

# Set-Valued Shadow Matching Using Zonotopes for 3D-Map-Aided GNSS Localization

Sriramya Bhamidipati | Shreyas Kousik | Grace Gao

Department of Aeronautics and  
Astronautics  
Stanford University  
Stanford, CA, USA

## Correspondence

Grace Gao,  
Department of Aeronautics  
and Astronautics, Stanford University,  
Stanford, CA, USA, 94305  
Email: [gracegao@stanford.edu](mailto:gracegao@stanford.edu)

## Abstract

Unlike many urban localization methods that return point-valued estimates, a set-valued representation enables robustness by ensuring that a continuum of possible positions obeys safety constraints. One strategy with the potential for set-valued estimation is GNSS-based shadow matching (SM) in which one uses a three-dimensional (3D) map to compute GNSS shadows (where line-of-sight is blocked). However, SM requires a point-valued grid for computational tractability, with accuracy limited by grid resolution. We propose zonotope shadow matching (ZSM) for set-valued 3D-map-aided GNSS localization. ZSM represents buildings and GNSS shadows using constrained zonotopes, a convex polytope representation that enables propagating set-valued estimates using fast vector concatenation operations. Starting from a coarse set-valued position, ZSM refines the estimate depending on the receiver being inside or outside each shadow as judged by received carrier-to-noise density. We demonstrate our algorithm's performance using simulated experiments on a simple 3D example map and on a dense 3D map of San Francisco.

## Keywords

3D-map-aided GNSS, constrained zonotope, localization, set intersection, set-valued estimate, shadow matching

## 1 | INTRODUCTION

GNSS-based localization is often unreliable in dense urban areas. As illustrated in Figure 1, direct line-of-sight (LOS) GNSS signals can be blocked or reflected by tall buildings (Hofmann-Wellenhof et al., 1992), creating non-line-of-sight (NLOS) and multipath effects, thereby lowering the number of visible GNSS satellites available for localization (Zhu et al., 2018). In particular, LOS satellite signals in the cross-street direction are more likely to be blocked or reflected by buildings than signals along the street. Consequently, positioning accuracy is degraded in urban areas, especially in the cross-street direction.

Robustness to degraded accuracy is important for safety-critical GNSS applications such as autonomous driving and drone delivery. As detailed later in our discussion of related work, set-valued position estimates enable robustness guarantees by ensuring an entire set of possible positions lies outside of obstacles or inside

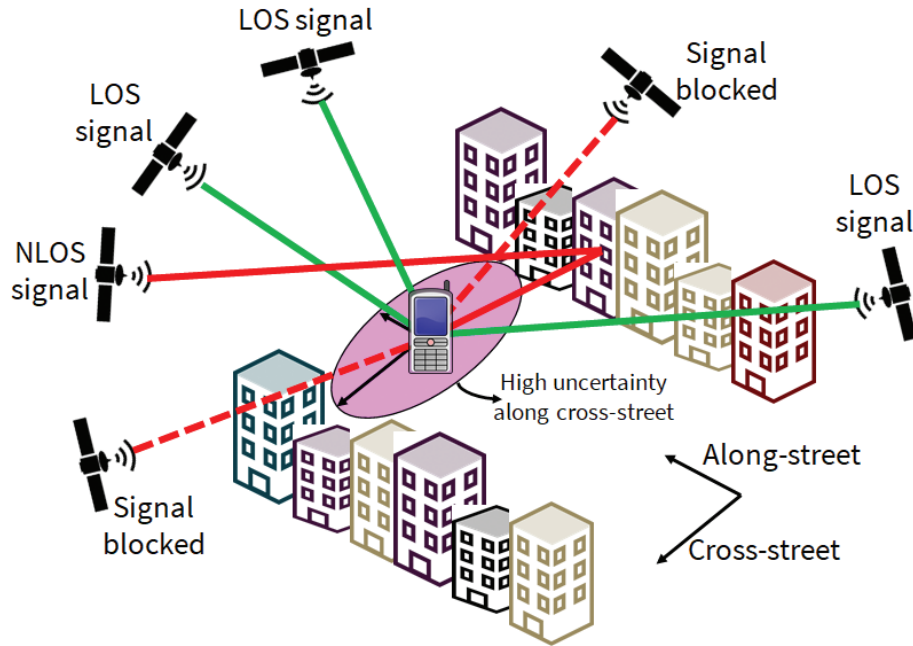


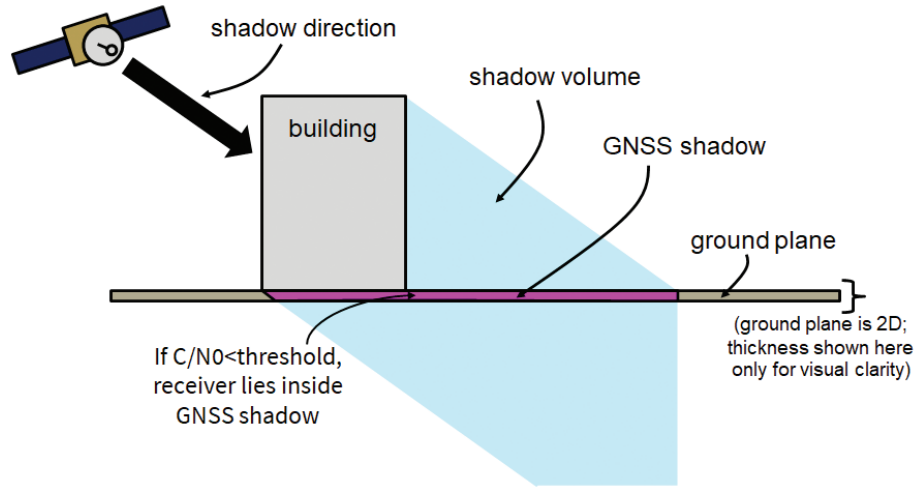
FIGURE 1 NLOS and multipath effects in GNSS satellites lead to high localization uncertainty in urban areas, especially along the cross-street direction. The pink ellipse depicts uncertainty in the receiver position estimate. This figure is adapted from Wang et al. (2013b).

user-specified bounds, for example. Unfortunately, generating such position estimates is often challenging, requiring approximation with a discrete collection of points. In this work, we propose a novel method of generating set-valued estimates without discretization by combining recent three-dimensional (3D)-map-aided techniques from the GNSS community with recent geometric set representations from the controls community.

## 1.1 | Related Work

A variety of approaches exist to address the challenge of positioning accuracy by leveraging a 3D map of an urban environment and buildings. For example, in ray tracing, one can create discrete position candidates with a particle filter and find all the possible paths from each particle to the GNSS satellite, while considering a limit on the number of interactions with nearby buildings per each transmitted ray (Miura et al., 2013; Suzuki, 2016). Essentially, ray tracing estimates the reflection route of NLOS signals and uses this to correct the bias in GNSS pseudoranges. While there exist commercial implementations of this approach (Iland et al., 2018), ray tracing is computationally-expensive and, instead, requires being offloaded to the cloud. One strategy to reduce the computational complexity is to apply machine learning (ML)-based GNSS to evaluate a predefined grid of position candidates with a neural network by collectively processing different heatmaps of pseudorange residuals (van Diggelen & Wang, 2020). This enables real-time performance with server-based processing, but achieves onboard tractability by considering a trade-off between the cross-street positioning accuracy and the computations executed to estimate receiver position.

An alternative approach is shadow matching (SM; Groves, 2011; Groves et al., 2015; Wang et al., 2013b), which can avoid offloading computation to the cloud



**FIGURE 2** Our proposed technique on set-valued shadow matching using zonotopes to leverage constrained zonotopes and efficiently represent buildings, shadows, and a set-valued receiver position estimate; note that the 2D group plane is shown as having volume only for illustration's sake.

(Wang et al., 2013a, 2014). SM uses a 3D map to estimate GNSS shadows of buildings, which are regions of the map where direct LOS signals are blocked. A GNSS shadow is depicted in Figure 2. The received carrier-to-noise density ( $C/N_0$ ) of GNSS satellite signals enables us to determine if a receiver is inside or outside the GNSS shadow of each building. However, to be computationally tractable, SM requires precomputing building skylines for a predefined grid of position candidates. These skylines are compared against the azimuth and elevation of GNSS satellites at runtime (Groves et al., 2015; Wang et al., 2013a, 2014). Each point-valued position candidate is evaluated by comparing the received  $C/N_0$  with the precomputed building skyline to estimate a visibility score that can be used to identify the most likely position candidate (or candidates) and also a weighted empirical covariance for downstream use (Wang et al., 2013a).

A common aspect of the above methods is that they represent position estimates in a discretized way. Since accuracy depends on the number of discrete points, these methods face a challenge in the case of robust urban localization, or ensuring that a position estimate obeys user-specified protection levels or safety bounds. That is, one can use a finer discretization to achieve a more accurate estimate at the expense of more computation.

To avoid the point-valued discretization trade-off, one can instead consider a set-valued approach to compute a continuum of state estimates given the initial set of states and known measurement bounds (Combettes & Civanlar, 1991; Scott et al., 2016; Shi et al., 2015; Shiryayev & Podivilova, 2015). This enables robustness by ensuring that the set lies within, e.g., user-specified safety bounds. Recently, there has been growing interest in set-valued representations of receiver states and measurements for robust localization and motion planning in urban environments (Bhamidipati & Gao, 2020; Kousik et al., 2019; Shetty & Gao, 2020). Unfortunately, these methods typically require assuming a set-valued representation of uncertain measurements by overapproximating a confidence level set of Gaussian distribution using a polytope (Bhamidipati & Gao, 2020; Shetty & Gao, 2020), for example. In other words, one must make an assumption about the underlying distribution of measurements. Therefore, it remains an open challenge to generate set-valued

receiver position estimates for the distributions that are common in urban environments (e.g., multi-modal with disconnected components).

One strategy for modeling arbitrary set-valued state estimates is to use zonotopes. Zonotopes are convex, symmetrical polytopes that can propagate set-valued state estimates using fast vector concatenation operations. Zonotopes are constructed as Minkowski sums (Althoff & Dolan, 2014) of line segments in an arbitrary-dimensional Euclidean space. These objects are well known for representing reachable sets of dynamic systems (Althoff, 2015), which can enable formally-verified motion planning, fault detection, and navigation (Bhamidipati & Gao, 2020; Kousik et al., 2019; Shetty & Gao, 2020). These objects can also be extended to constrained zonotopes, which can represent arbitrary convex polytopes (Raghuraman & Koeln, 2022; Scott et al., 2016), avoiding the limitation of symmetry. While zonotopes are widely used in robotics for path planning and collision avoidance, they have not been applied in the field of GNSS localization for addressing multipath/NLOS effects.

## 1.2 | Proposed Method

Our current work proposes a novel paradigm for 3D-map-aided GNSS localization based on GNSS shadow matching: set-based shadow matching termed as zonotope shadow matching (ZSM). This work is based on our recent ION GNSS+ 2021 conference paper (Bhamidipati et al., 2021). We formulate buildings and GNSS shadows (regions where LOS signals are blocked) using constrained zonotopes wherein we compute the GNSS shadow for each satellite/building pair via the intersection of building shadow volume with that of ground plane. An illustration of these set operations is shown in Figure 2. The ZSM algorithm begins with a coarse set-valued position estimate, which we iteratively refine using the GNSS shadow of each satellite/building pair in an urban map. We use  $C/N_0$  values to decide whether the receiver is inside or outside each GNSS shadow and intersect the shadow with our position estimate if inside the shadow, or else subtract the shadow from our position estimate. The final set-valued position estimate from our proposed ZSM algorithm can be incorporated into downstream polytopic estimation or verification methods without requiring assumptions about the distribution of the estimate.

A key feature of our proposed ZSM is that it avoids gridding and, instead, directly estimates a set-valued receiver position estimate by leveraging constrained zonotopes and two-dimensional (2D) polytopes (i.e., polygons). Furthermore, given that zonotope operations need fewer computations, our approach requires significantly lower computational power during the offline step of shadow matching, while during online processing, the required computational power is on-par to that of conventional SM and lesser storage than that of conventional SM is incurred. Another advantage is that our formulation estimates a non-Gaussian distribution of possible receiver positions (a continuum) in an urban environment, based on many position statistics such as worst-case error, multiple/distinct ambiguous modes, error bounds of each mode, can be evaluated in an exact manner.

Given the recent availability of high-fidelity, open-source 3D building maps (Miura et al., 2015; Wang et al., 2013b), we assume the inaccuracies in the building boundaries and road lanes and the effect of building materials on GNSS signals to be negligible. We also consider an ideal LOS/NLOS classifier (i.e., with true positive 100% [LOS detection] and true negative 100% [NLOS detection]). Our choice of an ideal LOS classifier is justified as it provides an uncorrupted platform

to compare the performance of our proposed set-valued shadow matching (i.e., zonotope shadow matching) with conventional SM. Also note that, in this work, we do not explicitly account for multipath effects. While this work demonstrates great success in achieving the exact bounds of set-valued receiver positioning for various simulation scenarios (see Section 4), ZSM suffers from challenges similar to that of shadow matching, including brittleness to misclassification and inability to resolve multi-modal ambiguities without additional information. Note that addressing these other challenges, which are mentioned in a detailed manner in Groves et al. (2015), is beyond the scope of this manuscript. However, we have proposed extensions of this current work that account for brittleness to satellite misclassification in Neamati et al. (2022a) and for resolving multi-modal ambiguities in Neamati et al. (2022b). While the ZSM method proposed in this work cannot directly handle satellite misclassifications associated with the state-of-the-art, off-the-shelf classifiers (Haosheng et al., 2020; Haosheng et al., 2018), we can still implement this work in real-world settings by leveraging a subset of GNSS satellites, either LOS or NLOS, that exhibit high satellite classifier probabilities (see experiments in Section 4). Also, given that our proposed ZSM method is an anytime implementation that works in a sequential manner, we can achieve real-time performance via strategic ordering of GNSS satellites (e.g., with an objective to maximize the dilution of precision) so the more useful satellites are processed first and the algorithm is terminated when required to output the current estimate of the set-valued receiver position.

### 1.3 | Contributions and Paper Organization

Our key contributions are as follows:

1. We propose a novel set-valued approach to representing GNSS shadows using zonotopes. Offline, we represent buildings using constrained zonotopes. At runtime, we generate the GNSS shadow for each satellite/building pair by extending the building in the shadow direction from the satellite to the building, as shown in Figure 2. This operation is efficient due to our representation of buildings and GNSS shadows as constrained zonotopes.
2. We propose the ZSM algorithm. Given an initial set-valued receiver position estimate, we iteratively subtract or intersect GNSS shadows across satellites to refine the estimate depending on the associated LOS/NLOS characteristics as judged by received  $C/N_0$  values. To our knowledge, this is the first work utilizing set representations for 3D-map-aided GNSS.
3. We experimentally demonstrate that ZSM computes a set-valued receiver position estimate unlike conventional SM (Groves, 2011; Groves et al., 2015; Wang et al., 2013a, 2013b) in which point-valued position accuracy is limited by its precomputed grid resolution. We perform experiment simulations using (a) a simple 3D map comprised of two urban buildings and (b) a dense 3D building map of San Francisco. We validate ZSM performance in terms of computation load (for both offline and online computations), point-valued estimation error, and the position bounds (size of the final set-valued receiver position estimate) in along-street and cross-street directions.

The remainder of the paper is organized as follows. Section 2 introduces mathematical notations for the relevant set representations. Section 3 states our

proposed ZSM algorithm. Section 4 presents various experimental results that validate our proposed ZSM in simulation. Finally, Section 5 provides concluding remarks.

## 2 | PRELIMINARIES OF SET REPRESENTATIONS

We now introduce the mathematical notation and definitions used throughout this paper and present constrained zonotopes.

### 2.1 | Mathematical Notation and Definitions

We denote the natural numbers as  $\mathbb{N}$  and  $n$ -dimensional Euclidean space as  $\mathbb{R}^n$ . Points and scalars are italic lowercase; sets and matrices/arrays are italic uppercase. Sets of sets are script uppercase. An  $n \times n$  identity matrix is  $\mathbf{I}_n$ . An  $n \times n$  array of zeros is  $\mathbf{0}_n$ , and a same-sized array of ones is  $\mathbf{1}_n$ . Similarly, an  $n \times m$  array of zeros is  $\mathbf{0}_{n \times m}$ , and a same-sized array of ones is  $\mathbf{1}_{n \times m}$ . An empty vector or array is  $[\ ]$ .

Let  $A, B \subset \mathbb{R}^n$ . The Minkowski (set) sum (Althoff & Dolan, 2014) is  $\oplus$ , defined as  $A \oplus B = \{a + b \mid a \in A, b \in B\}$ . For example, the Minkowski sum of two (non-parallel) line segments in  $\mathbb{R}^2$  is a convex polygon, whereas the Minkowski sum of two (non-parallel) polygons in  $\mathbb{R}^3$  is a convex polyhedron. The convex hull is  $\text{convhull}(A) = \{\lambda a_1 + (1 - \lambda)a_2 \mid a_1, a_2 \in A\}$ . A convex polytope  $P \subset \mathbb{R}^n$  can be constructed as the convex hull of a set of vertices  $V \subset \mathbb{R}^n$ :  $P = \text{convhull}(V)$ , which we call its vertex representation.

### 2.2 | Constrained Zonotopes

We now define constrained zonotopes (Scott et al., 2016), which we use to represent buildings and GNSS shadows as shown in Figure 2. Consider a center  $c \in \mathbb{R}^n$ , a generator matrix  $G = [g_1, \dots, g_m] \in \mathbb{R}^{n \times m}$ , and linear constraints defined by  $A \in \mathbb{R}^{p \times m}$  and  $b \in \mathbb{R}^p$ . Per Scott et al. (2016), we define a constrained zonotope  $Z \subset \mathbb{R}^n$  as a set:

$$Z = \text{zono}(c, G, A, b) = \{c + G\beta \in \mathbb{R}^n \mid \beta \in [-1, 1]^m \text{ and } A\beta = b\} \subset \mathbb{R}^n \quad (1)$$

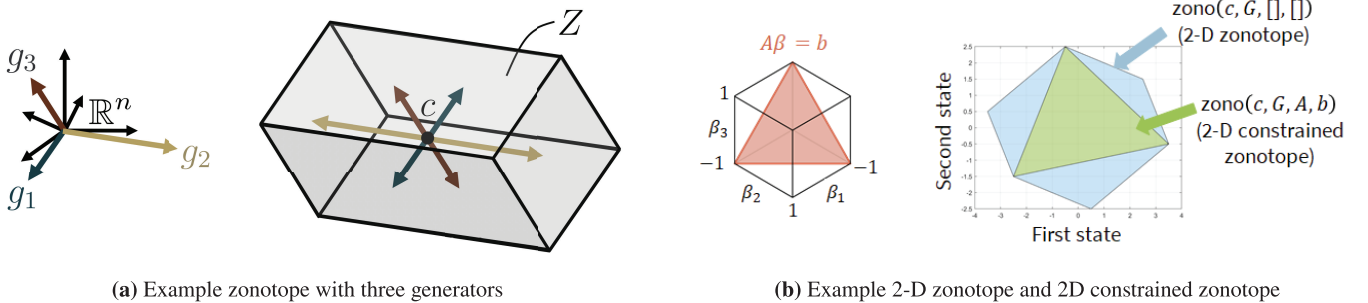


FIGURE 3 Subfigure (a) is a zonotope (grey volume) with no constraints and three generators; subfigure (b) shows a 2D zonotope (blue) and a constrained zonotope (green) with constraints shown in red on the left.

The columns of  $G$  are called generators. Also note that a zonotope is a constrained zonotope with no constraints defined, which we denote by  $\text{zono}(c, G, [], []) = \{c + G\beta \in \mathbb{R}^n \mid \beta \in [-1, 1]^m\}$ . An example 3D zonotope (i.e., no constraints) is shown in Figure 3(a).

For another example illustration with  $c = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ ,  $G = \begin{bmatrix} 1.5 & -1.5 & 0.5 \\ 1 & 0.5 & -1 \end{bmatrix}$ ,  $A = [1, 1, 1]$ ,

and  $b = 1$ , Figure 3(b) shows the 2D zonotope  $\text{zono}(c, G, [], [])$  in blue while the 2D constrained zonotope  $\text{zono}(c, G, A, b)$  is shown in green.

### 3 | PROPOSED ZSM ALGORITHM

Our proposed ZSM algorithm is summarized as follows. First, we create a 3D shadow volume for each satellite/building pair by extending the building in the shadow direction from the satellite to the building. Note, the building, shadow direction, and shadow volume are all represented as constrained zonotopes. Second, we intersect the shadow volumes with the ground plane (on which we assume the receiver is positioned) to compute GNSS shadows as 2D constrained zonotopes. Third, by comparing received  $C/N_0$  values against a user-specified threshold, we either subtract (if LOS signal) or intersect (if NLOS signal) each GNSS shadow with our current set-valued receiver position estimate. Starting from an initial set of position estimates, the final position estimate set is obtained by iterating through all buildings and satellites.

We note that this is a snapshot algorithm, meaning that we make use of only the instantaneous data available at any given time instance. Furthermore, the order in which GNSS satellites and buildings are considered does not affect the final set-valued estimate, meaning that the method is parallelizable. However, we leave extensions to time-varying shadows, receiver motion models, and parallelization as future work. To proceed, we describe our model and assumptions of the urban environment and GNSS satellite signals, then detail how we compute shadows and, finally, present the ZSM algorithm.

#### 3.1 | System Model

##### 3.1.1 | Urban Environment Map

We assume access to a 3D building map of the urban environment in which we plan to perform positioning, as is done in related work (Adjrad & Groves, 2017; Iland et al., 2018; van Diggelen & Wang, 2020). We assume such a map consists of a ground plane and a collection of buildings.

We assume that the ground plane, denoted  $\mathcal{G}$ , is either a single 2D constrained zonotope (as in Figure 2), or a collection of 2D constrained zonotopes:  $\mathcal{G} = \{G_k\}_{k=1}^{n_{\text{grnd}}}$ , where each  $G_k$  is a 2D constrained zonotope and  $n_{\text{grnd}} \in \mathbb{N}$ . Thus, our algorithm does not require the ground plane to be constant within an area of interest. We can account for the variations in the ground height by modeling the ground plane as a collection of triangles, wherein each triangle is converted to a constrained zonotope, and the ground plane is represented by a collection of constrained zonotopes. There exist many standard ways to extract ground height, of which one straightforward way is to extract the ground plane information from the 3D building map being utilized. The other is to leverage the open-source information on digital terrain models (DTMs), digital surface models (DSMs), or digital elevation

models (DEMs; Chen et al., 2017). We also assume the ground plane is bounded (implied by  $n_{\text{grnd}}$  being finite). In particular, we only consider a specific, bounded urban area that could be potentially identified by a coarse position estimate to within a few kilometers of the localization error.

We denote the set of all buildings as  $\mathcal{B} = \{B_i\}_{i=1}^{n_{\text{bldg}}}$ , where  $n_{\text{bldg}} \in \mathbb{N}$  is finite and  $B_i \subset \mathbb{R}^3$  denotes the  $i$ -th building. We further assume that each building can be represented as a union of constrained zonotopes:

$$B_i = \bigcup_{l=1}^{n_i} Z_l \quad \text{with each } Z_l = \text{zono}(c_{i,l}, G_{i,l}, A_{i,l}, b_{i,l}) \quad (2)$$

where  $n_i$  is the number of constrained zonotopes representing the building  $B_i$ . Note that any convex polytope is a constrained zonotope (Scott et al., 2016), so this is not a restrictive assumption. Also note that, since the union of constrained zonotopes cannot be represented as a constrained zonotope (because it is not necessarily a convex polytope), in practice, we represent each building as a list of constrained zonotopes (i.e., the union is treated implicitly).

### 3.1.2 | Preprocessing Standard 3D Maps

Representing buildings as constrained zonotopes requires preprocessing a 3D map. Standard 3D maps generated by computer-aided design software are often represented by a triangulation, or a union of triangles. That is, for a building  $B_i \in \mathcal{B}$ :

$$B_i = \bigcup_{l=1}^{n_{\text{trng}}} T_l \quad \text{with each } T_l = \text{convhull}(\{t_{l,1}, t_{l,2}, t_{l,3}\}) \quad (3)$$

such that  $n_{\text{trng}} \in \mathbb{N}$  is finite and each  $t_{l,j} \in \mathbb{R}^3$  is a vertex of the triangle. Assuming our map is in such a standard format, we preprocess a 3D map as follows to satisfy our assumption in Equation (2).

First, note that the convex hull of constrained zonotopes is given by Raghuraman and Koeln's (2022) Theorem 5 as follows. Let  $Z_1 = \text{zono}(c_1, G_1, A_1, b_1) \subset \mathbb{R}^n$  and  $Z_2 = \text{zono}(c_2, G_2, A_2, b_2) \subset \mathbb{R}^n$ . Suppose that  $G_1 \in \mathbb{R}^{n \times m_1}$ ,  $G_2 \in \mathbb{R}^{n \times m_2}$ ,  $A_1 \in \mathbb{R}^{k_1 \times m_1}$ , and  $A_2 \in \mathbb{R}^{k_2 \times m_2}$ . Then:

$$\text{convhull}(Z_1 \cup Z_2) = \text{zono}\left(\frac{1}{2}(c_1 + c_2), [G_1, G_2, \frac{1}{2}(c_1 - c_2), \mathbf{0}], A_{\text{CH}}, b_{\text{CH}}\right), \quad \text{where:} \quad (4a)$$

$$A_{\text{CH}} = \begin{bmatrix} A_1 & \mathbf{0}_{k_1 \times m_2} & -\frac{1}{2}b_1 & \mathbf{0}_{k_1 \times 2(m_1 + m_2)} \\ \mathbf{0}_{k_2 \times m_1} & A_2 & \frac{1}{2}b_2 & \mathbf{0}_{k_2 \times 2(m_1 + m_2)} \\ A_{3,1} & A_{3,2} & A_{3,0} & \mathbf{I}_{2(m_1 + m_2)} \end{bmatrix} \quad b_{\text{CH}} = \begin{bmatrix} \frac{1}{2}b_1 \\ \frac{1}{2}b_2 \\ -\frac{1}{2}\mathbf{1}_{2(m_1 + m_2) \times 1} \end{bmatrix} \quad (4b)$$

$$A_{3,1} = \begin{bmatrix} \mathbf{I}_{m_1} \\ -\mathbf{I}_{m_1} \\ \mathbf{0}_{2m_2 \times m_1} \end{bmatrix}, \quad A_{3,2} = \begin{bmatrix} \mathbf{0}_{2m_1 \times m_2} \\ \mathbf{I}_{m_2} \\ -\mathbf{I}_{m_2} \end{bmatrix} \quad \text{and} \quad A_{3,0} = \begin{bmatrix} -\frac{1}{2}\mathbf{1}_{2m_1 \times 1} \\ +\frac{1}{2}\mathbf{1}_{2m_2 \times 1} \end{bmatrix} \quad (4c)$$

Now, notice that each vertex can be represented as a constrained zonotope,  $t_{l,j} = \text{zono}(t_{l,j}, [], [], [])$ . So, we can represent  $T_l$  as a constrained zonotope by taking  $\text{convhull}(\text{convhull}(\{t_{l,1}, t_{l,2}\}, \{t_{l,3}\}))$ , meaning that we apply Equation (4) twice. Thus, we preprocess a 3D map by iterating over all of its triangles and representing



each one as a constrained zonotope. In this case,  $n_{\text{trng}} = n_i$ . Note that another series of convex hull operations can be performed to combine the constrained zonotopes (each corresponds to a triangle) to form a single constrained zonotope associated with each building, in which case  $n_i = 1$ .

### 3.1.3 | Receiver Assumptions

We assume that the receiver is on the 2D ground plane, which we find reduces the computational burden and is common in conventional SM (Groves et al., 2015). We plan to extend this method to 3D in future work.

To create a set-valued receiver position estimate, we consider an area of interest (AOI):

$$\mathcal{A} = \{A_k\}_{k=1}^{n_{\text{AOI}}} \subseteq \mathcal{G} \quad (5)$$

where  $A_k$  is a single ground plane-constrained zonotope. Since the ground plane is bounded, the AOI is bounded. While it is reasonable to treat the entire ground plane as the AOI, we make this distinction to enable our algorithm to take advantage of the non-shadow-based localization data obtained from various information sources. Note that the requirements for determining the size of the AOI remain the same as those for conventional SM (i.e., the AOI needs to cover the user's true location). Some potential methods for this include the following, among which a few are explained in a more detailed manner in Groves et al. (2015): (a) use of position estimate and uncertainty bounds from the conventional GNSS ranging solution, which is based on weighted analysis of all GPS satellites or only the LOS ranging signals; (b) use of other sensors, such as camera-based geotagging, WiFi-based positioning, sensor fusion of GPS-vision or GPS-lidar (GPS light detection and ranging), multi-agent system with inter-ranging; (c) use of predicted position estimates and uncertainty from onboard state estimation techniques, such as Kalman filters, particle filters; and (d) use of a heuristic approach that involves either considering a very large AOI and later narrowing down the correct mode from among the multiple ambiguous modes of shadow matching identified or starting with an initial AOI and increasing the size by a scale factor until a viable set-estimate of receiver position is obtained. Furthermore, the user can choose to eliminate the segments in the AOI that lie within building footprints since the user is expected to be outdoors and not inside the building.

### 3.1.4 | Satellites and Received $C/N_0$

We denote the observed set of GNSS satellites by  $\mathcal{S} = \{s_j\}_{j=1}^{n_{\text{sat}}}$ , where each  $s_j \in \mathbb{R}^3$  is a satellite location and  $n_{\text{sat}}$  is the total number. We obtain the received  $C/N_0$  values from the GNSS receiver; we denote these values by  $\mathcal{R} = \{r_j\}_{j=1}^{n_{\text{sat}}}$ , one for each satellite. In this work, we assume an ideal LOS/NLOS classifier and, thus, do not need to account for the associated false alarm or missed-detection rates.

## 3.2 | Computing Shadows

Now we detail how we compute shadows using operations on constrained zonotopes.

### 3.2.1 | Overview and Assumptions

To create shadows, first, we consider the shadow direction associated with each satellite/building pair. This is a unit vector pointing in the direction from the satellite's position to a point in the building. Then, we compute the 3D shadow volume for the satellite/building pair by extending the building in the shadow direction. Finally, we compute the 2D GNSS shadow for the satellite/building pair, which is a subset of the AOI in which the receiver does or does not lie depending on the received  $C/N_0$ .

Note, we have made a key assumption that the shadow direction can be treated as a single direction, as opposed to a cone of possible directions, because the distance from the satellite to any building is much larger than the size of the building. However, for a more accurate model, or if the satellite is at a low elevation angle that would produce a large shadow, we can instead consider a cone of shadow directions created by the convex hull of  $s_j$  and  $B_i$  using Equation (4). Since this convex hull strategy requires a more complicated presentation of the algorithm, one can discard low-elevation satellites while implementing ZSM.

### 3.2.2 | Computing Shadow Directions

We compute the shadow direction  $\ell_{i,j}$  as a unit vector from the satellite position  $s_j$  to a point  $b_i \in B_i$ . Recall from Equation (2) that each building is represented as a union of constrained zonotopes. Recall also that we assume the shadow direction is the same for any  $b_i \in B_i$ . So, we choose  $b_i$  as the mean of the vertices of all of the constrained zonotopes comprising  $B_i$ .

To get the vertices of  $B_i$ , we use the following procedure. Recall that  $B_i$  is a union of constrained zonotopes. Using techniques from Althoff (2015) and Matt (2021), one can convert a constrained zonotope  $Z = \text{zono}(c, G, A, b) \subset \mathbb{R}^n$  to a vertex representation as follows. Suppose  $Z$  has  $m$  generators (i.e.,  $G \in \mathbb{R}^{n \times m}$ ). Notice from Equation (1) that  $Z$  is an affine map of the polytope  $P = \{x \in \mathbb{R}^m \mid \|x\|_\infty \leq 1 \text{ and } Ax = b\}$ . That is, we can write  $Z = c + GP$ . To get a vertex representation of  $Z$ , we first apply the work of Matt (2021) to enumerate the vertices of  $P$  and get a set  $W \subset \mathbb{R}^m$  for which  $P = \text{convhull}(W)$ . Then, the vertices of  $Z$  are vertices of the set  $\text{convhull}(\{c + Gw \mid w \in W\}) \subset \mathbb{R}^n$ , which we compute using Althoff (2015).

We summarize the above procedure with a function that applies to the union of constrained zonotopes comprising  $B_i$ . Let  $\mathcal{Z} = \{Z_i\}_{i=1}^{n_z}$  be a set of constrained zonotopes. Then:

$$\{v_i\}_{i=1}^{n_{\text{verts}}} = \text{vertices}(\mathcal{Z}) \quad (6)$$

is a vertex representation of the polytope (not necessarily convex) created by the union  $\bigcup_{i=1}^{n_z} Z_i$ .

Now, consider the set  $\{v_k\}_{k=1}^{n_{\text{vert}}} = \bigcup_{l=1}^{n_z} \text{vertices}(Z_l)$ , where  $Z_l$  is as in Equation (2). We estimate the point  $b_i$  as  $b_i = \frac{1}{n_{\text{vert}}} \sum_{k=1}^{n_{\text{vert}}} v_k$ . Finally, we compute the shadow direction as:

$$\ell_{i,j} = \frac{s_j - b_i}{\|s_j - b_i\|_2} \quad (7)$$

Next, we use  $\ell_{i,j}$  to compute a building's shadow as a volume in 3D space.

### 3.2.3 | Computing Shadow Volumes

To compute a shadow volume for satellite/building pair  $(i, j)$ , we extend  $B_i$  in the shadow direction  $\ell_{i,j}$ . This extension is achieved by taking the Minkowski sum of the building and the shadow direction.

Since the building and shadow direction are constrained zonotopes, to proceed, we now define the Minkowski sum of constrained zonotopes. Consider the example zonotopes  $Z_1 = \text{zono}(c_1, G_1, A_1, b_1)$  and  $Z_2 = \text{zono}(c_2, G_2, A_2, b_2) \subset \mathbb{R}^n$ . Suppose that  $G_1 \in \mathbb{R}^{n \times m_1}$ ,  $G_2 \in \mathbb{R}^{n \times m_2}$ ,  $A_1 \in \mathbb{R}^{k_1 \times m_1}$ , and  $A_2 \in \mathbb{R}^{k_2 \times m_2}$ . Note,  $k_1$  is the number of constraints for  $Z_1$  and, similarly,  $k_2$  for  $Z_2$ , as per Equation (1). Per Scott et al. (2016), the Minkowski sum of constrained zonotopes is:

$$Z_1 \oplus Z_2 = \text{zono}(c_1 + c_2, [G_1, G_2], A_{\oplus}, b_{\oplus}), \quad A_{\oplus} = \begin{bmatrix} A_1 & \mathbf{0}_{k_1 \times m_2} \\ \mathbf{0}_{k_2 \times m_1} & A_2 \end{bmatrix} \quad b_{\oplus} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (8)$$

Note, this follows from the definition of constrained zonotopes.

Now we can compute the shadow volume. First, consider the zonotope:

$$L_{i,j} = \text{zono}(\mathbf{0}_{3 \times 1}, \epsilon \cdot \ell_{i,j}, [], []) \quad (9)$$

where  $\epsilon$  denotes a scaling factor that is greater than at least the height of the tallest building in the set  $\mathcal{B}$  (in practice, we use  $\epsilon = 10^5$  meters). In other words,  $L_{i,j}$  is a zonotope representing a long line segment extending in the shadow direction  $\ell_{i,j}$ . Then, the shadow volume of building  $B_i$  with respect to satellite  $j$  is:

$$V_{i,j} = B_i \oplus L_{i,j} = \bigcup_{l=1}^{n_i} \{Z_l \oplus L_{i,j}\} \quad (10)$$

where  $Z_l$  are the constrained zonotopes comprising  $B_i$  as in Equation (2). In practice, we represent the union of constrained zonotopes as a list of constrained zonotopes; so, we hold on to each:

$$V_l = Z_l \oplus \ell_{i,j} \quad (11)$$

where we have omitted the  $i, j$  indices from  $V_l$  for ease of notation.

### 3.2.4 | Computing GNSS Shadows

Finally, to compute a 2D GNSS shadow for the satellite/building pair  $(i, j)$ , we intersect the shadow volume with the 2D AOI.

Recall that the shadow volume and the AOI are represented by constrained zonotopes. Again let  $Z_1 = \text{zono}(c_1, G_1, A_1, b_1)$  and  $Z_2 = \text{zono}(c_2, G_2, A_2, b_2) \subset \mathbb{R}^n$  with  $G_1 \in \mathbb{R}^{n \times m_1}$ ,  $G_2 \in \mathbb{R}^{n \times m_2}$ ,  $A_1 \in \mathbb{R}^{k_1 \times m_1}$ , and  $A_2 \in \mathbb{R}^{k_2 \times m_2}$ . Per Scott et al. (2016), the intersection of constrained zonotopes is a constrained zonotope:

$$Z_1 \cap Z_2 = \text{zono}(c_1, [G_1, \mathbf{0}_{n \times m_2}]A_{\cap}, b_{\cap}), \quad A_{\cap} = \begin{bmatrix} A_1 & \mathbf{0}_{k_1 \times m_2} \\ \mathbf{0}_{k_2 \times m_1} & A_2 \\ G_1 & -G_2 \end{bmatrix} \quad \text{and} \quad b_{\cap} = \begin{bmatrix} b_1 \\ b_2 \\ c_2 - c_1 \end{bmatrix} \quad (12)$$

Therefore, for the shadow volume  $V_l$  and area of interest  $\mathcal{A} = \{A_k\}_{k=1}^{n_{\text{AOI}}}$ , we can compute a shadow  $S_{i,j}$  as:

$$S_{i,j} = \bigcup_{k=1}^{n_{\text{AOI}}} (V_l \cap A_k) \quad (13a)$$

$$= \bigcup_{k=1}^{n_{\text{AOI}}} \bigcup_{l=1}^{n_i} (\{Z_l \oplus L_{i,j}\} \cap A_k) \quad (13b)$$

where we use Equation (10) to expand  $V_{i,j}$  into a union of constrained zonotopes. As mentioned earlier, we represent a union of constrained zonotopes as a list of constrained zonotopes. So, in practice, we consider each  $k$ -th shadow

$$S_k = (Z_l \oplus L_{i,j}) \cap A_k \quad (14)$$

which is a single constrained zonotope corresponding to each  $A_k$ . We omit the  $i, j, l$  indices for ease of notation.

### 3.3 | Algorithm Details

We now describe how to initialize and perform ZSM, summarized in Algorithm 1. This is a modification of the conventional SM approach originally introduced by Groves (2011). In particular, we enable set-valued receiver position estimates. We emphasize again that this is a snapshot method, considering only the data available at a single time instance.

#### 3.3.1 | Algorithm Overview

We begin with the entire AOI  $\mathcal{A} \subseteq \mathcal{G}$  as an initial 2D set-valued estimate of our receiver position (Line 2). If we have no localization information available besides the fact that we are in an urban area, then we set  $\mathcal{A} = \mathcal{G}$ . Then, for each GNSS satellite (Line 3), we perform the following steps:

1. First, for each building (Line 5), we compute the 3D shadow volume as in Equation (10; Lines 6–9). Then, we intersect the shadow volume with the AOI to find a 2D GNSS shadow (Line 11), which we efficiently convert to a generic vertex representation (Line 12).
2. Second, we concatenate the vertex-represented regions of 2D GNSS shadows across all the buildings (Line 13).
3. Third, if the  $C/N_0$  value for the current satellite is below a user-specified threshold, then the current satellite is NLOS, so we intersect the concatenated 2D GNSS shadow with our current set-valued position estimate (Line 15 and Figure 6[c]); otherwise, the satellite is LOS, so we subtract the GNSS shadow from our set-valued position estimate (Line 17 and Figure 6[d]). We assume the initial AOI (Line 2) to contain the true receiver position and, hence, the intersection of GNSS shadow with our current set-valued position estimate is never an empty set.

Finally, after iterating through all buildings and satellites, the algorithm output is a polygon (2D polytope) on the ground plane representing the set-valued estimate of receiver positions (potentially with multiple disjoint components) given

**ALGORITHM 1**
 $P = \text{ZSM}(\mathcal{B}, \mathcal{S}, \mathcal{R}, \bar{r}_{\text{LOS}}, \mathcal{A}, \epsilon)$  // zonotope shadow matching (Snapshot)

---

```

1 /* Inputs:
  • Urban 3D map of buildings  $\mathcal{B} = \{B_i\}_{i=1}^{n_{\text{bldg}}}$  indexed by  $i$ , with each  $B_i$  as a set of
  constrained zonotopes  $\{Z_l\}_{l=1}^{n_i}$ 
  • Observed SVs  $\mathcal{S} = \{s_j\}_{j=1}^{n_{\text{sat}}}$  indexed by  $j$ 
  • Received  $C/N_0$  values  $\mathcal{R} = \{r_j\}_{j=1}^{n_{\text{sat}}}$  for each satellite
  • Prespecified  $C/N_0$  threshold  $\bar{r}_{\text{LOS}}$ 
  • Area-of-interest  $\mathcal{A} = \{A_k\}_{k=1}^{n_{\text{AOI}}}$  indexed by  $k$  and represented as a set of constrained
  zonotopes
  • Shadow length scale  $\epsilon \approx 10^5$  m.
*/
2  $P \leftarrow$  vertices( $\mathcal{A}$ ) // initialize coarse set-valued position estimate as a polygon (2D polytope)
3 for each  $s_j \in \mathcal{S}$  do
4    $C_k = \emptyset$  // initialize 2D GNSS shadow as an empty polytope in vertex representation
5   for each  $B_i \in \mathcal{B}$  do
6      $\ell_{i,j} \leftarrow$  makeShadowDirection( $B_i, s_j$ ) // make shadow direction as in Equation (7)
7      $L_{i,j} \leftarrow$  zono( $\mathbf{0}_{3 \times 1}, \epsilon \cdot \ell_{i,j}, [], []$ ) // make shadow direction zonotope as in Equation (9)
8     for each  $Z_l \in B_i$  do
9        $V_l \leftarrow Z_l \oplus L_{i,j}$  // construct shadow volume zonotope as in Equation (11)
10      for each  $A_k \in \mathcal{A}$  do
11         $S_k \leftarrow V_l \cap A_k$  // create 2D GNSS shadow using zonotope intersection as in
        Equation (14)
12         $S_k \leftarrow$  vertices( $\{S_k\}$ ) // convert 2D GNSS shadow to vertex representation with
        Equation (6)
13         $C_k \leftarrow C_k \cup S_k$  // concatenate vertex-represented regions of 2D GNSS shadow
        across buildings
14      if  $r_j < \bar{r}_{\text{LOS}}$  then
15         $P \leftarrow P \cap C_k$  // in shadow (NLOS), so intersect shadow with estimated position
16      else
17         $P \leftarrow P \setminus C_k$  // not in shadow (LOS), so subtract shadow from estimated position
18 return  $P$  // updated set-valued receiver position estimate

```

---

the current snapshot of GNSS signals (Line 18). An illustration of ZSM in simulation is shown later in Figure 6.

### 3.3.2 | Computational Considerations

We present Algorithm 1 without parallelization for ease of exposition. However, one can parallelize Algorithm 1 as follows. When iterating over each  $i$ -th building and each  $j$ -th satellite, one can create a separate position estimate  $P_{i,j}$  for each  $(i, j)$  pair, then intersect or subtract each  $P_{i,j}$  from the initial position estimate  $P$  in Line 2 by checking if the corresponding building/satellite pair is NLOS or LOS as in Lines 14–17. Note, we use the unparallelized version of Algorithm 1 in our numerical experiments.

Note that efficient numerical tools, such as the Multi-Parametric Toolbox (Herceg et al., 2013) and MATLAB’s polyshape, exist to perform polytope intersection and subtraction operations in 2D (Lines 11–14). However, to enable leveraging these 2D polygon tools, we require the constrained zonotope intersection operation as in Equation (12) to intersect the 3D building shadow zonotopes with the 2D ground plane zonotopes.

## 4 | EXPERIMENTAL RESULTS

We experimentally validated our proposed ZSM method in simulation. Particularly, we chose a simulated platform to conduct exhaustive analysis of our algorithm's performance. First, we performed a preliminary experiment to illustrate the utility of our constrained zonotope representation. Then, we evaluated ZSM in comparison to conventional SM (Groves, 2011; Wang et al., 2013a) for two simulation experiments.

Our ZSM implementation was in MATLAB. We used the open-source Continuous Reachability Analyzer (CORA) toolbox (Althoff, 2015) to represent constrained zonotopes. We used the polyshape tool to perform set operations on 2D GNSS shadows. We utilized received  $C/N_0$  from only GPS; however note that our algorithm is generalizable to other GNSS constellations as well. As explained earlier in Section 1, we consider an ideal LOS/NLOS classifier in this work and, thus, do not account for satellite misclassifications.

### 4.1 | Preliminary Experiment: Minkowski Sum Evaluation

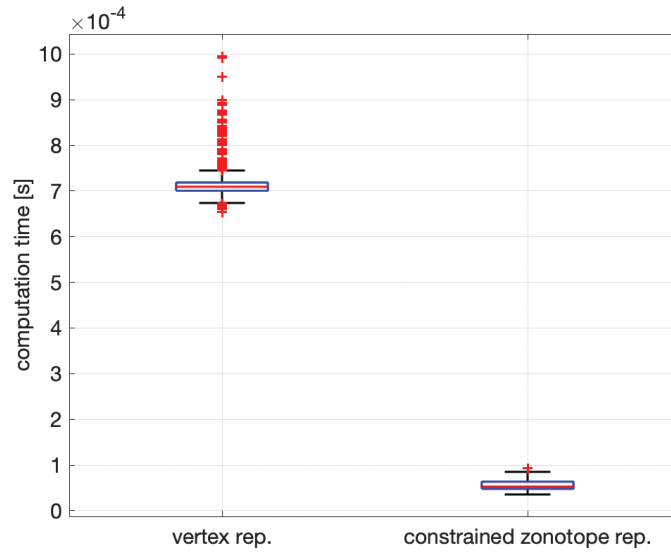
To illustrate the utility of using constrained zonotopes, we compared the speed of the Minkowski sum on 1,000 randomly-generated polytopes with up to 100 vertices, which we represented as both a standard vertex representation and as constrained zonotopes. Recall that a convex polytope  $P \subset \mathbb{R}^n$  can be represented as the convex hull of a set of vertices  $V \subset \mathbb{R}^n$ :  $P = \text{convhull}(V)$  as in Section 3.1.2. We used the vertex representation because 3D urban maps are typically represented as a triangulation of vertices as noted in Section 3.1.2. In other words, this comparison considered an alternative to ZSM wherein we would apply Algorithm 1 directly to a 3D map without converting it to constrained zonotopes.

We performed the Minkowski sum of both representations with a line segment to mimic the process of computing a building shadow zonotope and compared the average computation time as shown in Figure 4. We implemented the vertex representation with MPT (Herceg et al., 2013) and the constrained zonotope representation with CORA (Althoff, 2015). This experiment was performed on a laptop computer with an 8-core 2.4 GHz CPU and 32 GB RAM.

The constrained zonotope representation of this operation (executed once) was an order of magnitude faster on average (71.3 ms for MPT vs. 5.5 ms for CORA) and had a smaller standard deviation (3.3 ms for MPT vs. 1.1 ms with fewer outliers for CORA). Since this Minkowski sum was performed multiple times (specifically,  $n_{\text{sat}} \times \prod_{i=1}^{n_{\text{bidg}}} n_i$  times) in Algorithm 1, we see that the ZSM formulation enables a huge speed increase over using other polytope representations for SM. We also note that the Minkowski sum for a constrained zonotope with a line segment always resulted in one additional generator per Figure 4, whereas it is unclear how many additional vertices might be required for the vertex representation. So it is possible to preallocate memory for the constrained zonotope to further increase performance.

### 4.2 | Comparison Method and Validation Metrics

We validated our proposed ZSM via two simulation experiments: one using a simple 3D map comprised of two buildings and the other using a dense 3D building



**FIGURE 4** Constrained zonotopes improve Minkowski sum computation time by an order of magnitude over a standard vertex representation (each red plus denotes an outlier).

map of San Francisco. We compared ZSM’s performance with that of conventional SM, which was implemented in Groves (2011) and Wang et al. (2013a, 2013b). All computations were performed on a laptop with a 2-core 2.5 GHz CPU and 8 GB RAM. In these simulation experiments, we modeled each building as a constrained zonotope (i.e.,  $n_i = 1$  in Equation [2]).

#### 4.2.1 | Details of Conventional SM

We applied the conventional SM technique presented in Wang et al. (2013a) and explained earlier in Section 1. Offline, the method considers a prespecified uniform grid of position candidates and then performs a precomputation step by tracing evenly-distributed azimuth and elevation rays for each position candidate and each building in a 3D map to compute predicted satellite visibility. The number of discrete candidates in the uniform grid is determined by its grid size, which is defined as the distance between any two candidates either in cross-street or along-street directions. Note that the distance between any two position candidates in the along-street direction is the same as in cross-street, thus each grid represents a square.

At runtime (online computation), conventional SM computes a visibility score at each position candidate by comparing the received  $C/N_0$  (used for LOS/NLOS classification) for each satellite to that of predicted satellite visibility (from offline computation). Then, the position candidates with the highest visibility scores (can be more than one) are identified as the most likely receiver positions. The conventional method also computes the weighted empirical covariance (Martens et al., 2003) by analyzing the position candidates and their associated weights based on the normalized visibility scores.

To analyze ZSM performance for a given initial AOI, we precomputed the visibility map of conventional SM for various grid sizes. As mentioned earlier in Section 1, a finer discretization is attained with a lower grid size, thus increasing the position accuracy of the conventional SM technique, but at the expense of higher computation cost (both offline and online).

## 4.2.2 | Validation Metrics

We applied four validation metrics:

1. **Offline computation load:** For ZSM, this is the time required for converting buildings from vertices to constrained zonotopes. For conventional SM, this is the time required for precomputing the visibility map at each position candidate, and, thus, the predicted satellite visibility.
2. **Online computation load:** For ZSM, this is the time required to run Algorithm 1. For conventional SM, this is the time required to compute the visibility scores at all position candidates and compute the most likely position candidates (based on highest visibility scores) and the weighted empirical covariance.
3. **Point-valued estimation error in cross-street and along-street directions:** For ZSM, this is the error in cross-street and along-street directions between the true receiver location and centroid of the final set-valued position estimate (multiple centroid values obtained when disjoint components are present). For conventional SM, this is the error in cross-street and along-street directions between the most likely position candidates (highest visibility score) and the true receiver location.
4. **Bounds in cross-street and along-street directions:** For ZSM, this is the width of the bounding box that encloses the set-valued position estimate. Note that we report the width of individual bounding boxes when disjoint components are present in the final set-valued position estimate. For conventional SM, this is twice the  $3\sigma$  (three standard deviations) bound of the visibility-based weighted empirical covariance in the cross-street and along-street directions. In particular, we evaluated the weighted empirical covariance across all the position candidates based on their visibility scores.

## 4.3 | Simulated Experiment #1: Simple 3D Building Map

We first describe a simulation experiment that considers a simple 3D map comprised of two urban buildings. We use the real-world GPS data collected from an Android phone and, then, emulate the received  $C/N_0$  values with simulated NLOS effects.

### 4.3.1 | GPS Data Set and NLOS Emulation

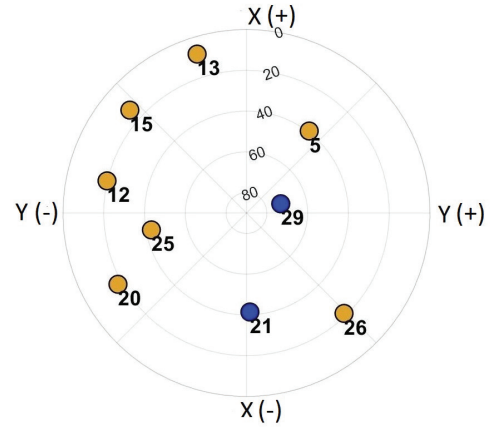
We used the openly available Google Android data set (Fu et al., 2020) with the data collection setup shown in Figure 5(a). To allow control over LOS/NLOS signals for each satellite, we used the data collected in open-sky conditions via a Pixel 4 XL Modded smartphone, which logged the GPS data, and a high-fidelity GNSS-RTK/IMU setup, which served as a reference ground truth.

To design an ideal LOS/NLOS classifier, we emulated NLOS effects as follows. First, we overlaid the test surroundings with simulated buildings to mimic an urban setting. We then chose a time when nine satellites were LOS, as judged by  $C/N_0$  values above 38-dB-Hz, which is representative of open-sky conditions based on the empirical studies conducted in Hetet (2000) and Kuusniemi et al. (2004). We induced simulated NLOS effects in a GPS satellite when the LOS vector between its





(a) Pixel 4 XL Modded smartphone used for data collection



(b) GPS satellite skyplot

**FIGURE 5** Subfigure (a) is the experiment test platform utilized to validate the proposed ZSM via publicly-available Google Android data sets in Fu et al. (2020). Subfigure (b) shows the skyplot for a particular receiver location with nine visible GPS satellites. The GPS satellites are represented by circles with LOS satellites in blue and NLOS (simulated effects) in dark yellow.

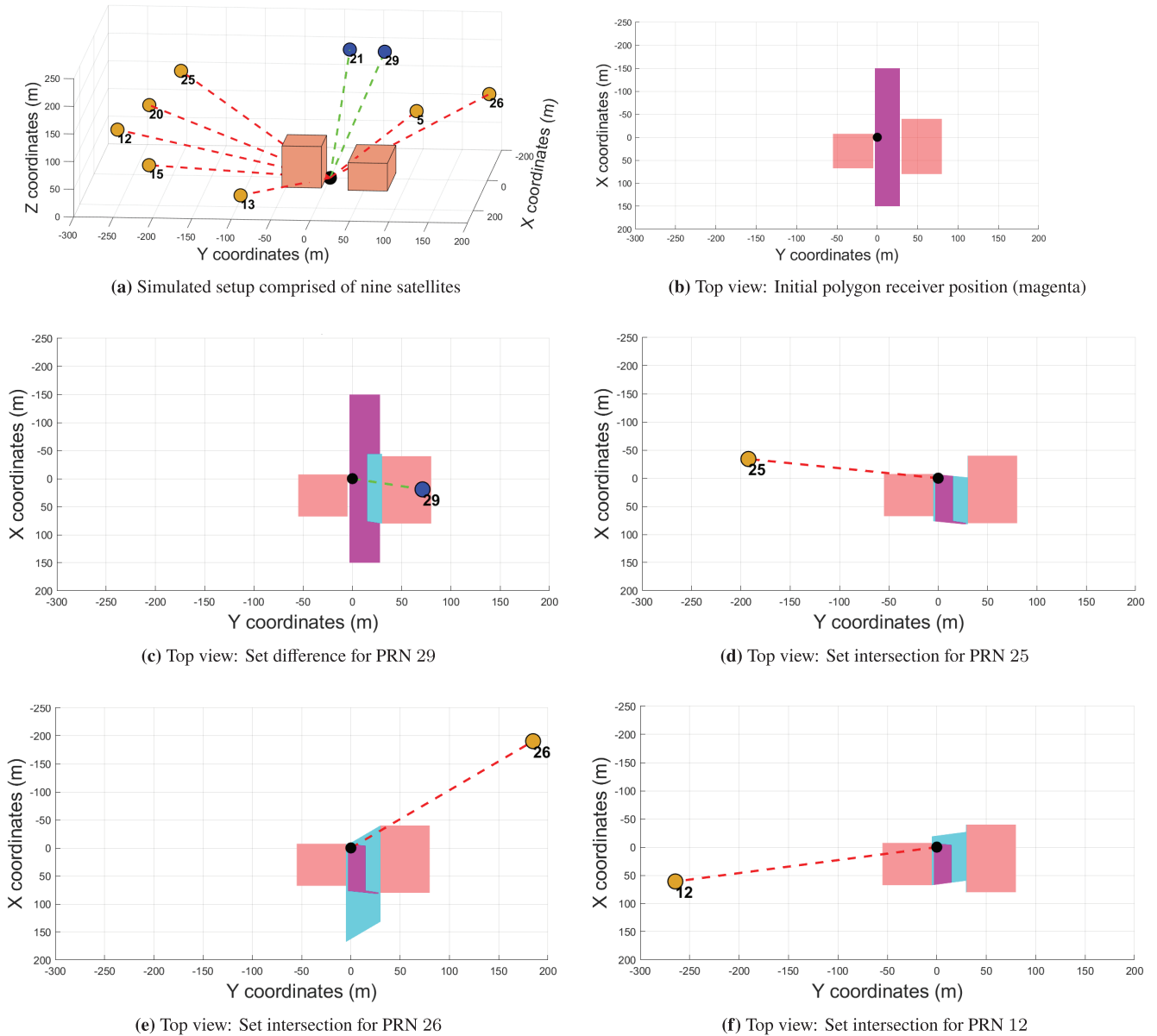
location and the true receiver position was obstructed by any simulated building. In particular, we attenuated the received  $C/N_0$  values for these identified NLOS satellites to be below the 38-dB-Hz threshold. Given that we are considering an ideal LOS/NLOS classifier that is based on binary thresholding against 38 dB-Hz, we do not account for the emulation and inclusion of multipath effects in this work.

### 4.3.2 | Experiment Setup

We set up the experiment as follows, also shown in Figure 6(a). For the urban 3D map, we considered two buildings. The receiver 2D position was initialized at (0, 0) m in local map coordinates and the initial AOI was chosen as the entire length and width of a street between the two buildings, shown later in Figure 6(b). We observed a total of nine visible GPS satellites whose skyplot is shown in Figure 5(b). As mentioned earlier, we induced simulated errors in received  $C/N_0$  to represent LOS/NLOS characteristics with respect to the true receiver position and 3D map. In Figure 5(b), the NLOS satellites are indicated by dark yellow circle markers whereas LOS ones are blue. For conventional SM, we considered three grid sizes with the distances between the discrete candidates in cross-street and along-street directions as follows: 15 m, 10 m, and 5 m. The number of position candidates for the grid sizes of 15 m, 10 m, and 5 m, were 63, 124, and 427, respectively.

### 4.3.3 | Results and Discussion

The first few steps (top-down view) of Algorithm 1 are shown in Figures 6(b)–6(f). In particular, we illustrate the satellites, buildings, initial AOI, GNSS shadows, and intersection/subtraction procedure. The full results are summarized in Table 1 and the final localization results for proposed ZSM and conventional SM are illustrated in Figure 7(a) and Figure 7(b). For each grid size of conventional SM reported in Table 1, we reported the top three most-likely position candidates (ranked based on the highest visibility score and the least point-valued estimation error with respect to ground truth).



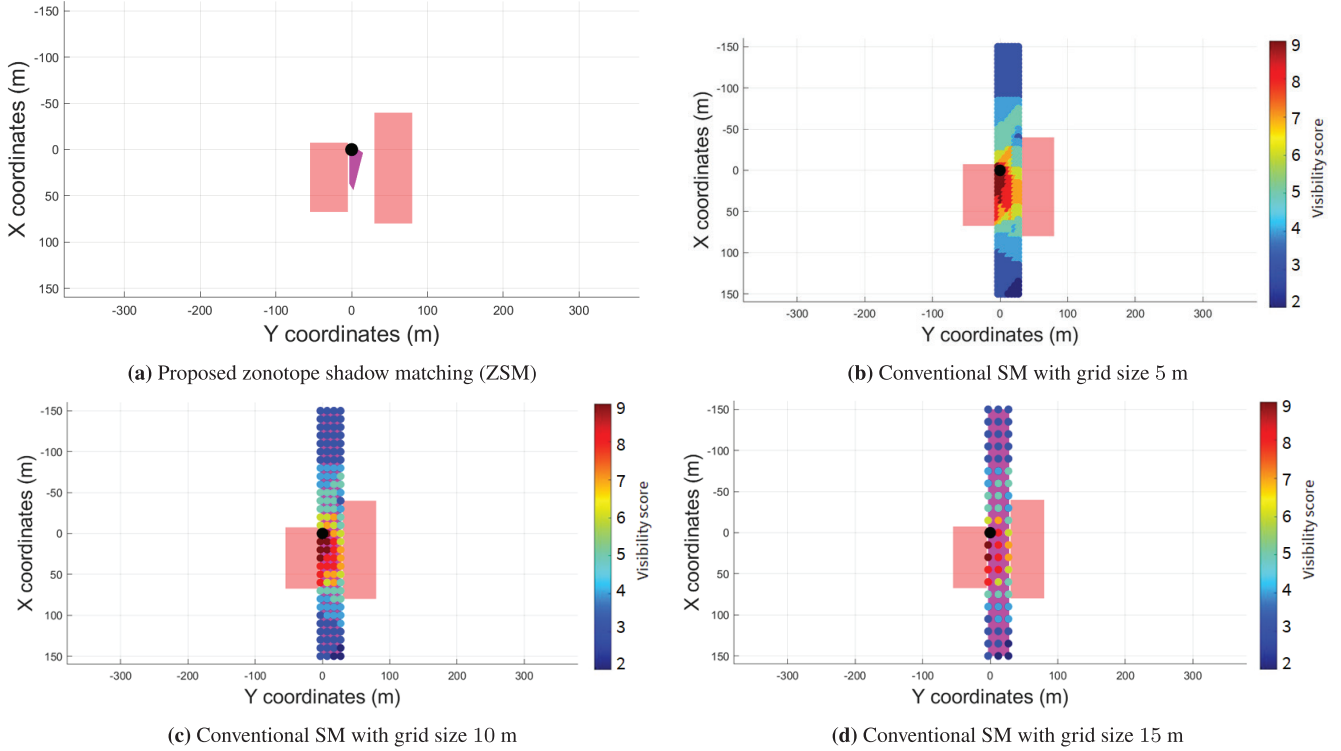
**FIGURE 6** Subfigure (a) is an illustration of the experiment setup, while subfigures (b)–(f) represent iterations of the proposed ZSM algorithm. The magenta area is the receiver position estimate and the blue areas are shadows.

Figure 6(a) shows the simulated setup with two urban buildings and nine GPS satellites. The black circle represents the true receiver position, blue circles represent LOS satellites, and LOS signals are represented by a green dashed line. NLOS satellites are represented by dark yellow circles with NLOS signals represented by a red dashed line. Figure 6(b) shows a top-down view with the initial AOI in magenta. Figures 6(c)–6(f) show the GNSS shadow for each satellite or building pair in cyan with the resulting receiver position estimate after the set intersection or difference in magenta. While we only illustrate the procedure for PRNs 29, 25, 26, and 12, the final result from all nine satellites is shown in Figure 7. Note that the satellites are plotted near the receiver for visualization only; the shadow directions as in Equation (7) were computed using the satellites' actual positions. Given the received  $C/N_0$  and 3D map, we demonstrated that the proposed ZSM computes a set-valued receiver position estimate.

**TABLE 1**

Performance Comparison Results for ZSM vs Conventional SM on a Simulation Experiment With two Buildings Seen in Figure 6(a) and Nine GPS Satellites seen in Figure 5(b).

Algorithm	Error w.r.t true location (m) [Cross-street, Along-street]	Bound (m) [Cross-street, Along-street]	Avg. computation load across 100 runs (s)	
			Offline	Online
<b>Proposed ZSM</b>	[3.46, 16.05]	[17.87, 50.11]	1.7e-4	0.39
<b>5 m</b>	[3.00, 5.00]	[59.58, 447.48]	1524.41	1.66
	[2.00, 0.00]			
	[3.00, 0.00]			
<b>Conventional SM with grid sizes</b>	[3.00, 0.00]	[66.72, 455.82]	405.33	0.08
	[7.00, 0.00]			
	[3.00, 10.00]			
<b>10 m</b>	[3.00, 0.00]	[73.54, 463.10]	296.32	0.04
	[3.00, 15.00]			
	[3.00, 30.00]			



**FIGURE 7** The ZSM position estimate in subfigure (a) has a comparable estimation error to the conventional SM results in subfigures (b)-(d). Note that the ZSM error does not depend on a predefined grid density.

Figure 7(a) shows the ZSM receiver position estimate in magenta. Figures 7(b)–7(d) show the conventional SM with grid sizes of 5 m, 10 m, and 15 m, respectively, where the position candidates are color-coded using a jet colormap with blue indicating a low visibility score of 2 and red representing a high value of 9 (number of available GPS satellites). For a lower online computation time of 0.39 s, ZSM demonstrates a comparable point-valued estimation error and a smaller width than the densest conventional SM case in both the cross-street and along-street directions.

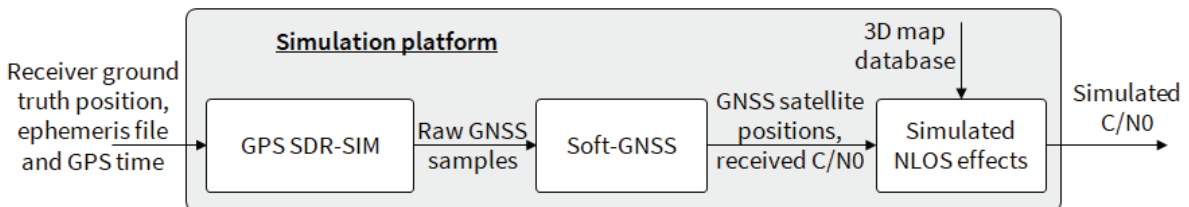
The results from the three cases of the conventional SM technique were used for validating the improved performance of our proposed ZSM algorithm. We saw a comparable point-valued estimation accuracy of ZSM (i.e., 3.46 m and 16.05 m in cross-street and along-street directions, respectively) as that of conventional SM with grid sizes of 5 m, 10 m, and 15 m, whose accuracies are reported in Table 1. Furthermore, unlike all the conventional SM cases, we showed that ZSM achieved a set-valued position estimate with an accuracy in width of only 17.87 m in the cross-street direction and 50.11 m in the along-street direction. We also found that the offline and online computations of ZSM only incurred an average runtime of  $1.7 \times 10^{-4}$  s and 0.39 s, respectively. Importantly, without the need for gridding, the ZSM result produced this point-valued positioning accuracy while returning a set-valued estimate, which contained the true receiver position. We anticipate that the result might be more accurate if we leveraged more GPS satellites and buildings that would provide additional geometry constraints in the shadow matching, thus reducing the centroid error and bounds of the disjoint component (ambiguous mode) containing the true receiver location, if any. This forms the premise of our next simulation experiment.

#### 4.4 | Simulated Experiment #2: Using 3D Building Map of San Francisco

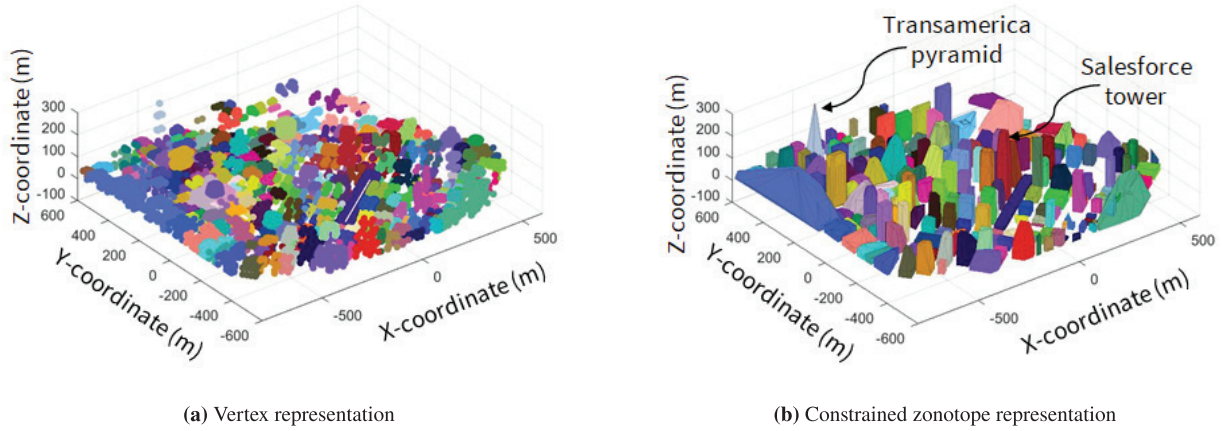
We performed our next simulation experiment using a publicly-available 3D building map of San Francisco. For this dense 3D urban map, we validated our algorithm's performance compared to that of the conventional SM technique using emulated GPS data. We also performed a sensitivity analysis of proposed ZSM by varying the number of buildings being considered.

##### 4.4.1 | GPS Data Set and NLOS Emulation

We generated GPS data using the simulation pipeline shown in Figure 8. Considering a desired static position (true receiver location), ephemeris file, and start time as inputs, we utilized a C++ language-based software-defined GPS simulator known as GPS-SIM-SDR (Bhamidipati et al., 2019; Ebinuma, 2018) to simulate the raw GPS samples, which are indicative of open-sky conditions. Later, we performed acquisition and tracking using a MATLAB-based software-defined radio known as SoftGNSS (Rojas, 2011) to generate received  $C/N_0$  values. Similar to NLOS emulation described in Section 4.3.1, we induced NLOS effects in



**FIGURE 8** Our GPS software simulation pipeline uses two publicly-available software: GPS-SDR-SIM (Ebinuma, 2018) and SoftGNSS (Rojas, 2011). For a given true receiver location, we simulated the  $C/N_0$  values and, thereafter, induced NLOS effects using a 3D building map of San Francisco.



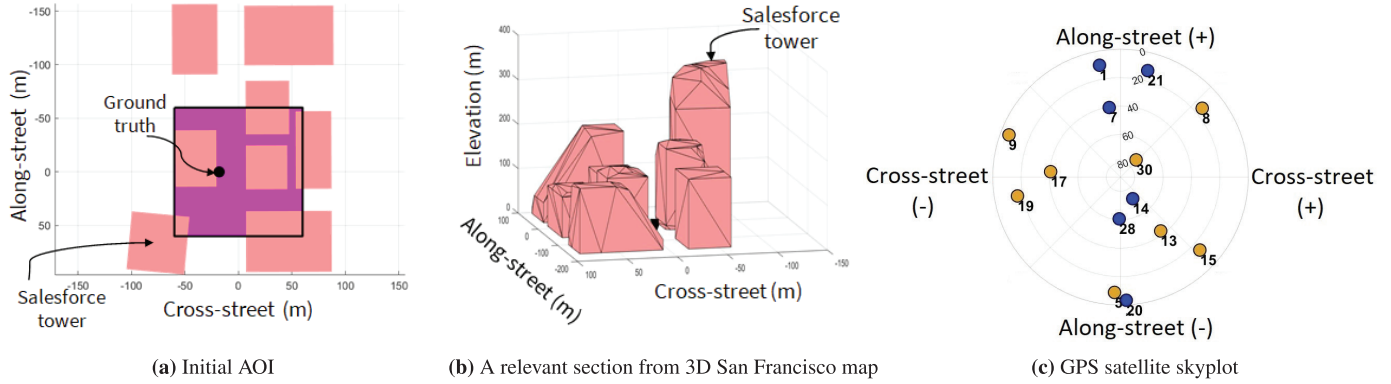
**FIGURE 9** Preprocessing 3D building map of San Francisco: (a) shows the vertex representation, wherein the vertices of each building are separately color-coded; and (b) represents each building as a constrained zonotope, which is given as input to our proposed ZSM algorithm.

$C/N_0$  values based on the simulated buildings in San Francisco and true receiver location.

#### 4.4.2 | Experiment Setup

We utilized a publicly available 3D building map of San Francisco (György, 2018). As explained in Equation (3) of Section 3.1.2, standard 3D building maps are represented as union of triangles, with each triangle comprised of three vertices. An illustration of the vertex representation for the 3D building map of San Francisco is shown in Figure 9(a), wherein we isolated the vertices of each building using the theory of connected graphs (König, 1990) and color coded them independently. We then converted the entire 3D map from vertex representation to constrained zonotope representation, which is visualized in Figure 9(b). Note that the prominent landmarks in San Francisco, such as the Transamerica pyramid and Salesforce tower, can be easily identified in Figure 9(b). The storage space required for building-constrained zonotopes is quite less. For instance, the .mat file size for the eight buildings discussed in Section 4.4.4 is 5 KB while, for 14 buildings, it is 8 KB, and for 20 buildings, it is 11 KB. Similarly, the entire San Francisco map (shown in Figure 9[b]) is 345 KB. While this is acceptable for many practical applications, based on the user platform specifications, if further reduction in file size is required, one could employ alternate techniques such as storing a reduced order representation of building-constrained zonotopes or storing a smaller section of the map at the original order of the building-constrained zonotopes. Note that the experiments conducted in this section were fully simulated and, therefore, inaccuracies of this 3D building map compared to that of the San Francisco city are not relevant. This is because we simulated the true receiver location and classified the satellites as LOS/NLOS relative to this open-source 3D building map.

The true receiver position was initialized at  $(0, -18)$  m in local map coordinates. The initial AOI was chosen to lie within a size of  $120 \text{ m} \times 120 \text{ m}$  while excluding the regions that were within the building footprints. In particular, we chose the size of the AOI in a heuristic manner by penalizing the position solution and uncertainty bounds ( $3\sigma$ ) estimated by the MATLAB-based SoftGNSS (Rojas,



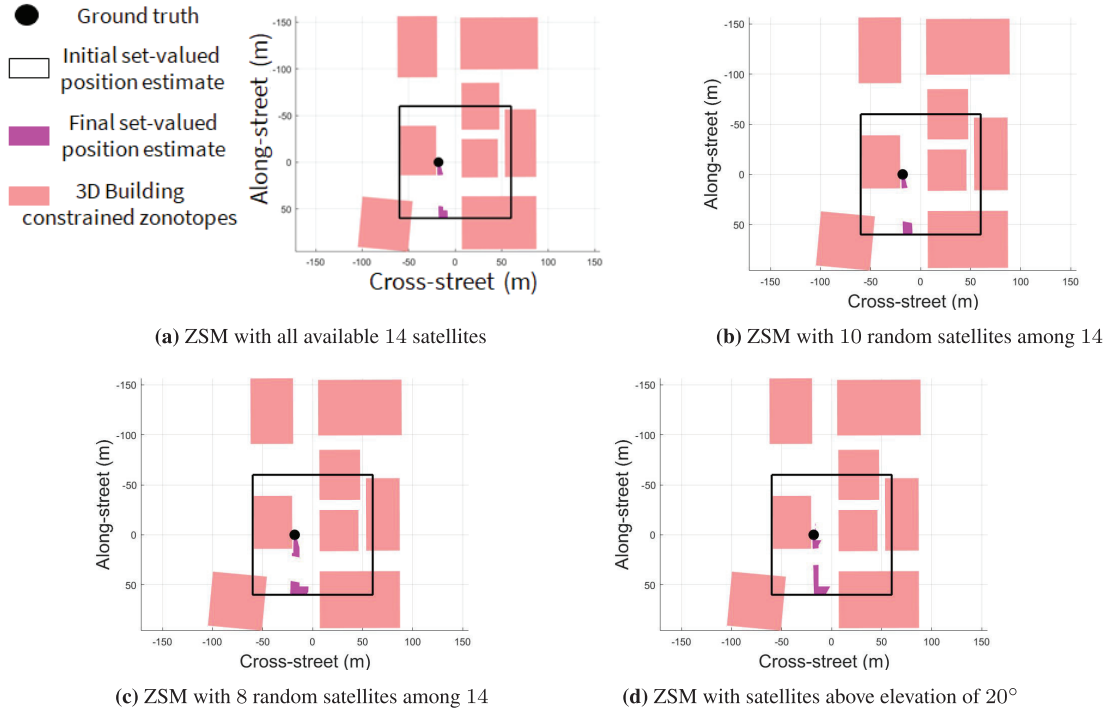
**FIGURE 10** Our second simulation experiment using a 3D map of San Francisco: Subfigures (a) and (b) show the initial AOI and surrounding buildings, while (c) shows the skyplot for the true receiver position with LOS satellites in blue and simulated NLOS in dark yellow.

**TABLE 2**  
Performance Comparison Between ZSM and Conventional SM Based on a 3D Map of San Francisco

Algorithm	Centroid error w.r.t true location (m) [Cross-street, Along-street]	Bounds (m) [Cross-street, Along-street]	Average computation load across 100 runs (s)		
			Offline	Online	
Proposed ZSM	14 satellites	[1.2, 8.1] [4.6, 58.8]	[5.8, 16.2] [9.8, 24.5]	162.5	4.4
	10 random satellites	[0.9, 6.8] [4.8, 53.8]	[6.1, 16.4] [9.7, 13.1]	162.5	4.2
	8 random satellites	[1.4, 13.1] [3.5, 54.3]	[18.3, 26.8] [7.6, 24.8]	162.5	3.0
	Satellites with elevation > 20°	[1.7, 10.2] [1.9, 7.0] [4.9, 48.5]	[0.5, 3.3] [8.7, 20.2] [16.8, 29.9]	162.5	2.4
Conventional SM (all available 14 satellites)	3 m	[0.0, 3.0], [0.0, 6.0], [0.0, 9.0] [0.0, 12.0], [3.0, 12.0], [0.0, 48.0] [3.0, 48.0], [0.0, 51.0], [3.0, 51.0] [0.0, 54.0], [3.0, 54.0], [3.0, 57.0]	[173.2, 214.5]	91204.0	2.8
	5 m	[2.0, 10.0] [2.0, 50.0], [3.0, 50.0], [3.0, 55.0]	[173.1, 215.6]	34389.9	1.2
	10 m	[2.0, 10.0] [2.0, 50.0]	[192.1, 221.6]	6874.1	0.3
	30 m	[12.0, 60.0]	[224.5, 261.8]	1433.4	0.1

Note: For proposed ZSM, we performed comparison analysis across different GPS satellite subsets, while for conventional SM we compared the performance for different grid resolutions. The 14 satellites can be seen in Figure 10(c)

2011), which was explained earlier in Section 4.4.1. A top-down view of this illustration is shown in Figure 10(a) and a relevant section of the 3D San Francisco map comprised of eight buildings (of which one of them was the Salesforce tower) is shown in Figure 10(b). We simulated 14 GPS satellites whose skyplot is shown in Figure 10(c), with LOS satellites indicated by blue circle markers and NLOS by dark yellow. For comparison with the conventional SM technique (explained in

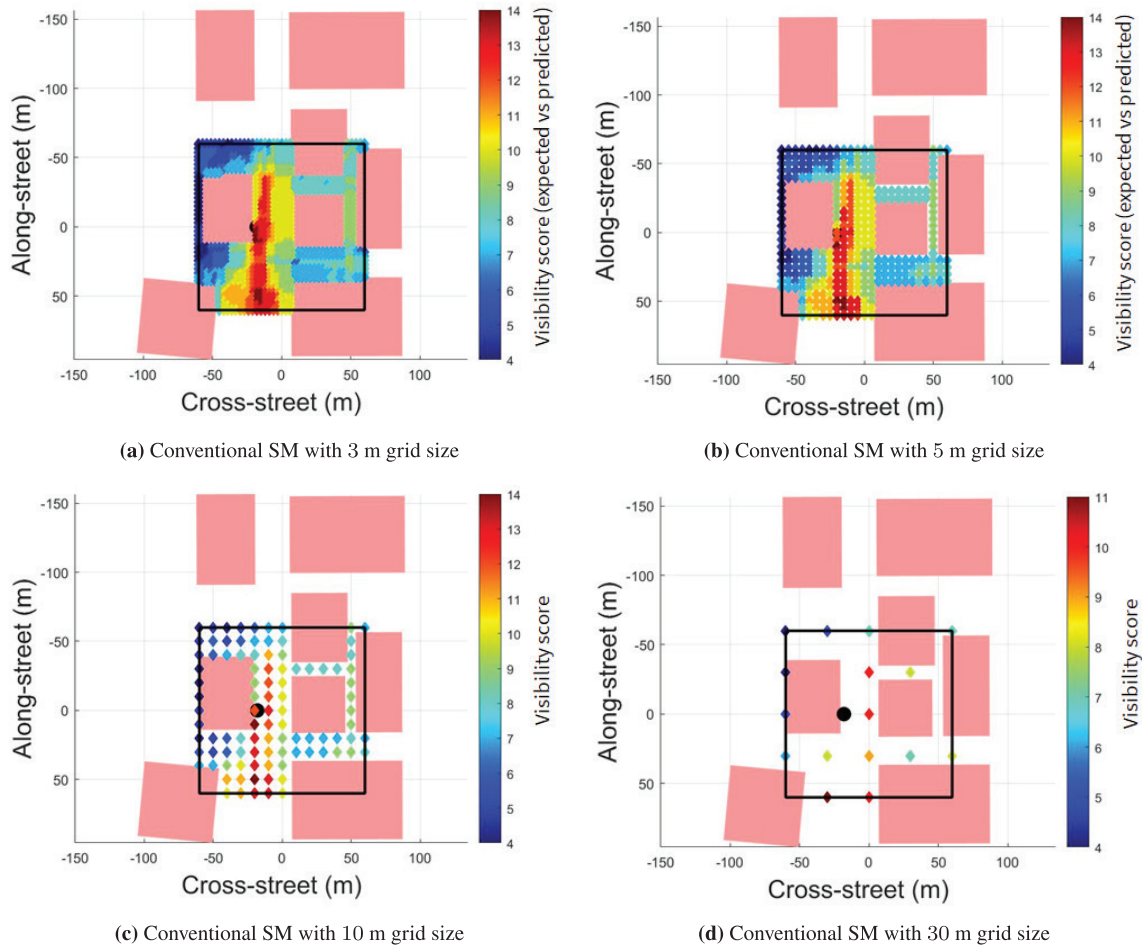


**FIGURE 11** Our proposed ZSM algorithm estimates the receiver position with high precision (i.e., the area of the receiver estimate set is small), but produces a bimodal output in this example. The mode containing the true receiver position is not majorly affected by the consideration of different subsets of the available GPS satellites.

Section 4.2), we considered four grid sizes of 30 m, 10 m, 5 m, and 3 m. The number of position candidates for the grid sizes of 30 m, 10 m, 5 m, and 3 m, were 16, 97, 346, and 902, respectively.

#### 4.4.3 | Comparison Results with Conventional SM

Table 2 reports comparison statistics between the proposed ZSM with different satellite subsets and conventional SM with varying grid sizes in terms of offline and online computation load, point-valued estimation error with respect to true receiver location, and the associated position bounds. A visualization of our ZSM with all available 14 satellites, a random subset of 10 satellites, a random subset of 8 satellites, and satellites with elevation  $> 20^\circ$  are shown in Figures 11(a), 11(b), 11(c), and 11(d), respectively. From among those illustrated earlier in Figure 10(c), the satellites with PRNs 1, 7, 14, and 15 were eliminated for the 10 random satellite case, while the satellites with PRNs 5, 7, 8, 9, 17, and 28 were omitted for the 8 random satellite case. Also, for the case with elevation  $> 20^\circ$ , only the GPS satellites with PRNs 7, 13, 14, 17, 28, and 30 were considered. Intuitively, the reduced number of GPS satellites mimic the cases when a non-ideal LOS/NLOS classifier is in a simulation setting, wherein the users can execute the proposed ZSM by leveraging only the GPS satellites exhibiting a high probability of being either LOS or NLOS while omitting the less certain ones. The detailed steps of convergence for our proposed ZSM algorithm for all available 14 satellites can be viewed here: <https://youtu.be/an1h4hd3ikw>. Similarly, the illustrations of conventional SM with grid sizes of 3m, 5 m, 10 m, and 30 m are shown in Figure 12(a), 12(b), 12(c), and 12(d), respectively.



**FIGURE 12** The results of conventional SM strongly depend upon grid size (a denser grid is more accurate). However, in this case, the uncertainty bounds in the conventional SM are over 100 m, indicating that ZSM is much more precise.

We validated that the proposed ZSM using all available 14 satellites successfully estimated two disjoint sets that showed a resemblance with the most-likely position candidates estimated using conventional SM of different grid sizes. For the conventional SM technique with a grid size of 30 m, we observed one most-likely candidate (highest visibility score) that showcased a point-valued accuracy of 12 m and 60 m in the cross-street and along-street directions, respectively. By implementing the conventional SM technique for a grid size of 10 m, we identified two most-likely candidates, both of which exhibited point-valued estimation error in a cross-street direction of 2 m. Similarly, conventional SM with grid sizes of 5 m and 3 m identified 1–3 grid points with a high visibility score of 14 in the near-neighborhood of each of the two disjoint sets estimated via proposed ZSM. While increasing the grid resolution reduces the estimated bounds of conventional SM, we observed that these bound values were more than 100 m across all grid sizes in both along-street and cross-street directions. These large uncertainty bounds from conventional SM can be majorly attributed to the use of Gaussian distribution for approximating the non-linear, multi-modal shadow matching distribution.

In contrast, across all satellite subsets, our proposed ZSM successfully estimated the set containing the true receiver location as one of the outputs, without requiring discretizing. Particularly, when all 14 available satellites were utilized,



ZSM demonstrated a higher point-value accuracy of 1.2 m in cross-street and 8.1 m in along-street direction, while achieving a small bound of only 5.8 m in cross-street and 16.2 m in along-street directions. We observed that, as the number of GPS satellites utilized in our proposed ZSM increased, the centroid error and the bounds of the mode that contained the true receiver location marginally decreased. Similarly, as the number of utilized GPS satellites increases, the offline computation load remains the same, but the online computation load marginally increases.

An interesting thing to note here is that, while ZSM estimates multiple disjoint components (modes), the ones that do not contain the true receiver location can be ruled out by fusing proposed ZSM with other information sources, say inertial measurement units (IMUs) or GPS pseudoranges. In particular, these additional sources could provide temporal information about the vehicle motion that could be used to perform a consistency check (i.e., compute the predicted/expected modes from IMU/GPS pseudoranges), and compare it with the set-valued estimate from zonotope shadow matching to eliminate incorrect modes over time. For instance, IMU data can be utilized to perform consistency checks over multiple time instants in the relative domain, itself, while considering the IMU biases and drifts to be constant over a relatively shorter time window. Similarly, GPS pseudoranges from satellites identified as LOS with high certainty can provide useful information, particularly in resolving the along-street component, thereby narrowing down the correct mode. Also, note that parallelization can further reduce the computational load of both the proposed ZSM and conventional SM.

#### 4.4.4 | Sensitivity Analysis of ZSM

We performed a sensitivity analysis of ZSM to evaluate the increase in offline and online computation loads as the level-of-detail for each building increased or alternatively the number of buildings increased. As explained in Section 3.1.2, standard 3D maps can be represented as a union of triangles, wherein each triangle is converted into a constrained zonotope. Therefore, in theory, the computational load incurred by increasing the number of buildings (with sparser level of detail for each building) is on the same order of magnitude as increasing the level of detail for each building, given a lower number of buildings. This is justified by comparing Equations (2) and (3) as follows: (a) given  $n_{\text{trng}}$  triangles and  $n_i = n_{\text{trng}}$ , the total number of constrained zonotopes representing the buildings are  $n_{\text{trng}}$  (each building is sparser in the level of detail); and (b) given another  $n_{\text{trng}}$  triangles with  $n_i < n_{\text{trng}}$ , the total number of constrained zonotopes representing the buildings would be  $n_i$  (each building encompasses more level of detail). Given that the open-source 3D building map (György, 2018) is not dense enough to conduct a sensitivity analysis on level of detail explicitly, we analyzed the performance of the proposed ZSM as the number of buildings are varied. We also quantified our ZSM's performance by analyzing the following: (a) centroid error and bounds associated with the disjoint component (ambiguous mode) containing the true receiver location; and (b) number of ambiguous modes and their variation as the number of buildings increases.

As shown in Table 3, we analyzed our ZSM's performance by increasing the number of buildings from eight (discussed in Section 4.4.3) to 14 and 20. For all three cases, we considered the same initial AOI, which is indicated by the black box shown in Figure 13. Figure 13 illustrates the final set-valued position estimate obtained for the case with 20 buildings. We validated that our proposed ZSM

TABLE 3  
ZSM Estimation Accuracy and Computation Time vs. Building Density

No. of buildings	Centroid Error @ Bound (m) [Cross-street, Along-street]	Avg. computation load for 100 runs (s)	
		Offline	Online
8	a) [1.2, 8.1] @ [5.8, 16.2] b) [4.6, 58.8] @ [9.8, 24.5]	162.5	4.6
14	a) [1.2, 8.1] @ [5.8, 16.2] b) [4.6, 58.8] @ [9.8, 24.5] c) [102.4, 63.9] @ [4.9, 25.2]	165.4	9.4
20	a) [1.2, 8.1] @ [5.8, 16.2] b) [4.6, 58.8] @ [9.8, 24.5] c) [102.4, 63.9] @ [4.9, 25.2] d) [1.7, 10.1] @ [0.4, 3.3] e) [87.0, 155.9] @ [0.3, 0.5]	169.8	12.2

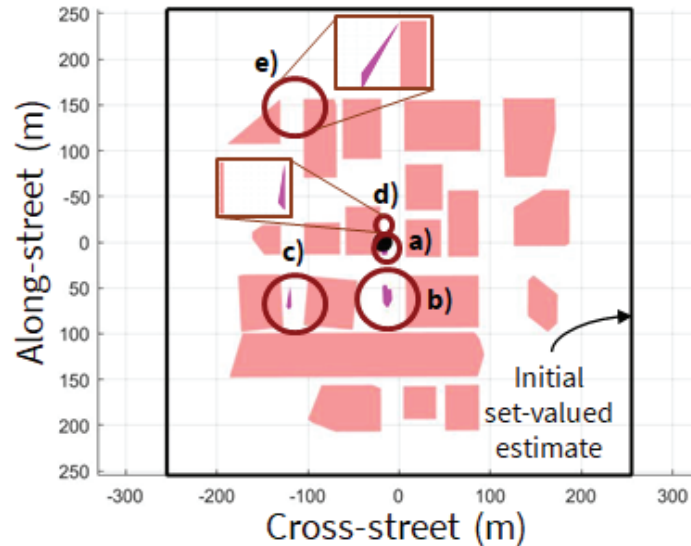


FIGURE 13 ZSM produces a set-valued estimate with many disjoint components (magenta) when considering many buildings (pink) and a large AOI. However, each disjoint component is small, so fusing a ZSM estimate with other estimates to eliminate multi-modality can produce an accurate receiver position estimate.

technique was able to successfully detect all the disjoint components (ambiguous modes) and their exact bounds, which can vary based on the number, density, and placement of buildings considered. From Table 3, we observe that the number of disjoint components in the final set-valued position estimate increases as the number of buildings increases, wherein the number of disjoint components is two when eight buildings are considered, three disjoint components for 14 buildings, and five disjoint components when 20 buildings are considered. As explained earlier, disjoint components not containing the true receiver location can be omitted by fusing proposed ZSM with other sources, such as IMUs and GPS pseudoranges. We demonstrated that our proposed ZSM successfully maintained a high estimation accuracy and a small position bound in both cross-street and

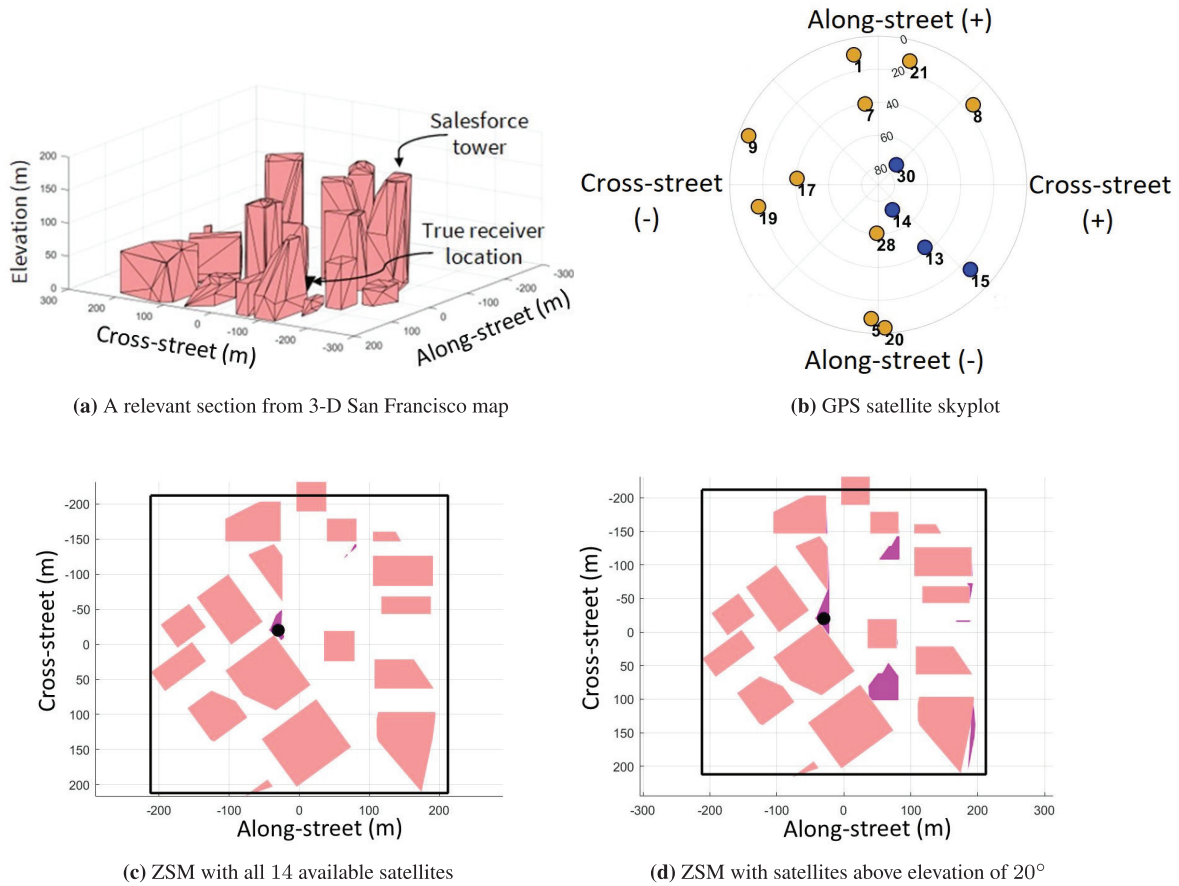
along-street directions. Another interesting point to note is that the eight-building and 14-satellite configuration already provides sufficient geometric diversity and, thus, increasing the number of buildings to 14 and 20 does not further reduce the centroid error and bound of the ambiguous mode containing the true receiver location.

Based on the results, we observed that the offline computation load (even with no parallelization as of now) demonstrated easy scalability with a number of buildings (i.e., no significant increase in computation time). While the online computation load increases with the number of buildings, note that there is an extensive scope for parallelization in the extraction of GPS shadows that can drastically reduce the computation time. For instance, one can parallelize the processing of analyzing each pair of constrained zonotope (associated with 3D buildings) and satellite to extract the GPS shadow, which is currently executed in a sequential manner as explained in Algorithm 1. However, note that parallelization incurs an increase in the computational resources required.

#### 4.4.5 | Performance Analysis of ZSM: Different Location

We evaluated our ZSM's performance in a different section of the 3D building map of San Francisco (i.e., near an intersection, where the distinction between the cross-street and along-street directions is not obvious). Given this, we labeled the directions as X and Y in the local map coordinates as shown in Figure 14(a), which comprised of 17 urban buildings in total. The true receiver position was initialized at  $(-20, -30)$  m in local map coordinates and the initial AOI (black box) was chosen to lie within a size of  $212.1 \text{ m} \times 212.1 \text{ m}$  while excluding the regions that lay within the building footprints. We considered 14 GPS satellites, which are same as those considered in earlier sections and whose skyplot is shown in Figure 14(b) with LOS satellites (a total of four) indicated by blue circle markers and NLOS by dark yellow. A visualization of our ZSM with all available 14 satellites and satellites with elevation  $> 20^\circ$  are shown in Figures 14(c) and 14(d), respectively. Among all the GPS satellites shown in Figure 14(b), for the case with elevation  $> 20^\circ$ , only the relevant six satellites with PRNs 7, 13, 14, 17, 28, and 30 were considered.

We demonstrated that our proposed ZSM estimated all the disjoint components (ambiguous modes). For the case in which all the GPS satellites are considered, three disjoint ones were identified with an online computation load of 10.9 s (no parallelization used). Similarly, for the case comprised of only the GPS satellites with elevation  $> 20^\circ$ , 13 disjoint ones were estimated using our proposed ZSM with an online computation load of 5.5 s. We also observed that, in both cases, our proposed ZSM successfully detected the mode that contained the true receiver location with a low centroid error of  $[0.9, 6.0]$  m and a bound of  $[21.4, 55.1]$  m in local map coordinates for the 14 GPS satellite case. On the other hand, a higher centroid error of  $[1.6, 10.8]$  m and bound of  $[21.4, 122.5]$  m in local map coordinates was observed for the six GPS satellite case. Unlike in Section 4.4.3 where increasing the number of satellites did not significantly affect the characteristics of the mode containing the true receiver location, in this case, we observed an improvement in our ZSM's performance. As explained before, this can be attributed to the further addition of useful geometric constraints in Figure 14(c) as compared to Figure 14(d) using the GPS shadows extracted from the remaining eight satellites.



**FIGURE 14** Our third simulation experiment in a different location using the 3D map of San Francisco; (a) shows a relevant section of 3D map near the Salesforce tower wherein the distinction between cross-street and along-street is not well-defined; (b) shows the skyplot with respect to the true receiver location with 14 visible GPS satellite, the GPS satellites are represented by circles with LOS satellites in blue and NLOS [simulated effects] in dark yellow; (c) shows the final set-valued estimate of the receiver position estimated by our ZSM when all 14 satellites are considered; and (d) shows this set-valued estimate using satellites with elevation  $> 20^\circ$ . We demonstrate that in both cases, our ZSM successfully detects the mode containing the true receiver location, while exhibiting a higher centroid error and bounds for the reduced six satellite case as compared to the full 14 satellite case.

## 5 | CONCLUSIONS

We presented zonotope shadow matching (ZSM), a novel approach to set-valued position estimation for 3D-map-aided GNSS. The method is achieved by leveraging constrained zonotopes, a recent advance in polytope representation. We computed the 3D buildings and 2D GNSS shadows using constrained zonotopes, and then refined the coarse set-valued receiver position estimate based on if the receiver lay inside or outside the GNSS shadow (which is judged by  $C/N_0$  values). Using simulated experiments on a simple 3D building map and a dense 3D map of San Francisco, we validated that the proposed ZSM achieves a high point-valued estimation accuracy with high certainty (small position bounds). Importantly, while achieving a comparable point-valued accuracy as that of the conventional shadow matching (SM) technique, ZSM can compute a set-valued state estimate that is independent of the point-valued discretization. We also demonstrated the easy scalability of ZSM with the number of buildings considered. This indicates that ZSM is a promising method for providing robustness guarantees for safety-critical GNSS applications, since such set-valued estimates can be used as measurements for set-valued robust estimators.

## ACKNOWLEDGEMENTS

We want to thank the National Science Foundation (NSF) and the Air Force Research Laboratory (AFRL) for funding this work. We also want to thank Adyasha Mohanty and Asta Wu for reviewing our paper.

## REFERENCES

- Adjrad, M., & Groves, P. D. (2017). Intelligent urban positioning: Integration of shadow matching with 3D-mapping-aided GNSS ranging. *Journal of Navigation*, 71(1). <https://doi.org/10.1017/s0373463317000509>
- Althoff, M. (2015). An introduction to CORA 2015. In G. Frehse & M. Althoff (Eds.), *ARCH14-15: 1st and 2nd International Workshop on Applied verification for continuous and hybrid systems* (Vol. 34, pp. 120–151). <https://doi.org/10.29007/zbkv>
- Althoff, M., & Dolan, J. M. (2014). Online verification of automated road vehicles using reachability analysis. *IEEE Transactions on Robotics*, 30(4), 903–918. <https://doi.org/10.1109/tro.2014.2312453>
- Bhamidipati, S., & Gao, G. X. (2020). Integrity-driven landmark attention for GPS-vision navigation via stochastic reachability. *Proc. of the 33rd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2020)*, 2311–2326. <https://doi.org/10.33012/2020.17546>
- Bhamidipati, S., Kim, K. J., Sun, H., & Orlik, P. V. (2019). GPS spoofing detection and mitigation in PMUs using distributed multiple directional antennas. *2019 IEEE International Conference on Communications (ICC)*, Shanghai. <https://doi.org/10.1109/icc.2019.8761208>
- Bhamidipati, S., Kousik, S., & Gao, G. (2021). Set-valued shadow matching using zonotopes. *Proc. of the 34th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2021)*, St. Louis, MO, 2373–2390. <https://doi.org/10.33012/2021.17933>
- Chen, Z., Gao, B., & Devereux, B. (2017). State-of-the-art: DTM generation using airborne lidar data. *Sensors*, 17(12), 150. <https://doi.org/10.3390/s17010150>
- Combettes, P. L., & Civanlar, M. R. (1991). The foundations of set theoretic estimation. *Proc. of the 1991 International Conference on Acoustics, Speech, and Signal Processing*, Toronto. <https://doi.org/10.1109/icassp.1991.151014>
- Ebinuma, T. (2018). GPS-SDR-SIM [Source code]. <https://github.com/osqzss/gps-sdr-sim>
- Fu, G. M., Khider, M., & van Diggelen, F. (2020). Android raw GNSS measurement datasets for precise positioning. *Proc. of the 33rd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2020)*, 1925–1937. <https://doi.org/10.33012/2020.17628>
- Groves, P. D. (2011). Shadow matching: A new GNSS positioning technique for urban canyons. *Journal of Navigation*, 64(3), 417–430. <https://doi.org/10.1017/s0373463311000087>
- Groves, P. D., Wang, L., Adjrad, M., & Ellul, C. (2015). GNSS shadow matching: The challenges ahead. *Proc. of the 28th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2015)*, Tampa, FL, 2421–2443. <https://www.ion.org/publications/abstract.cfm?articleID=12866>
- György, R. (2018). *3D Warehouse San Francisco*. <https://3dwarehouse.sketchup.com/model/4ad4796d-8102-4bdd-9bfc-5442dee9facf/San-Francisco>
- Xu, H., Angrisano, A., Gaglione, S., & Hsu, L. -T. (2020). Machine learning based LOS/NLOS classifier and robust estimator for GNSS shadow matching. *Satellite Navigation*, 1. <https://doi.org/10.1186/s43020-020-00016-w>
- Xu, H., Zhang, G., Xu, B., & Hsu, L. -T. (2018). GNSS shadow matching based on intelligent LOS/NLOS classifier. *The 16th World Congress of the International Association of Institutes of Navigation (IAIN)*, Chiba, Japan. [https://www.researchgate.net/publication/337825336\\_GNSS\\_Shadow\\_Matching\\_based\\_on\\_Intelligent\\_LOSNLOS\\_Classifier](https://www.researchgate.net/publication/337825336_GNSS_Shadow_Matching_based_on_Intelligent_LOSNLOS_Classifier)
- Herceg, M., Kvasnica, M., Jones, C. N., & Morari, M. (2013). Multi-parametric toolbox 3.0. *2013 European Control Conference (ECC)*, Zurich, Switzerland. <https://doi.org/10.23919/ecc.2013.6669862>
- Hetet, S. (2000). *Signal-to-noise ratio effects on the quality of GPS observations*. University of New Brunswick. <http://gauss.gge.unb.ca/papers.pdf/hetet.report.pdf>
- Hofmann-Wellenhof, B., Lichtenegger, H., & Collins, J. (1992). *Global positioning system*. Springer. <https://doi.org/10.1007/978-3-7091-5126-6>
- Iland, D., Irish, A., Madhow, U., & Sandler, B. (2018). *Rethinking GPS: Engineering next-gen location at Uber*. Uber Blog. <https://eng.uber.com/rethinking-gps>
- König, D. (1990). Theory of finite and infinite graphs. In D. König (Aut.), *Theory of finite and infinite graphs* (pp. 45–421). Birkhäuser. [https://doi.org/10.1007/978-1-4684-8971-2\\_2](https://doi.org/10.1007/978-1-4684-8971-2_2)
- Kousik, S., Holmes, P., & Vasudevan, R. (2019). Safe, aggressive quadrotor flight via reachability-based trajectory design. *ASME 2019 Dynamic Systems and Control Conference*, Park City, UT. <https://doi.org/10.1115/dscc2019-9214>

- Kuusniemi, H., Lachapelle, G., & Takala, J. H. (2004). Position and velocity reliability testing in degraded GPS signal environments. *GPS Solutions*, 8(4), 226–237. <https://doi.org/10.1007/s10291-004-0113-7>
- Martens, H., Høy, M., Wise, B. M., Bro, R., & Brockhoff, P. B. (2003). Pre-whitening of data by covariance-weighted pre-processing. *Journal of Chemometrics*, 17(3), 153–165. <https://doi.org/10.1002/cem.780>
- Matt, J. (2021). Analyze N-dimensional convex polyhedra. MathWorks (MATLAB). [https://www.mathworks.com/matlabcentral/fileexchange/30892-analyze-n-dimensional-convex-polyhedra?s\\_tid=srchtitle](https://www.mathworks.com/matlabcentral/fileexchange/30892-analyze-n-dimensional-convex-polyhedra?s_tid=srchtitle)
- Miura, S., Hisaka, S., & Kamijo, S. (2013). GPS multipath detection and rectification using 3D maps. *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, The Hague, Netherlands. <https://doi.org/10.1109/itsc.2013.6728447>
- Miura, S., Hsu, L.-T., Chen, F., & Kamijo, S. (2015). GPS error correction with pseudorange evaluation using three-dimensional maps. *IEEE Transactions on Intelligent Transportation Systems*, 16(6), 3104–3115. <https://doi.org/10.1109/tits.2015.2432122>
- Neamati, D., Bhamidipati, S., & Gao, G. (2022a). Mosaic zonotope shadow matching for risk-aware autonomous localization in harsh urban environments [Manuscript submitted for publication]. Stanford University. <https://stanford.box.com/shared/static/r0abx08okk5do2gbuhclhr1fuc4ayvij.pdf>
- Neamati, D., Bhamidipati, S., & Gao, G. (2022b). Set-based ambiguity reduction in shadow matching with iterative GNSS pseudoranges. *Proc. of the 35th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2022)*, Denver, CO. <https://www.ion.org/gnss/abstracts.cfm?paperID=11326>
- Raghuraman, V., & Koeln, J. P. (2022). Set operations and order reductions for constrained zonotopes. *Automatica*, 139. <https://www.sciencedirect.com/science/article/abs/pii/S0005109822000498>
- Rojas, C. P. P. (2011). *Soft GNSS* [Source code]. Github. <https://github.com/kristianpaul/SoftGNSS>
- Scott, J. K., Raimondo, D. M., Marseglia, G. R., & Braatz, R. D. (2016). Constrained zonotopes: A new tool for set-based estimation and fault detection. *Automatica*, 69, 126–136. <https://doi.org/10.1016/j.automatica.2016.02.036>
- Shetty, A., & Gao, G. X. (2020). Trajectory planning under stochastic and bounded sensing uncertainties using reachability analysis. *Proc. of the 33rd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2020)*, 1637–1648. <https://doi.org/10.33012/2020.17518>
- Shi, D., Chen, T., & Shi, L. (2015). On set-valued Kalman filtering and its application to event-based state estimation. *IEEE Transactions on Automatic Control*, 60(5), 1275–1290. <https://doi.org/10.1109/tac.2014.2370472>
- Shiryayev, V. I., & Podivilova, E. O. (2015). Set-valued estimation of linear dynamical system state when disturbance is decomposed as a system of functions. *Procedia Engineering*, 129, 252–258. <https://doi.org/10.1016/j.proeng.2015.12.045>
- Suzuki, T. (2016). Integration of GNSS positioning and 3D map using particle filter. *Proc. of the 29th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2016)*, Portland, OR, 1296–1304. <https://doi.org/10.33012/2016.14857>
- van Diggelen, F., & Wang, J. (2020). *Improving urban GPS accuracy for your app*. Android Developers Blog. <https://android-developers.googleblog.com/2020/12/improving-urban-gps-accuracy-for-your.html>
- Wang, L., Groves, P. D., & Ziebart, M. K. (2013a). GNSS shadow matching: Improving urban positioning accuracy using a 3D city model with optimized visibility scoring scheme. *NAVIGATION*, 60(3), 195–207. <https://doi.org/10.1002/navi.38>
- Wang, L., Groves, P. D., & Ziebart, M. K. (2013b). Urban positioning on a smartphone: Real-time shadow matching using GNSS and 3D city models. *Proc. of the 26th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2013)*, Nashville, TN, 1606–1619. <https://www.ion.org/publications/abstract.cfm?articleID=11339>
- Wang, L., Groves, P. D., & Ziebart, M. K. (2014). Smartphone shadow matching for better cross-street GNSS positioning in urban environments. *Journal of Navigation*, 68(3), 411–433. <https://doi.org/10.1017/s0373463314000836>
- Zhu, N., Marais, J., Bétaille, D., & Berbineau, M. (2018). GNSS position integrity in urban environments: A review of literature. *IEEE Transactions on Intelligent Transportation Systems*, 19(9), 2762–2778. <https://doi.org/10.1109/tits.2017.2766768>

**How to cite this article:** Gao, G., Bhamidipati, S., & Kousik, S. (2022). Set-valued shadow matching using zonotopes for 3D-map-aided GNSS localization. *NAVIGATION*, 69(4). <https://doi.org/10.33012/navi.547>