SkyScale: A Radio Tomographic Approach Towards Scaling UAV Network Deployments

Subrata Das and Ayon Chakraborty
Sense Lab, IIT Madras

ABSTRACT

A critical challenge in deploying UAV-based wireless networks is optimizing the UAV's position in aerial space to ensure robust connectivity. To achieve optimal positioning, the UAV must estimate the propagation loss for each ground terminal across the aerial space, also called Radio Environment Map (REM). Existing literature in this domain emphasizes accuracy enhancement of the REM estimations from sparse and noisy signal measurements – often leveraging deep learning techniques. However, these estimated REMs rapidly become outdated as UEs move, requiring repeated measurements and re-estimation. This introduces significant scalability challenges, especially in environments characterized by high UE mobility.

In this paper, we propose SkyScale, a first of its kind aerial network framework, that uses Radio Tomographic Imaging (RTI) to estimate the wireless attenuation characteristics of the underlying terrain. Such characteristics are fundamental properties of the terrain and are agnostic to UE locations, thus allowing us to estimate REMs for any arbitrary UE location. We evaluate SkyScale on our WiFi based UAV network testbed, traces from an LTE testbed, and realistic simulation studies. We demonstrate that our RTI-based SkyScale maintains REM accuracy within 3 dB while reducing measurement costs by 10× or more compared to state-of-the-art interpolation methods, making it highly scalable and adaptive to dynamic network conditions.

CCS CONCEPTS

• **Networks** → **Wireless access points, base stations and infrastructure**; *Network performance evaluation*; • **Human-centered computing** → Ubiquitous and mobile devices.

KEYWORDS

Aerial Communications, Network Deployment, Radio Environment Mapping, Radio Tomographic Imaging

ACM Reference Format:
Subrata Das and Ayon Chakraborty. 2024. SkyScale: A Radio Tomographic Approach Towards Scaling UAV Network Deployments. In *International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing (MobiHoc '24), October 14–17, 2024, Athens, Greece.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3641512.3686384

ACM ISBN 979-8-4007-0521-2/24/10
https://doi.org/10.1145/3641512.3686384

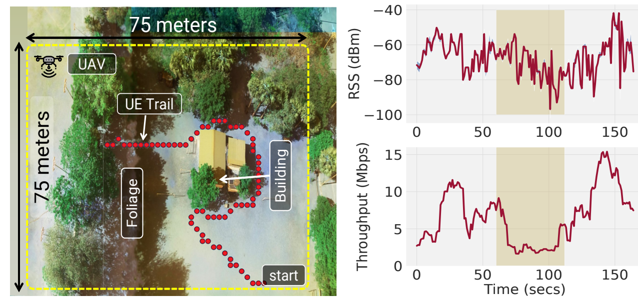

Figure 1: A top view of our deployment arena shows the UAV, acting as a WiFi base station at 40 meters height (top left). The figure illustrates the variation in Received Signal Strength (RSS) and TCP throughput (measured with `iperf3`) between a stationary UAV and a mobile UE following the marked trail. Note the 6–7× drop in throughput and ≈35 dB drop in RSS (shaded box) as the UE moves behind the building. To optimize performance for mobile UEs, the UAV must dynamically reposition based on real-time REM estimation.

1 INTRODUCTION

Recently, Unmanned Aerial Vehicle (UAV)-based platforms have emerged as a cost-effective and flexible solution for deploying wireless network infrastructure, particularly in ad-hoc or resource-constrained settings [9, 17, 27]. These UAVs function as *flying* base stations, providing wireless connectivity to clients (e.g., user equipments or UEs) located on the ground. A critical requirement for such deployments is to position the UAV in 3D airspace in a manner that optimizes the overall channel quality of the radio access network (RAN) across a set of spatially distributed UEs [24, 28]. Additionally, the UAV must continuously adjust its location in response to UE mobility [9]. This necessitates mapping the channel quality, such as signal-to-noise ratio (SNR) or received signal strength (RSS), throughout the aerial space for each UE on the ground. Such maps are commonly referred to as Radio Environment Maps or REMs. The problem of estimating REMs is extensively studied in the wireless literature, including empirical research specifically focused on optimizing UAV network deployments [9, 17, 27].

The goal is to determine the UAV's optimal operating point, which hinges on accurate REM estimation. It is important to note that only the uplink transmissions from the UEs to the UAV are relevant for REM estimation, as all wireless channel information must be available on the UAV side. At each point within the discretized aerial space, a *utility metric* is calculated by statistically aggregating the estimated REMs for each UE, such as by using the median RSS across the UEs. The point at which the utility metric assumes the highest value is regarded as the optimal operating position of the UAV. With slight variations in algorithmic details, most work done on aerial network deployments adopt more or less similar strategies [20, 21, 32]. For instance, some research focuses on designing efficient measurement trajectories to enhance REM accuracy [9, 17], while others aim to improve interpolation techniques for predicting patches within the REM that lack sufficient measurements [32]. A common constraint in all such optimization efforts is the limited flight endurance of UAVs, typically few tens of minutes. Therefore, it is preferable for UAVs to spend more time providing communication services rather that gathering measurements for REM estimation.

Lack of scalability. UAV positioning strategies are effective when UEs remain relatively static. However, as UEs move, REMs must be re-estimated, and the utility metric recalculated across the aerial space. In fig. 1, we demonstrate the effect of such UE movement on the network capacity (keeping the UAV stationary). This requires the UAV to gather fresh measurements corresponding to the re-positioned UEs, update the relevant REMs and move to its new optimal position. For a network with moderate UE dynamics, natural for any real deployment setting, such approaches clearly do not scale well. As the demand for UAV-based network deployments grows, addressing these scalability challenges is crucial for fully realizing the potential of this technology.

Our contribution in this paper is driven by a key insight that directly addresses the scalability challenge. While UE mobility causes REMs to vary significantly, the propagation environment or the terrain remains unchanged. By estimating the wireless attenuation characteristics of the terrain, we can predict REMs for *any arbitrary* UE position. Unlike existing methods that require fresh measurements for each new REM estimation, our approach leverages cumulative historical measurements (§5.1, see figs. 11 and 8). Although initial measurements are necessary, the need for additional measurements diminishes drastically with subsequent UE mobility, eventually approaching near-zero (§5.4, see figs. 7 and 10).

In this paper, we leverage the key observation outlined above to develop SkyScale, a robust and scalable REM estimation strategy that adapts to arbitrary UE mobility. SkyScale utilizes Radio Tomographic Imaging (RTI) to model the wireless attenuation properties of the surrounding environment. RTI based techniques discretize the 3D space into voxels and predict the attenuation coefficient for each such voxel. The attenuation coefficient denotes the extent by which a signal's power is degraded (aka, propagation loss) as it passes through such a voxel. The attenuation coefficients tagged onto each voxel in the 3D space make up what is referred as the *attenuation map* or *attenuation image*. This map enables the precise estimation of total wireless propagation loss for signals traveling from a UE to a UAV, facilitating accurate REM estimation

RTI techniques rely on a spatially distributed set of transmitter-receiver pairs at known locations, where each pair records the total propagation loss from the transmitter (UE) to the receiver (UAV). As previously mentioned, estimating REMs from the attenuation map is relatively straightforward, known as the *forward problem*). However, RTI attempts to solve the *inverse problem* – reconstructing the attenuation map from partial measurements of a few REMs (further details in §2.2). While RTI appears to be a promising approach, the vast number of voxels in 3D space poses significant challenges, making this solution computationally prohibitive. For instance, considering an area with dimensions 100 m×100 m and an overall height of 50 m, a resolution of 1 m results in half a million voxels! This is prohibitive not only in terms of the computation time but also in terms of onboard compute available on such UAVs (see fig. 3).

We deploy SkyScale on a real testbed consisting of a custom-built UAV hosting a WiFi network. Additionally, we evaluate the performance of SkyScale on (a) a real LTE dataset (SkyRAN [9]) and (b) two large scale 5G datasets obtained using the Nvidia Sionna [2], a state-of-the-art wireless PHY simulation framework. Overall, SkyScale can reduce the UAV's measurement effort by upto 10× when compared to state of the art interpolation based techniques (fig. 11). Also SkyScale by design leverages UE mobility to obtain spatially diverse measurements without requiring any movement of its own. We make the following key contributions in this paper.

- To our knowledge, *this is the first work* to harness RTI for optimizing ground-to-air channels in aerial networks. With SkyScale, the UAV adapts rapidly to dynamic network conditions, seamlessly supporting mobile UEs with *zero* additional overhead. Such diminishing overhead makes SkyScale superior to non-RTI approaches, ensuring scalability and sustainability for long-term deployments.
- We tackle the computational challenges of running RTI-based algorithms directly onboard the UAV. Our efficient processing of imagery data reduces the computational load by 50–100×, enabling real-time performance.
- We design and implement SkyScale prototype using a custom-built UAV that provides WiFi connectivity. Additionally, we demonstrate the effectiveness of SkyScale across diverse terrains and wireless traces. Compared to state-of-the-art approaches, SkyScale contains the REM estimation error with ≈3 dB, while reducing additional measurement data by 10× or more in the long run.

2 MOTIVATION AND BACKGROUND

2.1 Adapting REM Estimation to UE Mobility

As discussed earlier, much of the existing literature in this research area focuses on predicting REMs from scratch each time UEs relocate. These predictions often involve intelligent interpolation from newly obtained measurements, such as Gaussian Process Regression/Kriging [20], or rely on machine learning (ML) and deep learning (DL) techniques that handle sparser or noisier data. Additional cues, such as satellite or UAV imagery, are sometimes incorporated to enhance accuracy. While these methods are effective and produce nearly accurate REMs, a key question remains: *do historical measurements or trained models retain their relevance as UEs move?* A straightforward answer is *no*, as real-world terrains often experience significant REM changes due to non-line-of-sight (NLoS) blockages when a UE relocates (see fig. 1). Consequently, historical measurements are not *directly* effective in enhancing the accuracy of the current REM. In our deployment spanning a 75m × 75m terrain, we observe that ≈2 minutes worth of fresh in-flight measurements are necessary to reliably reconstruct the REM for a fixed set of UEs (median error ≤3 dB compared to a separately collected ground truth REM with more granular measurements). Given the limited endurance of UAVs – 20 minutes on a single battery cycle in our case – devoting a significant portion of flight time to gathering measurements is impractical.

We also explore popular deep generative models (e.g., U-Net) to evaluate their potential for REM estimation. We train a U-Net model similar to the one in [17], using UAV imagery of the terrain along with UE locations as inputs to predict the resulting REM. During training, we provide complete REM instances for seven different UE locations, encompassing ≈10K measurements from ≈20 minutes of flight time. However, we observe that the model was unable to generalize REM predictions for arbitrary UE locations, with a median prediction error of 8—10 dB. This outcome highlights a critical limitation: the substantial training overhead and data requirements undermine the scalability of UAV network deployments. Moreover, attempting to improve the U-Net model with additional measurements seems impractical and unjustified. In scenarios with significant UE churn or mobile UEs, such methods are poorly suited to updating REMs in real time and adaptively repositioning the UAV. Additionally, training such models exceeds the computational capacity of the onboard systems available on typical UAVs.

These challenges prompt us to seek a more effective representation primitive of the REM that can generalize across historical measurements and scale accordingly. One promising approach is to focus on the spatial attenuation characteristics of the terrain, which can be accurately estimated using Radio Tomographic Imaging (RTI) techniques.

2.2 Radio Tomographic Imaging (RTI) Primer

Wireless signals attenuate as they propagate through a medium, with significant losses occurring as they pass through physical objects such as buildings or foliage [14]. The extent of this attenuation is influenced by both the distance the signal travels and the material properties (e.g., permittivity) of the medium it traverses. To establish an analytical framework for RTI, we begin by discretizing the 3D space into N volume elements or voxels each with dimension δ^3 (where the resolution is δ). RTI then estimates the volumetric image, $\mathbf{X} \in \mathbb{R}^N$, where each $\mathbf{x_i} \in \mathbf{X}$ represents the attenuation coefficient of the i^{th} voxel. We define a 3D coordinate system with an arbitrary origin within the region of interest, such that each voxel corresponds to a single unit along each dimension. Fig. 2 shows a schematic of the setup – the UAV collects RSS measurements for each UE along its trajectory.

Let the k^{th} UE, denoted by UE_k, be located at L^k_{ue}. We assume that the UAV has knowledge of the UE locations. A common approach involves the mobile UAV estimating its distance or range to the UEs on the ground. This ranging process can be integrated directly into the communication protocol, for instance, WiFi Fine Time Measurement (FTM) [15]. As the UAV moves along its trajectory, it forms a synthetic aperture [9, 13], which can then be used to multilaterate and accurately determine the locations of individual UEs. The trajectory of the UAV is represented as $[L^1, L^2, L^3 \cdots]$, where L^n denotes the n^{th} voxel along the trajectory. The UAV's trajectory forms a synthetic aperture where, at each voxel, it records the received signal strength (RSS) from the set of available UEs. In such a setup, let $\mathbf{M}$ denote the measurement vector, where $m^k_n \in \mathbf{M}$ represents the RSS recorded for UE_k at L^n. For a given measurement m^k_n, the transmitter and receiver being located at L^k_{ue} and L^n

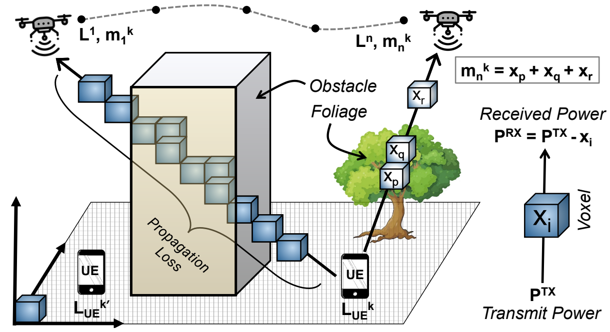

Figure 2: RTI schematic for SkyScale. The signal from each UE reaches the UAV through multiple voxels, each contributing to the propagation loss based on its attenuation coefficient, such as free space or foliage.

respectively, let $V_{k,n}$ denote the set of voxels traced by the straight line connecting them. We model m^k_n (i.e., the total propagation loss at L^n) as a sum of the attenuation coefficients of the voxels in $V_{k,n}$,

$$m^k_n = \sum_{j \in V_{k,n}} \mathbf{x_j} \tag{1}$$

For different UEs, the set $V_{k,n}$ varies (depending on k) and adds spatial diversity to the RSS measurements. Additionally, when a UE changes its location, say, from $L^{k_1}_{ue}$ to $L^{k_2}_{ue}$ we consider this to be equivalent to a *new* UE at location $L^{k_2}_{ue}$. We define a projection matrix $\mathbf{A} \in \{0,1\}^{R \times N}$, where R is the total number of measurements in $\mathbf{M}$ and N denotes the total number of voxels. Each row $\mathbf{a_i} \in \{0,1\}^N$ (an N-bit vector) corresponds to a single measurement from a specific UE, where, $\mathbf{a_{ij}} = 1$ if $j \in V_{k,n}$, else $\mathbf{a_{ij}} = 0$. Intuitively, each row of $\mathbf{A}_{R \times N}$ marks the voxels that contribute to the signal's attenuation corresponding to the measurement entry in the same row of $\mathbf{M}$. Note that each row of $\mathbf{A}/\mathbf{M}$ are independent and may correspond to any relevant UE (marked as '$*$' in $m^*_i \in \mathbf{M}$ in eqn. 2) for which measurements are performed.

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 1 & \cdots & 0 \\ 1 & 0 & 0 & 1 & \cdots & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}_{R \times N} \quad \mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}_{N \times 1} \quad \mathbf{M} = \begin{bmatrix} m^*_1 \\ m^*_2 \\ \vdots \\ m^*_R \end{bmatrix}_{R \times 1}$$

$$\mathbf{AX} = M \tag{2}$$

The system of linear equations in (2) are our RTI equations, the solution to which is the volumetric attenuation image ($\mathbf{X}$).

Inverse problem and ill-posedness. Given $\mathbf{A}$ and $\mathbf{X}$, it is straightforward to compute $\mathbf{M}$, aka, the *forward problem*. Rather, RTI solves the *inverse problem*, which is to estimate $\mathbf{X}$ from the measurement set and the projection matrix (computed from location traces). Since the RSS measurements are typically contaminated with noise, hence we can solve for $\mathbf{X}$ in the least squares sense,

$$\hat{X}_{ls} = \underset{X}{\arg\min} \, ||\mathbf{AX} - \mathbf{M}||^2 \tag{3}$$

The solution to eqn. 3 is equivalent to taking the *Moore-Penrose Pseudoinverse* on $\mathbf{M}$, i.e., $\hat{X} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{M}$. However, $\mathbf{A}$ needs to be full-rank which may not always hold good. This renders the problem *ill-posed*, where even slight variations in the input **M** cause the solution $\hat{X}$ to vary drastically. To contain the magnitude of variation, a regularization term is added to eqn. 3, shown in eqn. 4,

$$\hat{X}_r = \underset{X}{\arg\min} ||\mathbf{AX} - \mathbf{M}||^2 + \beta||L\mathbf{X}||^2 \tag{4}$$

Eqn. 4 is referred as the generalized Tikhonov regularization [29] and is a common tool for solving ill-posed inverse problems. $\beta > 0$ is a hyperparameter and L denotes the regularization operator. Note that such regularized inversion can also be expressed in the least squares form as,

$$\hat{X}_r = ((\mathbf{A}^T\mathbf{A} + \beta L^T L)^{-1}\mathbf{A}^T)\mathbf{M} \tag{5}$$

SkyScale adopts the solution demonstrated by eqn. 5 for estimating the attenuation image.

Computational challenges. To provide context, we present preliminary results on computational resource usage while running the vanilla RTI algorithm in our testbed environment, which covers a volume of 75 m×75 m×40 m. We vary the voxel dimension from 3.5 m down to 1 m and observe the computation time and the runtime memory usage for the UAV's onboard computer (a Raspberry Pi 4, 1.5 GHz clock with 8 GB main memory) and a ground station laptop computer (3 GHz clock with 16 GB main memory). As shown in fig. 3, the onboard computer is unable to process voxel dimensions smaller than 2.5 m.

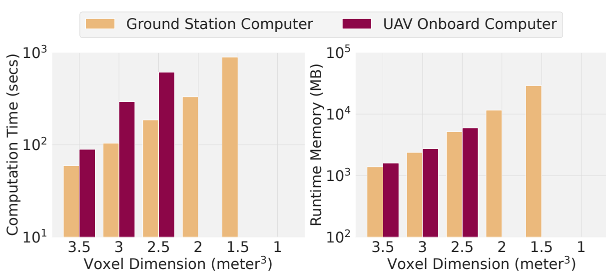

Figure 3: Benchmark results for computation time and runtime memory usage for running the vanilla RTI scheme (eqn.5) on the UAV's onboard Raspberry Pi computer and a well provisioned laptop. Due to memory constraints, the onboard computer is not able to run the solution if the voxel dimension is less than 2.5 m.

3 DESIGN OF THE SKYSCALE SYSTEM

SkyScale addresses the critical gap between the measurement overhead for re-estimating REMs as UEs relocate and the overall communication performance. Under the hood, SkyScale uses an RTI based approach with improved computational efficiency to estimate the RF attenuation image. To achieve such efficiency, we reduce the effective number of voxels in the RTI computation atleast by two orders of magnitude, based on the similarity among adjoining voxels. This enables SkyScale operate at real time on a modest amount of compute available onboard the UAV. The similarity among the adjoining voxels is estimated from the stereoscopic imagery of the underlying terrain taken from the UAV. Next, SkyScale computes a trajectory that is optimized to improve the attenuation image reconstruction for a fixed flight time budget. SkyScale's novelty lies in the fact that future measurements corresponding to relocated UEs will still be relevant and further improve the accuracy of the attenuation image. Overall, the availability of the attenuation image massively scales the REM estimation for any UE location, including mobile UEs. Fig. 4 shows a schematic of the different stages involved in SkyScale which we discuss in the following.

3.1 Dimensionality Reduction for RTI

As discussed in §2.2, the attenuation image $\mathbf{X} \in \mathbb{R}^N$ effectively captures the attenuation coefficient $\mathbf{x_i}$ for each of the N voxels in the 3D space. The vanilla RTI method, which solves this N dimensional problem, can become computationally prohibitive as N increases with finer resolution δ (see fig. 3). To address the issue of such bloated dimentionality, we leverage the UAV's stereoscopic imagery. Specifically, we compute a pixel-based depth map of the underlying terrain and perform image segmentation on it. These segments identify regions with similar obstacle characteristics, such as dense foliage, building tops, or open spaces.

We use the Watershed algorithm [18] on the depth map to perform segmentation. Watershed is a simple and relatively lightweight algorithm that simulates the flow of water within a terrain from higher to lower elevation. We found it particularly useful for segmenting elevation or depth maps. SkyScale is agnostic to any specific choice of the segmentation algorithm as long as it does not require site-specific training and the segmentation has a low computational footprint.

Segments in 3D space. Note that the above mentioned segments only identify 2D regions on the depth map. Voxels corresponding to such regions are marked as occupied in all 2D slices that make up the 3D volume, stacked up from the ground level to the average height of that particular segment. Such voxels are grouped together and mapped to specific 3D segments. Hereafter, *segment* implicitly refers to its 3D counterpart unless otherwise mentioned.

We assume that all voxels within a segment share the same average attenuation coefficient. This introduces a tradeoff between the number of segments and the computational resources required. Generally, smaller segment sizes result in a larger number of segments, improving the accuracy of the final REM but also increasing computational costs (see fig. 13). For instance, we observe that segments estimated by the Watershed algorithm offer a good balance between accuracy and computational efficiency. Segments corresponding to the maximum depth are treated as open or ground areas and are combined into a single *freespace segment*, where all voxels exhibit freespace propagation loss.

3.2 Attenuation Image Estimation

We now solve a smaller version of the RTI problem, where we estimate $\mathbf{X} \in \mathbb{R}^K$ (as opposed to $\mathbb{R}^N$ with $K \ll N$). Here K denotes the number of segments or the reduced dimension of the problem and $\mathbf{x_s} \in \mathbf{X}$ denotes the attenuation coefficient of all voxels within the s^{th} segment. We denote the K segments as $S_1, S_2, \cdots S_K$.

RTI approach for SkyScale. From §2.2, recall that the UAV traverses through the path $[L^1, L^2, L^3 \cdots L^n]$ collecting propagation loss measurements, where L^n denotes the n^{th} voxel along the trajectory. Let m_n^k denote the UAV's measurement at location L^n for the k^{th} UE located at L_{ue}^k. For the straight line connecting L^n and L_{ue}^k,

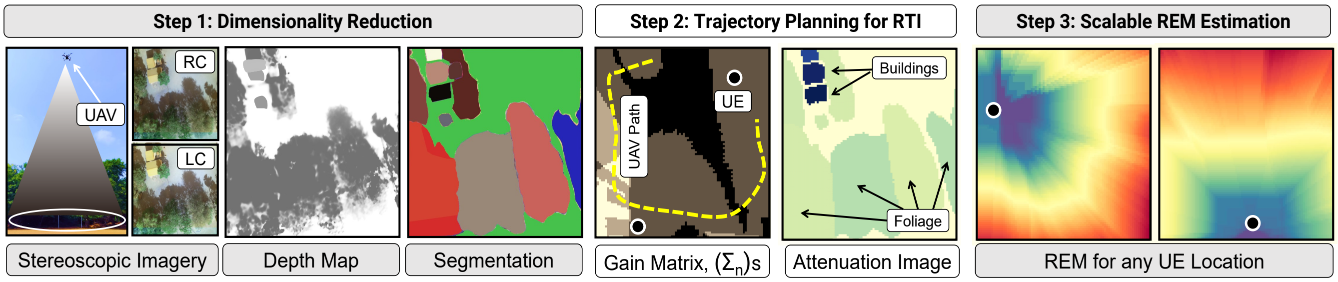

Figure 4: The different stages of SkyScale. LC and RC indicate the left and right camera perspectives from the stereo cam. First the segments are identified on which to perform the RTI. Second, measurements are taken covering the maximal number of segments and the attenuation image is compute. Next, for any given UE location the corresponding REM can be estimated.

let $d_i^n, \cdots d_j^n$ be the non-zero distances intercepted by segments $S_i, \cdots, S_j$ in terms of voxel units. We denote such collection of segment identifiers pertaining to UE$_k$ by the set $\sigma_{k,n} = \{i, \cdots, j\}$. Eqn. (1) can now be re-written as,

$$m_n^k = \sum_{s \in \sigma_{k,n}} d_s^n \mathbf{x_s} \tag{6}$$

$\mathbf{x_s}$ denotes the attenuation coefficient of each of the voxels within S_s and d_s is the length of the signal path (in voxel units) through S_s. For SkyScale, the RTI equations are,

$$\mathbf{A} = \begin{bmatrix} d_1^1 & d_2^1 & d_3^1 & \cdots & d_K^1 \\ d_1^2 & d_2^2 & d_3^2 & \cdots & d_K^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_1^R & d_2^R & d_3^R & \cdots & d_K^R \end{bmatrix}_{R\times K} \quad \mathbf{X} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_K \end{bmatrix}_{K\times 1} \quad \mathbf{M} = \begin{bmatrix} m_1^* \\ m_2^* \\ \vdots \\ m_R^* \end{bmatrix}_{R\times 1}$$

$$\mathbf{AX} = \mathbf{M} \tag{7}$$

In the above equation, each row in the projection matrix $\mathbf{A}$ denotes a measurement and each column corresponds to one of the K segments. $d_j^i \in \mathbf{A}$ denotes the length of the signal path intercepted by the j^{th} segment S_j corresponding to the i^{th} measurement m_i^* ('$*$' is generically used to indicate any relevant UE location). d_i^j is zero if the signal is not intercepted by the j^{th} segment. We use the regularized least-square formulation (eqn. 5) to solve for $\mathbf{X}$, with Identity matrix as the regularization operator, $\mathbf{L}$.

Trajectory planning. Given the limited flight time endurance of the UAV, it is critical to optimize the length of the measurement trajectory while improving the accuracy of the attenuation image. Let $\mathbf{U}$ denote the set of navigable voxel identifiers where the UAV can fly and take measurements. Considering k UEs in our system, we initialize the segment sets, Σ_n for each voxel $n \in \mathbf{U}$ as,

$$\Sigma_n = \bigcup_{\forall u \in \{1\cdots k\}} \sigma_{u,\,n} \tag{8}$$

Effectively, $|\Sigma_n|$ indicates the number of segments that can be 'seen' by the UAV from the n^{th} voxel, i.e., this ensures that such segments has non-zero column entries in the projection matrix $\mathbf{A}$ ($|\cdot|$ is the set cardinality operator). We refer to the collection of all $|\Sigma_n|$ superimposed on the aerial as the `gain` matrix. SkyScale's trajectory planning algorithm chooses a path, $\mathbf{T}$ (defined by a specific voxel order), that maximizes the total number of intercepted segments while keeping the length minimal or upper bounded by a budget, C. Specifically, SkyScale has the following goal.

$$\mathbf{T}_{opt} = \underset{\mathbf{T}}{\arg\max} \Big| \bigcup_{\forall n \in \mathbf{T}} \Sigma_n \Big|, \quad s.t., \quad \texttt{path\_length}(\mathbf{T}) \le C \tag{9}$$

$\mathbf{T}_{opt}$ denotes the optimal trajectory with the total path length given by the function `path_length()` in eqn. 9. Considering $\Sigma = \{1, 2, \cdots, K\}$ as the set of all segment identifiers, and $\Sigma_n \subset \Sigma$, $\forall n \in U$, eqn. 9 reduces to a variation of the Set Cover or the Max-Coverage problem which is well-known to be NP-Hard. In the following, we propose a greedy heuristic, specifically keeping the distance constraint in mind. The intuition is simple – We intend to minimize the cost per 'unseen' segment, where the cost depends on the distance between the current (L^{cur}) and the next voxel (see `line 5` of Algorithm 1).

Algorithm 1: SkyScale RTI Trajectory Planner

1 **Input:** L^0, Dmin, C, Sets: $\mathbf{U}$, Σ and $\Sigma_n \forall n \in U$
2 **Output:** $\mathbf{T}$
3 $\mathbf{T} \leftarrow [L^0]$, $L^{cur} \leftarrow L^0$
4 **while** $\Sigma \neq \emptyset$ and $C \geq 0$ **do**
5 $\quad i \leftarrow \underset{n \in \mathbf{U}}{\arg\min} \frac{\texttt{dist}(L^{cur},\ L^n)}{\Sigma \bigcap \Sigma_n}$, s.t., $\texttt{dist}(L^{cur},\ L^n) \geq$ Dmin
6 $\quad \Sigma \leftarrow \Sigma \setminus \Sigma_i$
7 $\quad C \leftarrow C - \texttt{dist}(L^{cur},\ L^n)$
8 $\quad$ `uav_path` $\leftarrow$ `Bresenham-3D` (L^{cur}, L^i)
9 $\quad$ **for** *voxel* $n \in$ `uav_path` **do**
10 $\quad\quad \Sigma \leftarrow \Sigma \setminus \Sigma_n$
11 $\quad\quad$ $\mathbf{T}$`.add_to_list(`L^n`)`
12 $\quad$ **end**
13 $\quad L^{cur} \leftarrow L^i$
14 **end**
15 **return** $\mathbf{T}$

The UAV's starting voxel in the aerial space, once it reaches the desired height, is denoted by L^0 and $\mathbf{U}$ denotes the set of navigable voxels. The total allocated distance budget is C. `dist`$(L^i,\ L^j)$ computes the Euclidean distance between the voxels i and j, whereas `Bresenham-3D` (L^i, L^j) enumerates the voxel path approximating the straight line path connecting voxels i and j [7]. The distance threshold Dmin used in `line 5` trades off between the *exploration*

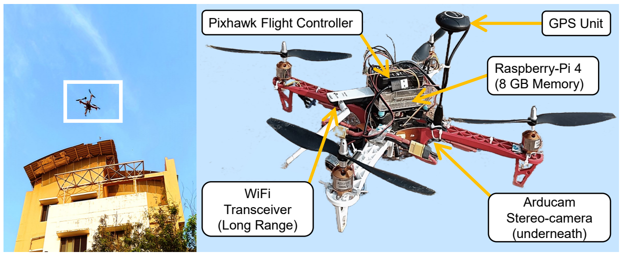

Figure 5: Our SkyScale UAV acts as a flying WiFi base station and can simultaneously provide network connectivity to 8+ UEs on ground.

of unseen segments away from L^{cur} versus *exploitation* of segments in the close proximity of L^{cur}, for solving the RTI equations. Algorithm 1 uses the weighted version of the Greedy Set Cover [16] where the weight is determined by the distance from L^{cur} to the next chosen voxel, L^i. Additionally, note that the journey from L^{cur} to L^i encounters new segments enriching the explored pool of segments further.

3.3 REM Estimation and Incremental Updates

The attenuation image can now be used to estimate the REM in aerial space for any given position of the UE (eqn. 7). For a real deployment, neither the network operation lasts for a single flight nor do the UEs remain at fixed locations. As we demonstrate in §5.1 and §5.4, the UEs are subject to continuous churn and the RTI equations should continuously be updated with incremental measurements.

4 UAV TESTBED AND EVALUATION DATASETS

We evaluate SkyScale in an arena of size 75 m×75 m using our custom-built UAV platform. Additionally, we also use a few more traces that add diversity to our dataset in terms of terrain complexity and wireless frequencies.

4.1 UAV Implementation and Testbed Dataset

The UAV, shown in fig. 5, is a lightweight custom-built quadcopter with a flight endurance of ≈ 20 minutes. It uses a `pixhawk 2.4.8` flight controller, programmable for controlled flights along custom trajectories. The UAV is equipped with a Raspberry Pi 4 (`RPI`) with 8 GB RAM, which gathers sensory and telemetry data from the flight controller and executes all trajectory planning algorithms. We use the Robot Operating System (ROS) on the UAV's `RPI` to handle communication and control. Additionally, the majority of in-flight power consumption is dedicated to the motors that keep the UAV airborne, with less than 5% of the power allocated to computational tasks such as depth imagery processing, segmentation, and attenuation image estimation.

Depth maps. The UAV is also equipped with an `Arducam` stereo camera [6] that provides synchronized frames for depth map estimation. We utilize OpenCV libraries, specifically the `StereoBM` block matching algorithm, to compute the disparity map of the scene. `StereoBM` allows us to specify the number of disparity levels, representing different gradations (or relative distance buckets) in the depth map. We have calibrated our system to convert the disparity map into a true depth map. Additionally, as discussed in §3.1 we perform image segmentation using the Watershed algorithm on the depth map to initialize the RTI problem. By adjusting the disparity settings or inputs for the segmentation algorithm (e.g., the number of markers for Watershed), we can control the number of segments generated.

WiFi Network and Coverage Arena. The `RPI` is configured to host a WiFi base station, using an external WiFi dongle paired with a high-gain (4 dBi, omnidirectional) antenna, which achieves a line-of-sight (LoS) range of approximately 150 meters. We operate on WiFi channel 6 in the 2.4 GHz band, free from interference within our test arena. The network is deployed across a 75 m×75 m area, with the UAV maintaining an altitude of 40–45 meters. The arena features a mix of open spaces, two medium-sized buildings, and dense foliage (see fig. 1) providing a diverse set of LoS and non-line-of-sight (NLoS) scenarios.

SkyScale WiFi dataset (Terr_1). We deploy seven `RPI`s as UEs at different ground locations, all connected to SkyScale's WiFi network. To collect ground truth measurements, the UAV flies a crisscrossed path across the entire 75 m×75 m area at an altitude of ≈40 meters (fig. 6). Using the `iperf3` tool, we continuously measure TCP throughput between the UAV and each UE. For each UE, we log GPS coordinates, WiFi RSS, and average TCP throughput. In SkyScale, we assume the locations of UEs are known. In total, we collect over 10K measurements for seven UEs across a ≈20 minute UAV flight.

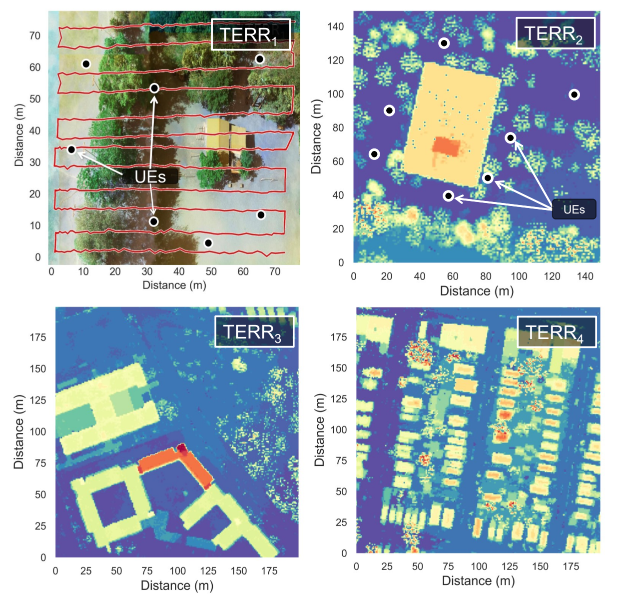

Figure 6: (Top-Left) Location of the seven `RPI` UEs are marked on the map where SkyScale testbed is deployed. A sample UAV trajectory is also shown while collecting groundtruth measurements. (Top-Right and Bottom) The three remaining figures represent the depth maps for Terr_2 (seven UEs marked), Terr_3 and Terr_4. We obtained these maps from the LiDAR traces publicly available at SRTM archives.

4.2 Other Datasets

Towards making our empirical analysis more comprehensive, we use a few datasets as mentioned below.

SkyRan LTE dataset (Terr$_2$). SkyRan [9] addresses a similar placement problem in an LTE network, where the UAV hosts an onboard LTE eNodeB operating over an undisclosed frequency in the sub-1 GHz band. Seven LTE enabled smartphones are deployed in a 200 m×200 m arena (fig. 6 *top-right*). Due to the absence of UEs in one section of the area, we use a slightly smaller region for our deployment. The arena features a large office building and sparsely distributed foliage. Highly granular REM data is available for all seven UE locations, enabling trace-driven simulations on the collected data.

Nvidia Sionna 5G dataset[1] (Terr$_3$ and Terr$_4$). In the SkyScale WiFi and the SkyRan LTE datsets we have access to the groundtruth signal map which can be used to estimate the REM prediction errors, however, the true attenuation coefficients are unknown. We use Nvidia Sionna [2], that provides state-of-the-art GPU based ray tracing framework to simulate wireless coverage incorporating configurable terrains maps and material properties of the obstacles. We choose two types of areas, Terr$_3$: a 200 m×200 m relatively sparser area with tall buildings (fig. 6 *bottom-left*), and, Terr$_4$: a 200 m×200 m residential area with a complex topography (fig. 6 *bottom-right*). Terr$_3$ and Terr$_4$ are based on true locations – the structural information of such locations are available as 3D *shape files* from OpenStreetMap [3]. Sionna's simulation framework can directly integrate structural information to create 5G coverage maps (frequency used 3.59 GHz). It also provides an option to attach material properties to the structures (as recommended by ITU P.2040 [1]). For our simulation, we assign different combinations of properties (*brick*, *concrete*, *wood* and *foliage*).

Depth Maps. For datasets Terr$_2$, Terr$_3$ and Terr$_4$, we do not have access to stereoscopic images, hence we directly used a LiDAR-based depth map from the SRTM database [19]. The same depth maps are presented in fig. 6.

5 PERFORMANCE EVALUATION

We evaluate the overall performance of SkyScale using the datasets described above. In the following analysis, we focus on the key performance metrics that are central to the design and operation of our system. Throughout this section, we assume a UAV speed of ≈2 m/sec.

5.1 Scalability: RTI versus REM Interpolation

Through figs. 7 – 11, we demonstrate how SkyScale outperforms interpolation based methods by an appreciable margin making such deployments sustainable for longer term operations. We design experiments where the UAV takes a series of flights and in each successive flight a fraction of the UEs relocate. We study the total flight time spent in collecting measurements to keep the average REM estimation error below 3 dB.

Experiment 1. Specifically, for Terr$_1$ and Terr$_2$, we choose a random subset of four UEs out of the seven deployed UEs. We perform REM estimation for six consecutive flights under various degrees of UE churn. UE churn here refers to the fraction of UEs that *relocate at the start of each flight*. We vary the UE churn from 0 to 100% in steps of 25%. 25%, 50%, 75% and 100% UE churn indicates that one, two, three and all four out of the set of four UEs change their position.

[1]Our datasets are available at https://cse.iitm.ac.in/~sense/skyscale/

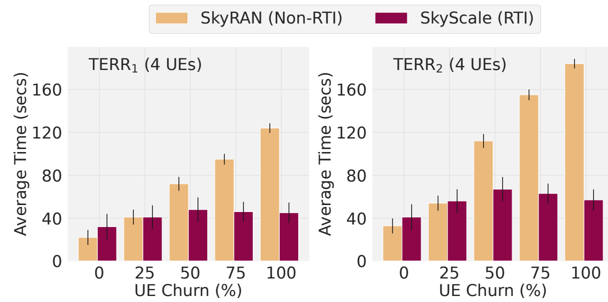

Figure 7: Even under high UE churns, SkyScale offers a 3–4× reduction in average flight budget compared to interpolation based techniques like SkyRan

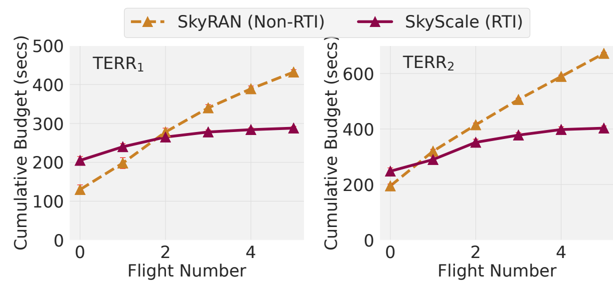

Figure 8: For an UE churn of 50%, SkyScale can maintain an average REM accuracy of 3 dB with ≈300 s worth of measurements.

Fig. 7 shows the average flight time of six consecutive flights while collecting measurements in order to contain the average REM estimation error within 3 dB. We use SkyRan as a candidate for our baseline interpolation scheme which also incorporates its own intelligent trajectory design. Although in absence of UE churn, SkyRan's average flight time is ≈8–10% lesser compared to SkyScale, it soon worsens off as UEs relocate. For an extreme case, where the network is highly dynamic with 100% UE churn, SkyScale cuts down the average flight time by a factor of 3–4×. Additionally, note that for a static setup (0% UE churn), interpolation based techniques work well as the REMs do not change and one-time measurement is sufficient. RTI incurs some additional overhead in such cases to estimate the attenuation image. In fig. 8, we drill down further onto the specific scenario with 50% UE churn and present the cumulative flight time used over the six successive flights. For an RTI based technique like SkyScale, requirement of additional measurements diminishes to almost nil beyond a certain point. However, interpolation based methods fundamentally require fresh measurements in order to re-estimate the REM, irrespective of how intelligent and optimized the trajectory planning is.

Experiment 2. For Terr$_3$ and Terr$_4$, we present a similar set of results in figs. 10 and 11, but with a relatively scaled up setup involving 16 UEs. Here we show an average of ten successive flights to reinforce the point how RTI based REM estimation gets massively scalable and sustainable for longer term operations. Interestingly, fig. 10 demonstrates a reduction of average flight time upto 8–10× for high UE churns. Fig. 11 shows a micro-benchmark for the case with 50% UE churn, i.e., eight out of the sixteen UEs relocate before each flight. As expected, SkyScale learns the attenuation

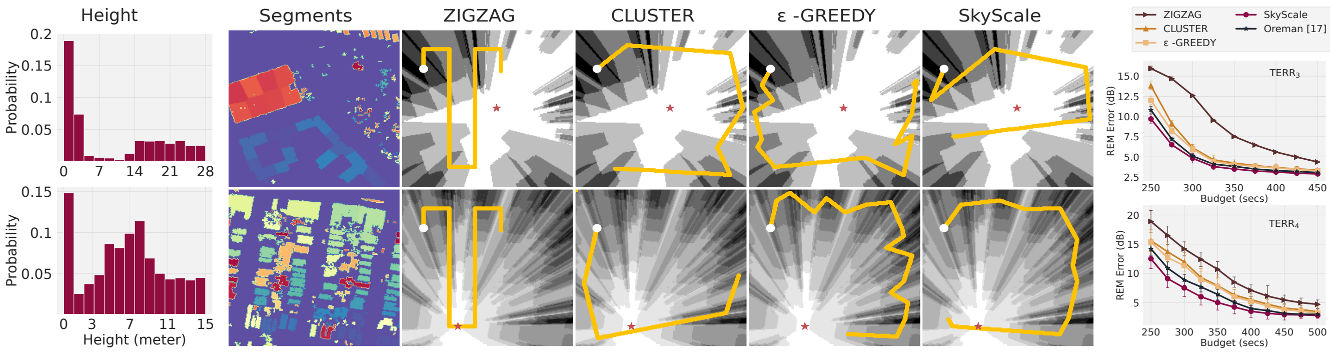

Figure 9: Height histograms for terrains Terr$_3$ and Terr$_4$, as obtained from the SRTM provided depth maps. Segmentation of the depth map is shown. The four rightmost plots on both rows present computed trajectories based on the `gain` matrix (Σ). Total trajectory length for both case is 500 meters. The initial `gain` matrix is also shown in the background. On the extreme right, we show an overall comparison of the REM error obtained for the different SOTA methods.

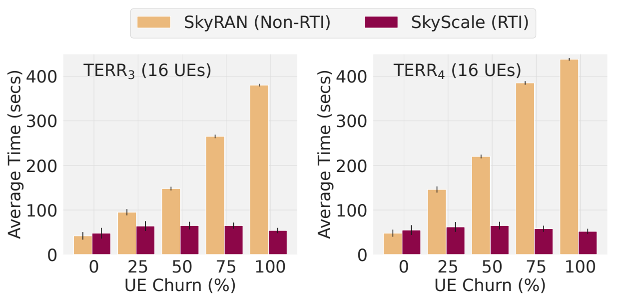

Figure 10: As the number of flights increase (10 flights), the relative benefit of SkyScale over SkyRan is highlighted.

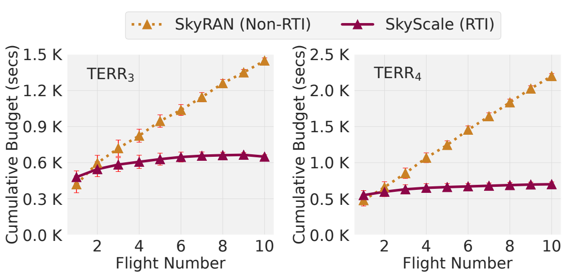

Figure 11: Cumulative budget for an UE churn of 50%. The requirement of fresh measurements beyond a point is nil for SkyScale which bolsters its sustainability for relatively longer term deployments.

image within a cumulative flight time of ≈600 seconds and further measurements are unnecessary to reliably predict the REM, unlike SkyRan.

5.2 Trajectory Planning and Segment Discovery

SOTA baseline algorithms. We evaluate SkyScale's trajectory planning scheme (Algorithm 1) and compare its performance with some state-of-the-art (SOTA) baseline approaches. Note that, such baselines are originally reported in literature in the context of REM interpolation. We demonstrate their performance limitation in an RTI setting. The three baseline approaches are as follows.

(a) **`ZIGZAG`**: The default waypoints path that many UAV applications use to systematically scan a region of interest in form of horizontal or vertical raster lines.
(b) **`CLUSTER`**: Such techniques cluster regions [9, 12] on the REM based on measurement values. Next, it assigns an uncertainty value to each cluster where a higher value necessitates more measurements. Definition of the uncertainty metric is subject to specific works in literature (SkyRAN uses the spatial gradient of RSS values [9], Oreman estimates it from aerial imagery [17] and so on). Further, a shortest path (*Dijkstra's algorithm* [17]) or route (*Traveling Salesman Problem* [9]) is constructed connecting clusters with higher uncertainty values. We use the SkyRan specific path planning algorithm.
(c) **ϵ-`GREEDY`**: Such techniques [26, 31] add a small amount of randomness in their exploration procedure. For instance, the UAV either visits the next location that maximizes some gain with a probability of $(1-\epsilon)$ (exploitation), or visits a random location with probability ϵ (exploration). We use an ϵ value of 0.1.

All the above baselines along with SkyScale take as input the `gain` matrix, Σ and computes a path. In fig. 9, we show sample paths constructed by the four algorithms for terrains Terr$_3$ and Terr$_4$. In the rightmost figure (top and bottom), we demonstrate how SkyScale outperforms state-of-the-art trajectory planning techniques, achieving a lower RSS estimation error for the same traversed path length.

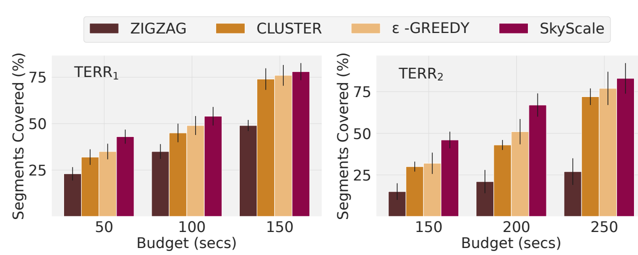

Figure 12: With increasing flight budget the UAV discovers new segments and incorporates them to the RTI equations.

Segment discovery. We show (fig. 12) that the Greedy Set Cover based approach in SkyScale is able to discover more segments based on a fixed budget while compared to its counterparts. The commonly used ZIGZAG strategy shows a relatively poor performance. In this context, we also discuss the implication of the depth map segmentation algorithm, in particular the number of segments. Fig. 13 shows the REM estimation accuracy and the associated resource usage for RTI ($\propto |\mathbf{A}|$, projection matrix) as a joint dependency on the number of segments and the number of measurements taken. First, using larger number of segments without insufficient measurements leads to poor accuracy. Second, using lesser number of segments make the attenuation image coarse grained leading to poorer accuracy. Given the computational constraints (fig. 13 (*right*)) and desired REM accuracy, one can choose the right number of segments.

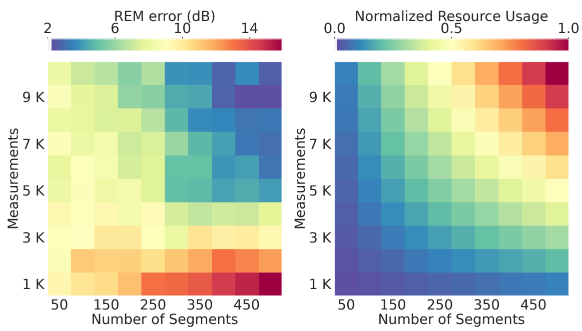

Figure 13: (*Left*) The joint dependency of the number of segments and the amount of measurements in determining an accurate attenuation image and hence, accurate REMs. (*Right*) Normalized resource usage for the same set of parameters. This analysis helps optimize accuracy based on available computational resources.

5.3 Overall REM Prediction Performance

We now discuss some results related to the REM prediction performance. Both for terrains Terr_1 and Terr_2, we see a quick improvement in the REM accuracy with a reasonable budget (fig. 14).

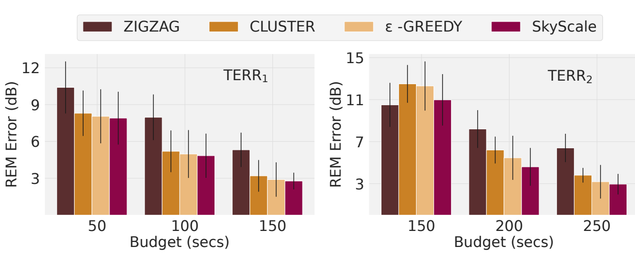

Figure 14: Overall REM errors with increasing measurements for terrains Terr_1 and Terr_2

Experiment 3, (Terr_1). We do another elaborate scalability study on end-to-end performance. For Terr_1, we split the seven UEs into two *disjoint* groups containing four and three UEs each in all possible combinations ($\binom{7}{3} = 35$). The attenuation map is always estimated with the first group of UEs and the UAV is positioned to serve the second group of UEs. In fig. 15, we present the CDF of the drop in RSS (median 3–4 dB) and TCP throughput (median 1–3 Mbps) as compared to that for the optimal position. The variance can be primarily attributed to the asymmetry in the UE positions within the arena. These results based on our real UAV network testbed demonstrates the effectiveness of SkyScale.

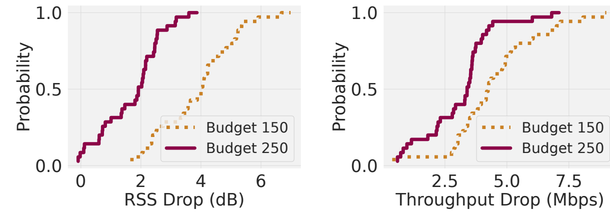

Figure 15: Results related to Experiment 3. TCP Throughput (iperf3) and RSS drop under UE churn.

Attenuation coefficient. In fig. 16, for the simulated datasets Terr_3 and Terr_4, we present the average prediction error of the attenuation coefficient of the segments. After a reasonable budget (as the REM estimation error $\leq$ 3 dB), the attenuation coefficient prediction error is less than 0.2 dB. The true coefficients were fixed by us while setting up the Sionna simulation.

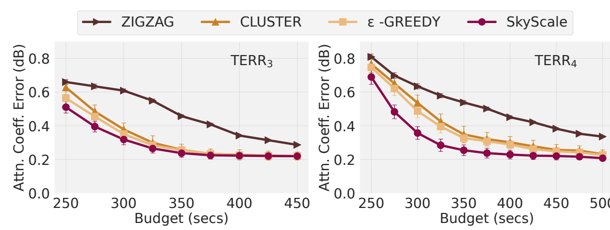

Figure 16: SkyScale is able to predict the attenuation coefficient within an accuracy of 0.2 dB.

5.4 Adapting to UE Mobility in Real-Time

Finally, we report one of the crucial contribution that SkyScale makes, i.e., adapting to UE mobility in real-time.

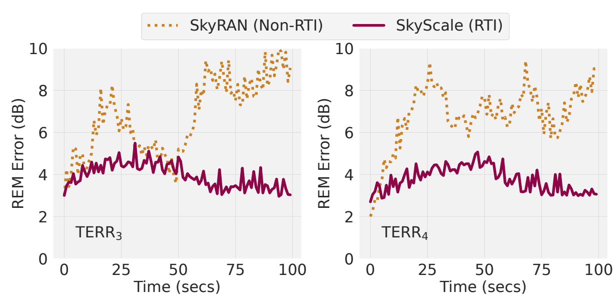

Figure 17: Results related to Experiment 4. SkyScale makes the best out UE mobility to keep a stable performance, unlike SkyRAN.

Experiment 4, (Terr_3 and Terr_4). The UAV initially flies for 500 seconds, keeping the REM error for each of the 16 UEs within 3 dB. Over the next 100 seconds, UEs move randomly within the arena. In SkyRan (non-RTI), no new data updates the REM after the initial flight. However, SkyScale, even without further UAV movement, leverages UE mobility to incorporate new measurements into the RTI framework, incrementally updating the attenuation image whenever new segment information is added to the system and improving accuracy. Consequently, the UAV then moves to the new optimal location. Fig. 17 shows SkyScale maintaining the REM error around 3–4 dB, with some fluctuations due to simulator-introduced AWGN noise.

5.5 Limitations of SkyScale

Although SkyScale demonstrates superior performance compared to its state-of-the-art counterparts, there are scenarios where this advantage diminishes. In relatively static environments with minimal UE churn, new information is not introduced into the RTI framework unless the UEs move. This limits the system's ability to update and improve the attenuation image over time. Observe in fig. 7 and fig. 10, for cases the network is static (zero UE churn), direct interpolation (e.g., SkyRan) outperforms SkyScale.

6 RELATED WORKS

Recent research in UAV-based wireless networks explores various aspects of communication, networking, and sensing. This discussion focuses on efficiently constructing REMs and optimizing UAV placement to enhance overall communication performance. Most existing literature relies on simulations for evaluation, while prototyping and testbed deployments are increasingly rare.

Deep learning (DL) based REM generation. While classical interpolation based methods (e.g., Kriging [10]) have been around, recent works capitalizes on the advances of DL techniques (`U-Net`,`cGAN` - [11], [23]) to efficiently reconstruct the REMs. DL techniques, e.g., CNNs or GANs, require access to huge volume of 'groundtruth' information for training which makes it impractical to adopt in such scenario. The trajectory optimization problem has been benefited by some recent progress in Reinforcement Learning([26, 31] and Multi-Arm Bandits [5]. Although these algorithms can operate in real-time, they require significant time to stabilize.

Structural reconstruction using UAV sensing. Some works in this area focus on UAV-based sensing for reconstructing terrain or structures in 3D. This includes techniques such as mmWave sensing [4], photogrammetry from aerial images [8], and recent advancements using Neural Radiance Fields (NeRFs) [22]. While these methods provide detailed 3D reconstructions, they do not directly create attenuation maps, making them unsuitable for direct REM generation.

REM and RTI related to UAV based deployments. A few works exist that directly compute the 2D or 3D REM [9, 17, 25, 30] based on intelligent spatial sampling and trajectory optimization [20, 24, 32]. [21] uses an existing tomographic map for UAV placement but does not discuss any RTI technique.

7 CONCLUSION

In this work, we introduced SkyScale, a scalable UAV positioning system tailored for dynamic networks. Our approach is among the first to experimentally leverage RTI in UAV-based wireless networks. SkyScale minimizes additional measurements during long-term operations, outperforming state-of-the-art interpolation methods by reducing measurement costs by 10× or more. While SkyScale may underperform in static scenarios with few UEs, we suggest intelligently combining RTI with interpolation techniques to optimize performance based on network dynamics.

REFERENCES

[1] ITU, Effects of building materials and structures on radiowave propagation above about 100 MHz. https://www.itu.int/rec/R-REC-P.2040/en.
[2] NVIDIA Sionna Wireless Simulation Framework. https://nvlabs.github.io/sionna/examples/Sionna_Ray_Tracing_Introduction.html.
[3] OpenStreetMap. https://www.openstreetmap.org/.
[4] Ahmad, F., , et al. AeroTraj: Trajectory Planning for Fast, and Accurate 3D Reconstruction Using a Drone-based LiDAR. *ACM IMWUT 2023 7*, 3.
[5] Arani, A. H., et al. Reinforcement learning for energy-efficient trajectory design of UAVs. *IEEE Internet of Things Journal 9*, 11 (2021).
[6] Arducam. Stereo-camera for Raspberry Pi. https://www.arducam.com/product/b0195-synchronized-stereo-camera-hat-raspberry-pi/.
[7] Bresenham, J. E. Algorithm for computer control of a digital plotter. In *Seminal graphics: pioneering efforts that shaped the field*. 1998, pp. 1–6.
[8] Bylow, E., Maier, R., Kahl, F., and Olsson, C. Combining depth fusion and photometric stereo for fine-detailed 3d models. In *Scandinavian Conference on Image Analysis* (2019), Springer, pp. 261–274.
[9] Chakraborty, A., Chai, E., Sundaresan, K., and Rangarajan, S. SkyRAN: a self-organizing LTE RAN in the sky. In *Proc. ACM CoNEXT 2018*.
[10] Chakraborty, A., Rahman, M. S., Gupta, H., and Das, S. R. Specsense: Crowdsensing for efficient querying of spectrum occupancy. In *Proc. IEEE INFOCOM'17*.
[11] Chaves-Villota, A., and Viteri, C. A. DeepREM: Deep-Learning-Based Radio Environment Map Estimation from Sparse Measurements. *IEEE Access* (2023).
[12] Chen, J., Du, C., Zhang, Y., Han, P., and Wei, W. A clustering-based coverage path planning method for autonomous heterogeneous UAVs. *IEEE Transactions on Intelligent Transportation Systems 23*, 12 (2021), 25546–25556.
[13] Dhekne, A., Chakraborty, A., Sundaresan, K., and Rangarajan, S. TrackIO: Tracking First Responders Inside-Out. In *Proc. USENIX NSDI 2019*.
[14] Hampton, J. R., Yao, S., Kroeger, T., Carducci, C., and Murphy, C. Drone-based forest propagation measurements for ground-to-air EMI applications. *IEEE Antennas and Wireless Propagation Letters 18*, 12 (2019), 2627–2631.
[15] Ibrahim, M., Liu, H., Jawahar, M., Nguyen, V., Gruteser, M., Howard, R., Yu, B., and Bai, F. Verification: Accuracy evaluation of wifi fine time measurements on an open platform. In *Proc. ACM Mobicom 2018*.
[16] Ivanova-Rohling, V. N. Neighborhood-based strategies for widening of the greedy algorithm of the set cover problem. In *Proceedings of the 19th International Conference on Computer Systems and Technologies* (2018), pp. 27–32.
[17] Matson, N., and Karthikeyan, S. Online Radio Environment Map Creation via UAV Vision for Aerial Networks.
[18] Meyer, F. Color image segmentation. In *International conference on image processing and its applications* (1992), IET.
[19] NASA. SRTM Database. https://www.earthdata.nasa.gov/sensors/srtm.
[20] Romero, D., and Kim, S.-J. Radio map estimation: A data-driven approach to spectrum cartography. *IEEE Signal Processing Magazine 39*, 6 (2022), 53–72.
[21] Romero, D., Viet, P. Q., and Leus, G. Aerial base station placement leveraging radio tomographic maps. In *Proc. IEEE ICASSP 2022*.
[22] Rosas-Ordaz, et al. An Overview of NeRF Methods for Aerial Robotics. *Machine Learning for Complex and Unmanned Systems* (2024).
[23] Shawel, B. S., Woldegebreal, D. H., Pollin, S., et al. A Deep-Learning Approach to a Volumetric Radio Environment Map Construction for UAV-Assisted Networks. *International Journal of Antennas and Propagation 2024* (2024).
[24] Sheshadri, R. K., Chai, E., Sundaresan, K., and Rangarajan, S. SkyHAUL: A Self-Organizing Gigabit Network In The Sky. In *Proc. ACM MobiHoc 2021*.
[25] Shrestha, R., Romero, D., and Chepuri, S. P. Spectrum surveying: Active radio map estimation with autonomous UAVs. *IEEE Transactions on Wireless Communications 22*, 1 (2022), 627–641.
[26] Steiger, J., Lu, N., and Sorour, S. Learning for path planning and coverage mapping in UAV-assisted emergency communications. In *Proc. IEEE Globecom'20*.
[27] Sundaresan, K., Chai, E., Chakraborty, A., and Rangarajan, S. SkyLiTE: End-to-end design of low-altitude UAV networks for providing LTE connectivity. *arXiv preprint arXiv:1802.06042* (2018).
[28] Viet, P. Q., and Romero, D. Aerial base station placement: A tutorial introduction. *IEEE Communications Magazine 60*, 5 (2022), 44–49.
[29] Wilson, J., and Patwari, N. Radio tomographic imaging with wireless networks. *IEEE Transactions on Mobile Computing 9*, 5 (2010), 621–632.
[30] Wu, Q., Shen, F., Wang, Z., and Ding, G. 3D spectrum mapping based on ROI-driven UAV deployment. *IEEE Network 34*, 5 (2020).
[31] Zeng, Y., Xu, X., Jin, S., and Zhang, R. Simultaneous navigation and radio mapping for cellular-connected UAV with deep reinforcement learning. *IEEE Transactions on Wireless Communications 20*, 7 (2021), 4205–4220.
[32] Zhang, S., and Zhang, R. Radio map-based 3D path planning for cellular-connected UAV. *IEEE Transactions on Wireless Communications 20*, 3 (2020).