Agents for mobile radio tomography

Batenburg, K.J.; Helwerda, L.S.; Kosters, W.A.; Meij, T. van der

Agents for Mobile Radio Tomography

K. Joost Batenburg [a b] Leon Helwerda [c] Walter A. Kosters [c]
Tim van der Meij [c]

[a] *Mathematical Institute, Universiteit Leiden*
[b] *CWI, Amsterdam*
[c] *Leiden Institute of Advanced Computer Science (LIACS), Universiteit Leiden*

Abstract

Mobile radio tomography uses moving agents that perform wireless signal strength measurements in order to obtain a reconstruction of objects inside an area of interest. We propose a toolchain to facilitate the planning, data collection and reconstruction process. The main components are the automated planning of missions for the agents, and the dynamic tomographic reconstruction. Preliminary experiments show that the approach is feasible and results in smooth images that clearly depict objects at the expected locations, when using missions that sufficiently cover the area of interest.

1 Introduction

Radio tomography is a technique for measuring the signal strength of low-frequency radio waves which are exchanged between *sensors* around an area, and reconstructing information about objects in that area. We send a signal between a source and target sensor of a bidirectional *link*. The signal passes through objects that attenuate it, which results in a detectably weaker signal at the receiving end. This phenomenon makes it possible to determine where objects are located. The typical setup for radio tomography is illustrated in Figure 1 (left), in which the sensors are situated on the boundaries in an evenly distributed manner. Gray lines represent unobstructed links and red lines indicate links attenuated by the object.

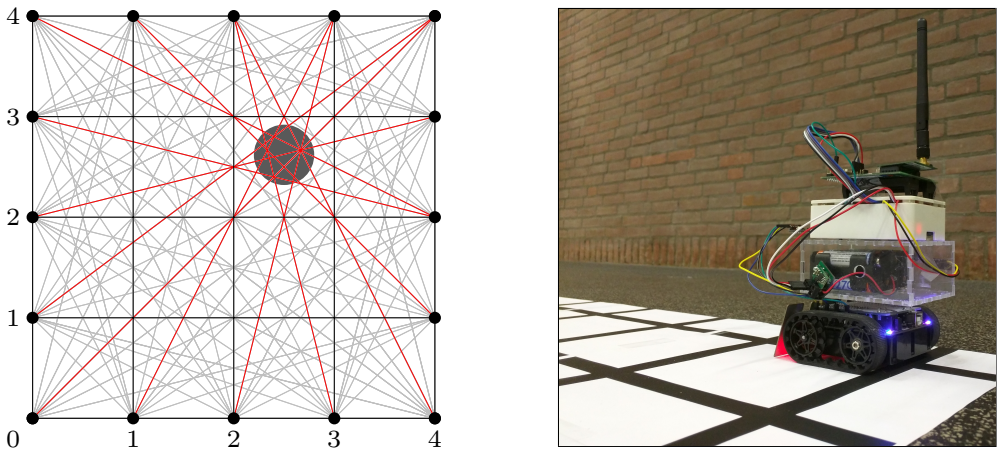


Figure 1: Network of sensors for radio tomography, and the physical realization of a vehicle.

There are benefits of radio tomography over other detection techniques. We can see through walls, smoke or other uninteresting obstacles. The technique is non-intrusive and does not require objects to carry sensor devices themselves. Furthermore, it is less privacy-invasive than optical cameras, because the possible level of detail is limited so that a detected object can only be pictured as an unidentifiable blob.

A static sensor network with a large number of affordable sensors, placed around an area of interest, can be used to reconstruct and visualize a smooth image in real time [12]. This discrete setup is in contrast to the more traditional continuous approach from, e.g., [9]. The drawbacks of such a static network are that it requires a large number of sensors, and there is no way to resolve gaps in the sensor coverage or to react to information obtained through the reconstruction.

One way to resolve these issues is to move the sensors around using *agents*, which are realized as autonomous *vehicles* as pictured in Figure 1 (right). We position them along a *grid* which defines the discrete sensor positions and resulting *pixels* in the reconstructed image. In comparison to the static setup, we require fewer sensors and less prior knowledge about the area. We may adapt the coverage dynamically, for example by zooming in on a part of the area. We name this concept *mobile radio tomography*, which includes both the agent-based measurement collection and the dynamic reconstruction approach.

In this paper, we present our toolchain for mobile radio tomography using intelligent agents. In Section 2 we describe the key challenges for mobile radio tomography and the components in our toolchain that address them. We then cover two such challenges in greater detail: (i) planning the paths of the agents in Section 3, and (ii) reconstructing an image from the measurements in Section 4. Results for real-world experiments are presented in Section 5, followed by conclusions and further research in Section 6. This paper is based on two master's theses on the subject of mobile radio tomography [8, 10].

2 Toolchain

Compared to existing radio tomographic imaging techniques that use statically positioned sensors, imaging using dynamically positioned agents leads to several new challenges. In particular, routes must be planned for each agent to obtain isotropic sampling of the network, short total scanning time and collision-free movement. Images must be reconstructed from the measurements in real time, requiring algorithms and models that work with highly limited and noisy data. Communication between the agents must be interleaved with data acquisition using a robust protocol. To deal with these challenges, we develop an open-source, component-based toolchain. The toolchain is written in Python, with low-level hardware components written in C. The diagram in Figure 2 shows the components in the toolchain.

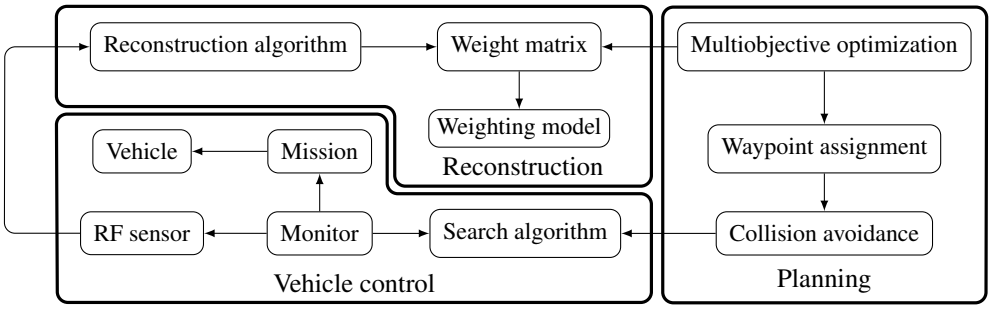


Figure 2: Diagram of components in the toolchain.

The planning components are responsible for automatic mission generation, which we discuss in Section 3. Creating a set of links to measure may be done manually or using an evolutionary multiobjective algorithm that uses a weight matrix to ensure that the links cover the entire network. Next, a waypoint assignment algorithm distributes the sensor positions for each link over the vehicles. Collision avoidance, using a path graph search algorithm, is applied to prevent the vehicles from clashing with each other.

Execution of the mission is taken care of by the vehicle control components. The monitor oversees the process and tracks auxiliary sensors on the vehicles, such as distance sensors for obstacle detection. It makes the RF (radio frequency) sensor perform the signal strength measurements and it may use the search algorithm for collision avoidance during a mission. The mission, consisting of the waypoints for the vehicle, instructs the vehicle to control its motors for moving to the next waypoint.

The reconstruction components convert the signal strength measurements to a two-dimensional image. We study this process in Section 4. The reconstruction algorithm outputs a reconstruction of the area of interest, which can be visualized. The weight matrix determines which pixels are intersected by a link, and a weighting model describes how the contents of pixels contribute to a measured signal strength.

3 Missions

We instruct the autonomous vehicles to travel to locations around the area of interest, that two-by-two correspond to the positions where sensor measurements are performed. The vehicles then execute this *mission*, consisting of *waypoints* that denote which locations they should visit in order. They should choose a short and safe path that does not conflict with any other concurrent route.

We wish to plan the mission algorithmically, rather than assigning the waypoints by hand. The vehicles should visit *sensor positions* such that they can perform a measurement together. This problem is related to other vehicle routing problems [7] with synchronization constraints [5]. We propose a two-stage algorithm, the first part described in textual format, and the second part also in pseudocode.

Assuming that we already know which links we want to measure for collecting tomographic data, we can distribute the positions of the sensors required for each link among the vehicles. Thus, our assignment algorithm is given as input a set $P = \{(p_{1,1}, p_{1,2}), (p_{2,1}, p_{2,2}), \ldots, (p_{\omega,1}, p_{\omega,2})\}$ with ω location pairs of *coordinate tuples* (two-dimensional vectors), and a set $V = \{v_1, v_2, \ldots, v_\eta\}$ of $\eta \geq 2$ vehicles, which are initially located at the predefined coordinate tuples $S_1, S_2, \ldots, S_\eta$. Now define U as the pairwise unique permutations of the vehicles, e.g., with two vehicles, this is $U = \{(v_1, v_2), (v_2, v_1)\}$.

Our greedy assignment algorithm then works as follows: for each vehicle pair $\vartheta = (v_a, v_b) \in U$ and each sensor pair $\rho = (p_{c,1}, p_{c,2}) \in P$, determine the distances $d_1(\vartheta, \rho) = \|S_a - p_{c,1}\|_1$ and $d_2(\vartheta, \rho) = \|S_b - p_{c,2}\|_1$. We use the L^1 norm $\|\cdot\|_1$ because we move only in cardinal directions on a grid; in other applications we may use the L^2 norm $\|\cdot\|_2$. Next, take the maximum of the distances, and finally select the overall minimal pair combination, i.e., solve the following optimization problem:

$$\underset{(\vartheta,\rho)\in U\times P}{\arg\min} \Big(\max(d_1(\vartheta,\rho), d_2(\vartheta,\rho))\Big) \tag{1}$$

The selected positions are then assigned to the chosen vehicle pair, and removed from P. Additionally, S_a becomes the first position and S_b becomes the second sensor position. The greedy algorithm then continues with the next step, until P is empty, thus providing a complete assignment for each vehicle.

Secondly, we design a straightforward collision avoidance algorithm that searches for routes between waypoints that do not *conflict* by crossing any concurrent route of another vehicle; see Algorithm 1. The algorithm is kept simple in order to incorporate it into an evolutionary algorithm (see [11] for more involved methods). We use a search algorithm to find a *safe route* that crosses no other routes. Only when a vehicle performs a measurement involving some other vehicle, their prior routes no longer conflict.

Algorithm 1 The collision avoidance algorithm.

let $V = \{v_1, v_2, \ldots, v_\eta\}$, and $W_1, W_2, \ldots, W_\eta$ be sets, in which $W_i = \{v_i\}$ for $i = 1, 2, \ldots, \eta$
initialize the graph G of possible positions and connections in the area
remove incoming edges of nodes in G that enter forbidden areas, and those of $S_1, S_2, \ldots, S_\eta$
initialize empty sequences of routes $R_1, \ldots, R_\eta$
procedure AVOID($S_1, S_2, \ldots, S_\eta, v_p, v_q, N_p$)
    for all $v_i \in V \setminus W_p$ **do**
        remove the edges for nodes in R_i from G
    end for
    let $R^* \leftarrow$ SEARCH(G, S_p, N_p) ▷ find a safe path R^* in G from S_p to N_p
    append R^* to R_p, reinsert the edges for S_p to G and remove incoming edges for the N_p node
    $S_p \leftarrow N_p$ and $W_p \leftarrow W_p \cup \{v_q\}$
    for all $v_i \in V$ **do**
        if $v_i \notin W_p$ **then**
            reinsert the edges for nodes in R_i to G
        end if
        if $v_i \neq v_p \wedge W_i = V$ **then**
            clear the sequence R_i
            $W_i \leftarrow \{v_i\}$
        end if
    end for
end procedure

Let v_p be the vehicle that we currently assign the position N_p to, and v_q the vehicle that will visit the other sensor position. We also initialize sets $W_1, W_2, \ldots, W_\eta$, where each W_i indicates with which other vehicles the given vehicle v_i has recently performed a measurement. We assume that the search algorithm is given as input a graph G, start point S_p and end point N_p, and outputs a route of *intermediate* points R^*, or an empty sequence if there is no safe path. We can use the collision avoidance algorithm every time the waypoint assignment algorithm assigns a position to a vehicle, so twice per step. Thus, we detect problematic situations as they occur, which are either solved via detours (although the vehicle might also search for a faster safe path during the actual mission), or by rejecting the complete assignment.

In order to supply sensor positions to the waypoint assignment and collision avoidance algorithms, we use an evolutionary multiobjective algorithm [6] to iteratively generate sets of positions that converge toward a theoretical optimal assignment. We keep a population $(X_1, X_2, \ldots, X_\mu)$ of multiple *individuals*, each of which contains variables that encode the positions in adequate form. After a random initialization, the algorithm performs iterations in which it selects a random individual X_i and slightly mutates it to form a new individual [2].

In our case, the variables encode coordinates for positions, of which $m^{(i)}$ are correctly placed such that they intersect the network. From this, we can deduce other information, such as a weight matrix $A^{(i)}$ (containing link influence on pixels, see Section 4) for each individual X_i. Then the algorithm removes an individual that is infeasible according to domains or the *constraints* in Equations 2 and 3 such as a minimum number of valid links ζ, wrapped into a combined feasibility value in Equation 4:

$$(2)\quad Q_1^{(i)} : \exists j : \forall k : A_{j,k}^{(i)} \neq 0 \qquad (3)\quad Q_2^{(i)} : m^{(i)} \geq \zeta \qquad (4)\quad f_i = \begin{cases} 0 & \text{if } \neg Q_1^{(i)} \vee \neg Q_2^{(i)} \\ 1 & \text{if } Q_1^{(i)} \wedge Q_2^{(i)} \end{cases}$$

If all constraints and domain restrictions are met, the multiobjective algorithm uses a selection procedure based on the *objectives*. We remove an individual if its objective values are strictly higher than those of one *dominating* individual in the population. If no individuals are dominated, we remove the one with the least area around it when placed in a plotted function, the *Pareto front*. The objectives to minimize are the two functions (the algorithm favors two over more than two objectives) in Equations 5 and 6:

$$(5)\quad g_1(X_i) = -\sum_{j=1}^{m^{(i)}} \sum_{k=1}^{n} A_{j,k}^{(i)} \qquad (6)\quad g_2(X_i) = \delta \cdot \left(\sum_{j=1}^{m^{(i)}} \left\| p_{j,1}^{(i)} - p_{j,2}^{(i)} \right\|_2 \right) + (1 - \delta) \cdot T^{(i)}$$

For the selection step of the evolutionary algorithm, we use the reconstruction, waypoint assignment and collision avoidance algorithms to check that a new individual adheres to the constraints and to calculate the objective values. Aside from the weight matrix $A^{(i)}$ for one individual X_i, we calculate the pairwise L^2 norms between sensor positions, and $T^{(i)}$, the sum of all minimized distances of Equation 1, weighted by the factor δ. These algorithms generate missions that provide sufficient network coverage.

4 Reconstruction

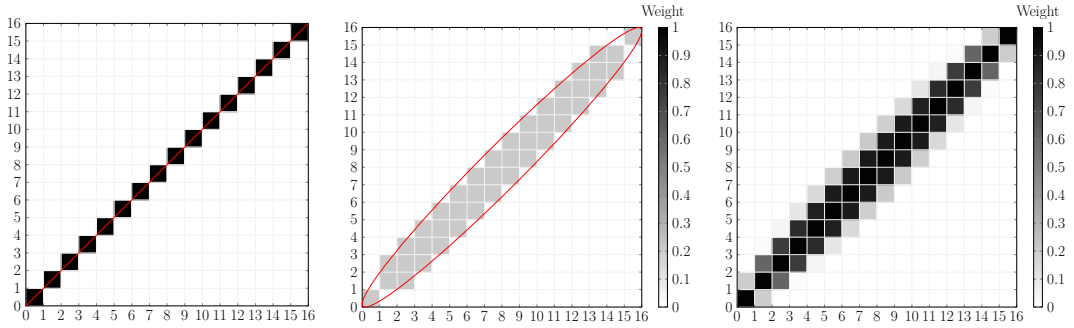
The reconstruction phase converts a sequence of signal strength measurements to a two-dimensional image of $m \times n$ pixels that may be visualized. Let $M = \{(s_1, t_1, r_1), \ldots, (s_k, t_k, r_k)\}$ be the input, in which s_i and t_i are pairs of integers indicating the x and y coordinates on the grid for the source and target sensor i, respectively, r_i is the *received signal strength indicator* (*RSSI*) and k is the total number of measurements. Performing the conversion is done by expressing the problem algebraically as $A\vec{x} = \vec{b}$, in which $\vec{b}$ is a column vector of RSSI values $[r_1, r_2, \ldots, r_k]^T$, $\vec{x}$ is a column vector of $m \cdot n$ pixel values (in row-major order) and A is a weight matrix that describes how the RSSI values are to be distributed over the pixels that are intersected by the link, according to a weighting model.

In general, signal strength measurements contain a large amount of noise due to *multipath interference*. The wireless sensors send signals in all directions, and thus more signals than those traveling in the line-of-sight path may reach the target sensor, causing interference. This phenomenon is especially problematic in indoor environments due to reflection of signals. Techniques exist to include an estimate of the contribution of noise in the model [12], but we suppress noise outside the model using calibration measurements and regularization algorithms. The difficulty lies in the fact that the reconstruction problem is an ill-posed inverse problem, which is primarily caused by the unstable nature of the measurements.

The *weight matrix* A defines the mapping between the input $\vec{b}$ and the output $\vec{x}$. If and only if the contents of a pixel with index i attenuate a link with index ℓ, then the weight $w_{\ell,i}$ in row ℓ and column i is nonzero. Given a link ℓ, a *weighting model* determines which pixels have an influence on the link and are thus assigned nonzero weights, which may be normalized using the link length d_ℓ to favor shorter links [13]. The variable $d_{\ell,i}$ is the sum of distances from the center of pixel i to the endpoints of link ℓ. The *line model* in Equation 7 assumes that the signal strength is determined by objects on the line-of-sight path, as shown in Figure 3a; the *ellipse model* in Equation 8 is based on the definition of Fresnel zones:

$$(7)\quad w_{\ell,i} = \begin{cases} 1 & \text{if link } \ell \\ & \text{intersects pixel } i \\ 0 & \text{otherwise} \end{cases} \qquad (8)\quad w_{\ell,i} = \begin{cases} 1/\sqrt{d_\ell} & \text{if } d_{\ell,i} < d_\ell + \lambda \\ 0 & \text{otherwise} \end{cases} \qquad (9)\quad w_{\ell,i} = e^{-\frac{(d_{\ell,i}-d_\ell)^2}{2\sigma^2}}$$

Fresnel zones, used to describe path loss in communication theory, are ellipsoidal regions with focal points at the endpoints of the link and a minor axis diameter λ. Only pixels inside this region are assigned a nonzero weight, as depicted in Figure 3b. Moreover, we introduce a new *Gaussian model* in Equation 9 which assumes that the distribution of noise conforms to a Gaussian distribution. The log-distance path loss model is a signal propagation model that describes this as well [1]. We use a Gaussian function that assigns most weight to pixels on the line-of-sight path and less weight to pixels farther away from this path, based on the parameter σ, as indicated in Figure 3c.



(a) Line model (b) Ellipse model (c) Gaussian model

Figure 3: Illustration of weight assignment for a link from (0, 0) to (16, 16) by the weighting models.

Due to the ill-posed nature of the problem, in general there exists no exact solution for $A\vec{x} = \vec{b}$ because A is not invertible. Instead, we attempt to find a solution $\vec{x}_{min}$ that minimizes the error using *least squares approximation* [3] as defined in Equation 10:

$$\vec{x}_{min} = \arg\min_{\vec{x}} \left(\left\| A\,\vec{x} - \vec{b} \right\|_2^2 + R(\vec{x}) \right) \tag{10}$$

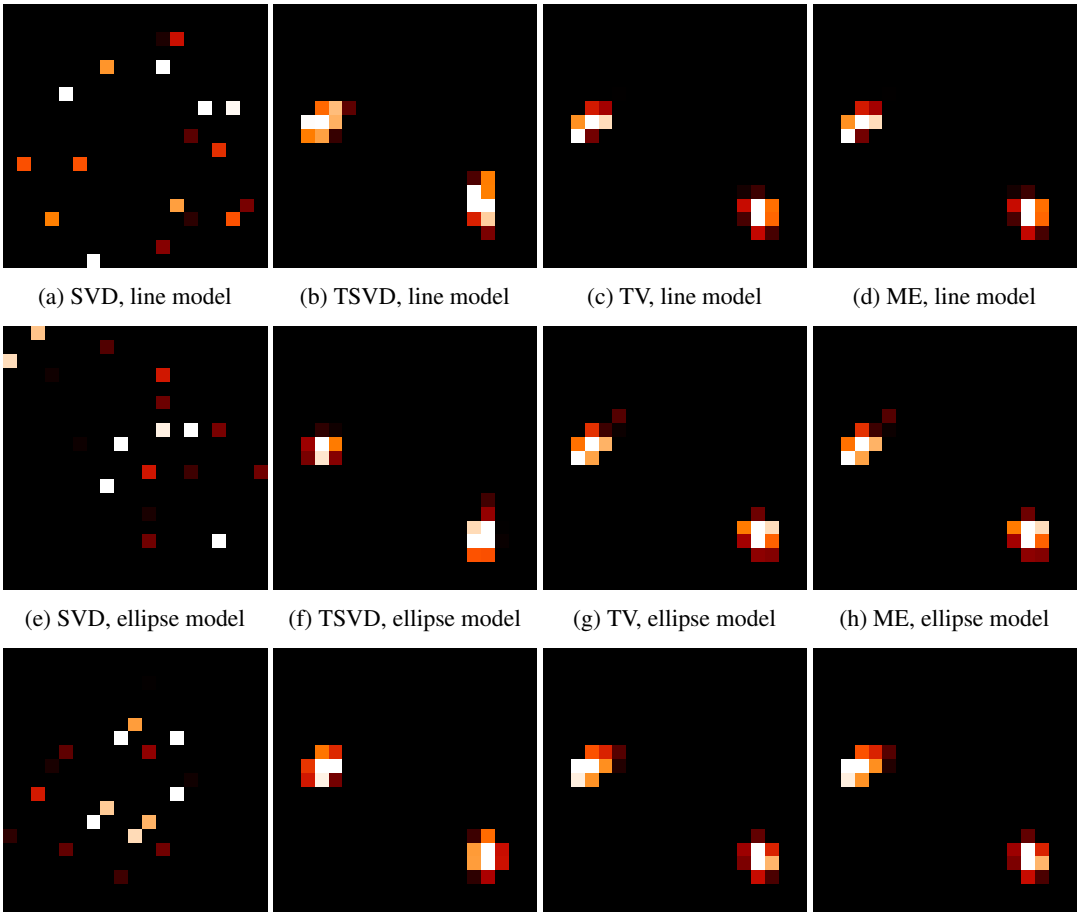
The *singular value decomposition* (*SVD*) of A may be used to solve this and is defined as $A = U\Sigma V^T$, in which U and V are orthogonal matrices and Σ is a diagonal matrix with singular values [13]. Singular value decomposition is exact and does not apply any regularization, so the regularization term $R(\vec{x}) = 0$. *Truncated singular value decomposition* (*TSVD*) is a regularization method that only keeps the τ largest singular values in the SVD and is defined as $A = U_\tau \Sigma_\tau V_\tau^T$ [13]. Small singular values have low significance for the solution and become erratic when taking the reciprocals for Σ.

Iterative regularization methods incorporate desired characteristics of the reconstructed images. *Total variation minimization* (*TV*, see [13]) enforces that the reconstructed images are smooth, i.e., that the differences between neighboring pixels are as small as possible, by penalizing slow changes. The gradient $\nabla\vec{x}$ of $\vec{x}$ is a measure of the variability of the solution. The regularization term in Equation 10 is set to $R(\vec{x}) = \alpha \sum_{i=0}^{\xi-1} \sqrt{(\nabla\vec{x})_i^2 + \beta}$, in which ξ is the number of elements in $\nabla\vec{x}$. The parameter α indicates the importance of a smooth solution and leads to a trade-off as a high value indicates more noise suppression, but less correspondence to the actual measurements. The term is not squared, so we need an optimization algorithm for the minimization. The parameter β is a small value that prevents discontinuity in the derivative when $\vec{x} = 0$, as that generally needs to be supplied to optimization algorithms.

Finally, we consider another variability measure. *Maximum entropy minimization* (*ME*) smoothens the solution by minimizing its entropy. The *Shannon entropy* is defined as $H = -\sum_{i=0}^{\gamma-1} q_i \log_2(q_i)$, in which γ is the number of unique gray levels in the solution and q_i is the probability that gray level i occurs in the solution. Low entropy indicates a low variation in gray levels (which we observe as noise). While this regularization technique is well-known [4], we have found no previous work about its application to radio tomographic imaging. The regularization term in Equation 10 is set to $R(\vec{x}) = \alpha H$. We calculate a numerical approximation of the derivative. The described reconstruction methods and weighting models allow us to obtain a clear image of the area.

5 Experiments

To study the effectiveness of our approach, we perform a series of experiments. Two vehicles drive around on the boundaries of a 20×20 grid in an otherwise empty experiment room. We use hand-made missions that apply common patterns used in tomography, such as fan beams. With this setup we create a dataset with two persons standing in the middle of the left side and in the bottom right corner of the network, and a dataset with one person standing in the top right corner of the network. A separate dataset is used for calibration. The first experiment compares all combinations of regularization methods and weighting models to determine which pair creates the most accurate reconstructions. We use the dataset with the two persons, so both must be clearly visible. The outcome of this experiment is presented in Figure 4, in which darker pixels indicate low attenuation and brighter pixels indicate high attenuation.



(a) SVD, line model (b) TSVD, line model (c) TV, line model (d) ME, line model

(e) SVD, ellipse model (f) TSVD, ellipse model (g) TV, ellipse model (h) ME, ellipse model

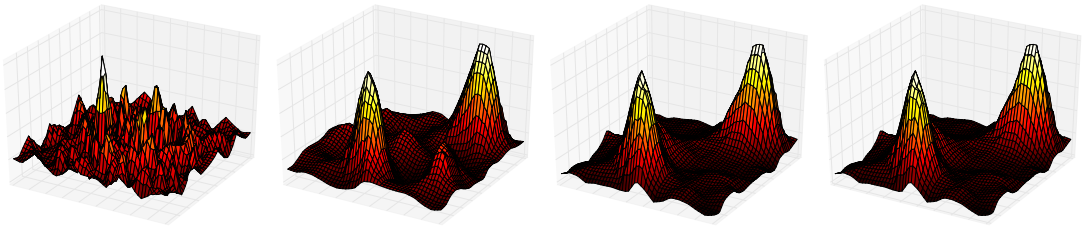
(i) SVD, Gaussian model (j) TSVD, Gaussian model (k) TV, Gaussian model (l) ME, Gaussian model

Figure 4: Reconstructions combining regularization methods and weighting models (two persons dataset).

SVD indeed leads to major instabilities because of the lack of regularization. Noise is amplified to an extent that the images do not provide any information about the positions of the persons. TSVD

provides more stable results that clearly show the positions of the two persons. The ellipse model and the new Gaussian model yield similar clear results, whereas the use of the line model leads to slightly more noise compared to the former two. Even though TV and ME use different variation measures, the reconstructions are visually the same.

Besides providing a clear indication of where the persons are located inside the network, it is important that the reconstructed images are smooth. The second experiment studies the smoothening effects of the regularization methods using 3D surface plots of the raw grayscale images, i.e., without any additional coloring steps applied. The only difference between the experiment runs are the regularization method, so any other parameters remain the same, such as the Gaussian weighting model and the dataset with the two persons that we use. The results for this experiment are shown in Figure 5.



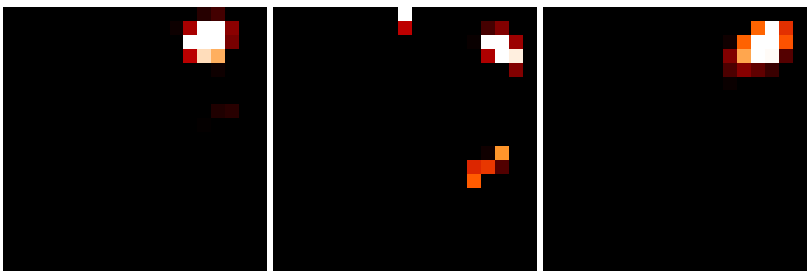
(a) SVD (b) TSVD (c) TV (d) ME

Figure 5: 3D surface plots of the reconstructions for each regularization method (two persons dataset).

The ideal surface plot consists of a flat surface with two spikes exactly at the positions of the persons. The surface plot for SVD is highly irregular, which leads to noticeable noise in the image as there is a high variance in pixel values. In contrast, the surface plot for TSVD is smooth and the two spikes are clearly distinguishable. However, there are still small instabilities. The surface plots for TV and ME are, again, practically the same and have even fewer instabilities.

We now create a mission using the evolutionary multiobjective algorithm from Section 3 and compare it to a hand-made mission. The algorithm places sensors for at least 320 and up to 400 valid measurements. Other parameters are tuned such that they result in lower objective values. At 7000 iterations, we end the run, and pick a solution that optimizes both objectives. The collision avoidance algorithm determines that this assignment is safe.

In Figure 6, we show the images resulting from the tomographic reconstruction of the dataset with one person. The reconstruction, which is run in real time during the collection of measurements, uses TV and the Gaussian model. We can see how long it takes for each mission to provide a smooth and correct result in terms of quality and realism. In Figure 6a, we are around halfway through the planned mission, with 204 out of 382 measurements collected. The reconstructed image clearly shows the person standing in the top right corner, so we end the mission here. Figure 6b shows the reconstructed image obtained from the handmade mission after 413 out of 800 measurements. Although this mission is executed faster due to fewer erratic movements, it is not stable enough to clearly show one person at this point. Figure 6c shows the end result, where the hand-made mission does provide an acceptable reconstructed image.



(a) Algorithmically planned mission, partial result (b) Hand-made mission, partial result (c) Hand-made mission, final result

Figure 6: Reconstructions for algorithmically planned and hand-made missions (one person dataset).

6 Conclusions and further research

We propose a mobile radio tomography toolchain that collects wireless signal strength measurements using dynamic agents, which are autonomous vehicles that move around with sensors. We plan missions, which consist of the locations that the agents must visit and in what order. Novel algorithms provide us with generated missions, guaranteeing that two sensors are at the right locations to perform a measurement. The algorithms avoid conflicts between the routes and provide an optimized coverage of the network.

The measurements are then passed to the reconstruction algorithms to create a visualization of the area of interest that corresponds to the patterns in the data as best as possible. Regularization methods suppress noise in the measurements and increase the smoothness of the resulting image. We introduce a new Gaussian weighting model and apply maximum entropy minimization to the problem of radio tomographic imaging. Results from our preliminary experiments show that the mobile radio tomography approach is effective, i.e., it is able to provide smooth reconstructed images in a relatively short amount of time using algorithmically planned missions.

There is much potential for further research. One interesting topic is to replace the agents, that currently operate on the ground using small-scale robotic rover cars, with drones that fly at different altitudes. This leads to 3D reconstruction, e.g., by performing a reconstruction at different altitudes and combining the images, which are slices of the 3D model. The reconstruction algorithms could also be modified to allow performing measurements anywhere in the 3D space, although this makes the problem more difficult to solve in real time. Finally, improvements can be made to the evolutionary multiobjective algorithm, such as altering the objectives or using predetermined patterns that we encode in the variables. It is also not entirely clear yet how the network coverage, or the lack thereof, influences the quality of the reconstruction. The greedy waypoint algorithm could factor in the time that certain actions take, such as turning around. Altering missions dynamically helps making adaptive scanning a viable approach.

References

[1] J. B. Andersen, T. S. Rappaport, and S. Yoshida. Propagation measurements and models for wireless communications channels. *IEEE Communications Magazine* 33 (1995), pp. 42–49.

[2] T. Bäck. Evolutionary algorithms in theory and practice: Evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press, 1996.

[3] A. Björck. Numerical methods for least squares problems. SIAM, 1996.

[4] A. C. Bovik. Handbook of image and video processing. Academic Press, 2005.

[5] D. Bredström and M. Rönnqvist. Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. *European Journal of Operational Research* 191.1 (2008), pp. 19–31.

[6] M. Emmerich, N. Beume, and B. Naujoks. An EMO algorithm using the hypervolume measure as selection criterion. In: *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization*. LNCS 3410. Springer, 2005, pp. 62–76.

[7] B. L. Golden, S. Raghavan, and E. A. Wasil. The vehicle routing problem: Latest advances and new challenges. Springer, 2008.

[8] L. Helwerda. Mobile radio tomography: Autonomous vehicle planning for dynamic sensor positions. Master's thesis. LIACS, Universiteit Leiden, 2016.

[9] A. C. Kak and M. Slaney. Principles of computerized tomographic imaging. SIAM, 1999.

[10] T. van der Meij. Mobile radio tomography: Reconstructing and visualizing objects in wireless networks with dynamically positioned sensors. Master's thesis. LIACS, Universiteit Leiden, 2016.

[11] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence* 219 (2015), pp. 40–66.

[12] J. Wilson and N. Patwari. Radio tomographic imaging with wireless networks. *IEEE Transactions on Mobile Computing* 9 (2010), pp. 621–632.

[13] J. Wilson, N. Patwari, and F. Guevara Vasquez. Regularization methods for radio tomographic imaging. In: *Virginia Tech Symposium on Wireless Personal Communications*. 2009.