SkyScale: A Radio Tomographic Approach Towards Scaling UAV Network Deployments

Subrata Das and Ayon Chakraborty
IIT Madras

ABSTRACT

A critical requirement for deploying Unmanned Aerial Vehicle (UAV) based wireless networks is to position the UAV optimally in the aerial space, ensuring robust connectivity. Such optimal positioning requires the UAV to estimate the propagation loss for each ground terminal or user equipment (UE) across the aerial space – a mapping typically referred to as the Radio Environment Map (REM). Existing literature in this area primarily focuses on improving the accuracy of REM estimation from sparse and noisy signal measurements at the UAV (e.g., by training deep learning models). However, the estimated REMs lose relevance as the UEs relocate, necessitating the UAV to collect fresh measurements to re-estimate the REMs. Such repetitive measurements make the deployment challenging to scale, particularly with a higher number of UEs or with UE mobility. In this paper, we propose SkyScale 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 saves measurement costs by 10× or more compared to state-of-the-art interpolation based schemes, making it highly scalable and seamlessly adaptive to dynamic network conditions.

1 INTRODUCTION

Recently Unammed Aerial Vehicle (UAV) based platforms provide a low-cost and flexible alternative to deploy wireless network infrastructure, particularly in adhoc or resource constrained settings [9, 17, 28]. Such UAVs serve as *flying* base stations providing wireless connectivity to clients (e.g., user equipment or UEs[1]) located on the ground. A critical requirement for such deployments is to position the UAV in 3D airspace that optimizes the overall channel quality of the radio access network (RAN) across different UEs [25, 29]. Second, the UAV must constantly adjust its location in response to the UEs' mobility [9]. This requires mapping the channel quality, e.g., signal to noise ratio (SNR) or received signal strength (RSS), over all points in the aerial space, corresponding to each UE on the ground. Such maps are typically referred as Radio Environment Maps or REMs. The problem of estimating REMs is well studied in the wireless literature including the empirical works specifically geared towards optimizing UAV network deployments [9, 17, 28].

In a nutshell, the optimal operating position of the UAV is determined as follows. The UAV moves along a specified trajectory gathering signal level measurements corresponding to each UE and estimates the REMs, for instance, by interpolating such measurements spatially. Note that only the uplink transmissions from the UEs to the UAV are relevant for the REM estimation, as all the wireless channel related information needs to be available on the UAV side. At each point in the discretized aerial space, a *utility metric* is calculated by statistically aggregating the estimated REMs per UE – e.g., 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, much of the work done on such aerial network deployments assume similar strategies [21, 33]. It is evident that accurate estimation of the REMs is critical for optimal communication performance [9, 22]. For instance, a body of work solely focuses on efficient measurement trajectories for the UAV to improve the accuracy of the REMs [9, 17], while another body of work aims at enhancing the interpolation techniques used in predicting parts of the REM that lacks measurements. A common forcing function for all such optimization efforts is the limited flight endurance of the UAVs; the UAVs would be better off spending more time providing communication services than gathering measurements to estimate REMs.

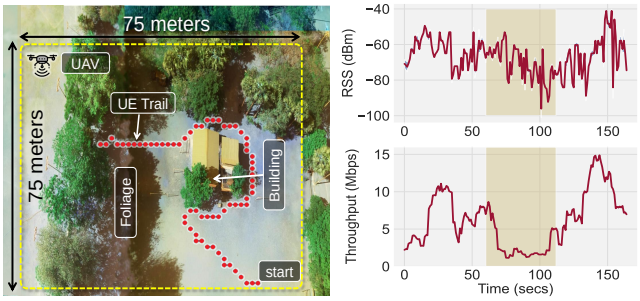

Figure 1: A bird's eye view of our deployment arena with the UAV, acting as a WiFi base station, operating at a height of 40m (top left). We present the variation in the Received Signal Strength (RSS) and TCP throughput between the stationary UAV and a mobile UE (using `iperf3`) as the latter follows the trail marked on the figure. Note the 6–7× drop in throughput and ≈40 dB (shaded box) as the UE moves behind the building. Optimizing the wireless performance for such mobile UE(s) requires the UAV to adaptively reposition itself based on the real-time REM estimation.

Lack of scalability. Note that such UAV positioning works well assuming the set of UEs to be reasonably static. Nevertheless, the REMs vary when the UEs move necessitating the system to re-estimate the REMs and the utility metric across the aerial space. In

[1] Although the term UE is often specified in the context of 4G/LTE technology, we use it to indicate any generic wireless client that connects to the UAV network.

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 an approach clearly does not scale well. As the demand for UAV-based network deployments continues to grow, addressing these challenges will be paramount in realizing the full potential of this technology.

Our contribution in this paper is motivated by a key observation that directly addresses such scalability issue. Even if the UE mobility causes the REMs to vary significantly, what remains invariant is the propagation environment or the underlying terrain. Estimating wireless attenuation characteristics of the terrain provides a direct way to estimate REMs for *any arbitrary* UE position. Unlike existing approaches, where every new REM estimation requires collecting independent measurements from scratch, such terrain estimation benefits cumulatively from historical measurements (§5.1, see figs. 11 and 8). Although, initially, it is still required to collect measurements, however, with subsequent UE mobility the need for further measurements diminishes drastically to almost nil (§5.4, see figs. 7 and 10).

In this paper, we build upon the above observation towards designing SkyScale, a robust and scalable REM estimation strategy that adapts to arbitrary UE mobility. SkyScale uses Radio Tomographic Imaging (RTI) to estimate the wireless attenuation properties of the surrounding environment. RTI based techniques discretize the 3D space into voxels (at some resolution) 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*. The total wireless propagation loss that occurs while the signal traverses from the UE to a UAV location can be easily estimated from the attenuation map.

RTI techniques make use of 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 stated earlier, given the attenuation map it is relatively straightforward to estimate the REMs (aka *forward problem*). However, RTI attempts to solve the *inverse problem*, i.e., given partial measurements pertaining to a few REMs it reconstructs the attenuation map or the attenuation image (more on this later, see §2.2). Although RTI seems to be reasonable approach, the sheer number of voxels in the 3D space makes such solution challenging and 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 yields 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] 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 contributions in this paper.

- We explore RTI as a key approach that can significantly reduce the UAV's time probing for new measurements. SkyScale can adapt to dynamic network settings faster and support mobile UEs with *zero* additional overhead.
- We address the computational bottlenecks to run RTI based algorithms onboard the UAV. Our lightweight processing of the imagery data (§3.1) helps reduce the computational load by 50–100×.
- We demonstrate our idea by designing and building SkyScale, an end-to-end system to deploy and operate a WiFi based UAV network. Additionally, we test our algorithm on wireless traces related to LTE and 5G networks. Overall, SkyScale is not only scalable but also sustainable for longer term network operations.

2 MOTIVATION AND BACKGROUND

2.1 Adapting REM Estimation to UE Mobility

As mentioned earlier, much of the literature in this research area attempts to predict the REMs from scratch every time the UEs relocate. This prediction can either effectively be an intelligent interpolation from freshly obtained measurements (e.g., Gaussian Process Regression/Kriging [21]) or involve machine learning (ML)/deep learning (DL) tools that work with sparser or noisy measurements. Often additional cues like satellite or UAV imagery data are used. Although such approaches perform well and produce nearly accurate REMs, a key question that remains unclear is whether such historical measurements or trained models continue to be relevant as the UEs relocate. A straightforward answer is *no*, as for most real-life terrains there is a non-trivial change in the REM caused by non-line-of-sight (NLoS) blockages as a UE moves(see, fig. 1). Hence, 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). With limited endurance of UAVs – e.g., 20 minutes in our case on a single battery cycle – spending a substantial fraction of the flight time in gathering measurements seems prohibitive.

Second, we also experiment with some of the popular deep generative models (e.g., U-Net) to understand their viability towards REM estimation. We train a U-Net model similar to [17], that takes the UAV imagery of the terrain along with the UE location as inputs, and predicts the resulting REM. At the training phase, we provide complete instances of REM for seven different UE locations spanning ≈10K measurements from 20+ minutes of flight. We observe that the model was unable to generalize the REM generation for an arbitrary UE location, the median prediction error being 8–10 dB. At this point, it is critical to note that such training overhead and data requirement defeats the whole purpose of scaling UAV network deployments. Also, we find it unjustified and impractical to attempt improving the U-Net model with further measurements.

For instance, with a considerable UE churn or UEs being mobile such methods are not well suited to update the REMs in real time and adaptively reposition the UAV. Also training such models is beyond the capacity of on-board compute available on a relatively modest UAV.

The above attempts motivate us to look for a better representation primitive of the REM that can generalize across historical measurements, and hence scale. One such primitive is the spatial attenuation characteristics of the terrain. Techniques based on Radio Tomographic Imaging or RTI precisely attempts to estimate such characteristics.

2.2 Radio Tomographic Imaging (RTI) Primer

Wireless signals get attenuated as it traverses along a medium and significantly so, as it passes through physical objects – e.g, buildings, foliage etc. [14]. The extent of such attenuation depends on the length of path traversed by the signal and material properties (e.g., permittivity) of the traversing medium itself. Towards setting up an analytical framework for RTI, we first discretize the 3D space into N volume elements or voxels each of dimension δ^3 (resolution being δ). RTI estimates the volumetric image, $\mathbf{X} \in \mathbb{R}^N$, where $\mathbf{x_i} \in \mathbf{X}$ specifies the attenuation coefficient of the i^{th} voxel. We choose an arbitrary origin in the region of interest and specify our 3D coordinate system where a 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.

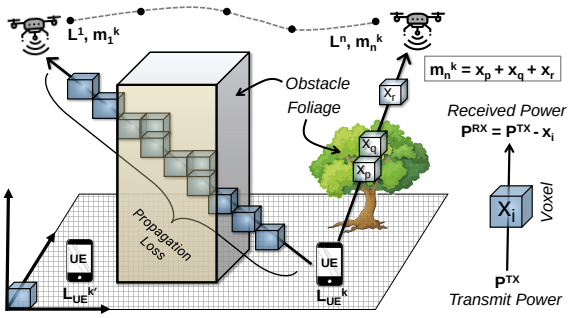

Figure 2: RTI schematic for SkyScale. The signal from each UE reaches the UAV traversing through a host of such voxels experiencing propagation loss. Each voxel contributes to the propagation loss determined by its attenuation coefficient - for instance, free space or foliage etc.

Let the k^{th} UE, denoted by UE_k, be located at L^k_{ue}. It is trivial to assume that the UAV is aware of the UE location [9, 13]. 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 matrix, 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 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,

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

The solution to eqn. 3 is equivalent to taking the *Moore-Penrose Pseudoinverse* on $\mathbf{M}$, i.e., $\hat{X} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T M$. 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 $\mathbf{M}$ 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,

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

Eqn. 4 is referred as the generalized Tikhonov regularization [30] and is a common tool for solving ill-posed inverse problems. $\beta > 0$ is a hyperparameter and L denotes the regularization operator whose choice, as we show later, affects the quality of the solution considerably. 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 calibrate the reader, we present some preliminary results on computational resource usage while running vanilla RTI algorithm for our testbed environment, 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 run when the voxel dimension is lesser than 2.5 m.

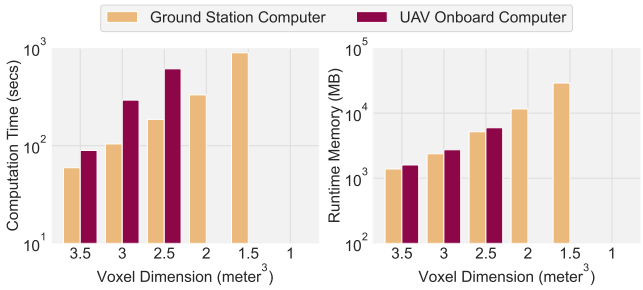

Figure 3: Benchmark results for computation time and runtime memory usage for running the vanilla RTI scheme (eqn.5) on the UAV's onboard Raspberry Pi computer and a well provisioned laptop. Due to memory constraints, the onboard computer is not able to run the solution if the voxel dimension is less 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 $\mathbf{x_i}$ of each voxel for each of the N voxels. Effectively, the vanilla RTI method solves this N dimensional problem. However, N can be very large depending upon the resolution δ and thus result in prohibitively heavy computational load (see fig. 3). We make use of the UAV's stereoscopic imagery data to reduce such bloated dimensionality. In particular, we compute a pixel based depth map for the underlying terrain and perform image segmentation on the same. Such segments roughly identify regions with similar obstacle characteristics - for instance, areas with dense foliage, building tops, open spaces and so on. We use the Watershed algorithm [19] 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 exhibit the same attenuation coefficient on an average. In this regard, there exists a tradeoff between the number of segments versus the computation resources required. In general, smaller segment size tends to create larger number of segments thereby increasing both the accuracy of the eventual REM as well as computational costs (see fig. 13). For instance, we observe that Watershed estimated segments provide the reasonable sweetspot between accuracy and compute. The segments pertaining to the maximum depth are treated as open or ground area and are merged together as the *freespace segment*. All voxels within such segment 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, 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,

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

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

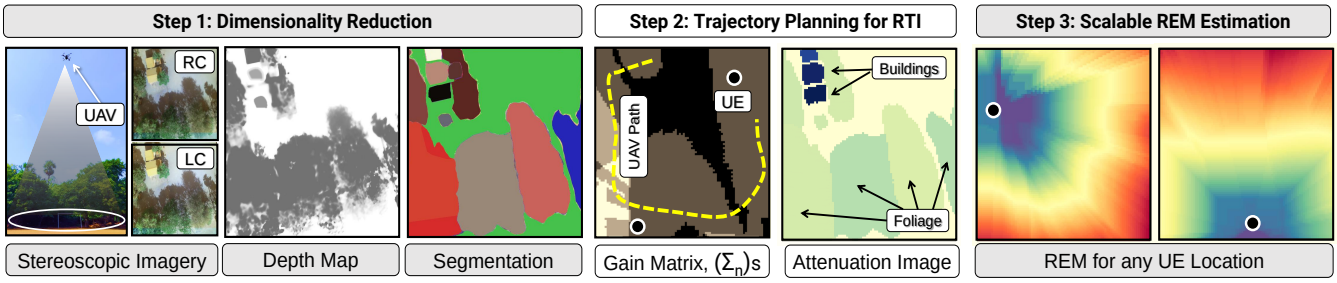

Figure 4: The different stages of SkyScale. LC and RC indicate the left and right camera perspectives from the stereo cam. First the segments are identified on which to perform the RTI. Second, measurements are taken covering the maximal number of segments and the attenuation image is compute. Next,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

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. $d^i_j \in \mathbf{A}$ 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^j_i 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.

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

$\mathbf{T}_{opt}$ 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).

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* 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 [15] 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. The weighted Greedy Set Cover is known to have an approximation ratio of the n^{th} Harmonic number, $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} \cdots + \frac{1}{n}$, i.e., $\mathbf{T} = \mathbf{T}_{opt}.H_n$, where $n = |\mathbf{U}|$.

Algorithm 1: SkyScale RTI Trajectory Planner

```
1  Input: L^0, Dmin, C, Sets: U, Σ and Σ_n ∀n ∈ U
2  Output: T
3  T ← [L^0], L^cur ← L^0
4  while Σ ≠ ∅ and C ≥ 0 do
5      i ← arg min_{n∈U} dist(L^cur, L^n) / (Σ ∩ Σ_n), s.t., dist(L^cur, L^n) ≥ Dmin
6      Σ ← Σ \ Σ_i
7      C ← C − dist(L^cur, L^n)
8      uav_path ← Bresenham-3D (L^cur, L^i)
9      for voxel n ∈ uav_path do
10         Σ ← Σ \ Σ_n
11         T.add_to_list(L^n)
12     end
13     L^cur ← L^i
14 end
15 return T
```

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.

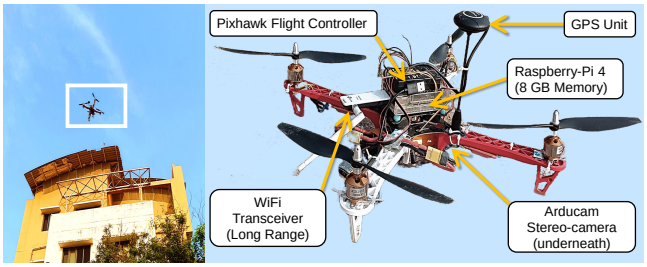

Figure 5: Our SkyScale UAV acts as a flying WiFi base station and can simultaneously provide network connectivity to 8+ UEs on ground.

Low-rank incremental updates. Note that fresh measurements, not considering noise filtering purposes, effectively contributes to the attenuation image reconstruction only if it improves the rank of the projection matrix, **A**. As **A** gets larger with time, the singular valued decomposition (SVD) for rank computation gets expensive. We resort to a low-rank incremental update, where we take a new batch of measurements ($\mathbf{A}_{new}$) and perform SVD($\mathbf{A}_{new}$) = $U_{new}S_{new}V_{new}^{\mathsf{T}}$. We retain the top r singular values of S_{new} to obtained the truncated S_{new_r}. The low-rank matrix $\mathbf{A}_{new\_lowrank}$ is computed as $U_{new}S_{new\_lowrank}V_{new}^{\mathsf{T}}$, and finally **A** is updated as $\mathbf{A} + \mathbf{A}_{new\_lowrank}$. This helps SkyScale to work in real-time and support mobile UEs without overwhelming the onboard compute.

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 quadcopter custom-built for SkyScale, with a flight time endurance of ≈20 minutes. It uses a pixhawk 2.4.8 flight controller which can be programmed for any controlled flight along a custom trajectory. The UAV houses a dedicated compute unit, a Raspberry Pi 4 (RPI) with 8 GB of RAM, for gathering sensory and telemetry related information from the flight controller and executing all trajectory planning related algorithms. We use the popular Robot Operating System or ROS framework hosted on the UAV's RPI to communicate with the flight controller to exchange all control information.

Depth maps. The UAV is also equipped with an Arducam stereo-camera [6] that provides highly synchronized frames for depth map estimation. We use OpenCV libraries, specifically the 'block matching algorithm' implementation (StereoBM [16]), for computing the disparity map of the scene. StereoBM takes the desired number of disparity levels as input, i.e., the number of gradations (relative distance buckets) in the depth map. Additionally, we have calibrated our setup to estimate the true depth map from the disparity map. Further, as discussed in §3.1 we perform image segmentation (using the Watershed algorithm) of the depth map to initialize the RTI problem. Depending on the disparity value or inputs passed to the segmentation algorithm (e.g., number of markers for Watershed), we can achieve a desired number of segments.

WiFi Network and Coverage Arena. The RPI is configured to host a WiFi base station. We use an external WiFi dongle along with a high gain antenna (4 dBi, omnidirectional) that achieves a LoS range of approximately 150 m. We use the WiFi channel 6 in the 2.4 GHz band which is devoid of any interference around our arena. We deploy the network in a 75 m×75 m and the height of the UAV is maintained between 40 –45 m. The arena has a good mix of open spaces, two medium sized buildings and also a distribution of dense foliage (see fig. 1) – all these contribute to an interesting combination of LoS and NLoS scenarios.

SkyScale WiFi dataset (Terr$_1$). We deploy seven RPIs as UEs at seven different locations on the ground that connect to our SkyScale's WiFi network. For collecting groundtruth measurements, the UAV traverses the entire arena through a crisscrossed path cruising at a height of ≈40 m (fig.6). We use the iperf3 tool to continuously measure TCP throughput between the UAV and all UEs. For each UE, we log the GPS coordinates, WiFi RSS and average TCP throughput. For SkyScale, we assume the location of the UEs to be known. Totally, we collected 10K measurements for seven UEs across a ≈20 min UAV flight.

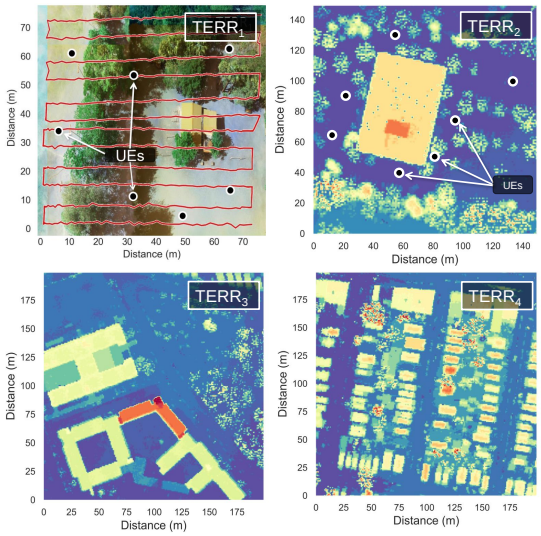

Figure 6: (Top-Left) Location of the seven RPI UEs are marked on the map where SkyScale testbed is deployed. A sample UAV trajectory is also shown while collecting groundtruth measurements. (Top-Right and Bottom) The three remaining figures represent the depth maps for Terr$_2$ (seven UEs marked), Terr$_3$ 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 demonstrates 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*). We use a slightly smaller

area compared to the one used in SkyRan due to complete unavailability of UEs on one portion of the arena. The area contains a large office building along with a sparse distribution of foliage. Highly granular REM information is available corresponding to all seven UE locations for performing any trace driven simulation on the data.

Nvidia Sionna 5G dataset (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 [20]. The same depth maps are presented in fig. 6.

5 PERFORMANCE EVALUATION

We evaluate the overall performance of SkyScale based on the above mentioned datasets. In the following evaluation, we highlight the key performance aspects that is central to the design of our system. Throughout this section, we assume the UAV speed to be ≈ 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.

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

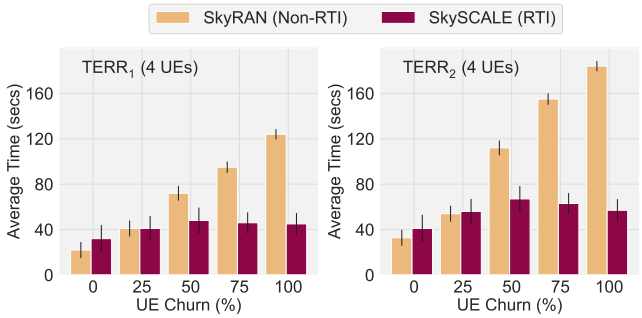

Figure 7: Even under high UE churns, SkyScale offers a 3–4× reduction in average flight budget compared to interpolation based techniques like SkyRan

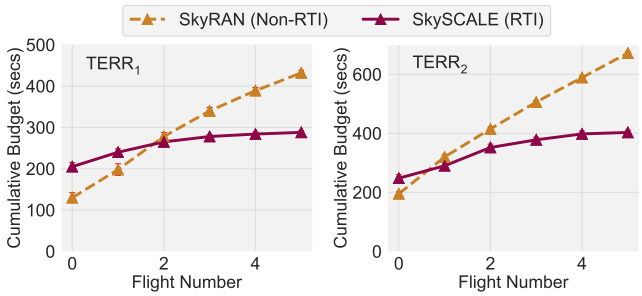

Figure 8: For an UE churn of 50%, SkyScale can maintain an average REM accuracy of 3 dB with ≈300 s worth of measurements.

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 image within a cumulative flight time of ≈600 seconds and further

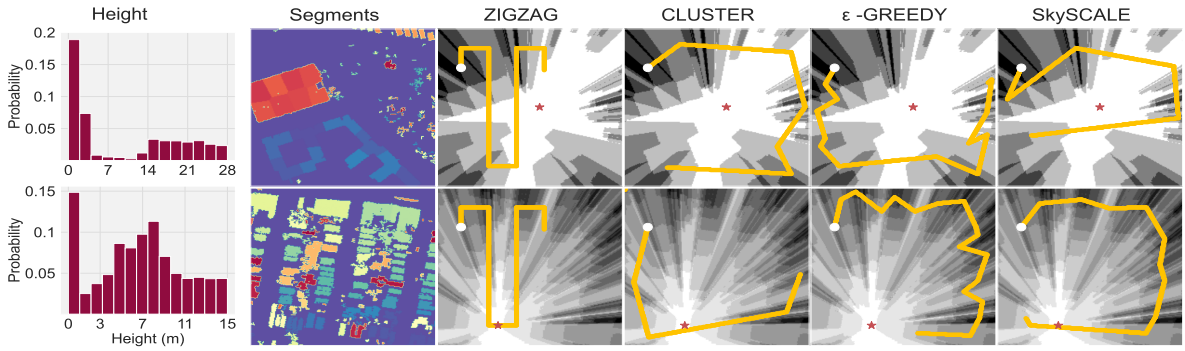

Figure 9: Height histograms for terrains TERR$_3$ and TERR$_4$, as obtained from the SRTM provided depth maps. Segmentation of the depth map is shown. The four rightmost plots on both rows present computed trajectories based on the `gain` matrix (Σ). Total trajectory length for both case is 500 meters. The initial `gain` matrix is also shown in the background.

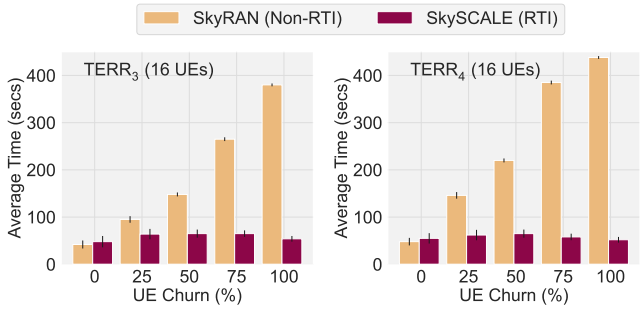

Figure 10: As the number of flights increase (10 flights), the relative benefit of SkyScale over SkyRan is highlighted.

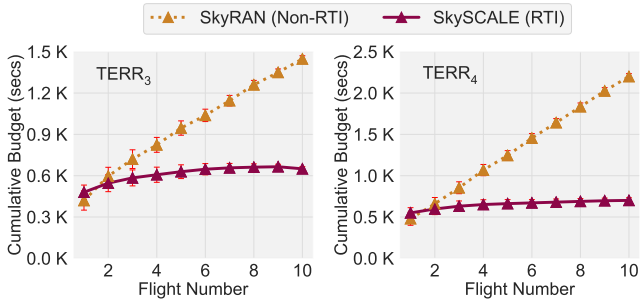

Figure 11: Cumulative budget for an UE churn of 50%. The requirement of fresh measurements beyond a point is NIL for SkyScale which bolsters its sustainability for longer term deployments.

measurements are unnecessary to reliably predict the REM, unlike SkyRan.

5.2 Trajectory Planning and Segment Discovery

Baseline algorithms. We evaluate SkyScale's trajectory planning algorithm (algorithm 1) and compare its performance with some standard 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 [27, 32] 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$_4$.

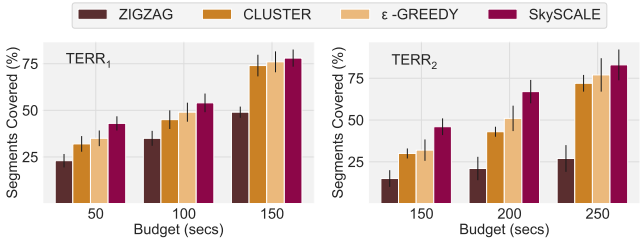

Figure 12: The plot shows how with increasing flight budget the UAV discovers new segments to incorporate them to the RTI equations.

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 course grained leading to poorer accuracy. Given the computational constraints (fig. 13 (*right*)) and desired REM accuracy, one can choose the right number of segments.

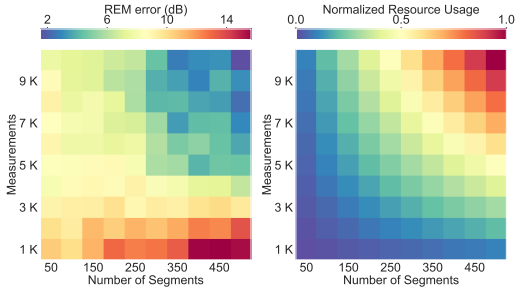

Figure 13: (*Left*) The joint dependency of the number of segments and the amount of measurements in determining an accurate attenuation image and hence, accurate REMs. (*Right*) Normalized resource usage for the same set of parameters. Overall one can use such analysis to maximize accuracy based on 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).

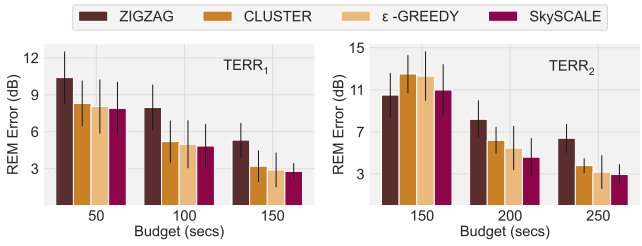

Figure 14: Overall REM errors with increasing measurements for terrains Terr_1 and Terr_2

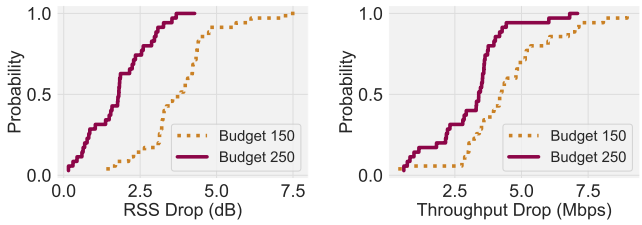

Figure 15: Results related to Experiment 3. TCP Throughput (`iperf3`) and RSS drop under UE churn.

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.

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 ≤ 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.

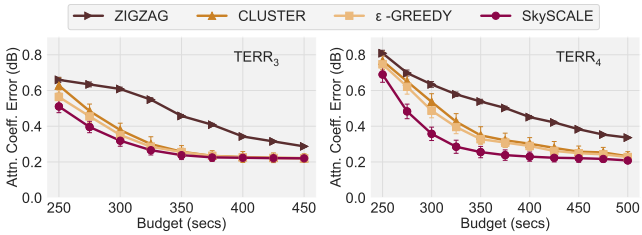

Figure 16: SkyScale is able to predict the attenuation coefficient within an accuracy of 0.2 dB.

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.

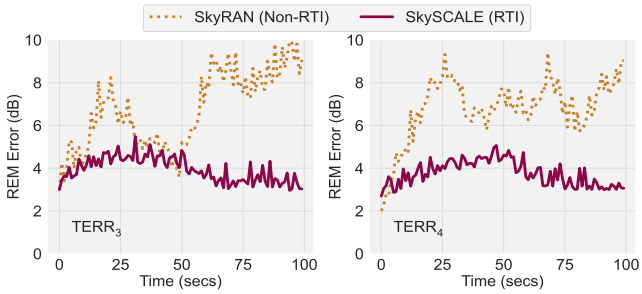

Figure 17: Results related to Experiment 4. SkyScale makes the best out UE mobility to keep a stable performance, unlike SkyRAN.

Experiment 4, (Terr_3 and Terr_4). First, the UAV takes a flight with a budget of 500 secs that contains the average REM error for each of the sixteen UEs within 3 dB. For each of the next 100 seconds, every UE in the simulation moves randomly within the arena. In case of SkyRan (non-RTI), no new information is added to the original REM available at the zeroth second. However, *even without any UAV movement*, SkyScale leverages the mobility of the UEs

and incorporates new measurements into the RTI framework. This is used to perform incremental updates to the attenuation image improving its accuracy. The SkyScale UAV seamlessly moves to the new optimal location. In fig. 17, we show that SkyScale is able to restrict the REM error reasonably well to around 3–4 dB – some of the fluctuations are due to the AWGN noise introduced by the simulator.

6 RELATED WORKS

Recently, UAV based wireless networks is an active area of research touching many interesting aspects in communication, networks and sensing. We restrict the discussion here to efficient construction of Radio Environment Maps (REMs) and related algorithms for optimal UAV placement for improving overall communication performance.

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], [24]) 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([27, 32] and Multi-Arm Bandits [5].Many of such algorithms operate in real-time but takes non-trivial time to achieve stability in performance.

Structural reconstruction using UAV sensing. Some works in this area include UAV based various sensing to reconstruct underlying terrain or structures in 3D, including mmWave [4] or photogrammetry from aerial images [8] or the most recent works on Neural Radiance Fields (NeRFs) [23]. Although such approaches perform a 3D reconstruction, however they do not serve the purpose of creating an attenuation map and hence direct REM creation is not be possible.

REM and RTI related to UAV based deployments. A few works exist that directly compute the 2D or 3D REM [9, 17, 18, 26, 31] based on intelligent spatial sampling and trajectory optimization [21, 25, 33]. The only work that performs 3D RTI in a UAV setting is [22] (that we are aware of). [22] presents an analysis of the placement problem – no extensive performance results are showcased.

Additionally, the majority of the existing literature performs an evaluation with simulation studies – prototype implementation and testbed deployment have been increasingly rare.

7 CONCLUSION

In this work, we presented the design and development of SkyScale, a lightweight UAV positioning system that can scale to large number of users and seamlessly adapt to dynamic network settings. To the best of our knowledge, this is one of the very first works that experimentally demonstrates the usage of RTI for a UAV based wireless network. SkyScale is fundamentally designed to scale and uses minimal to zero additional measurements in the longer term operation over successive flights. Compared to state-of-the-art interpolation based schemes,SkyScale reduces measurement costs by 10× or even more depending on how long the network is operational. However, in case of too few UEs or a relatively static scenario, SkyScale's performance is marginally poorer compared to its interpolation based counterparts. In such cases, we recommend an intelligent fusion of RTI and interpolation schemes, using the best of both depending on the dynamics of the network.

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 *ACM CoNEXT 2018*.
[10] Chakraborty, A., Rahman, M. S., Gupta, H., and Das, S. R. Specsense: Crowdsensing for efficient querying of spectrum occupancy. In *IEEE INFOCOM 2017*.
[11] Chaves-Villota, A., and Viteri-Mera, 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 *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] 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.
[16] Konolige, K. Projected texture stereo. In *IEEE ICRA 2010*.
[17] Matson, N., and Karthikeyan, S. Online Radio Environment Map Creation via UAV Vision for Aerial Networks. *IEEE INFOCOM 2024*.
[18] Mela, L., et al. Radio tomographic imaging with drones. Master's thesis, 2023.
[19] Meyer, F. Color image segmentation. In *International conference on image processing and its applications* (1992), IET.
[20] NASA EarthData. Shuttle Radar Topography Mission Database. https://www.earthdata.nasa.gov/sensors/srtm.
[21] 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.
[22] Romero, D., Viet, P. Q., and Leus, G. Aerial base station placement leveraging radio tomographic maps. In *IEEE ICASSP 2022*.
[23] Rosas-Ordaz, et al. An Overview of NeRF Methods for Aerial Robotics. *Machine Learning for Complex and Unmanned Systems* (2024).
[24] 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).
[25] Sheshadri, R. K., Chai, E., Sundaresan, K., and Rangarajan, S. Skyhaul: A self-organizing gigabit network in the sky. In *ACM MOBIHOC 2021*.
[26] 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.
[27] Steiger, J., Lu, N., and Sorour, S. Learning for path planning and coverage mapping in uav-assisted emergency communications. In *IEEE GLOBECOM 2020*.
[28] 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).
[29] Viet, P. Q., and Romero, D. Aerial base station placement: A tutorial introduction. *IEEE Communications Magazine 60*, 5 (2022), 44–49.
[30] Wilson, J., and Patwari, N. Radio tomographic imaging with wireless networks. *IEEE Transactions on Mobile Computing 9*, 5 (2010), 621–632.
[31] Wu, Q., Shen, F., Wang, Z., and Ding, G. 3d spectrum mapping based on roi-driven uav deployment. *IEEE Network 34*, 5 (2020).
[32] 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.
[33] Zhang, S., and Zhang, R. Radio map-based 3D path planning for cellular-connected UAV. *IEEE Transactions on Wireless Communications 20*, 3 (2020).