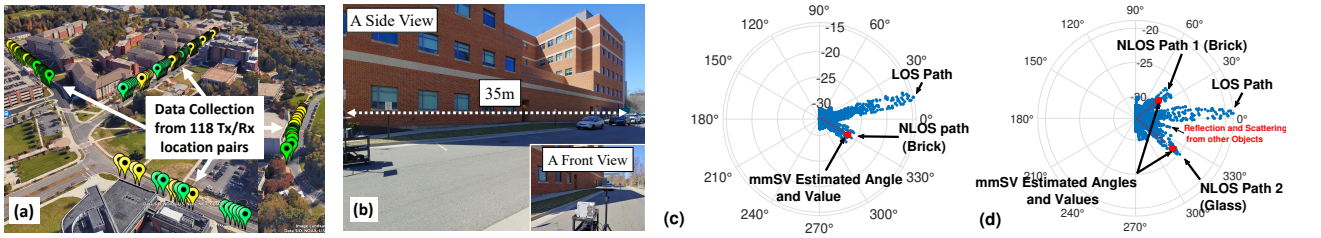


Fig. 9: (a-b) 3D material map measurements, (c-d) Measured ground truth and predicted power angular profile.

parameters ($\theta_p, \theta_{ab}, \theta_L, w_p$) and over-segmentation parameters ($cluster\_size, \sigma, minimum\_segment\_size$) on validation set and find the best-performing combination to use.

Fig. 7a shows the pixel classification accuracy (number of pixels for which the material is correctly classified or total number of pixels) and mean material class accuracy with and without over-segmentation. Fig. 7b shows material class accuracy for each material. The results clearly show the impact that over-segmentation has (improvement of 8.96%) on material identification. Fig. 8 shows examples of how existing street images are converted to their corresponding material images using our model.

6.2 mmWave Networking with 3D Material Maps

6.2.1 Testbed and experiment setup We consider two types of 60 GHz mmWave devices in our evaluation (Fig. 10a): (i) a software radio system that consists of a phased-array based RF frontend from Sivers Semiconductors [13] and USRP as baseband processor, and (ii) Commercial off-the-shelf 802.11ad AP from MikroTik [12]. The Sivers 60 GHz RF frontend has two phased antenna arrays (one for sending and the other one for receiving) with 64 antenna elements each. We use the default codebook from Sivers with 63 beams where the mainlobe angle of every beam is approximately every 1.5° in Azimuth. We perform a separate measurement campaign to measure the beam patterns of each beam in the codebook. Here, we use another horn antenna based 60 GHz system (VubIQ [14]) to measure the gain of every beam of Sivers while rotating it on a mechanical rotator. We use Sivers+USRP as Tx and Rx for our 3D map evaluation including power angular profile, reflection path assessment, and loss prediction as it provides better control over beams (i.e., fewer sidelobes, finer granularity of angles, etc.).

Apart from the software radio system, we also use MikroTik 802.11ad AP that has three antenna array patches of 32 elements and 64 beams as Tx. Here, we use Sivers RF frontend with an Oscilloscope (Keysight MSO9404A) to receive the data. This setup enables us to measure the 802.11ad beacons, beamforming frames along with PHY preambles to measure CIR and calculate PDP and ToF, along with SNR. We use the MikroTik Tx and Sivers+Oscilloscope Rx for restricted beam-searching evaluation including localization and throughput.

Outdoor experiments. We perform our evaluation through outdoor measurements using our mmWave testbed on four road segments on a university campus. As shown in Fig. 10b, the four road segments are 350 m, 120 m, 167 m and 85 m long, respectively. The roads are surrounded by buildings (2-5 floors) resembling a typical urban scenario. The buildings include different types of materials including brick, concrete, glass, stone, and metal. We use publicly available street view images from Google for the four road segments.

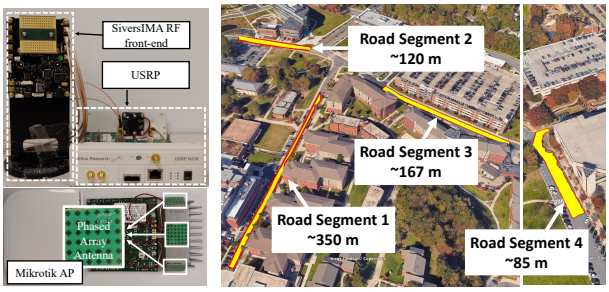
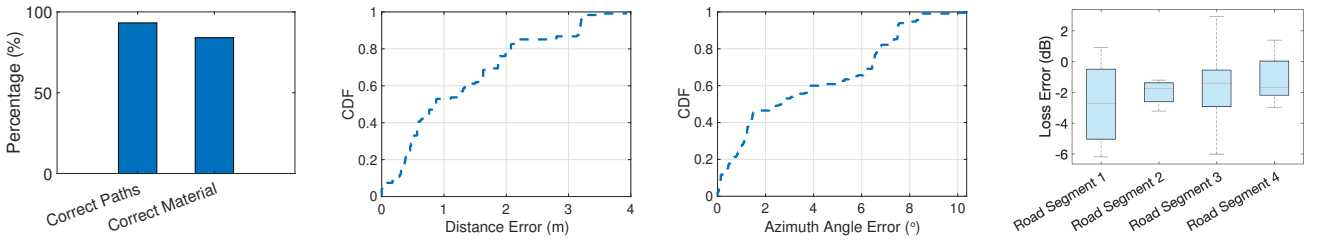


Fig. 10: (a) Our mmWave testbed and (b) Four road segments used in evaluation.

6.2.2 3D material map In this section, we evaluate how accurately our model reconstructs the 3D material map from the SV images. We also evaluate what are the errors when we use our constructed 3D material map to detect mmWave reflection paths. In order to do these, we collect mmWave channel data at 118 Tx-Rx location pairs in the four road segments as shown in Fig. 9a. These channel measurements include AoA and RSSI for different mmWave paths (LoS and paths reflecting from buildings) for the Tx and Rx locations. We use our Sivers software radio system to perform exhaustive beam scanning on both Tx and Rx at each location to measure all possible mmWave paths in the environment. These measurements act as ground truth to our predicted reflection loss based on 3D material maps. We also collect the actual physical distance between Tx, Rx, and buildings. Fig. 9b shows our mmWave data collection setup on tripods.

Power angular profile. For each of the road segments, we use publicly available street view images from Google and reconstruct their 3D material map using the process described in Section 4. We then use our environment-driven raytracing to identify the mmWave paths, their AoA/AoD, and RSSI. Figs. 9c and 9d show the power angular profile for two receive locations based on our measurements and 3D material map-based prediction. We find that mmSV correctly captures the angle and RSSI for LoS and reflected paths (from brick and glass walls in Figs. 9c and 9d). We note that when the reflective object is not part of our 3D building material map (e.g., a car passing by), its corresponding reflection is not predicted by our model as expected. Fig. 9d shows such



(a) Path and material detection **(b) Distance estimation error** **(c) Azimuth angle error** **(d) Reflection loss estimation error**

Fig. 11: 3D material map evaluation.

an example. We find that such reflections are short-lived as well as they provide very small RSS.

Number of reflected paths and their material. We further compare the ground truth and output of our environment-driven raytracing. Fig. 11a shows the number of mmWave paths correctly detected through our model compared to the ground truth measurements. We find that mmSV detects on average 93.2% paths correctly. We find that apart from the paths created by dynamic objects such as cars, errors in OSM 3D building information lead to missing paths in our prediction, especially on the edge of the buildings. In terms of the material of the reflected paths, we find that 84.07% of the material is detected correctly by our model.

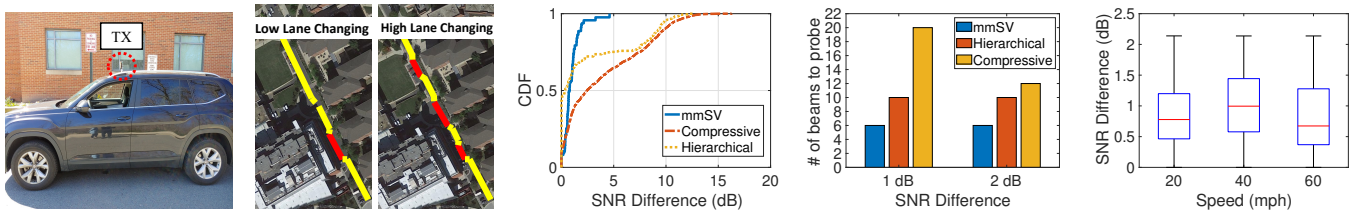
Distance, angle and loss prediction. We now compare the distance error which is the distance difference between the measured distance for a signal path compared to the predicted distance in our model. Fig. 11b shows the CDF of distance error. We find that the mean distance error is 1.27 m. Similarly, Fig. 11c shows the CDF of the angle difference between the ground truth and predicted NLoS paths from our raytracer. We find that the mean (azimuth) angle error is 3.53°. Both distance and angle errors show that our reconstructed 3D material map closely matches reality. The inaccuracy in OSM 3D building models is the primary source of these errors. Lastly, Fig. 11d compares the predicted and measured reflection loss. We find the measured reflection loss by subtracting the distance-based path loss and Tx and Rx antenna gains from the received power. The predicted reflection loss is based on the predicted material and signal incident angle. We find that the predicted loss is on average 2.74 dB, 1.88 dB, 1.66 dB and 1.14 dB lower than the ground truth for the four road segments, respectively. The difference is due to the complexity of the two road segments and the materials used as well as the OSM building information error. We note that the first and third road segments are longer and have more diversity in terms of building materials, resulting in slightly higher angle estimation errors compared to the other two road segments. The maximum error of 6 dB happens due to the angle prediction error in which mmSV predicts the path to reflect from a brick wall instead of glass.

Raytracing and beamsearching time. Our environment-driven raytracing should run in real-time to support the beamforming. Our results on two road segments show that compared to SBR, our method achieves a dramatic reduction in the number of rays to trace. We find that just knowing the presence of objects and focusing the rays on specific walls as per probability of reflection reduces the number of rays from $\approx 1.1\,M$ in SBR to $\approx 2.1\,K$ in mmSV (close to 99× reduction). In terms of time, when simulating this on a quad-core Intel CPU, mmSV takes 0.4 ms for its raytracing. On the latest GPUs equipped with dedicated ray-tracing cores with the capability to process tens of millions of rays per second, our raytracing can be done within microseconds. The localization method used in the work increases the CPU utilization only by $\approx$ 2% on the same CPU. The average number of beams to search at each time step in mmSV is 6 (Fig. 12d) which takes 0.17 ms to probe and get feedback as per 802.11ad SSW. This means that mmSV takes on average 0.57 ms for raytracing and beamsearching on CPU (0.17 ms on GPU) which can be implemented in real-time on vehicles even with moderate compute capabilities.

6.2.3 End-to-end evaluation on vehicle trajectories

We now evaluate how mmSV's performance in terms of restricted beamsearching, Rx localization, link layer throughput, and beamforming overhead. We collect mmWave measurement data at additional 81 Tx-Rx location pairs on road segment 1. The locations are selected on both lanes of the road to understand the impact of realistic traffic characteristics such as lane changes. We use the 81 locations to create 5 vehicle trajectories. In two of the vehicle trajectories, the vehicles use low lane changing behavior (1 lane change per minute) while in the other three trajectories, the vehicles exhibit high lane changing behavior (3 lane changes per minute). Here, we use the Mikrotik 802.11ad APs as Tx and Sivers software radios as Rx. This setup enables us to collect CIR, PDP, ToF, and SNR as the 802.11ad AP scans different beams depending on the scheme. Figs. 12a and 12b show our experiment setup and 2 sample trajectories, respectively.

Restricted beamsearching. We use the above-mentioned experiment setup with vehicle trajectories to evaluate mmSV's performance in terms of predicting the correct Tx beams to probe and localizing the Rx based on those beam measurements. The subset of beams is derived based on the 3D material maps created from street-view images and environment-driven raytracing. We compare mmSV's restricted beamsearching with (i) hierarchical beamforming where Tx utilizes a multi-level codebook and probes the beams with different beamwidths at different levels iteratively. We choose

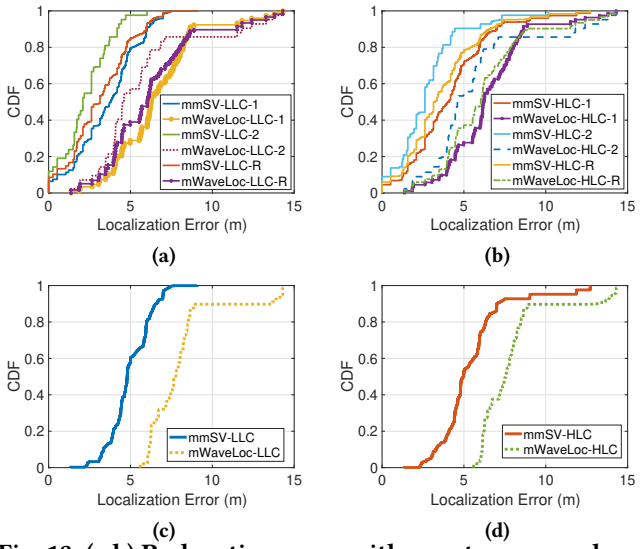


(a) (b) (c) (d) (e)

Fig. 12: (a) Data collection using vehicle-mounted mmWave radios (b) lane changing behavior(c) SNR difference from the highest SNR beam for mmSV, compressive and hierarchical beamforming, (d) Number of beams to probe to achieve a given SNR difference from the highest SNR beam, and (e) SNR difference of mmSV beamforming at difference mean vehicle speeds.

this two-level approach as it is considered for 5G NR initial access in [28, 33, 39, 78]. In our implementation on 802.11ad MikroTik, the mean half-power beamwidth of the first and second levels is 42° and 15°, respectively, (ii) compressive beamforming [68] where the Tx probes a randomly chosen subset of beams (in the order of $log(n)$ where n is the number of beams) to determine the AoD and selects the beam with the maximum gain in AoD direction, and (iii) exhaustive beamsearching (as in 802.11ad) where all Tx beams are probed to determine the best beam.

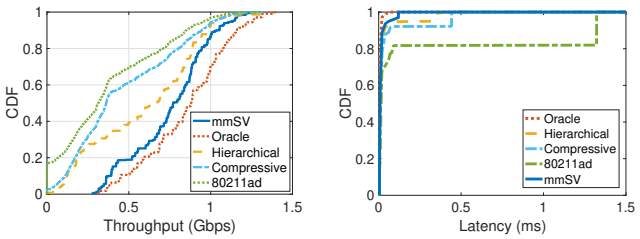
SNR. Fig. 12c shows the SNR difference CDF for mmSV, hierarchical and compressive beamforming compared to the best (highest SNR) beam found through the exhaustive search. For a fair comparison, we probe an equal number of beams for both mmSV and compressive beamsearching. We find that the average SNR difference for mmSV is 0.96 dB. mmSV performs better than both the hierarchical and the compressive beamsearching where the average SNR difference is observed to be 2.65 dB and 4.06 dB, respectively. We note that hierarchical beamforming performs better than compressive beamforming and sometimes even mmSV. However, it suffers from misidentification of the best beam often when the reflected paths do not have high SNR when probed using wider beamwidth beams. Fig. 12d shows the minimum number of beams that need to be probed by all three methods in order to find a beam that has 1 dB and 2 dB difference from the highest SNR beam. mmSV significantly reduces the number of beams to search primarily due to its knowledge of the environment and environment-driven raytracing that directly operates on 3D material maps. On the other hand, since initial levels of hierarchical beamforming often do not provide the required SNR, it probes the same number of beams in both cases to get to high-gain beams at the lowest level. Fig. 12e shows the performance of mmSV in terms of SNR difference for three different mean vehicle speeds (20, 40 and 60 miles per hour). We use the collected measurements to create vehicle trajectories at different speeds and evaluate their impact on mmSV's beamforming. Since mmSV takes as low as 0.17 *ms* for beamsearching, there is no conceivable impact in real-time as the vehicle speeds increase and SNR difference at different vehicle speeds remains comparable.



(a) (b) (c) (d)

Fig. 13: (a,b) Rx location error with one, two or random (R) number of available paths; (c,d) Rx location error with NLoS paths only. LLC and HLC stand for Low and High Lane Changing, respectively.

Rx location. We now evaluate the performance of Rx localization from Tx when it uses mmSV's restricted beamsearching in the presence of blockages. We compare the performance with a state-of-the-art mmWave localization scheme mWaveLoc proposed in [57]. mWaveLoc scans all 64 beams to measure the CIR and perform the localization. On the contrary, mmSV only probes a subset of beams (on average 6, max. 9) as part of the restricted beamsearching. Figs. 13a and c show the location error for both methods. We note that since this evaluation uses vehicle trajectories, it includes LoS as well as NLoS paths available between the Tx and Rx. To demonstrate the impact of blockage, we include the results for different scenarios: (1) only one path is available (not blocked) and all remaining paths are blocked, (2) two paths are available, and (3) a more realistic (R) scenario where the number of available paths is random. Furthermore, Figs. 13c and d show the location error only for NLoS situations. We find that mWaveLoc performs significantly poorly in NLoS situations as it is primarily designed for LoS path. On the other hand, mmSV, with its knowledge of the environment, can distinguish between the LoS and NLoS paths, and use the reflection path information (such as angle



(a) (b)

Fig. 14: Comparison of (a) link throughput and (b) link latency including the beamforming overhead.

and distance) to better estimate the Rx. Since mmSV's multi-direction localization accuracy depends on the number of available paths, we observe a decrease in accuracy when only one path is available compared to two or more. Furthermore, we observe a slight increase in the location error in scenarios with high lane changing behavior due to the dependency of mmSV's localization on the location in the previous step.

Throughput and latency. We now evaluate the link-layer throughput and latency overhead of beamforming for the vehicle trajectories. We compare mmSV with (i) compressive beamforming, (ii) hierarchical beamforming, (iii) 802.11ad which probes all beams, and (iv) an Oracle scheme that selects the best beam without any overhead. Fig. 14a shows the throughput comparison. The throughput is calculated for every beacon interval (BI) over the trajectory. It is affected by two factors: current SNR and the amount of time spent in beamforming, both of which differ based on the scheme. mmSV achieves a better throughput compared to compressive beamsearching. This is due to two reasons. First, as shown earlier, mmSV is able to find a beam with better SNR due to its knowledge of the environment. Second, mmSV does so while probing fewer beams, leaving more time for data communication. Compared to hierarchical beamforming which can result in misidentification of the best beam as mentioned earlier, the advantage mostly comes from mmSV's ability to select high SNR beams. While probing one sector can take no more than 20 μs, vehicular channels require frequent beamsearching due to relative mobility. In such a case, a significant amount of time can be wasted on beamforming when mmSV is not used. Fig. 14b shows the measured link layer latency. Here, latency is calculated as the amount of time a frame has to be buffered at the transmitter while the link is established through beamforming. The latency includes packet loss and the beamforming overhead, i.e., the time taken to perform beamforming and establish a link in the presence of vehicle mobility. The results clearly show a reduction in latency for mmSV compared to other schemes due to fewer beams probed to find a high-SNR beam.

7 Related Work

While a range of beamforming approaches [46] such as hierarchical beamforming [17, 38, 71], compressive sensing [54, 60–62], PDP based [70] and with co-phasing [40, 77] have been proposed, their primary focused has been on indoor non-vehicular channels. Performance of mmWave networking in the vehicular environment is investigated through experiments and simulations in [63], [51] and [76]. [51] showed that leveraging road geometry can help reduce the beamforming overhead. Similarly, authors in [76] propose a pruned codebook to limit the selected beams to the road geometry to achieve high coverage with low overhead. These works focus on V2I communication where blockages are shown to be infrequent [76]. On the other hand, we show that V2V channels are highly dynamic with frequent blockages due to other vehicles. Also, limiting beam search to road geometry is not effective in V2V as many of the reflections are available through the surrounding environments.

Numerous studies explore out-of-band data to improve vehicular mmWave communication. Techniques include the use of DSRC [34], radars [16, 35], LiDAR [29], and cameras [65] on BS, RSU, or elsewhere. [66] uses LiDAR, GPS and camera images from each vehicle. In [64], vehicles broadcast GPS and type, combined with BS camera data, lowering beamsearching overhead. Compared to mmSV, these works require additional sensors on vehicles or base stations to collect out-of-band information. On the other hand, mmSV relies on already publicly available street view images and does not require any real-time vision data collection. Recently, authors in [80] proposed the use of a LIDAR sensor for scanning the indoor environment and using it to perform reduced overhead raytracing and mmWave beamforming. In comparison, mmSV focuses on highly mobile V2V channels in outdoor settings where pre-scanning the environment using LIDAR to build a reflection profile for a large area requires significant effort. Also, compared to SV images, publicly available outdoor LIDAR datasets [19, 22, 44, 69] are limited to few locations with terrain plan views only.

8 Conclusion and Discussion

In this paper, we presented a first-of-its-kind V2V mmWave system that leverages street view images to efficiently find reflections and maintain reliable connectivity. We note that mmSV's performance is limited by the publicly available 3D environment maps (e.g., OSM). The 3D OSM models lack granularity in terms of building microstructures. However, such detailed structure information is not available unless exhaustive 3D depth data is collected using sensors such as LiDARs through cumbersome efforts. Although we find that the impact is not significant because the majority of building reflections are limited to elevation angles around the horizontal plane of Tx/Rx vehicle antennas, it is also possible that the SV images can be used to detect the microstructures and improve the reflection predictions.

Acknowledgment

We thank our shepherd and anonymous reviewers for their insightful comments. This research is supported by ARL grant W911NF22-1-0216 and NSF grant CNS-2045885.

References

[1] 2020. IEEE Draft Standard for Wireless Access in Vehicular Environments (WAVE) – Networking Services. *IEEE P1609.3v4/D11, June 2020* (2020), 1–184.

[2] 2021. IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks–Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016)* (2021), 1–4379. https://doi.org/10.1109/IEEESTD.2021.9363693

[3] 2021. Remcom Wireless InSite 3D Wireless Prediction Software. . https://www.remcom.com/wireless-insite-em-propagation-software.

[4] 2022. 3GPP Enhancement of 3GPP support for V2X Scenarios. https://www.3gpp.org/ftp/Specs/archive/22_series/22.186/.

[5] 2022. AUTONOMOUS VEHICLES. https://content.naic.org/cipr-topics/autonomous-vehicles.

[6] 2022. Connected car fleet by region 2021-2035 . https://www.statista.com/statistics/1155517/global-connected-car-fleet-by-market/.

[7] 2023. California Department of Transportation (Caltrans) . https://dot.ca.gov/programs/traffic-operations/census.

[8] 2023. Google Street View static API . https://maps.googleapis.com/maps/api/streetview.

[9] 2023. NYS Department of Transportation . https://www.dot.ny.gov/tdv.

[10] 2023. Sources of photography. https://www.google.com/streetview/how-it-works/.

[11] 2023. The Bing Maps REST Imagery API. http://dev.virtualearth.net/REST/v1/Imagery/MetaData/Streetside/.

[12] [n. d.. Mikrotik wAP 60Gx3 AP. https://mikrotik.com/product/wap_60gx3_ap.

[13] [n. d.. The Evaluation Kits (EVKs) EVK06002. https://www.sivers-semiconductors.com/sivers-wireless/evaluation-kits/evaluation-kit-evk06002/.

[14] [n. d.. Vubiq, Irvine, CA, USA. 60 GHz Systems and Modules. https://www.ettus.com/product/details/N210-KIT.

[15] 3GPP. 2022. 5G;Service requirements for enhanced V2X scenarios (3GPP TS 22.186 version 17.0.0 Release 17). https://portal.etsi.org/webapp/workprogram/Report_WorkItem.asp?WKI_ID=65210.

[16] Anum Ali, Nuria González-Prelcic, and Amitava Ghosh. 2020. Passive Radar at the Roadside Unit to Configure Millimeter Wave Vehicle-to-Infrastructure Links. *IEEE Transactions on Vehicular Technology* 69, 12 (2020), 14903–14917. https://doi.org/10.1109/TVT.2020.3027636

[17] A. Alkhateeb, O. El Ayach, G. Leus, and R. W. Heath. 2014. Channel Estimation and Hybrid Precoding for Millimeter Wave Cellular Systems. *IEEE Journal of Selected Topics in Signal Processing* 8, 5 (2014), 831–846.

[18] Muhammad Alrabeiah, Andrew Hredzak, and Ahmed Alkhateeb. 2020. Millimeter Wave Base Stations with Cameras: Vision-Aided Beam and Blockage Prediction. In *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. 1–5. https://doi.org/10.1109/VTC2020-Spring48590.2020.9129369

[19] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. 2019. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*.

[20] Sean Bell, Paul Upchurch, Noah Snavely, and Kavita Bala. 2015. Material recognition in the wild with the Materials in Context Database. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3479–3487. https://doi.org/10.1109/CVPR.2015.7298970

[21] Syed Hashim Raza Bukhari, Mubashir Husain Rehmani, and Sajid Siraj. 2015. A survey of channel bonding for wireless networks and guidelines of channel bonding for futuristic cognitive radio sensor networks. *IEEE Communications Surveys & Tutorials* 18, 2 (2015), 924–948.

[22] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*.

[23] Mathilde Carlier. 2022. Sport utility vehicles worldwide - Statistics & Facts. https://www.statista.com/topics/6185/suv-market-worldwide/#topicOverviewhttps://www.statista.com/topics/6185/suv-market-worldwide/#topicOverview.

[24] Ashok Chaluvadi. 2020. Regional Variations on Most Popular Exterior Wall Materials. https://nahbnow.com/2020/10/regional-variations-on-most-popular-exterior-wall-materials/?_ga=2.123814095.474119564.1618212154-524437145.1617955526.

[25] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. 2017. Rethinking Atrous Convolution for Semantic Image Segmentation. *CoRR* abs/1706.05587 (2017). arXiv:1706.05587 http://arxiv.org/abs/1706.05587

[26] Junil Choi, Vutha Va, Nuria Gonzalez-Prelcic, Robert Daniels, Chandra R. Bhat, and Robert W. Heath. 2016. Millimeter-Wave Vehicular Communication to Support Massive Automotive Sensing. *IEEE Communications Magazine* 54, 12 (2016), 160–167. https://doi.org/10.1109/MCOM.2016.1600071CM

[27] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[28] Vip Desai, Lukasz Krzymien, Philippe Sartori, Weimin Xiao, Anthony Soong, and Ahmed Alkhateeb. 2014. Initial beamforming for mmWave communications. In *2014 48th Asilomar Conference on Signals, Systems and Computers*. 1926–1930. https://doi.org/10.1109/ACSSC.2014.7094805

[29] Marcus Dias, Aldebaro Klautau, Nuria González-Prelcic, and Robert W. Heath. 2019. Position and LIDAR-Aided mmWave Beam Selection using Deep Learning. In *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. 1–5. https://doi.org/10.1109/SPAWC.2019.8815569

[30] Haichuan Ding and Kang G. Shin. 2022. Accurate Angular Inference for 802.11ad Devices Using Beam-Specific Measurements. *IEEE Transactions on Mobile Computing* 21, 3 (2022), 822–834. https://doi.org/10.1109/TMC.2020.3015936

[31] Pedro F Felzenszwalb and Daniel P Huttenlocher. 2004. Efficient graph-based image segmentation. *International journal of computer vision* 59, 2 (2004), 167–181.

[32] Martin A. Fischler and Robert C. Bolles. 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* 24, 6 (jun 1981), 381–395. https://doi.org/10.1145/358669.358692

[33] Marco Giordani, Marco Mezzavilla, and Michele Zorzi. 2016. Initial Access in 5G mmWave Cellular Networks. *IEEE Communications Magazine* 54, 11 (2016), 40–47. https://doi.org/10.1109/MCOM.2016.1600193CM

[34] Nuria González-Prelcic, Roi Méndez-Rial, and Robert W. Heath. 2016. Radar aided beam alignment in MmWave V2I communications supporting antenna diversity. In *2016 Information Theory and Applications Workshop (ITA)*. 1–7. https://doi.org/10.1109/ITA.2016.7888145

[35] Andrew Graff, Yun Chen, Nuria González-Prelcic, and Takayuki Shimizu. 2023. Deep Learning-Based Link Configuration for Radar-Aided Multiuser Mmwave Vehicle-to-Infrastructure Communication. *IEEE Transactions on Vehicular Technology* (2023), 1–15. https://doi.org/10.1109/TVT.2023.3239227

[36] Mordechai Haklay. 2010. How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. *Environment and planning B: Planning and design* 37, 4 (2010), 682–703.

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[38] S. Hur, T. Kim, D. J. Love, J. V. Krogmeier, T. A. Thomas, and A. Ghosh. 2013. Millimeter Wave Beamforming for Wireless Backhaul and Access in Small Cell Networks. *IEEE Transactions on Communications* 61, 10 (2013), 4391–4403.

[39] Aleksandar Ichkov, Simon Häger, Petri Mähönen, and Ljiljana Simić. 2022. Comparative Evaluation of Millimeter-Wave Beamsteering Algorithms Using Outdoor Phased Antenna Array Measurements. In *2022 19th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 497–505. https://doi.org/10.1109/SECON55815.2022.9918162

[40] Ish Kumar Jain, Raghav Subbaraman, and Dinesh Bharadia. 2021. Two Beams Are Better than One: Towards Reliable and High Throughput MmWave Links. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference* (Virtual Event, USA) *(SIGCOMM '21)*. Association for Computing Machinery, New York, NY, USA, 488–502. https://doi.org/10.1145/3452296.3472924

[41] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, Zeng Yifu, Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Victor Sonck, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Dhruv Nair, Max Strobel, and Mrinal Jain. 2022. *ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation.* https://doi.org/10.5281/zenodo.7347926

[42] Aldebaro Klautau, Nuria González-Prelcic, and Robert W. Heath. 2019. LIDAR Data for Deep Learning-Based mmWave Beam-Selection. *IEEE Wireless Communications Letters* 8, 3 (2019), 909–912. https://doi.org/10.1109/LWC.2019.2899571

[43] Philipp Krähenbühl and Vladlen Koltun. 2013. Parameter Learning and Convergent Inference for Dense Random Fields. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28* (Atlanta, GA, USA) *(ICML'13)*. JMLR.org, III–513–III–521.

[44] Sriram Krishnan, Christopher Crosby, Viswanath Nandigam, Minh Phan, Charles Cowart, Chaitanya Baru, and Ramon Arrowsmith. 2011. OpenTopography: A Services Oriented Architecture for Community Access to LIDAR Topography. In *Proceedings of the 2nd International Conference on Computing for Geospatial Research & Applications* (Washington, DC, USA) *(COM.Geo '11)*. Association for Computing Machinery, New York, NY, USA, Article 7, 8 pages. https://doi.org/10.1145/1999320.1999327

[45] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (Eds.), Vol. 25. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[46] S. Kutty and D. Sen. 2016. Beamforming for Millimeter Wave Communications: An Inclusive Survey. *IEEE Communications Surveys Tutorials* 18, 2 (2016), 949–973.

[47] O. Landron, M.J. Feuerstein, and T.S. Rappaport. 1996. A comparison of theoretical and empirical reflection coefficients for typical exterior wall surfaces in a mobile radio environment. *IEEE Transactions on Antennas and Propagation* 44, 3 (1996), 341–351. https://doi.org/10.1109/8.486303

[48] B. Langen, G. Lober, and W. Herzig. 1994. Reflection and transmission behaviour of building materials at 60 GHz. In *5th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Wireless Networks - Catching the Mobile Future.*, Vol. 2. 505–509 vol.2. https://doi.org/10.1109/WNCMF.1994.529141

[49] Byung-Hyun Lee, Jong-Hwa Song, Jun-Hyuck Im, Sung-Hyuck Im, Moon-Beom Heo, and Gyu-In Jee. 2015. GPS/DR Error Estimation for Autonomous Vehicle Localization. *Sensors* 15, 8 (2015), 20779–20798. https://doi.org/10.3390/s150820779

[50] H. Ling, R.-C. Chou, and S.-W. Lee. 1989. Shooting and bouncing rays: calculating the RCS of an arbitrarily shaped cavity. *IEEE Transactions on Antennas and Propagation* 37, 2 (1989), 194–205. https://doi.org/10.1109/8.18706

[51] Adrian Loch, Arash Asadi, Gek Hong Sim, Joerg Widmer, and Matthias Hollick. 2017. mm-Wave on wheels: Practical 60 GHz vehicular communication without beam training. In *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*. 1–8. https://doi.org/10.1109/COMSNETS.2017.7945351

[52] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. 2018. Microscopic Traffic Simulation using SUMO, In The 21st IEEE International Conference on Intelligent Transportation Systems. *IEEE Intelligent Transportation Systems Conference (ITSC)*. https://elib.dlr.de/124092/

[53] David Martín-Sacristán, Sandra Roger, David Garcia-Roger, Jose F Monserrat, Panagiotis Spapis, Chan Zhou, and Alexandros Kaloxylos. 2020. Low-latency infrastructure-based cellular V2V communications for multi-operator environments with regional split. *IEEE Transactions on Intelligent Transportation Systems* 22, 2 (2020), 1052–1067.

[54] Z. Marzi, D. Ramasamy, and U. Madhow. 2016. Compressive Channel Estimation and Tracking for Large Arrays in mm-Wave Picocells. *IEEE Journal of Selected Topics in Signal Processing* 10, 3 (April 2016), 514–527. https://doi.org/10.1109/JSTSP.2016.2520899

[55] Barrett Mohrmann. 2023. Height of a car and why it matters! https://www.way.com/blog/height-of-a-car-and-why-it-matters/#:~:text=The%20average%20car%20in%20the%20U.S%20is%20between%20five%20and%20six%20feet%20high&text=But%20when%20driving%20a%20vehicle,above%20the%20maximum%20passable%20height.

[56] OpenStreetMap contributors. 2017. Planet dump retrieved from https://planet.osm.org . https://www.openstreetmap.org.

[57] Ioannis Pefkianakis and Kyu-Han Kim. 2018. Accurate 3D Localization for 60 GHz Networks. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems* (Shenzhen, China) *(SenSys '18)*. Association for Computing Machinery, New York, NY, USA, 120–131. https://doi.org/10.1145/3274783.3274852

[58] Nikos Platis and Theoharis Theoharis. 2003. Fast Ray-Tetrahedron Intersection Using Plucker Coordinates. *Journal of Graphics Tools* 8, 4 (2003), 37–48. https://doi.org/10.1080/10867651.2003.10487593 arXiv:https://doi.org/10.1080/10867651.2003.10487593

[59] Hang Qiu, Po-Han Huang, Namo Asavisanu, Xiaochen Liu, Konstantinos Psounis, and Ramesh Govindan. 2022. AutoCast: Scalable Infrastructure-Less Cooperative Perception for Distributed Collaborative Driving. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services* (Portland, Oregon) *(MobiSys '22)*. Association for Computing Machinery, New York, NY, USA, 128–141. https://doi.org/10.1145/3498361.3538925

[60] D. Ramasamy, S. Venkateswaran, and U. Madhow. 2012. Compressive adaptation of large steerable arrays. In *2012 Information Theory and Applications Workshop*. 234–239.

[61] D. Ramasamy, S. Venkateswaran, and U. Madhow. 2012. Compressive tracking with 1000-element arrays: A framework for multi-Gbps mm wave cellular downlinks. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. 690–697.

[62] M. E. Rasekh and U. Madhow. 2018. Noncoherent compressive channel estimation for mm-wave massive MIMO. In *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. 889–894. https://doi.org/10.1109/ACSSC.2018.8645127

[63] Francesco Raviglione, Marco Malinverno, Stefano Feraco, Giuseppe Avino, Claudio Casetti, Carla Fabiana Chiasserini, Nicola Amati, and Joerg Widmer. 2021. Experimental Assessment of IEEE 802.11-Based V2I Communications. In *Proceedings of the 18th ACM Symposium on*

Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks (Alicante, Spain) *(PE-WASUN '21)*. Association for Computing Machinery, New York, NY, USA, 33–40. https://doi.org/10.1145/3479240.3488506

[64] Guillem Reus-Muns, Batool Salehi, Debashri Roy, Tong Jian, Zifeng Wang, Jennifer Dy, Stratis Ioannidis, and Kaushik Chowdhury. 2021. Deep Learning on Visual and Location Data for V2I mmWave Beamforming. In *2021 17th International Conference on Mobility, Sensing and Networking (MSN)*. 559–566. https://doi.org/10.1109/MSN53354.2021.00087

[65] Batool Salehi, Mauro Belgiovine, Sara Garcia Sanchez, Jennifer Dy, Stratis Ioannidis, and Kaushik Chowdhury. 2020. Machine Learning on Camera Images for Fast mmWave Beamforming. In *2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. 338–346. https://doi.org/10.1109/MASS50613.2020.00049

[66] Batool Salehi, Jerry Gu, Debashri Roy, and Kaushik Chowdhury. 2022. FLASH: Federated Learning for Automated Selection of High-band mmWave Sectors. In *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*. 1719–1728. https://doi.org/10.1109/INFOCOM48880.2022.9796865

[67] Lavanya Sharan, Ruth Rosenholtz, and Edward Adelson. 2009. Material perception: What can you see in a brief glance? *Journal of Vision* 9, 8 (2009), 784–784.

[68] Daniel Steinmetzer, Daniel Wegemer, Matthias Schulz, Joerg Widmer, and Matthias Hollick. 2017. *Compressive Millimeter-Wave Sector Selection in Off-the-Shelf IEEE 802.11ad Devices*. Association for Computing Machinery, New York, NY, USA, 414–425. https://doi.org/10.1145/3143361.3143384

[69] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. 2020. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[70] Sanjib Sur, Ioannis Pefkianakis, Xinyu Zhang, and Kyu-Han Kim. 2018. Towards Scalable and Ubiquitous Millimeter-Wave Wireless Networks. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking* (New Delhi, India) *(MobiCom '18)*. Association for Computing Machinery, New York, NY, USA, 257–271. https://doi.org/10.1145/3241539.3241579

[71] Sanjib Sur, Vignesh Venkateswaran, Xinyu Zhang, and Parmesh Ramanathan. 2015. 60 GHz Indoor Networking through Flexible Beams: A Link-Level Profiling. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems* (Portland, Oregon, USA) *(SIGMETRICS '15)*. Association for Computing Machinery, New York, NY, USA, 71–84. https://doi.org/10.1145/2745844.2745858

[72] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going Deeper With Convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[73] Alexandru Telea. 2004. An Image Inpainting Technique Based on the Fast Marching Method. *Journal of Graphics Tools* 9, 1 (2004), 23–34. https://doi.org/10.1080/10867651.2004.10487596 arXiv:https://doi.org/10.1080/10867651.2004.10487596

[74] Aysim Toker, Qunjie Zhou, Maxim Maximov, and Laura Leal-Taixe. 2021. Coming Down to Earth: Satellite-to-Street View Synthesis for Geo-Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6488–6497.

[75] Ries Uittenbogaard, Clint Sebastian, Julien Vijverberg, Bas Boom, Dariu M. Gavrila, and Peter H.N. de With. 2019. Privacy Protection in Street-View Panoramas Using Depth and Multi-View Imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[76] Song Wang, Jingqi Huang, and Xinyu Zhang. 2020. Demystifying Millimeter-Wave V2X: Towards Robust and Efficient Directional Connectivity under High Mobility. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking* (London, United Kingdom) *(MobiCom '20)*. Association for Computing Machinery, New York, NY, USA, Article 51, 14 pages. https://doi.org/10.1145/3372224.3419208

[77] Song Wang, Jingqi Huang, Xinyu Zhang, Hyoil Kim, and Sujit Dey. 2020. X-Array: Approximating Omnidirectional Millimeter-Wave Coverage Using an Array of Phased Arrays. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking* (London, United Kingdom) *(MobiCom '20)*. Association for Computing Machinery, New York, NY, USA, Article 5, 14 pages. https://doi.org/10.1145/3372224.3380882

[78] Lili Wei, Qian Li, and Geng Wu. 2017. Exhaustive, Iterative and Hybrid Initial Access Techniques in mmWave Communications. In *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. 1–6. https://doi.org/10.1109/WCNC.2017.7925666

[79] E. WEISZFELD. 1937. Sur le point pour lequel la Somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal, First Series* 43 (1937), 355–386.

[80] Timothy Woodford, Xinyu Zhang, Eugene Chai, Karthikeyan Sundaresan, and Amir Khojastepour. 2021. SpaceBeam: LiDAR-Driven One-Shot MmWave Beam Management. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services* (Virtual Event, Wisconsin) *(MobiSys '21)*. Association for Computing Machinery, New York, NY, USA, 389–401. https://doi.org/10.1145/3458864.3466864

[81] Jianxiong Xiao and Long Quan. 2009. Multiple view semantic segmentation for street view images. In *2009 IEEE 12th International Conference on Computer Vision*. 686–693. https://doi.org/10.1109/ICCV.2009.5459249

[82] Erich Zöchmann, Ke Guan, and Markus Rupp. 2017. Two-ray models in mmWave communications. In *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. 1–5. https://doi.org/10.1109/SPAWC.2017.8227681