ABSTRACT

Title of Dissertation: IMMERSIVE VISUAL ANALYTICS OF WI-FI SIGNAL PROPAGATION AND NETWORK HEALTH

Alexander Rowden

Dissertation Directed by: Professor Amitabh Varshney
Department of Computer Science

We are immersed in waves of information. This information is typically transmitted as radio waves in many protocols and frequencies, such as WiFi, Bluetooth, and Near-Field Communications (NFC). It carries vital information such as health data, private messages, and financial records. There is a critical need for systematic and comprehensive visualization techniques to facilitate seamless, resilient, and secure transmission of these signals. Traditional visualization techniques are not enough because of the scale of these datasets. In this dissertation, we present three novel contributions that leverage advances in volume rendering and virtual reality (VR): (a) an outdoor volume-rendering visualization system that facilitates large-scale visualization of radio waves over a college campus through real-time programmable customization for analysis purposes, (b) an indoor, building-scale visualization system that enables data to be collected and analyzed without occluding the user's view of the environment, and (c) a systematic user study with 32 participants which shows that users perform analysis tasks well with our novel visualizations.

In our outdoor system, we present the Programmable Transfer Function. Programmable Transfer Functions offer the user a way to replace the traditional transfer function paradigm with a more flexible and less memory-demanding alternative. Our work on indoor WiFi visualization is called WaveRider. WaveRider is our contribution to indoor-modeled WiFi visualization using a virtual environment. WaveRider was designed with the help of expert signal engineers we interviewed to determine the needs of the visualization and who we used to evaluate the application. These works provide a solid starting point for signal visualization as our networks transition to more complex models.

Indoor and outdoor visualizations are not the only dichotomy in the realm of signal visualization. We are also interested in visualizations of modeled data compared to visualization of data samples. We have also explored designs for multiple sample-based visualizations and conducted a formal evaluation where we compare these to our previous model-based approach. This analysis has shown that visualizing the data without modeling improves user confidence in their analyses. In the future, we hope to explore how these sample-based methods allow more routers to be visualized at once.

IMMERSIVE VISUAL ANALYTICS OF WI-FI SIGNAL PROPAGATION AND NETWORK HEALTH

BY

ALEXANDER ROWDEN

DISSERTATION SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL OF THE
UNIVERSITY OF MARYLAND, COLLEGE PARK IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
2023

ADVISORY COMMITTEE:
PROFESSOR AMITABH VARSHNEY, CHAIR/ADVISOR
PROFESSOR JOSEPH JAJA
DR. ERIC KROKOS
PROFESSOR NIRUPAM ROY
DR. KIRSTEN WHITLEY

Table of Contents

List of Tables

List of Figures

List of Abbreviations and Initialisms

AR	Augmented Reality
BSSID	Basic Service Set Identifier
CAVE	Cave Automatic Virtual Environments
CPU	Central Processing Unit
CSV	Comma-Separated File
CT	Computed Tomography
dBm	Decibel-milliwatts
FPS	Frames Per Second
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HMD	Head Mounted Display
LIC	Line Integral Convolution
MAC	Media Access Control
MRI	Magnetic Resonance Imaging
PTF	Programmable Transfer Function
RAM	Random Access Memory
RF	Radio Frequency
SSID	Service Set Identifier
SUS	System Usability Scale
TLX	Task Load Index
VR	Virtual Reality
WiFi	Wireless Fidelity
WLAN	Wide Local Area Network

Chapter 1: Introduction

Our most sensitive information and urgent communications happen wirelessly over the radio frequency (RF) spectrum. When doctors pull up patient information, they often do so wirelessly using WiFi. Secure facilities often use passive Radio Frequency Identification (RFID) to protect their spaces. Therefore, ensuring that these networks are healthy, fast, constantly available, and secure is vital. In our works, we use WiFi as our source of RF signal data but strive for generalizability to other RF signal sources.

Signal data is notoriously difficult to visualize comprehensively as signals propagate in three dimensions unevenly, and sensors are noisy. Users of these visualizations need to understand the context of the signal in the physical environment and its relationship to the other communications around it. Immersive technology, such as virtual reality (VR), allows one to visualize signals more effectively than traditional displays. It enables data to be displayed directly in its environmental context. This allows analysts to make decisions holistically and develop insights that would have been difficult to come to otherwise. It allows users to interact with the data in three dimensions: their natural environment.

The visualization of large datasets is a significant challenge in the field of visual analytics. Signal data falls under this category because signal datasets often contain thousands of multidimensional data points that cover several routers, and each router has many attributes a user may be interested in, such as its frequency, service set identifier (SSID), basic service set identifier (BSSID), and security capabilities. Thus, the challenges in the visualization of large datasets apply to the visualization of RF signals. For instance, we must tackle the issues of overdrawing, environmental and self-occlusion, and visualizing multiple data series. Overdrawing occurs when glyphs overlap each other. Occlusion is a related challenge, where glyphs block the user from seeing additional information. In the case of environmental occlusion, the visualization interferes with the user's ability to see the environment. Self-occlusion is where one part of a visualization blocks the user's ability to see another section of the visualization. Finally, there is the more general issue of visualizing multiple data series, like overlapping RF access points.

Large outdoor areas, such as campuses and smart cities, are challenging to manage as a network administrator. Due to their scale, testing the network health at every location in the region is physically and mentally demanding. Further, ensuring that the most critical areas have enough bandwidth without causing interference is also vital. Our first work [1], which we present in Chapter 2, attempted to tackle this challenge. We developed a large-scale visualization using a specialized form of volume rendering that utilized a novel transfer function. Our work allows the lightweight computation of multiple large-scale datasets while allowing the user to customize the visualization for their analysis.

Indoor areas are also challenging for network visualization. As the environment is more closed off, occlusion becomes a more significant issue. With that in mind, we developed a new visualization strategy, which we present in Chapter 3. In this work, we created an application called WaveRider [2] to visualize a pre-collected dataset in a digital twin of the environment. This allowed us to focus on the visualization design rather than the complex challenge of implementing a real-time AR system. We developed our strategy with a team of signal experts and developed a set of requirements for an indoor visualization. Following that, we developed four visualizations and packaged them in an application called WaveRider. Our experts determined that our visualizations were intuitive and significantly improved state-of-the-art.

The WaveRider visualizations were intuitive and allowed users to localize to the router efficiently. However, they were computationally expensive, visually dense, and relied on a RF propagation model. All this combined meant that while users could obtain a solid understanding of a few routers based on a computational model, they could not easily survey a space looking at many routers. Additionally, any analysis using these visualizations would depend highly on the propagation model. Chapter 4 presents our work tackling these problems with two more visualizations, which we designed to be computationally light-weight and require no propagation model [3]. We then conducted a user study with 32 participants to compare these visualizations to the state-of-the-art in four common analysis tasks. Our novel visualizations compared favorably, matching or exceeding the state-of-the-art in every case.

1.1 Programmable Transfer Functions

1.1.1 Introduction

Large outdoor networks are common and vital to our world. The cellphone network is one such network, where coverage is sometimes of life-and-death importance. Ensuring that our outdoor networks are secure, efficient, and have their intended range is crucial. Here, we present a summary of how we tackled this challenge. Details can be found in Chapter 2. In this work, we developed a novel transfer function for volume rendering, which we employed to visualize the networks on the campus of the University of Maryland. Our technique allows lightweight and

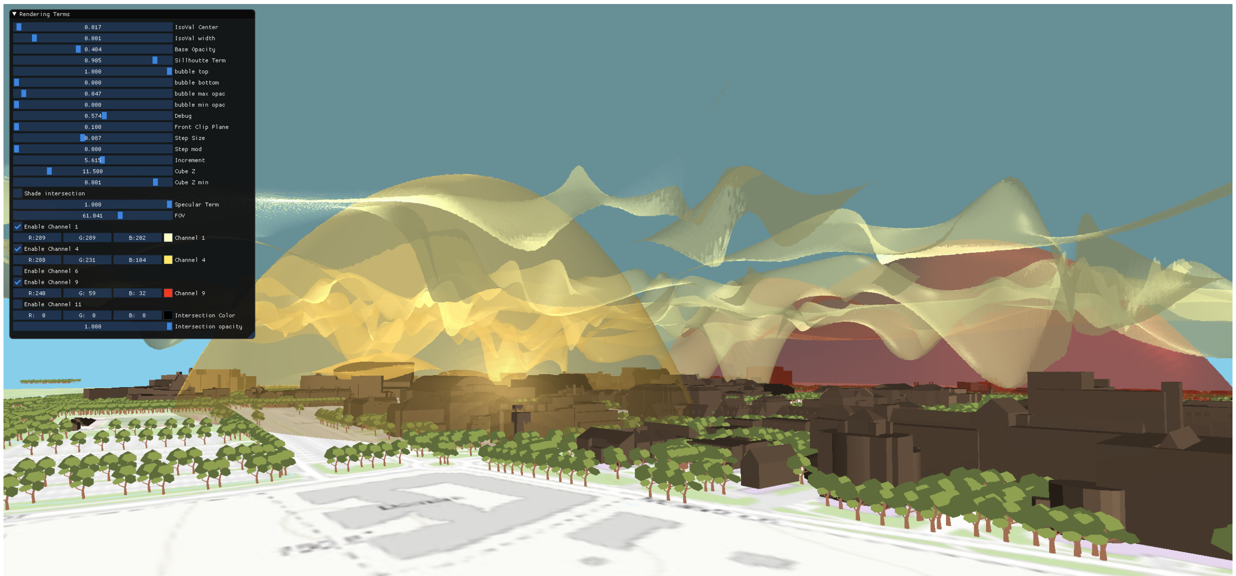


Figure 1.1: An example of the full-campus rendering we achieved using our programmable transfer function. Here, we show three networks with different properties. Also included in the screenshot is our GUI, which allows the user to customize the rendering, to-scale models of the campus buildings, and the campus plant inventory to increase user immersion and aid in user localization.

customizable renderings of any three-dimensional scalar field dataset.

1.1.2 Approach

To visualize these large-scale networks, we decided to model them as three-dimensional volumes and render them using volume rendering. Modeling the networks this way was a natural choice, as the RF signals propagate in three-dimensional space. To limit the occlusion of the system, we decided to render the volumes as thin transparent isosurfaces. Isosurfaces are parts of a three-dimensional scalar field where all points have the same scalar value. This allows the viewers of our visualization to see all points in a router's coverage that have the same signal strength. This gives a very intuitive grasp of the shape of a router's signal propagation. We decided to reinvent the transfer function to store our large datasets and have highly customizable multidimensional transfer functions, which have become the standard in volume rendering. Tra-

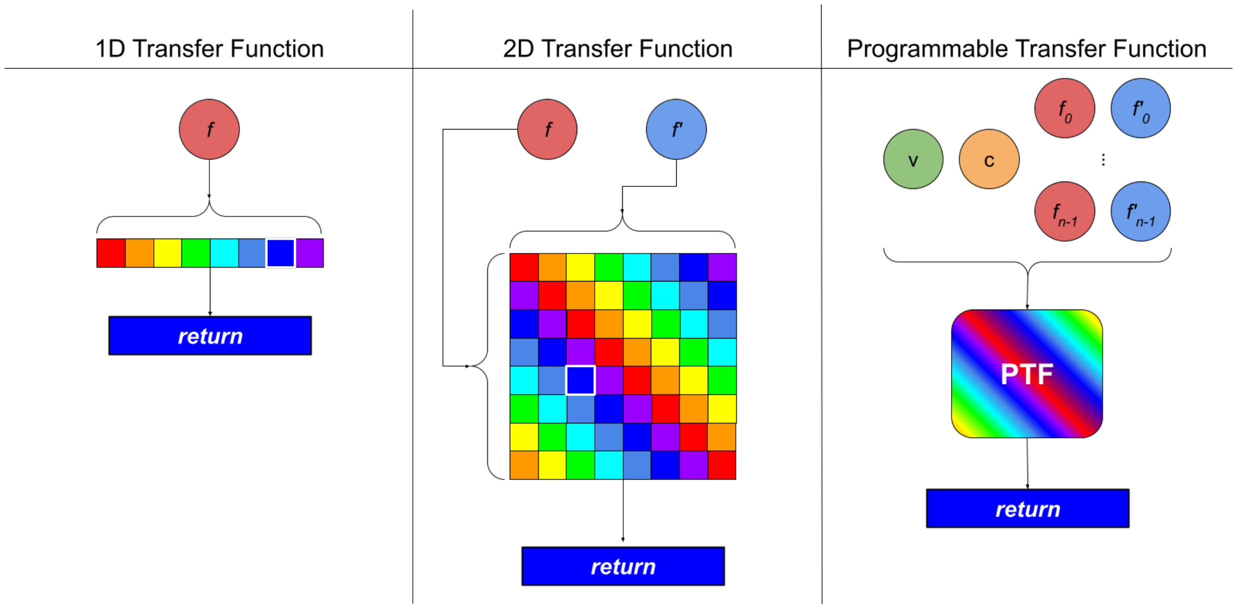


Figure 1.2: Visualization of the trade-off between single-dimensional, two-dimensional, and programmable transfer functions.

ditional transfer functions use large look-up tables, which must be computed on the CPU and then transferred and stored on the GPU for rendering. This can become a heavy burden on GPU memory for high-quality renderings if the resolution and dimensionality of the function become large. Thus we replaced these burdensome tables with customizable shader code. This trade-off is visualized in Figure 1.2. Programmable Transfer Functions benefit from the creation of exceptionally fast Graphics Processing Units (GPUs) as they replace the memory overhead with added computation. This trade-off also gives us several additional advantages. Firstly, we no longer need to calculate the function ahead of time. Transfer functions can be created and modified on the fly without incurring the additional expense of sending the data from the CPU to the GPU. Secondly, GPU memory is becoming increasingly in demand as environmental models and textures are becoming more detailed in modern rendering practices. We also make room for more complicated and higher-dimensional transfer functions by removing the memory burden. Multidimensional transfer functions often elucidate more information from volumes but come at a high memory cost. With programmable transfer functions, the cost of adding a dimension is negligible.

1.1.3 Results

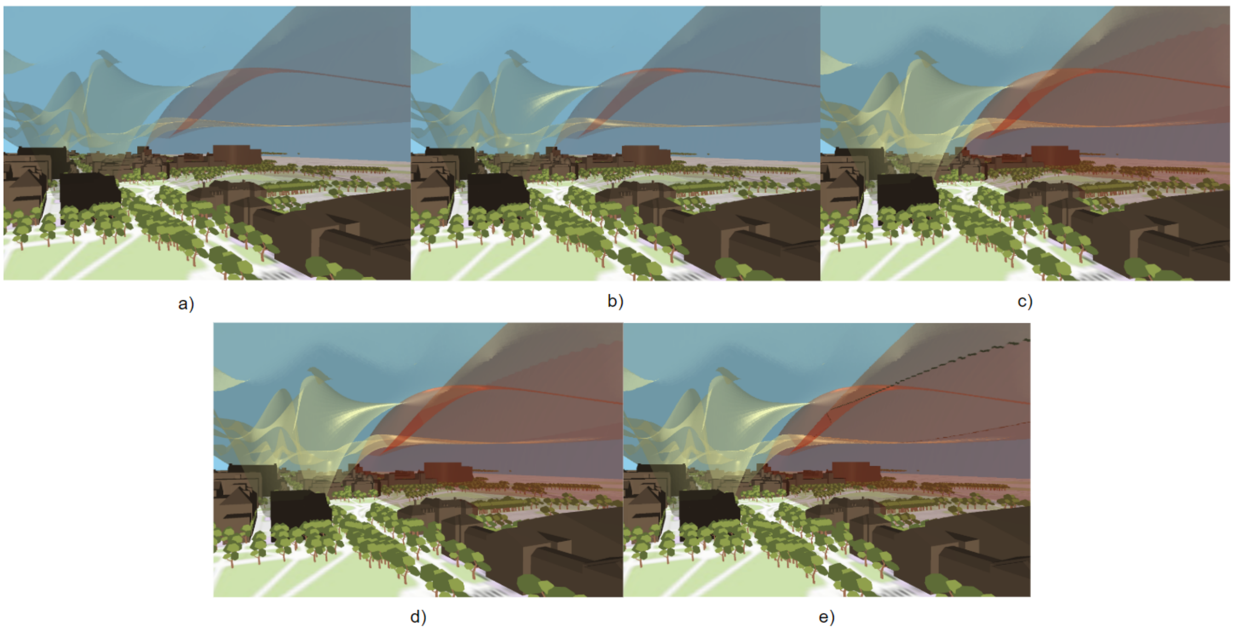


Figure 1.3: Examples of the customization that PTFs can provide.

To evaluate the usability of our programmable transfer functions, we implemented a few custom shaders to show off the utility and versatility they offer. These transfer functions demonstrated different ways PTFs are advantageous over traditional transfer functions. Our first custom PTF utilized the normal of the isosurface and the direction from the light to the surface to add opacity to the volume in the region of the specular highlight. This allows users to more easily see the curvature of the volume since the specular highlight is often washed out in thin, translucent surfaces. Thus, they have a more intuitive understanding of its geometry. This specular augmentation PTF highlights the versatility of the technique because it can utilize additional variables and dimensions, such as the light direction. Similarly, we created a PTF that increases the isosurface's opacity at its boundaries, which we call Silhouette Shading. This technique requires access to the surface normal and the direction from the viewer to the shaded position. Finally, to highlight PTFs' ability to enable specialized rendering for multiple scalar fields, we created a PTF to highlight the intersection between two volumes. The result of these PTFs can be seen in Figure 1.3.

	Base	Specular	Silhouette	Spec+Silh	Intersection	All 3
1 Volume (interior)	105	105	105	105	NA	NA
1 Volume (exterior)	181	181	181	181	NA	NA
5 Volumes (interior)	85	85	85	85	85	85
5 Volumes (exterior)	116	116	116	116	116	116

Table 1.1: The frame rates of our example Programmable Transfer Functions on one volume and five volumes. These rates were recorded both inside and outside the volumes.

We also conducted two additional tests of their usability. In one, we tested Programmable Transfer Function's practical utility by recording the frame timings in various conditions to prove that these renderings can achieve real-time interactive rates, see Table 1.1. We also developed a VR experience that allowed users to walk, move around the campus, and be immersed in the campus's networks. We believe that such an outdoor visualization can increase the quality of our large-scale networks by giving network administrators and planners ways to evaluate network coverage and see how the signals interact with their environment.

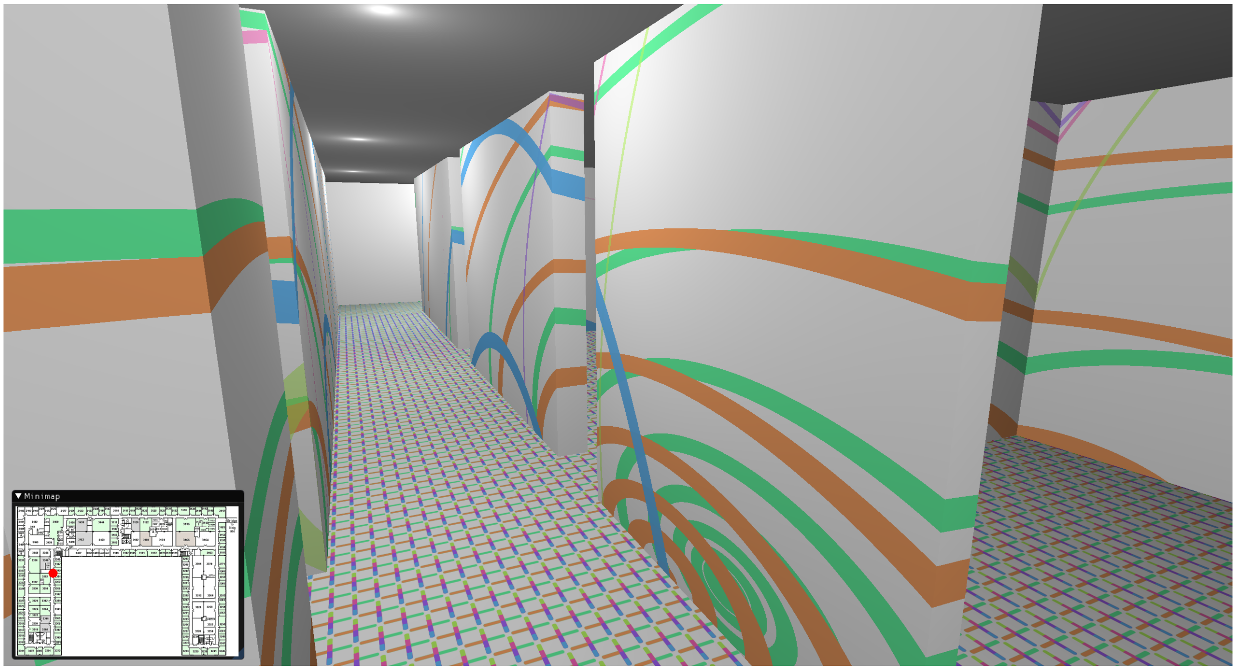


Figure 1.4: Screenshot of the desktop version of WaveRider shows several routers simultaneously visualized in an indoor environment.

1.2 WaveRider: Immersive Visualizations of Indoor Signal Propagation

1.2.1 Introduction

Indoor signal visualization is arguably more important. Many of our most vital systems exist in our home and work environments or places like hospitals and courtrooms. Visualizing networks in these environments poses its own set of issues. Namely, how does one visualize so much data without blocking the user's line of sight to critical environmental features? Not only would this occlusion impede an analyst's ability to make decisions about the network based on its environment, but it also would limit the utility of the visualization in Augmented Reality (AR), where virtual objects are placed on top of the real world. AR visualizations are increasing in popularity partly due to the mass commercial availability of headsets and because they allow users to augment their view of the world without hindering their ability to perceive it.

Since routers must be placed close together to ensure dense coverage and high bandwidth in an indoor context, routers are more likely to interfere with one another. There are two kinds of frequency interference. The first form of interference is adjacent channel interference, where routers whose frequency bands overlap with each other can cause each other's signals to modulate. This modulation will cause packet loss and slow communication. Adjacent channel interference is uncommon, however, as most routers are configured on non-overlapping frequency bands, and frequency modulation is often recoverable. The other form of frequency interference is cochannel interference. In cochannel interference, multiple routers communicate on the same frequency band. In this case, a router must wait to communicate until the band is clear. This can cause long stalls. If a router's communication is interrupted by another router trying to communicate on its frequency, it will likely need to be repeated. This interference is common and a common source of network inefficiency, so our visualization must account for it.

Here, we present a summary of Chapter 3, where we introduce WaveRider. WaveRider is the application we created that introduces several visualization techniques to the world of signal visualization and fuses them to tackle the tasks required for indoor signal analysis. We also provide an evaluation by signal experts and offer directions for the future of indoor signal visualization.

1.2.2 Approach

To develop WaveRider, we spoke with professional network engineers to establish the requirements for an indoor signal visualization. Using those requirements, we gathered visualization techniques from other areas, like the visualization of multiple overlapping scalar fields and

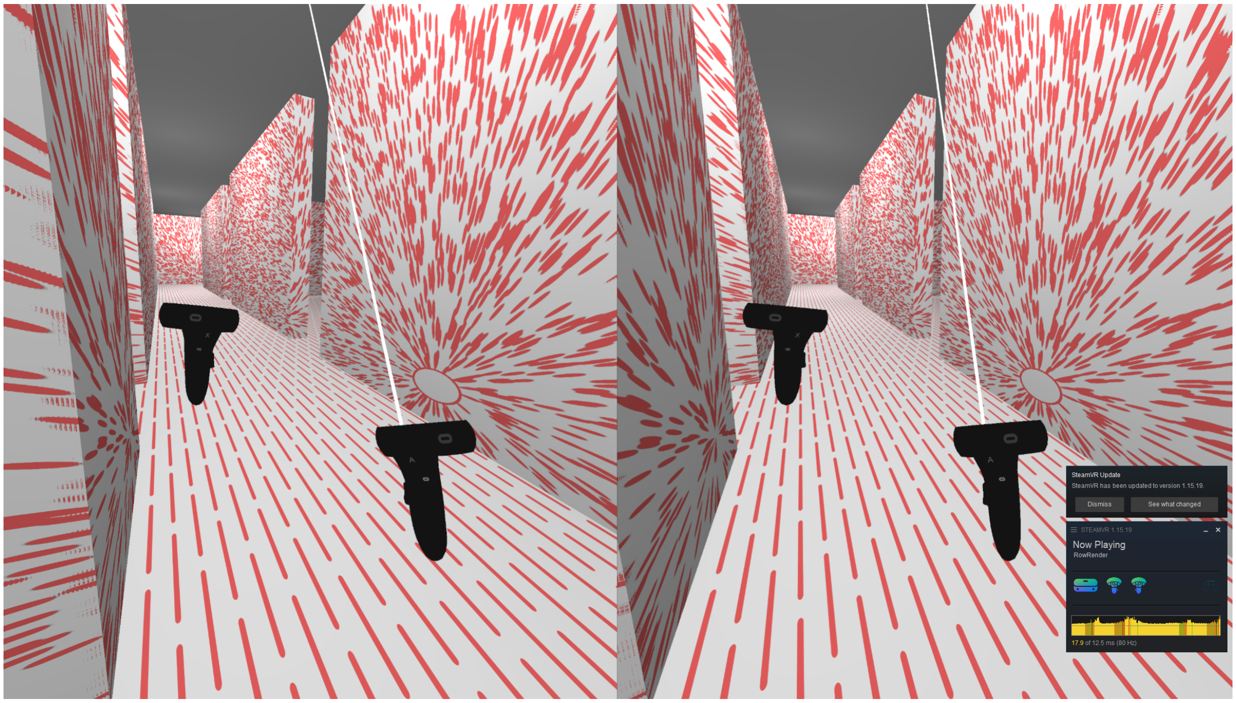


Figure 1.5: Screenshot of WaveRider visualizing a single router using Line Integral Convolution in Virtual Reality

fluid flow, to confront the challenge from a new perspective. When put together, we developed four visualizations. Three tackled the challenge of visualizing signal strength and router coverage, while one visualization allowed the user to visualize the frequency each router utilized. We fused these visualizations by using the natural segmentation of the walls and floor. Users view the signal data on the walls, allowing them to view the signals in three dimensions, while the floor provides them with supplementary frequency knowledge, should they need it.

To visualize our data, we decided to model it using Principle Component Analysis. We chose to model our data so that the user could see the signal level at locations between data samples and to avoid glyph-based visualizations, which would overdraw, as the routers share the same set of sample locations. This gave us ellipsoids we could visualize instead of raw data. We decided to use ellipsoids as they are simple to understand and do a fair job of fitting our data. Nevertheless, to ensure WaveRider's utility, we developed our visualizations to be model-agnostic. The only requirement on the router model is that a function must exist to map the world position to signal strength and a direction to the router.

1.2.2.1 Contour Lines

With our model in place, we were free to develop our visualizations. The first visualization we created was contour lines based on topographical maps. In contour lines, the wall is shaded at fixed signal strengths as the router's signal decays. This would produce a series of concentric ellipses on each wall. The thickness of the contour line, transparency, and the number of contours to the center determine the router's signal strength. The direction to the router's source is conveyed in the curvature of the ellipse. An example of the contour lines can be seen in Figure 1.4.

1.2.2.2 LIC

We designed an alternative visualization based on the fluid flow visualization technique, line integral convolution (LIC). LIC is used for visualizing vector fields. We utilize it to visualize the field of vectors that point to the router's location. LIC produces an image consisting of lines pointing in the direction of the vector field. This allows the user to navigate directly to the signal source without needing to estimate the direction themselves. The signal strength is encoded in the transparency of the lines. An example of LIC can be seen in Figure 1.5. This LIC technique works for one router.

We developed two techniques for visualizing multiple routers. The first is called Layered LIC, where separate LIC images are composited on top of one another to produce the final visualization. This visualization allows the user to visualize multiple routers in every position, but it suffers from visual clutter and overlap as the number of routers increases. It also suffers from performance issues, as each router's LIC image needs to be calculated individually. The other technique is called MaxLIC. In MaxLIC, we produce one LIC image. Instead of visualizing the vector to one router, we visualize the vector to the router with the highest signal strength in the visualized router set. This technique has a few interesting properties. Firstly, it allows for a constant computation time, as no matter how many routers we visualize, we only need to compute one LIC image. Another benefit is that MaxLIC produces a boundary line where two routers are equal. This feature has some analytical value as it highlights the region where two routers have similar strengths, which is where they are more likely to interfere. MaxLIC does, however, have the disadvantage that by looking at a single segment of the wall, a user will not know how many routers and what routers are present, as only one router is shown at a single point.

1.2.2.3 Textons

Our final visualization technique comes from the visualization of multiple scalar fields and is called Oriented Texture Slivers. Oriented texture slivers are a variety of textons, the base unit of precognitive texture perception, which has been used to visualize categorical data. Textons, in general, are useful in visualization because a user can easily distinguish between different textons without the mental overhead. Oriented texture slivers are a variety of textons consisting of thin lines pointing in different directions. The orientation of these lines can be used to encode categorical data. We, however, modify the usage. Our slivers are oriented based on the frequency of the router. However, we can encode more data by assigning a color to each texton that identifies the router it is associated with. Additionally, to reduce visual clutter and highlight routers operating on the same frequency, we group textons oriented in the same direction onto a single line. An example of textons can be seen in Figure 1.4.

1.2.3 Evaluation

Once we created WaveRider, we immersed our experts in the VR version of the application to gather their feedback on each of our visualizations. We guided each expert through the visualizations and asked them to complete tasks, such as localization and determining if routers could interfere with each other. During the application demo, we encouraged the experts to ask questions and asked if they wanted to tweak the visualization. After showing off the application, we took each expert through a semi-structured interview to get their feedback and suggestions for how to improve WaveRider and to see what additional features of a visualization they would be interested in. The experts were satisfied with the utility of our visualizations and were surprised by their intuitiveness.

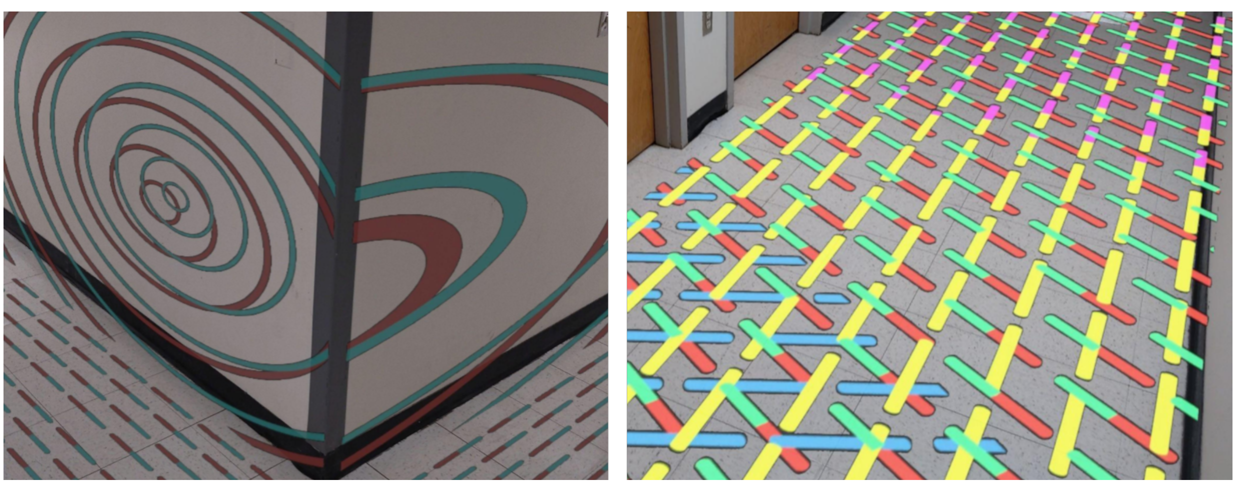
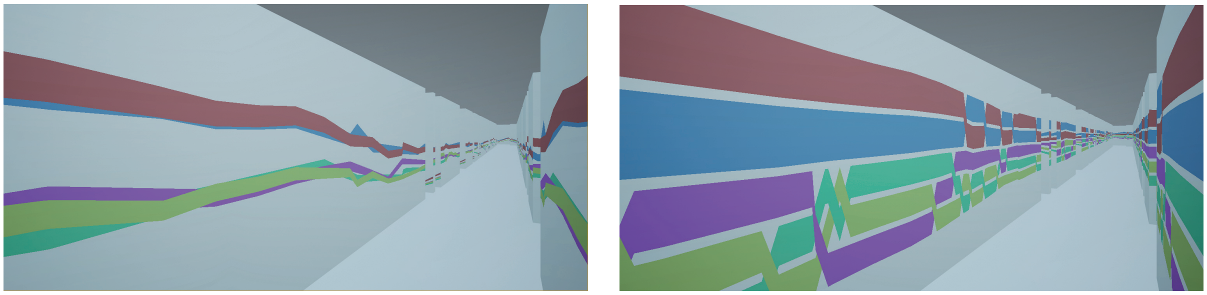


Figure 1.6: Screenshot of WaveRider visualizing a single router using Line Integral Convolution in Virtual Reality



(a) Wavelines (b) Stacked Bars

Figure 1.7: The novel visualizations created for light-weight surveying of large indoor environments. a: Visualization that emphasizes localization and estimating exact signal-strength values. May contain large areas of overlap. b: Visualization that emphasizes the ranking of signal strengths across multiple routers. Designed to contain minimal overlap.

We also implemented an AR version prototype of WaveRider to evaluate how WaveRider's design would transfer to the new medium. This prototype highlighted the fact that small details in a visualization do not come through as well in AR applications due to the complex textures and dynamics of the real world.

1.3 Exploring Effective Immersive Approaches to Visualizing WiFi

1.3.1 Introduction

Our work with WaveRider was a massive success. Our experts verified both the need for these visualizations and their efficacy. Still, there were a few shortcomings that provided an opening for iteration and improvement. Firstly, LIC was computationally expensive, preventing more than a few routers from being represented. Both visualizations suffered from visual cluttering, where data overwhelmed the user when analyzing many routers. Additionally, both of these visualizations rely on modeling the data before the data can be visualized. For our purposes, we used a relatively simple model, but more complicated models could be used. However, any analysis made with these visualizations will be limited by the correctness of the propagation model used.

The shortcomings of the visualizations used in WaveRider set the stage for our third work, which we present in Chapter 4. This work presents two more novel visualizations seen in fig. 1.7 - Stacked Bars fig. 1.7b and Wavelines fig. 1.7a. These visualizations require no propagation model and are not computationally intense, thus allowing for the real-time visualization of many routers (our system rendered up to 40 at interactive rates in VR). With these visualizations in hand, we ran a user study to see how these visualizations perform when compared to our previous work (represented by contour lines) by testing user performance at four different analysis tasks.

In the following summary, we will discuss our novel visualizations and our user study at a high level of detail. For more information, please review Chapter 4.

1.3.2 Visualizations

This work drew inspiration from our WaveRider work in two key ways. Firstly, our visualizations were represented solely on the natural features of the environment in order to reduce occlusion and visual clutter. Our visualizations also used WaveRider's model of using the walls for visualizing the signal strength of the routers and using the floor to represent channel information via textons. From there, we set out to design non-cluttered visualizations of signal data for the walls.

1.3.2.1 Wavelines

The first visualization we developed while keeping visual clutter in mind was Wavelines fig. 1.7a. We decided to draw a single line for each visualization at any point on the wall, thus minimizing visual clutter. From there, we decided to encode signal strength in the height and thickness of each line. In this way, Wavelines reads similarly to a bar graph with routers with higher signal strengths on the top and those with lower signal strengths near the floor. This allows the visualization to be incredibly intuitive to read and makes seeing a router's high signal strength area obvious from a distance. On the downside, when the routers have similar signal strengths, they overlap. This can make areas with many routers with equal signal strength difficult to interpret. With this in mind, we developed the Stacked Bars visualization to guarantee minimal overlap.

1.3.2.2 Stacked Bars

In order to guarantee that each router line does not overlap, we drew each line on top of one another with thickness proportional to signal strength. We placed the routers in order of their signal strength, with the weakest on the bottom. When the ranking changes, the order of the lines will change; thus, we allow a temporary overlap in these areas to create continuous lines for the user to trace over the environment. We call this visualization Stacked fig. 1.7b. As we intended, stacked contains minimal overlapping lines and gives the user the ability to see every router at all locations. It, however, loses Waveline's ability to see high signal areas at a distance because the signal strength is no longer encoded directly in wall height. After designing both of these visualizations, we were curious to see how they would perform in comparison to each other and the state of the art so we conducted a user study evaluation.

1.3.3 User Study

In order to validate our designs, we ran a user study with 32 participants. Each user performed four analysis tasks with each visualization. These tasks were localization, ranking, coverage, and interference. A full description of these tasks can be found in Chapter 4. At the end of the user study, we collected subjective feedback from the users. The results show that our novel visualizations perform as well as, if not better, than contour lines in every task. Wavelines outperform Stacked at the localization task, and Stacked does the best at the ranking task. Thus, our new light-weight visualizations are an improvement on the state-of-the-art that does not depend on a propagation model and reduces visual clutter.

Chapter 2: Programmable Transfer Functions

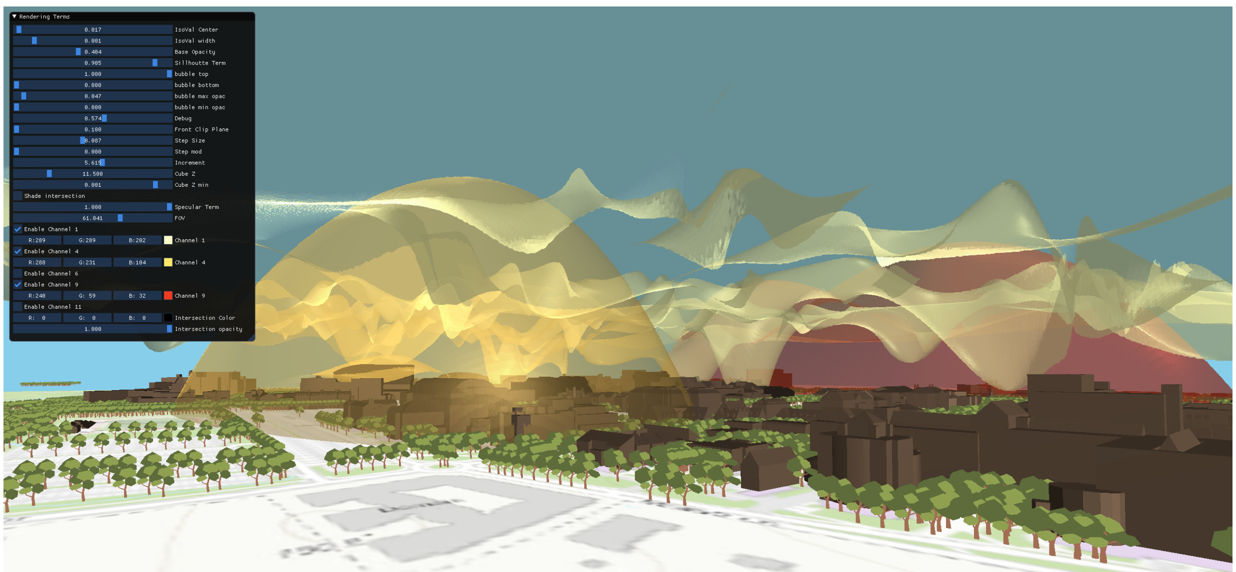


Figure 2.1: Screen capture of a rendering tool developed to utilize a Programmable Transfer Function, which offers user interaction to enhance a data analysis task. Here, five networks are shown over the University of Maryland campus to allow the user to assess signal coverage and frequency interference potential.

2.1 Introduction

Visualizing WiFi signal strength can help us engineer superior buildings, academic campuses, and smart cities by ensuring constant, secure, and reliable coverage. Data connection reliability in the information age is paramount to ensure critical data is not lost. It is thus necessary to look into and update how we analyze the coverage and effectiveness of a WiFi signal space. Current design practices analyze aggregate data and heat maps but do not allow an analyst to make decisions about the environment, such as planning router locations that maximize coverage while minimizing cost and potential interference. In this chapter, we propose a technique of WiFi visualization that utilizes direct volume rendering of sampled WiFi data to provide a better situational awareness of the complex WiFi signal space. We utilize WiFi as a stand-in for a generalized RF signal as it is easily accessible and widely understood.

Visualizing three-dimensional volumes using direct volume rendering involves the use of transfer functions. A transfer function maps the abstract voxel data to a human-interpretable color and opacity. Commonly, the volumes visualized through these means arise from scientific measurement or simulation and are thus complex, possibly involving a mixture of materials or interacting phenomena. Therefore, it is imperative to design a system to aid in creating simple and effective transfer functions. Good transfer functions reveal significant data regions, mask unimportant areas that may occlude regions of interest, and increase visual clutter. Designing transfer functions automatically, semi-automatically, and interactively is an open area of research. This chapter aims to strengthen the visual analysis process by replacing the traditional look-up-based transfer function with a higher-level Programmable Transfer Function that facilitates flexible rendering of multiple scalar fields *with* their mutual interactions to enable flexible and easy-to-understand visualizations for exploring real-world multifield volumetric data.

Motivated by our driving application of WiFi visualization, we propose a new concept called the *Programmable Transfer Function*. Programmable Transfer Functions can be implemented as shader functions that take multiple inputs, such as camera parameters, lighting parameters, and multiple scalar fields with their gradients, and generate a color and opacity. By replacing the traditional transfer function's single texture lookup, a Programmable Transfer Function enables:

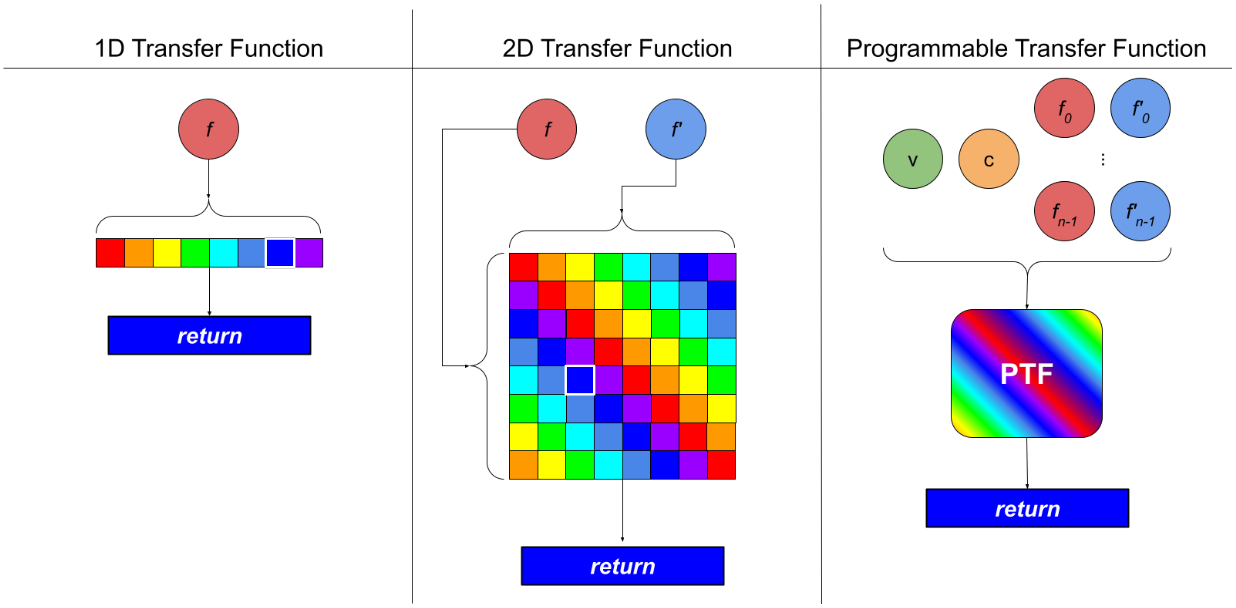


Figure 2.2: Comparison between (left) one-dimensional, (middle) two-dimensional, and (right) Programmable Transfer Function. From left to right, we observe how the transfer functions generalize to accept more input parameters. These include view-dependent parameters (v), class labels (c), the vector fields f_i and their gradients f_i', $0 \leq i \leq n-1$. Note how the Programmable Transfer Functions succinctly handle the increasing multi-dimensionality.

- superior design through user interaction by leveraging view-dependent attributes to enhance comprehensibility in real-time,
- compact representation as succinct and fast shader code and alleviate the need for large arrays of slow look-ups from memory and
- operation on multiple scalar fields and, therefore, can highlight features such as their intersection curves and regions.

Programmable Transfer Functions can assist in visually analyzing large amounts of three-dimensional volumetric data. These functions are easier to plan, analyze and visualize than traditional transfer functions and enable the user to design multifield visualizations without prohibitive memory cost. In addition to our driving application of WiFi visualization, we believe that Programmable Transfer Functions could be helpful for scientists who run large-scale simulations and healthcare professionals studying medical images.

This chapter presents how Programmable Transfer Functions are beneficial for our driving application of WiFi visualization. We first present an overview of the Programmable Transfer Function approach in Section 2.4, which includes the base rendering system's implementation in Section 2.4.1 and performance analysis in Section 2.4.5. We next present three use cases to validate the capabilities of the Programmable Transfer Function in subsections 2.4.2, 2.4.3, and 2.4.4. Finally, we discuss other possible use cases in Section 2.5 and present our Conclusions in Section 2.6.

The contents of this chapter come from our work featured in MDPI Information journal's special issue: Trends and Opportunities in Visualization and Visual Analytics in 2022 [1].

2.2 Related Works

2.2.1 WiFi Data Visualization

Current techniques for modeling WiFi signal strength over a large space are limited. An analyst could review the data in its raw form in a comma-separated values (CSV) format file, but the growing volume of data makes this intractable. Therefore, current approaches use statistical modeling and modern rendering techniques to gain an advantage. One such work by Kokkinos *et al.* [4] utilized upload and download speeds, throughput, and ping times as metrics of internet quality and collected their data via crowdsourcing. Therefore, they evaluated signal coverage over an area rather than at each sample location. Several authors have utilized two-dimensional heatmaps to represent their signal data over indoor [5,6] and outdoor [7,8] areas. Another commonly used strategy uses a network graph to represent the wireless network. Although this works quite well for visualizing network security and infrastructure, it fails to provide practical information to answer questions of region coverage or interference [9, 10]. This chapter presents our system that uses volume rendering of the WiFi signal strength and real-world geometry to help review regional signal coverage, find areas of potential co-channel interference, assess possible security vulnerabilities, and more. We chose to use direct volume rendering rather than other three-dimensional scalar field rendering techniques as it allows for manipulation of the scalar field without substantial overhead, as would be necessary for marching cubes.

2.2.2 Direct Volume Rendering

Direct Volume Rendering is a method for visualizing three-dimensional scalar fields. These fields often arise in the medical, engineering, and scientific fields due to various data acquisition technologies. In addition, many computer simulations process and output data in n-dimensional grids. Direct Volume Rendering was first introduced by Levoy [11] and improved by Drebin [12]. Volume rendering techniques have improved over the years with the introduction of various acceleration data-structures [13] and automated transfer function generation techniques [14].

Direct volume rendering is computationally intensive, and with an increase in the size of datasets, volume rendering performance quickly drops below interactive frame rates. This field has seen many improvements. These include hardware acceleration techniques [15] which use per-fragment texture fetch operations, texture rendering targets and per-fragment arithmetic to accelerate the rendering of volumetric data. Another source of frame rate improvements comes in the form of acceleration structures, such as the octree [15] to implement empty-space skipping, which steps over regions that do not contain renderable values. Multiresolution textures or mipmaps have also increased the interactivity of texture-based volume rendering [16] at the cost of memory. Roettger *et al.* [17] used an innovative technique known as *preintegration* to upsample only semantically significant areas in order to remove aliasing artifacts. The localized preintegration technique of Roettger *et al.* utilizes the second derivative to modify the step size adaptively and thus better enable step skipping. In order to avoid the cost of transfer function creation, several researchers have designed techniques that use a clustering algorithm to segment the volume into regions of interest and generate a transfer function to show these boundaries [18, 19].

2.2.3 Interaction in Volume Rendering

In the field of volume visualization interaction, Sharma *et al.* [20] use a graph-based approach to identify material boundaries and create a transfer function. Their graph represents the different materials based on how deep they are in the volume and its density. They then allow user interaction by allowing the transfer function to be modified for each individual segment. This technique allows the user to change the color and opacity of different segments. After each edit, the transfer functions must be calculated and stored as a texture.

Pflesser *et al.* [21] perform virtual cuts into volumes to simulate surgery to prepare surgeons in training and acts as a way to view internal structures of a volume. Carpendale *et al.* [22] increased user interactivity by producing three-dimensional distortion tools for data analysis by modifying the camera to make certain areas of a volume appear larger or smaller. Ip *et al.* [23] use normalized cut to create an interactive hierarchical structure for data exploration and transfer function creation. This technique allows users to automatically generate interesting data representations and interact with a fixed set of model variations.

Kniss *et al.* [24] present an elegant technique that uses multidimensional transfer functions to base the shading of the volume not just on the value at a specific location on the three-dimensional grid but also the gradient, or even further the Hessian, at that location. This chapter proposes a set of controls to interact with the multidimensional transfer function to aid in creating these transfer functions. These controls help the user explore the volume through different ways of looking at the volume, and they may edit the function by modifying the opacity for specific isovalues.

2.2.4 Non-photorealism

Another area of interest for us is non-photorealistic rendering. Our Programmable Transfer Functions modify the rendering to aid in data analysis, but these effects are not a realistic simulation of light transport and are non-photorealistic. Non-photorealistic volume rendering has innovative use cases. For example, an importance-based method proposed by Viola *et al.* [25] assigns each sample a level of sparseness during an importance compositing step. Despite their intrinsic structure and opacity, they make significant regions more visible during their final render than unimportant regions. Treavett and Chen [26] use a pen-and-ink style to render a three-dimensional or even two-dimensional representation of volume, which they compare to an architect's sketch. They showed that this sketch-like visualization helped analysts in specific tasks. Csebfalvi *et al.* [27] use non-photorealism to render the contours along a surface, thus providing a more comprehensible view of the overall structure of the volume.

2.2.5 Multifield Data

Multifield data consists of multiple values at each point. An example of one such multifield dataset would be a standard scalar field paired with the gradient at each point, or it could represent another volume entirely. Visualizing multifield data is essential to modern researchers as most scientific simulations and measurements yield multiple values at each point in three-dimensional space. One way to visualize high-dimensional data is to reproject it into three dimensions using clustering [28]. Another technique is to create a volume with a multidimensional transfer function [29]. While highly versatile, the dimensionality of the transfer function increases memory requirements. For instance, a one-dimensional transfer function over eight-bit characters would require 256 elements, whereas a four-dimensional transfer function would require 256^4 elements. This memory burden is the impetus of several performance improvements. One such technique is to use mixtures of analytical functions to represent the transfer functions, including Gaussians [30] or ellipsoids [31]. Multifield rendering can also visualize the mathematical properties of the inter-field relationships. For example, Multifield-Graphs visualize the correlation between multiple scalar fields [32].

2.3 Data

To evaluate the effectiveness of our Programmable Transfer Functions, we visualize WiFi signal strength data as it represents a varying volume over a large scale. The data used in our examples is a collection of WiFi signal data collected on the University of Maryland - College Park campus. Our dataset was collected using two handheld receivers moving across the campus over six one-hour data collection sessions. At each sample point, among the data collected was the router's Service Set Identifiers (SSIDs) and Basic Service Set Identifiers (BSSIDs) - (the names of the WLAN networks and the MAC addresses of the routers, respectively), WiFi signal strength in decibel-milliwatts (dBm), the GPS longitude and latitude of the sample, an estimate of GPS accuracy, the router's security capabilities, and the signal frequency (which contains channel information). We chose to analyze radio frequency (RF) signals as they presented a diverse set of volumes covering large areas whose propagation is affected by their environment in which analysts may be interested in seeing trends. Specifically, we are interested in studying campus coverage, signal propagation trends, and areas of potential co-channel interference, a source of signal loss due to the overlap of signal bands between channels. The methods developed and explored here will lead to new tools to analyze how various channels may interact on a large scale, such as over a university campus or a smart city.

The data used in this chapter has been interpolated after acquisition using Matlab's fit functions. We mapped the data to a two-dimensional uniform grid representing bins of latitude and longitude. We create a unique texture for each network, usually defined as a specific SSID or a specific SSID and frequency pair. The textures contain the RSSI (Received Signal Strength Indicator) value sampled at each latitude and longitude. We then used Matlab biharmonic spline fit to model the router over the whole campus. The resulting function is then output to a binary file for our rendering.

In addition to our volume data, we also use both 3D building models and a campus vegetation inventory with GPS-accurate positioning to help our users orient themselves in the virtual world. Since these building models form a one-to-one mapping to the real world, users will be able to make actionable conclusions from the information, such as where to place additional routers or which routers to move to a new channel to improve the signal landscape.

2.4 Programmable Transfer Functions

Programmable Transfer Functions significantly improve traditional transfer functions due to their malleability. In order to edit a conventional transfer function, the user must perform a memory swap, replacing the transfer function array or texture with new data. Data transfer between the CPU and GPU is a significant bottleneck for rendering pipelines. One of our contributions in this chapter is trading the transfer function lookup, which is memory intensive, for a function call at every sample location. Leveraging computation over memory fetches reduces lookup time and enables swift modifications to the transfer function through parameter modification. We can give the user far more customization and interaction options with this function call. Examples of this increased functionality are shown in Sections 2.4.2, 2.4.3, and 2.4.4.

The Programmable Transfer Function is well-suited for visualizing our complex WiFi signal space as we have many interactions across the dataset to analyze. The interactivity that the Programmable Transfer Function gives significantly benefits a signal analyst. In real-time, a user can update the isovalue to analyze stronger or weaker areas and efficiently recognize weak signal strength areas. Further, by using the specialized Programmable Transfer Functions listed below, an analyst can easily recognize the shape of the WiFi isosurfaces and find their regions of intersection. These intersection regions are noteworthy as they indicate areas of potential frequency interference or areas where WiFi packet loss may occur due to sharing space on the RF spectrum. These regions of interest in the three-dimensional space would be harder to find using aggregate data, heat maps, or even volume rendering with a traditional transfer function.

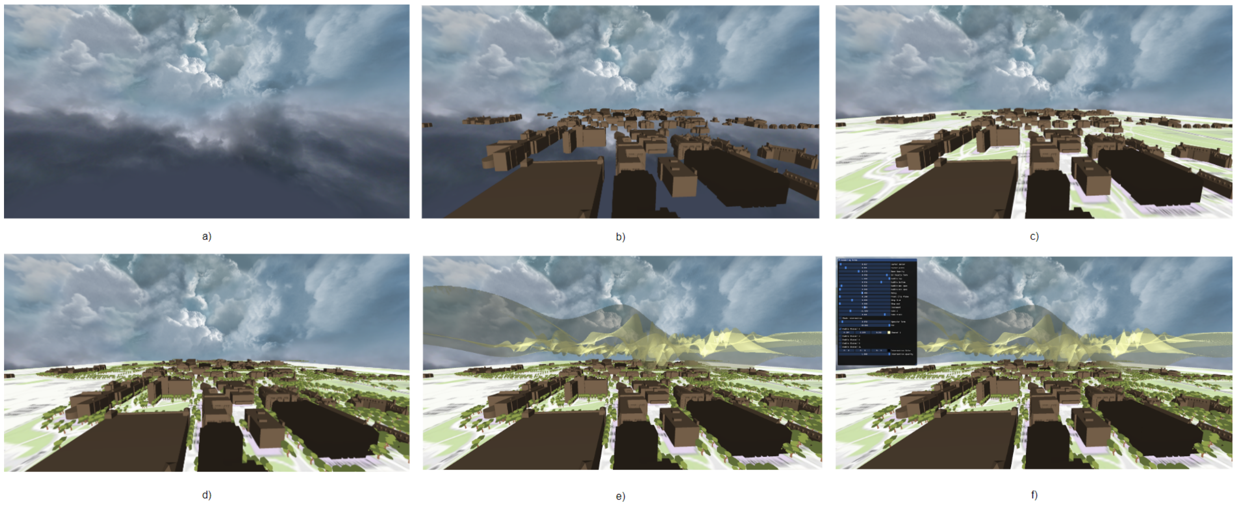


Figure 2.3: The base rendering pipeline with six steps: (a) skybox rendering, (b) model rendered with one mesh, (c) campus map modeled on a two-dimensional quad, (d) instance rendering of the vegetation directory, (e) volume rendering using ray marching, and (f) IMGUI window rendering.

2.4.1 Base Direct Volume Rendering

We render the volume rendering in this chapter in conjunction with traditional mesh rendering using rasterization. This rendering is a part of a multi-step rendering pipeline shown in Figure 2.3. First, we render a skybox as a background using the standard skybox shaders. We then render the buildings as one large mesh, the campus map as a single textured quad, and the vegetation inventory as multiple instances of a single tree object. Finally, we perform volume rendering with a GUI interface.

We decided to use volume rendering to visualize the isosurfaces of the WiFi signals. We chose to visualize isosurfaces because they are easy for users to interpret. Due to the limited overlap region, they are also compact enough to allow the simultaneous rendering of multiple networks. This enables users to determine total coverage on campus and make decisions regarding co-channel interference. Please note that the decision to use isosurfaces was made for our use case

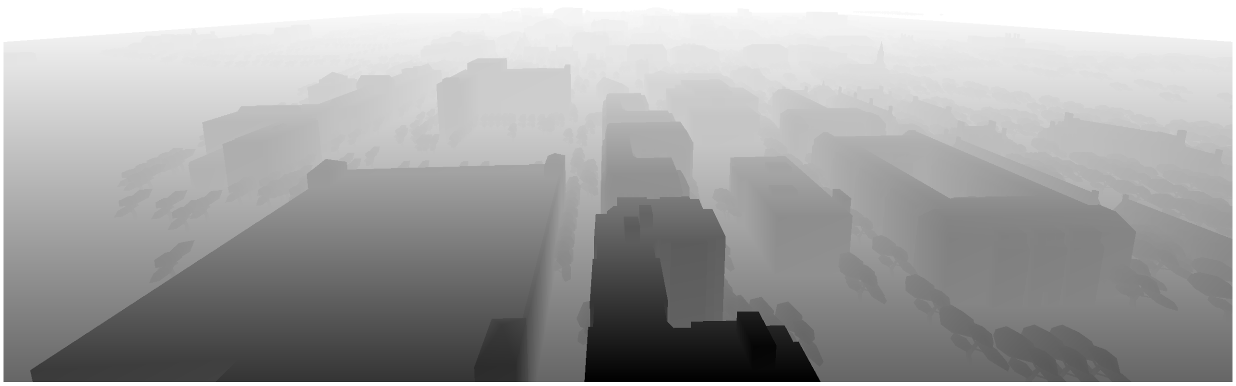


Figure 2.4: Depth mask used to terminate the volume rendering raycasts in order to enable occlusion and early ray termination. Occlusion allows the user to maintain their sense of depth, while early ray termination boosts rendering performance.

of WiFi visualization and does not represent a fundamental limitation to Programmable Transfer Functions. Programmable Transfer Functions enable us to customize isosurfaces' appearance in real-time fully. However, indirect volume rendering using traditional isosurface extraction techniques, such as Marching Cubes, limits how flexible our visual depiction can be. It would also require us to re-extract the surfaces whenever our visualization parameters are updated.

Both the mesh rendering and the volume rendering are OpenGL implementations. We use a bounding cube as an acceleration structure and render the front and back hit points of the bounding cube to a framebuffer object. Then, we use the front- and back-hit points to create the ray representing the light path for that pixel. We step through the volume sampling at each point along the ray. We store the volume data as a flat two-dimensional array. When sampling, we indexed the array based on the latitude and longitude of the sample point and subtracted a value proportional to the z component of the sample position to represent the signal strength fall-off. A traditional volume renderer would take these samples and access a transfer function texture to get a color and opacity at each point on the ray. However, we instead call a function to receive the same information. This function is the Programmable Transfer Function. We then composite all the colors with opacities along the ray to create the final rendering. In order to terminate rays early and allow buildings to occlude the volume, we use the depth buffer from our building rendering; see Figure 2.4. Early ray termination improves frame rates and provides the occlusion necessary for proper depth perception in the environment. When appropriate, we process multiple volumes by sampling each volume in turn and calculating their respective contribution at each point along the ray. Figure 2.1 shows the results of this approach. We can implement additional features using various Programmable Transfer Functions from this volume rendering.

When analyzing these renderings, the user should interpret the isosurfaces as indicating equal signal strength. The higher the isosurface in an area is, the higher the WiFi signal strength on the ground beneath it. Any area without a surface overhead is a dead zone with no measurable signal strength. Where volumes overlap, there is a potential for co-channel interference.

2.4.2 Silhouette Shading

A common challenge for many volume renderings is unclear boundaries. Specifically, a soft fall-off where a volume ends. This boundary is a concern as it makes it difficult for an analyst to discern where features are in a volume. We introduce our first use case for a Programmable Transfer Function to address this issue. We use the idea that the silhouette of an isosurface has surface normals perpendicular to the viewing angle. In contrast, the central parts of the

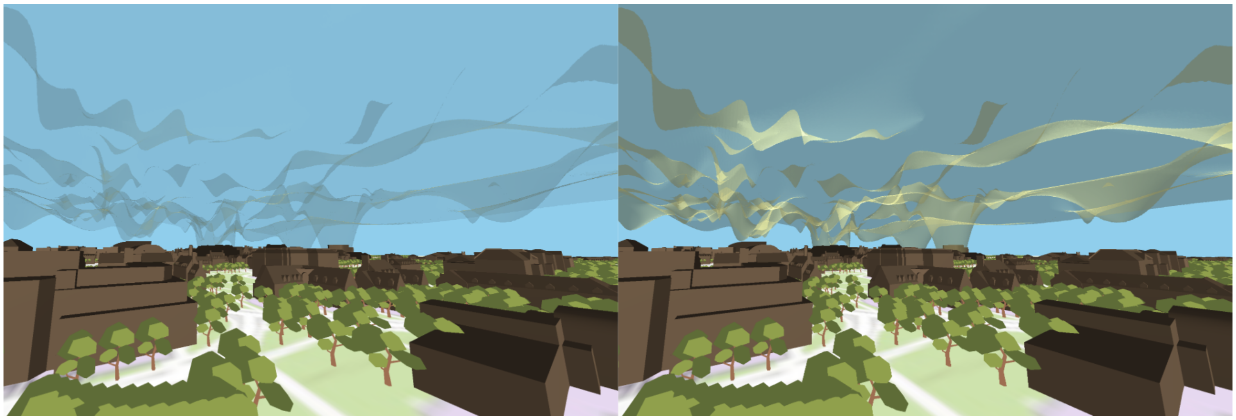


Figure 2.5: Volume rendering with (Right) and without (Left) silhouette shading. Notice how this makes it easier to distinguish different parts of the volume and accentuates details. The increased comprehensibility allows the user to make decisions regarding router coverage at a glance, as it is more clear where a volume, and thus a router's signal, ends.

isosurface has normals aligned with the view direction. We can therefore use the dot-product of the isosurface normal vector and the view direction to modify the color and opacity to accentuate the silhouettes of the isosurface. Pseudocode for this approach is in Algorithm 1. This method produces a bubble-like shading effect. This effect is described in Demir *et al.* [33]. Programmable Transfer Functions can not only implement the silhouette shading effect they also enable the user to manipulate the silhouette parameters in real time. In traditional volume rendering, silhouette shading often requires a multidimensional transfer function, increasing storage requirements. However, with a Programmable Transfer Function, no additional cost is incurred.

Algorithm 1: Silhouette Shading

for *Each Pixel* **do**

  RayDir $\leftarrow$ (FrontHit - BackHit)

  for *Each Sample Along RayDir* **do**

    $\mu \leftarrow 1-$ `Dot` (*viewDir, normal*)

    if $\mu \leq silh_{min}$ **then**

      $\mu \leftarrow silh_{min}$

    else if $\mu \geq silh_{max}$ **then**

      $\mu \leftarrow silh_{max}$

    $\alpha_{sample} \leftarrow \alpha_{base} + \mu'$

    $color_{sample} \leftarrow color_{base} + \mu * (1, 1, 1)$

    ...

  end

end

Note that the silhouette coefficient described above is tuned based on two ranges. One range $[silh_{min}, silh_{max}]$ represents how thick the silhouette augmentation band should be. Typically $silh_{max}$ is set to 1.0 as this defines the absolute edge. The other range $[\alpha_{min}, \alpha_{max}]$ defines how much the silhouette should be augmented. The results of this algorithm are shown in Figure 2.5. The silhouette shading would allow a network analyst to see the network's features more clearly, making it easier to draw conclusions from the data. For instance, in Figure 2.5 an analyst can not see much of the flat features without silhouette shading. With silhouette shading, the analyst can determine that the flat regions represent relatively high signal strength and thus are not worrisome.

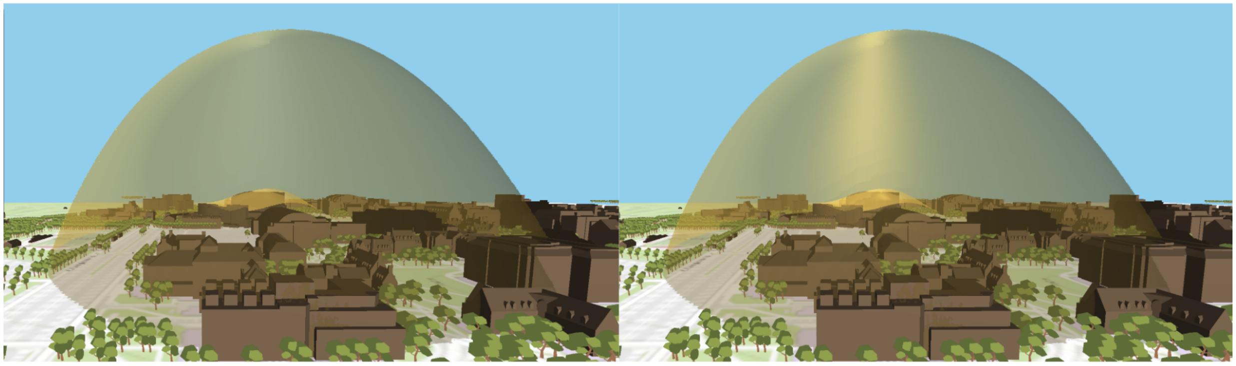


Figure 2.6: Volume rendering without (Left) and with (Right) specular highlight augmentation. This highlight allows the user to understand the curvature of the surface better and make decisions about the environment more intuitively.

2.4.3 Specular Highlight Augmentation

When illuminating thin semi-transparent surfaces, it is typical for the specular highlight to be lost. This loss is due to the lack of opacity, diluting the specular contribution. The Programmable Transfer Function can selectively boost the opacity in a region of high specular highlight to mitigate this. The added specular highlight help users orient themselves in the virtual world and effectively discern the volume's features. Specifically, the specular highlight can help elucidate the curvature of the surface. The utilization of specular highlight in volume rendering is discussed in Fernando [34]. When implementing specular highlights using a Programmable Transfer Function, users can further enhance comprehensibility by selectively increasing the opacity where a significant specular effect exists. This can emphasize surface shape by reducing the loss of visual appearance of highlights due to volumetric transparency. The specular highlight augmentation is tunable via the parameter μ_{spec} We have observed that μ_{spec} value depends on the thickness of the surface, the base opacity, and the ray-stepping size.

Algorithm 2: Specular Highlight Augmentation

for *Each Pixel* **do**

RayDir ← (FrontHit - BackHit)

for *Each Sample Along Ray* **do**

dotLightNorm ← `Dot` (*lightDir, normal*)

spec ← `Pow` (*dotLightNorm, shininess*)

$\alpha_{sample} \leftarrow \alpha_{base} + spec * \mu'_{spec}$

...

end

end

With the aid of specular highlight, the curvature of the volume becomes much more easy to understand, and analysis of the volume becomes more intuitive.

2.4.4 Multi-volume Interaction

It can be challenging to distinguish among independent volumes when rendering multifield data, notably when the volumes are semi-transparent. The Programmable Transfer Function can aid the user by highlighting their interaction as done by Jankowai and Hotz [35]. For instance, we can visualize where two volumes intersect and shade these regions a particular color as in Figure 2.7. In this example, we are coloring the intersections of the two volumes black to highlight where they meet, but more elaborate interaction visualizations are possible. It is important to note that this style of visualization is best applicable for thin isosurfaces. A different shading model may be necessary for more complicated volumes. We suggest some of these possibilities in Section 2.5. This feature is valuable, especially in the case of our data where routers commu-

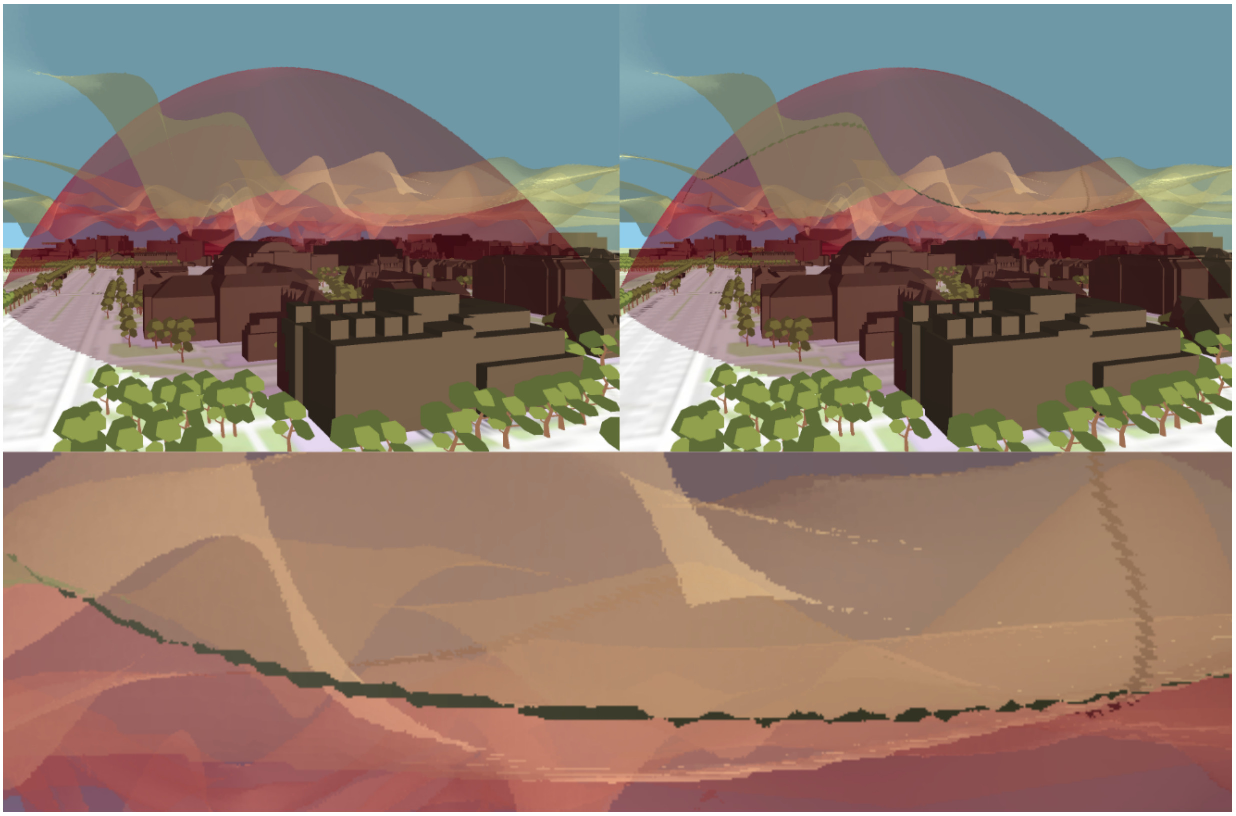


Figure 2.7: Volume rendering without (Top Left) and with (Top Right) intersection highlighting. Underneath is a zoomed-in picture of the area of intersection with the highlighting. Intersection highlighting allows a user to better determine the depth ordering of the volumes, and understand the arrangement of the surfaces. In our use case, it also serves to highlight regions where co-channel interference is likely. In regions such as the one highlighted here, a network analyst could determine that due to the large region of overlap depicted in this figure, one of these networks, should be configured to communicate on another frequency channel.

nicate on the same frequency band creating a higher probability of destructive interference and, thus, worse signal coverage and lower bandwidth in that area. In general, it also helps analysts determine the depth ordering of the volumes.

Algorithm 3: Multivolume Intersection Shading

```
for Each Pixel do
    RayDir ← (FrontHit - BackHit)
    for Each Sample Along RayDir do
        hit_volume ← false
        for each volume do
            sample ← Texture (vol_u, vol_v, vol_w)
            if IsInRange(sample) then
                if hit_volume then
                    color_sample ← (1,1,1)
                    break
                else
                    hit_volume ← true                    // Shade Normally


                    ...
                end
            end
        end
    end
end
```

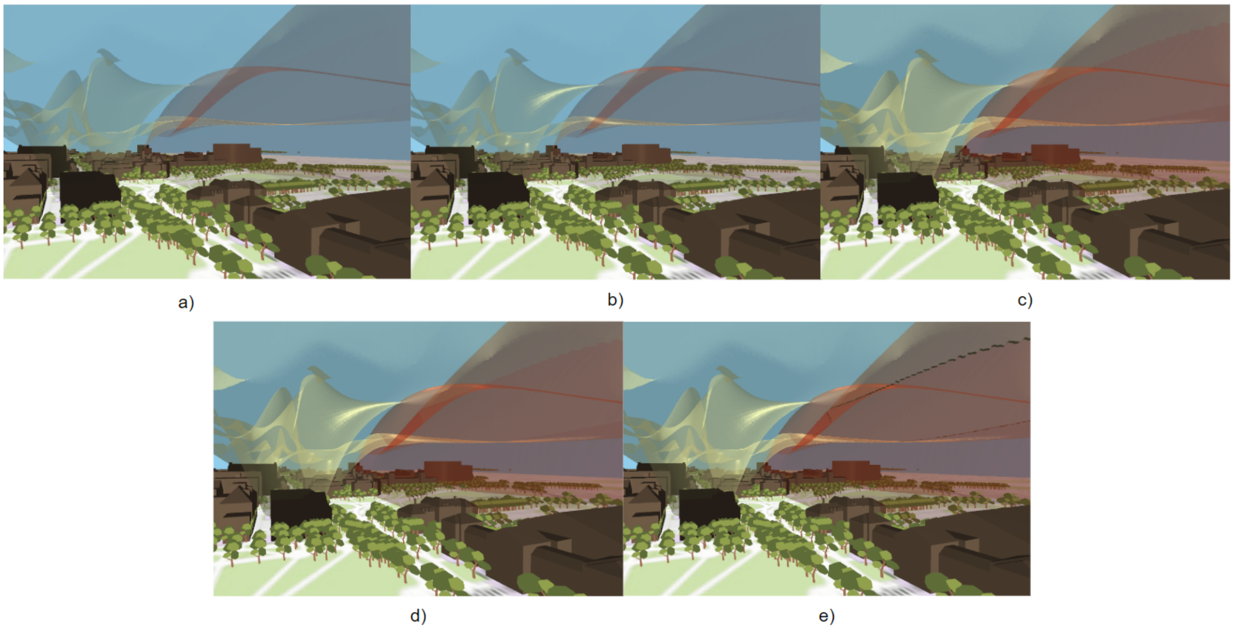


Figure 2.8: Representation of (a) the base volume rendering, (b) the rendering with specular highlight augmentation, (c) the rendering with silhouette highlighting, (d) the base rendering with both silhouette rendering and specular highlight augmentation, (e) the full suite with silhouette highlighting, specular augmentation and intersection highlighting.

2.4.5 Performance

The benefit of Programmable Transfer Functions All rendering occurs on a single NVIDIA RTX 2080 with 48 GB of RAM and an Intel Core i7 CPU, and we report frame rates in Table 2.1. Interestingly, the addition of the specialized Programmable Transfer Functions does not result in any significant frame rate loss. The lack of a performance dip may be due to how OpenGL handles branching conditionals since all functions are implemented in one shader and toggled through conditional statements. Notice, however, that even when rendering at its worst, the frame rates remain consistently well above our application's interactive threshold of 60 fps. We test each rendering configuration as shown in Figure 2.8.

	Base	Specular	Silhouette	Spec+Silh	Intersection	All 3
1 Volume (interior)	105	105	105	105	NA	NA
1 Volume (exterior)	181	181	181	181	NA	NA
5 Volumes (interior)	85	85	85	85	85	85
5 Volumes (exterior)	116	116	116	116	116	116

Table 2.1: Performance measured in frames per second (FPS). Each volume represents the signal strength of a single WiFi channel over the campus. Each test case in the single-volume case uses the same volume. For the five-volume cases, we simultaneously render all channels from this dataset in the 2.4 GHz region. We captured interior and exterior frame rates to note the difference in performance from within and without the volume.

2.4.6 Interaction

We have shown the versatility of the Programmable Transfer Functions for our driving application of WiFi visualization. We have also implemented a GUI in our renderer so that variables in our Programmable Transfer Functions can be changed dynamically, and the user can tune them to produce the rendering they need. In our example, we created an IMGUI window that can modify many rendering aspects. From this window, the user can modify the volume rendering terms such as the isovalue used to render the surface, the color used to shade each surface, and the volume step size. The user can also modify all of the Programmable Transfer Function parameters such as the silhouette term, the silhouette coefficients, the coefficient of specular highlight augmentation. In addition, the user can toggle the intersection shading on or off. We also control several acceleration techniques from this GUI.

To tune the rendering, we can manipulate variables in the GUI with a simple widget, such as a slider or a checkbox (Figure 2.9). This ease of use contrasts with how interaction traditionally works, where a new transfer function would have to be computed and stored into a texture.

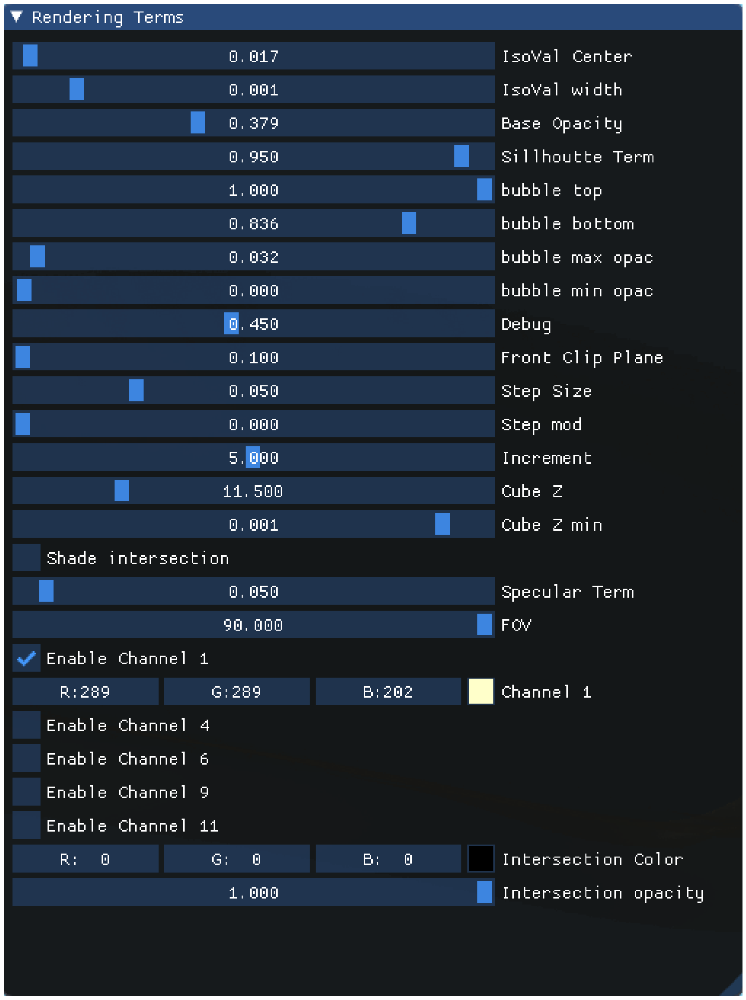


Figure 2.9: The IMGUI interface for user interaction. This GUI allows the user to take advantage of the flexibility afforded by Programmable Transfer Functions, by allowing them the ability to manipulate rendering parameters, and turn on and off various features such as intersection highlighting.

2.5 Limitations and Future Work

The application presented in this chapter is just an example of what Programmable Transfer Functions can do, but their potential extends beyond what we have presented here. As with all WiFi visualization approaches, our approach is most effective at visualizing a limited number of networks. We have visualized up to six networks at a time. This limit, however, is due to the high-frequency details and scale of WiFi data. In general, Programmable Transfer Functions are scalable. In this chapter, we have shown that we can visualize WiFi data using Programmable Transfer Functions. An exciting future work would be to examine the effectiveness of Programmable Transfer Functions for other application domains such as medical visualization or large-scale simulations.

There are many ways in which we are looking to leverage the potential of Programmable Transfer Functions. For ease of discussion, we have divided these into four categories; customized rendering, Section 2.5.1; data analytics features, Section 2.5.2; visual enhancements, Section 2.5.3; and multivolume tools, Section 2.5.4.

2.5.1 Customized Rendering

A promising direction for Programmable Transfer Functions is through fully customized rendering. In this method, an expert user would design their shader function during runtime, and the rendering would update in real-time. This way, users could interact with the code in whichever way they saw fit. For example, one could imagine a system reminiscent of Unreal Engine material shaders, where users specify inputs and visually program their desired rendering output. This system would allow arbitrary functionality and complete customization and be an exciting area for future work.

2.5.2 Data Analytics

The simplest form of data augmentation with a Programmable Transfer Function is data highlighting. For example, one could shade all values above 90% as red and everything else blue, thus highlighting the strongest signals. The Programmable Transfer Function can also leverage supplemental data. For example, in our WiFi signal data, we could shade the region based on which router had the strongest signal strength in that region, or we could mask the signal strengths in a specific region if we knew that the data there is corrupted or proprietary. In addition, several interesting geometric properties that one could highlight using Programmable Transfer Functions such as curvature and gradient.

2.5.3 Visual Enhancements

The silhouette shading and specular highlight augmentation from our implementation section fall into this category. Programmable Transfer Functions can aid in data analytics, but they also generally improve the visual component of the rendering. As an example, we could utilize additional textures to store class-based masks. For instance, a volume from a CT or MRI segmented into known tissues and organs could leverage Programmable Transfer Functions to hide organs and tissues that may be obscuring some feature of interest or highlight a region of interest. This functionality could be a valuable tool for visualizing multi-class data.

2.5.4 Multivolume Tools

Programmable Transfer Functions could be invaluable in analyzing how multiple volumes interact. This functionality is particularly useful in simulation and sensor data. The data is often in more than three dimensions, and analyzing two variables at once may help analysts identify previously unseen patterns. Algorithm 3 shows our implementation of this method. Instead of just viewing the intersection, one could consider any mathematical formulation of multiple fields, such as their difference or correlation. Further, we could design Programmable Transfer

Functions to render any feature level-set as defined by Jankowai and Hotz [35]. Programmable Transfer Functions enable the user to choose features on the fly and unconstrained by the need to compute new scalar fields for rendering. For example, a Programmable Transfer Function could visualize the intersection depth for two volumetric scalar fields for any given point. An exciting possibility is to form a conditional operation based on multiple volumetric fields. For instance, one could view the signal strength of a particular SSID and only shade it red if it is not the maximum signal over a set of SSIDs, therefore, representing the set of all SSIDs with the maximum signal strength at each location.

2.6 Conclusion

Direct volume rendering has made many strides since its origins. With the advances in graphics processing hardware, we can now mathematically calculate the transfer function on the fly rather than storing it in a predefined lookup table. This method allows an analyst to modify the transfer function on the graphics hardware and interact with the volume more efficiently. This new freedom allows for the development of new kinds of transfer functions. No longer constrained by dimensionality limits, transfer functions enable analysts to utilize other data sources. We have implemented three specific cases in which a Programmable Transfer Function can be used and suggested many others, but there are far more than we could mention here. We have shown the usefulness of the Programmable Transfer Function for WiFi signal analysis. In particular, we have used direct volume rendering to allow a user to assess the signal coverage at the University of Maryland Campus to conclude the interaction between the signals and their environment. Programmable Transfer Functions can aid in understanding and interpreting the WiFi volumes. Using the multivolume intersection transfer function, we have also allowed analysts to evaluate the potential for co-channel interference. Programmable Transfer Functions have allowed us to get both of these benefits from one rendering technique. Programmable Transfer Functions also allow for a data-specific transfer function. For example, a function could use one scalar field to mask another, or the interactions between two fields can be visualized expressly in the transfer function. Programmable Transfer Functions offer a new way of thinking for designers of multifield volume visualizations. They enable data scientists to explore their data in a flexible and efficient way while still providing all the functionality of a traditional transfer function. We believe that the use of Programmable Transfer Functions is likely to benefit several other fields beyond WiFi signal analysis.

Chapter 3: WaveRider: Immersive Visualizations of Indoor Signal Propagation

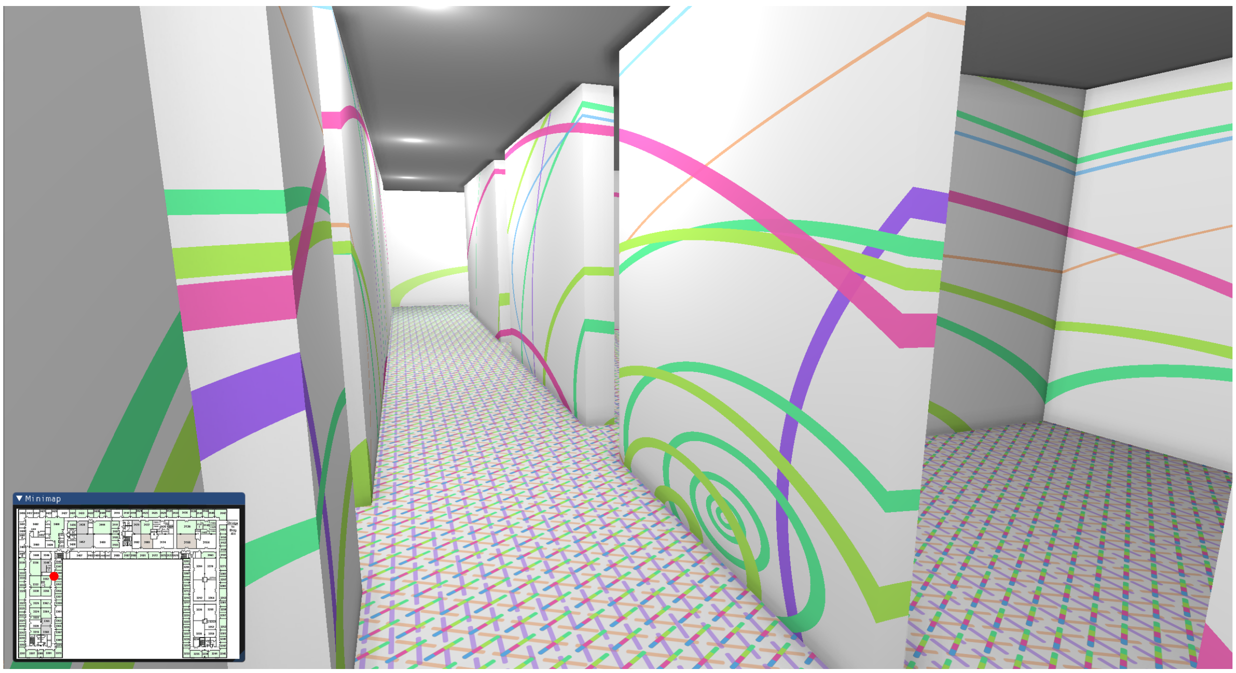


Figure 3.1: WaveRider with a multi-router contour visualization representing six routers on three different frequencies. Note the mini-map in the bottom left corner to assist the user in self-localization.

3.1 Introduction

Electromagnetic signals permeate the space around us, carrying information to and from wireless devices. These invisible communications make up an essential part of our lives, facilitating communication, data transmission, and collection. However, network health is challenging to assess due to the complex way signals interact with their environment. These signals represent various protocols such as Bluetooth, cellular, and Zigbee. In this chapter, we examine WiFi signals as they are the most complex and understandable network we have access to. Specifically, rather than looking at our outdoor networks, here we look at indoor WiFi signals. Indoors, we are presented with a completely different environment defined by walls that block a user's line of sight. In this work, we will take advantage of these natural surfaces to display our data in an efficient and structured way.

By visualizing WiFi signals, a systems analyst can monitor the network's design and health, essential components for the system's reliability and user satisfaction. In addition, WiFi signal analysis software is also critical in ensuring coverage, comparing available bandwidth at peak usage, minimizing the overlap of adjacent frequencies, and identifying available frequency bands for new routers to prevent overlap when upgrading the network. Unfortunately, as we present, the state-of-the-art WiFi signal visualization software is limited to just one or two dimensions [36, 37], significantly constraining the amount of information conveyed to the analyst. Specifically, it makes localization and coverage analysis difficult as signals propagate in all three dimensions unevenly.

Many WiFi visualizations focus on data at a single location [36, 38]. Single sample visualizations take a snapshot of the network environment at one position and display that data to the user. Since they lack environmental context, single sample visualizations delegate much of the tasks, such as source localization and coverage analysis, to the user, making the decision-making process much harder. Other WiFi visualizations look at data over the entire environment [6, 39, 40]. While these provide environmental knowledge, they are also limited by causing visual clutter and occlusion or providing a limited view of the WiFi signals.

Our application, WaveRider, allows users to view WiFi data from multiple routers over the whole environment. We designed and developed WaveRider by analyzing the current state-of-the-art signal visualizations, reviewing the inherent needs of the domain, and creating multiple visualization strategies to tackle those needs using the strengths of prior techniques. We then recruited five subject matter experts to get their detailed, informed opinions of our methods and see what changes they would recommend for our system moving forward. Through these dialogues, we have collected a large amount of feedback which has significantly enhanced WaveRider.

WaveRider provides six main contributions to state-of-the-art signal visualization:

1. WaveRider depicts more signal sources than today's state-of-the-art visualizations while representing the sources over the entire environment,

2. WaveRider introduces the visualization techniques of line-integral convolution and textons to WiFi signal space analysis for the first time,

3. WaveRider extends line integral convolution to environments with discontinuous surface normals (such as the adjacent walls of a building) via a new rendering technique.

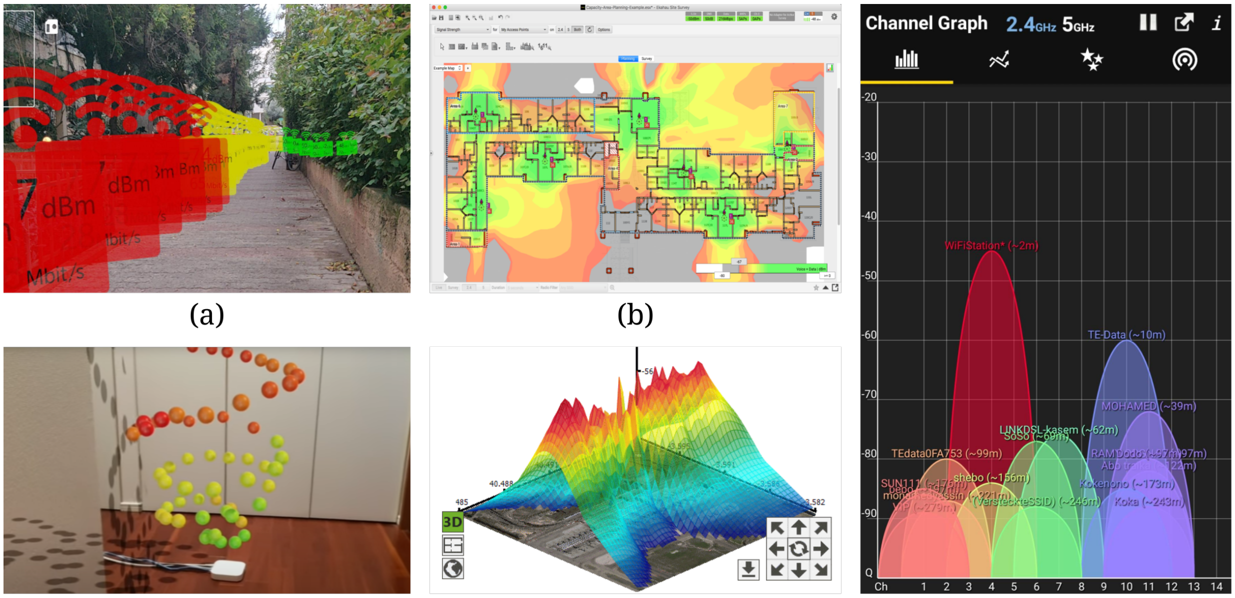
4. WaveRider enables the user to localize a router and determine its coverage and interference potential while preserving the user's ability to locate themselves in the environment.

5. WaveRider uses the natural indoor surfaces (walls and floors) for depicting the signals – this balances the conflicting goals of conveying information over the entire volume of interest with the need to reduce visual occlusion and clutter, and

6. WaveRider provides an Augmented Reality (AR) prototype, demonstrating that these design principles can work in real spaces.

This chapter presents our work published in ACM Spatial User Interaction 2022 [2].

3.2 Related Work



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

Figure 3.2: The current state of the art in WiFi visualization tools. Each image represents a commercial application for visualizing WiFi Signal Strength. *a)* WiFi AR [41] is an Android app that uses a color-coded data glyph to represent sample points. *b)* Ekahau Survey [42] is a heatmap-based visualization product for MAC OS. *c)* AR Sensor [43] is a basic glyph visualization that uses colored spheres to represent signal strength. *d)* Acrylic Wi-Fi Heatmaps [37] uses a heatmap that dual encodes height. *e)* WiFi Analyzer [36] is a simple channel graph visualization that uses simplified shapes to bin signals by frequency to make reading the spectrum easier and help visualize channel congestion. None of these visualizations can represent multiple networks while maintaining a sense of the environment.

3.2.1 Embedded Data Representations

Immersive technologies have been around for a long time. The first head-mounted display was developed in 1968 [44]; however, it was not until the Binocular Omni Orientation Monitor was developed in 1990 [45] that immersive data visualization really took off with the Virtual Wind Tunnel [46] in 1991. From there, we have seen an outpouring of work in the realm of immersive data visualizations [47] with head-mounted displays returning to the forefront of immersive visualization work in 1997 with the Virtual Data Visualizer [48]. These works utilize the ease of exploration and natural sense of positioning in space to aid the analyst [49].

A subset of immersive visualization has emerged where data is overlaid on the physical environment it represents called embedded visualization [50]. These embedded representations have provided analysts with the necessary context to make decisions about their data and how it relates to the world [51–53].

3.2.2 WiFi Visualization

WiFi visualization is still relatively young, with few contributions either in peer-reviewed papers or commercial applications such as in Figure 3.2. The current state-of-the-art methods for WiFi visualization are generally used for either visualizing coverage, as a user may need to ensure access to the network throughout the environment, or for visualizing interference potential, as a network administrator may need to guarantee reliability and speed. Unfortunately, no methods exist which can handle both at the same time.

3.2.2.1 2D Heatmaps

Heatmaps are a widespread way of encoding WiFi signal strength [6, 8, 54] as they are widely known and commonly understood, see Figures 3.2b and 3.2d. Notably, the field of Electromagnetic Compatibility (EMC) analysis, the assessment of external interference on electromagnetic radiation data, uses heatmaps for visualizing signal interference [40]. This method, unfortunately, does not scale well. If we represent each router in an environment with a heatmap with the same color map, it becomes difficult to distinguish between two adjacent routers visually.

However, mixing the colors can easily cause confusion and ambiguity if we use different color maps, especially when we have more than three colors. Even without the blending concerns, monochromatic heatmaps struggle to show slight changes in value. Hence, its ability to represent signal strength is limited, making comparisons of magnitude even more difficult. Several techniques exist to mitigate the layering issue in the two-dimensional case. One common approach is Small Multiples [55], where multiple heatmaps are shown in a grid, each representing the same area. Small Multiples has been used successfully in immersive visualizations [38,56], but they do not allow the data to be nested in the environment. Heatmaps also have the drawback of only representing the data in two dimensions, obscuring the actual distribution of the data in space. This limitation motivates the creation of 3D volumetric heatmaps rendered using volume rendering.

3.2.2.2 Volume Rendering

As mentioned above, heatmaps can visualize data in two dimensions, but the user can see all three with volume rendering. Volume Rendering involves projecting a cone of rays into a three-dimensional scalar field. We discretely sample the scalar field along each ray and use a transfer function to map the scalar values to color and opacity for each sample [39]. We then composite the color and opacity samples in a front-to-back order along each ray to calculate the final color for each pixel. This process allows us to convert volumes to rendered images for scientific visualization [57, 58]. Volume visualization can thus be viewed as a natural extension of heatmaps and overcomes the hurdles associated with heatmap's two-dimensionality and has been used in WiFi visualization for that purpose [1]. Volume rendering, however, does not overcome the layering difficulty of heatmaps and introduces new complications, including self-occlusion and computational complexity. If the visualization's data is noisy, as with most real-world data that is captured with sensors, there are often be many peaks and troughs in the isosurfaces representing constant scalar values in the volume. Such high-frequency features can add visual clutter and block the user's view of the dataset through self-occlusion. While transparency can reduce self-occlusion, it also makes it more challenging to analyze the curvature and shape of the surface. Further, since the rendering algorithm must trace at least one ray per pixel and the resolution of the visualization is dependent on the step size used in ray casting, the number of computations for each frame is significant.

3.2.2.3 Channel Graphs and Spectrograms

A critical use case for WiFi visualization is examining the channel distribution of the routers in a region. Modern WiFi transmits data over the RF spectrum in predefined bands known as channels. The technical specifications of channel allocation are defined in the IEEE 802.11 Standard [59]. Notably, for 2.4 GHz WiFi, channels are 22 MHz wide, and their centers are 5 MHz apart. It is important to note that these channels overlap. If overlapping channels transmit data simultaneously, they can distort each other. We discuss Channel Interference and its risks to network health in Section 3.3.3. However, the user must know each router's channel configuration and signal strength to assess the risk of channel interference. Channel graphs or spectrograms offer a convenient way to visualize channel interference by representing each router's signal strength and frequency channel at a specific location in a two-dimensional graph. A spectrometer can acquire the frequency and strength of a signal over time and is often used to visualize RF signals [38, 60]. Channel Graphs are simplified versions of spectrographs. An example can be seen in Figure 3.2e. The main shortcoming of this representation is its constraint to a single location. Thus, users must look at several channel graphs and infer their relationships to analyze signal coverage. With its high cognitive burden, such a process quickly becomes infeasible when exploring large regions with many routers.

3.2.2.4 Glyphs

Glyphs, or markers, have been used in EMC [40] and WiFi visualization to describe distinct samples [61–65]. Glyphs can be as simple as a sphere placed at the sample position, colored to identify the router, and sized to represent the signal strength. Several modern visualization approaches use this style of visualization, see Figures 3.2a and 3.2c. More sophisticated glyphs include a data cube visualization [66] which could depict a router's cryptographic capabilities and manufacturer. Glyphs are valuable because they can represent data with multiple encodings, such as color, shape, size, and position. Still, they are inherently discrete and thus leave gaps in regions that lack information samples. Such gaps can impose an additional cognitive burden on the user in inferring where the signal strength is strongest or where it falls off entirely. A related challenge is determining the placement of these glyphs in the scene. Since glyphs are often opaque markers, they can occlude themselves and their environment.

3.2.3 Line Integral Convolution

Line Integral Convolution (LIC) [67] is commonly used in visualizing vector fields, especially in fluid dynamics. LIC works by advecting random noise in the direction of the vector field, resulting in streaks called streamlines. These streamlines act as a visual indicator for the direction of the local vector field. The overall visualization looks similar to dye injected into a moving water source. The dye follows the direction of the water and indicates the currents. A specialized form of Line Integral Convolution called Fast Surface Line Integral Convolution [68–70] combines the principles of Fast LIC [68,70], a performance improvement on LIC that takes advantage of redundant calculations and Surface LIC [71–73], a technique for applying LIC to 3D objects. One system, [74], uses screen space to parameterize a surface for LIC. Three-dimensional vector fields have been used to visualize EMF before by [75], which aggregates signal readings by frequency and visualizes them by drawing lines connecting every point with its neighbor. This visualization nicely represents the topology of the EMF signals but introduces substantial visual clutter.

3.2.4 Textons

The base unit of precognitive texture perception is called the texton [76]. Textons are popular for use in training image classifiers, but they have also been used to model multivariate data [77]. Since a user can easily distinguish among textons, we can represent different categories of data with different textons. Oriented Slivers [78] is one such texton that has been used to visualize multiple overlapping scalar fields. Oriented Slivers are composed of rectangular capsule shapes which are rotated about their center. When layered, they produce a clear segmentation of the visual field.

3.3 Use Cases

To inform our visualization strategy, we have designed a set of WiFi use cases to aid analysts in monitoring a network. In order to develop these use cases, we formed a relationship with the expert community by reaching out to a group of professional telecommunications researchers. We interviewed these experts, performed a literature review, and embedded ourselves in a professional training course for signal analysts. Through these means, we learned the day-to-day tasks related to signal analysis that network administrators must perform and what the state-of-practice visualizations do and do not provide. We discussed their current techniques and what they would like to see in a novel visualization. We examine the use cases we developed in Sections 3.3.1– 3.3.4.

3.3.1 Localization

Localization aims to help a user identify the router's location. In real-life scenarios, localization is helpful for maintenance, service, and security reasons. For example, if a secured network router is in a public, unsupervised location, it could be subject to tampering. If a router is in an unexpected area, it could indicate that someone is attempting to impersonate that router as part of a man-in-the-middle attack. In addition, localization can help direct a repair person to a signal source that is acting up somehow or allow a network administrator to reconfigure the router to operate on a better frequency channel with lower interference. The current state-of-the-art visualizations tend to put a significant cognitive burden on the user to localize the router, or in the case of channel visualizations like Figure 3.2e, do not support localization at all.

3.3.2 Signal Coverage

The region where a user can communicate with a router is its signal coverage area. For network planning, a user must identify the undesirable dead zones with no signal coverage to guide the placement of additional routers. Another use case is that of containing a network's signal coverage. For instance, a network's coverage could go beyond the intended bounds, potentially causing a security vulnerability. While heatmaps do a fine job at visualizing coverage, they fail at attributing the coverage to specific routers due to the limitations of their color maps. Glyph visualizations rely heavily on their sample points and can easily miss coverage areas by not sampling the region. As with localization, channel visualizations do not support signal coverage analysis.

3.3.3 Interference Potential

Communication among many routers and devices in a limited space can lead to interference. Each router typically communicates on a single frequency channel, a region of the RF spectrum. In adjacent channel interference, devices communicate on nearby channels that overlap, leading to signal distortion and packet loss. In co-channel interference, routers must wait for their channel to be open before sending their packets. As a result, users may experience slowed network access when there is high network demand. Therefore, the ability to monitor a network for areas with high interference potential would be a vital asset to any network administrator. The only visualization from Figure 3.2 that aids the user in detecting Frequency interference is the channel visualization. However, it fails to encapsulate environmental knowledge.

3.3.4 Signal Awareness

Our visualization must be grounded in the real world to allow users to make actionable decisions from their analyses. For example, discovering a coverage gap can guide the placement of a new router. Therefore we must develop a visual paradigm that allows users to maintain their sense of location as they evaluate the network performance. Further, given the overwhelming amount of signal information, we also need to make sure an analyst can view the most salient network information at any given time to avoid having to sift through irrelevant information. While heatmaps and glyph visualizations can maintain some environmental knowledge, heatmaps are constrained to two dimensions, and glyphs heavily occlude their environment. Channel visualizations are far worse, containing no knowledge of the environment.

3.4 Visualizations

Our goal for WaveRider was to develop a WiFi network planning tool for a multi-floor environment. WaveRider is built to assist a network manager in creating and maintaining a wireless network that provides a stable and fast connection for their customers. Because WiFi analysis involves domain knowledge, WaverRider is built for users with this domain expertise, not for laypeople. We also designed WaveRider as an immersive Virtual Reality (VR) application to allow for intuitive movement and interaction with the environment. To facilitate real-time decision-making while in the data's environment, we designed WaveRider with embedded data representations. The use of immersive technology allows the user to conduct their data analysis in the environment and thus make actionable decisions in-place including decisions about where to focus data collection. Our goal is to enable experts to understand interactions between their physical spaces and the “signal space” while making decisions about router placement and data collection in the environment. Going into the design phase, we determined seven features that we required for our novel visualizations. We derived these requirements from our use cases and through discussion with signal experts. Our new visualizations must,

- Maintain environmental context for the user
- Show multiple routers simultaneously
- Allow the user to identify a specific router
- Delineate a clear boundary for each router’s coverage area
- Show the direction to each router
- Encode signal strength to allow for comparison between routers
- Represent the frequency a router uses to communicate

3.4.1 Data Representation

From our background literature survey on visualizing WiFi, we realized an inherent trade-off in direct three-dimensional visualizations between impeding the user’s view and the size and opacity of a visualization. If we increase our visualization’s opacity, we occlude the scene and cause overwhelming visual clutter, but if we make it more transparent, we will lose the ability to see and interpret our visualization. Therefore, we decided to use the natural features of an indoor scene - the walls, floor, and ceiling - to our advantage by depicting the information directly on them. Further, as the user can easily distinguish between a wall and the ceiling, we could display different information on each surface. Our goal was to visually segment the two spaces to facilitate greater user comprehension of more data without being overwhelmed. This strategy would allow opaque data rendering without creating occlusions.

3.4.2 Contour Lines

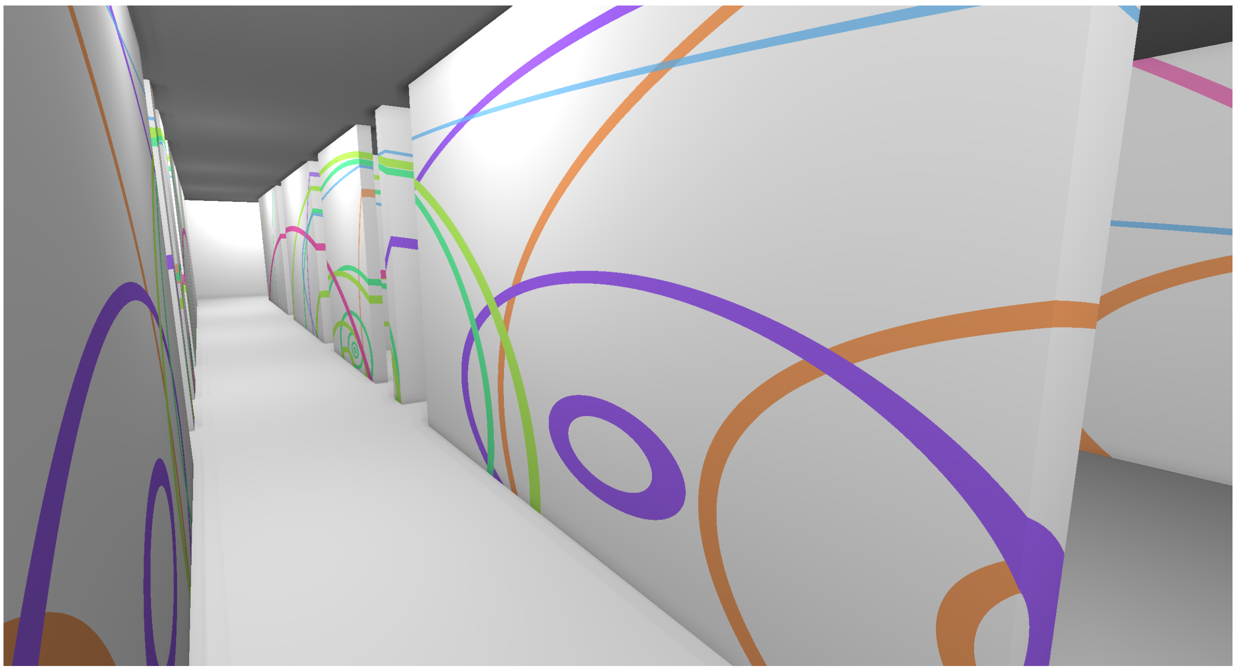


Figure 3.3: Contour line visualization with six routers. Replacing layered heatmaps with these contours makes interpreting signal strength simpler. Also, contours preserve the original colors for easier router identification. This method allows us to show more routers than with heatmaps.

Heatmaps inspired the first visual technique we developed. We assigned each router a color to represent multiple routers and visualized it with a monochromatic heatmap. Monochromaticity, however, made distinguishing signal strength more difficult. Additionally, when heatmaps stack on top of one another, their colors can blend, making it challenging to determine which routers are present. An example of the blending issue can be found in Figure 3.4. This ambiguity would not be an issue for well-separated signal sources. Still, the overlap is inevitable

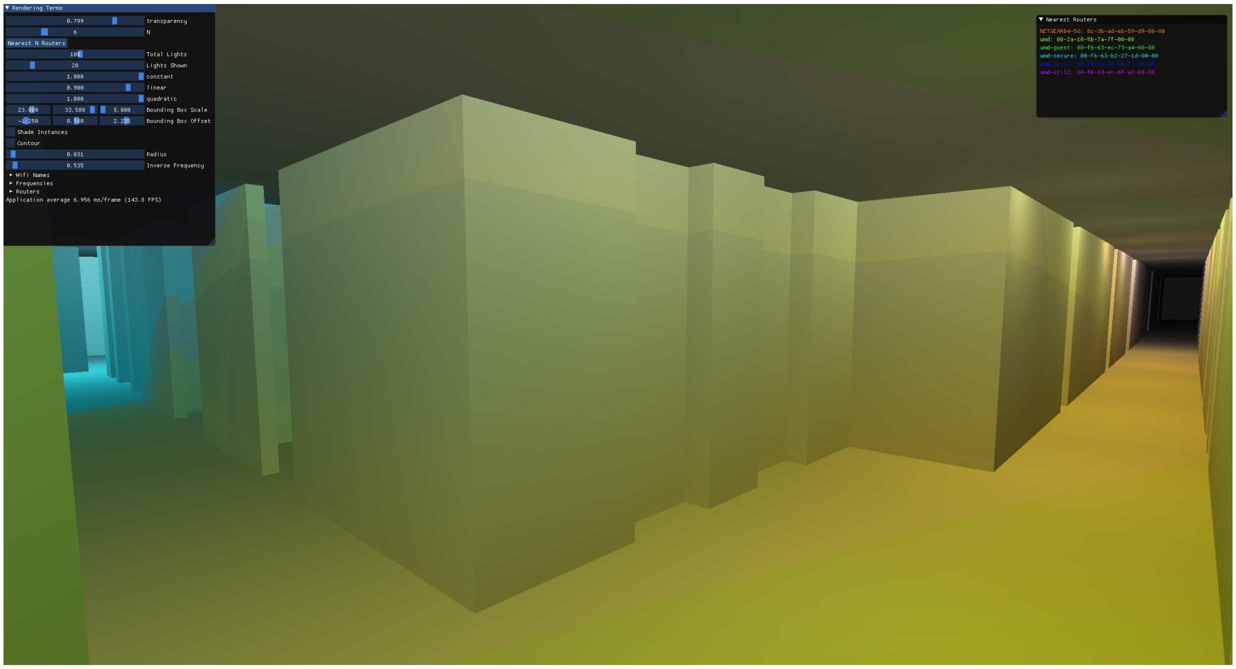


Figure 3.4: This visualization represents six routers. A unique color represents each router, with opacity indicating its signal strength. Unfortunately, the router colors blend due to high overlap, making it challenging to interpret the signal coverage and router position accurately.

since network planning aims to maximize high signal strength coverage, and overlapping signal coverage results in fewer dead zones. We realized, however, that we could tackle both of these issues at once if we only looked at isocontours or segments where the signal strength of a router is the same. We refer to isocontours as contour lines as they reveal the contours of the underlying surface - like on a topographical map. We allow the user to adjust the thickness and transparency of the contour lines to represent the signal strength better. This method can more comprehensibly depict the overlapping signal coverage from proximal routers, as seen in Figure 3.3.

The routers are each assigned a unique color in the contour line visualization. We then encode their signal strength in three ways: the thickness of the contour, the number of contours from the center, and the opacity. Additionally, the curvature of the contour line represents the direction to the signal source allowing for localization. Another benefit we found with the contour

Algorithm 4: Pseudo-code for basic contour line visualization

```
for Each Pixel do
    for Each Router do
        dist ← distanceFunc(worldPos, routerInfo)
        if dist < extent && fmod(dist, extent * freq) < thickness then
            color ← shadePixel(color, routerInfo)
        end
    end
end
```

lines is their ability to represent additional information. We initially started using dashed lines and then extended this idea to placing text inside the contours (see Figure 3.5). The contour labels were helpful but can sometimes make the visualization cluttered, so we decided to run our experiments without these labels.

Table 3.1: Visual Encoding - Contour Lines

Information	Visual Encodings
Router ID	Color
Signal Strength	Contour thickness, distance between contours, opacity
Direction to signal source	Curvature

Contour lines encode the signal source direction in their curvature. Estimating the curvature is relatively easy for our application. However, when the user is far away from the router source, the lines may appear almost straight, making it harder to identify the direction of the signal. Naturally, there is a trade-off between the number of contours, the maximum thickness of contours, and the number of routers shown at a time. For instance, thick contour lines make the change in signal strength more apparent, but they also increase the overlap with neighboring routers, causing ambiguity due to color blending. These trade-offs led us to develop the Layered LIC approach to improve our visualization by directly encoding the direction to signal source.

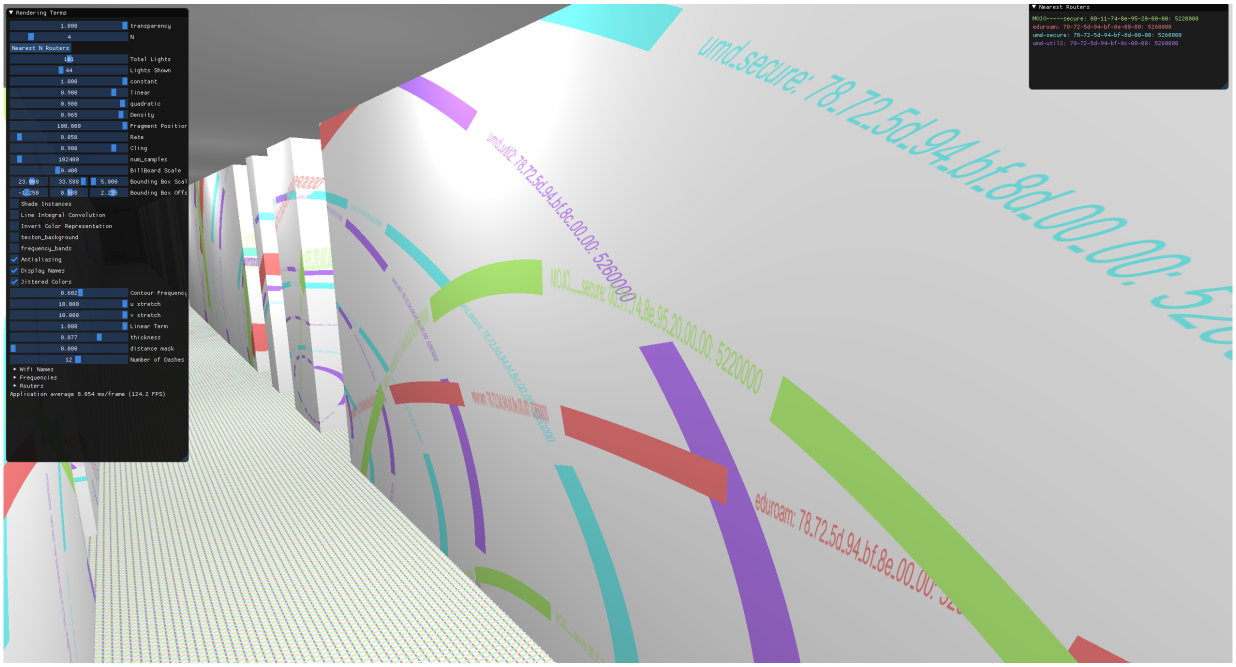


Figure 3.5: Variation of contour lines using text labels placed inside the contour lines. These lines preserve contour color and thickness while augmenting the visualization with frequency data, network names, and MAC addresses. These labels could, however, be generalized to show any information.

3.4.3 Layered LIC

To render a router's coverage, we want to visualize the field consisting of the direction to the router for all points in the router's coverage. In our original design, we encountered several challenges. The first was creating a three-dimensional noise field that was dense enough to be visually appealing and large enough to cover a building while still able to fit in memory. Another challenge was the lack of contrast from streamlines moving directly through a wall during advection. Since we are only visualizing the routers on the walls, we decided to project the vector field onto each wall. This decision allowed us to use two-dimensional noise, saving on memory. By using the projection, the image had improved contrast since the streamlines will not be pushed through the walls. A user must localize on two perpendicular walls to successfully localize the

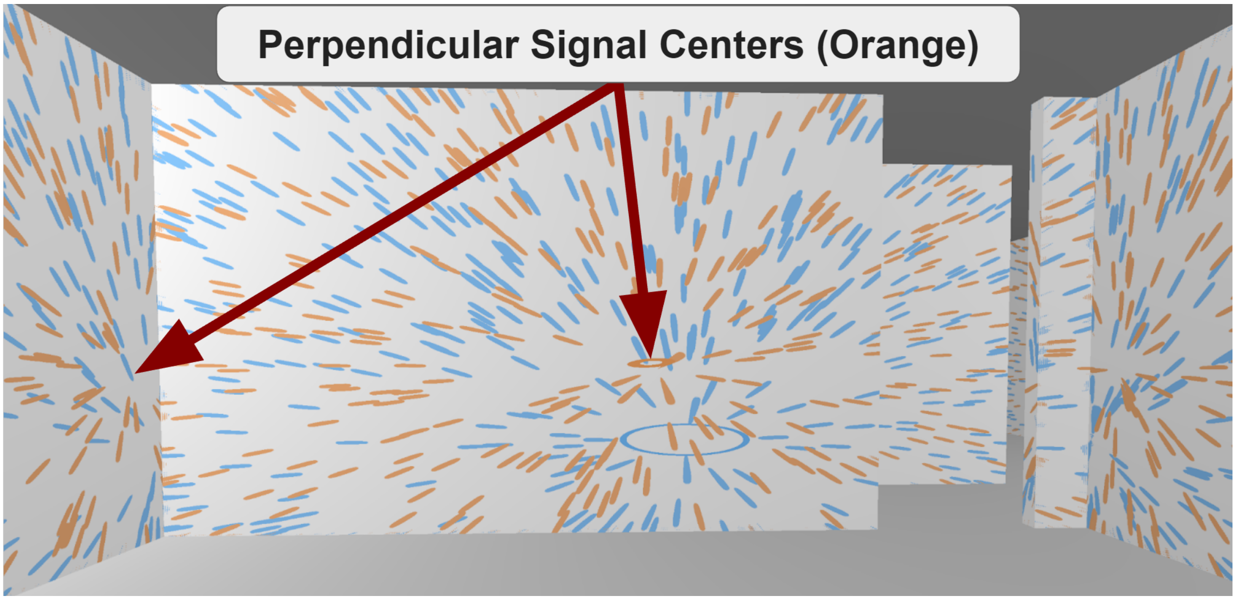


Figure 3.6: Layered LIC visualization of two routers. As seen, small ellipsoids mark the router's exact position. Users can find the router locations by following the streamlines on perpendicular walls.

router. We place a ring around the actual location of the router to inform the user that they have found the router's position.

By using two-dimensional noise placed onto each wall and running LIC on the projection of the vectors, we introduce a new challenge of continuity. The vector dramatically changes between wall segments since we project onto a different plane. When this happens, we get a sharp discontinuity in the advection of the noise. To produce visually pleasing and less distracting results, we developed a system that places the noise on the wall in (u, v)-space but performs line integration in screen space. The algorithm then detects discontinuity and modifies the integral to account for the change. This algorithm extends two different established techniques; screen-space LIC and surface LIC. We show the difference in rendering before and after this change in Figure 3.7.

When looking at multiple routers, we needed to establish a way to combine the LIC images.

Algorithm 5: Pseudo-code for basic layered LIC visualization

```
for Each Pixel do
    for Each Router do
        dist ← distanceFunc(worldPos, routerInfo)
        if dist < extent then
            counter ← 0
            while counter < numSteps do
                dir2router ← dirFunc(worldPos, routerInfo)
                mask ← getMask(counter)
                accum ← accum + mask * textureSample(worldPos - dir2router *
                  counter)
                total ← total + mask
                counter ← counter + 1
            end
            color ← shadePixel(color, routerInfo, accum/total)
        end
    end
end
```

Table 3.2: Visual Encoding - Layered LIC

Information	Visual Encodings
Router ID	Color
Signal Strength	Opacity
Direction to signal source	Streamline orientation

Our first technique was to calculate the LIC image for each router and layer them on top of each other as in Figure 3.6. This method provides the ability to visualize multiple routers at once. However, we noted that it is best to reduce the density of streamlines when showing multiple vector fields to minimize the visual clutter. This layering requires the system to calculate the entire LIC image for each router, which means a performance cost linear in the number of routers.

In the LIC technique, the direction to each router is encoded directly in the orientation of the streamlines. This direct encoding contrasts with the contours approach, in which the user must estimate the direction from the ellipsoid curvatures. As before, the color of the streamlines identifies the router, and opacity is now the sole encoding of signal strength.

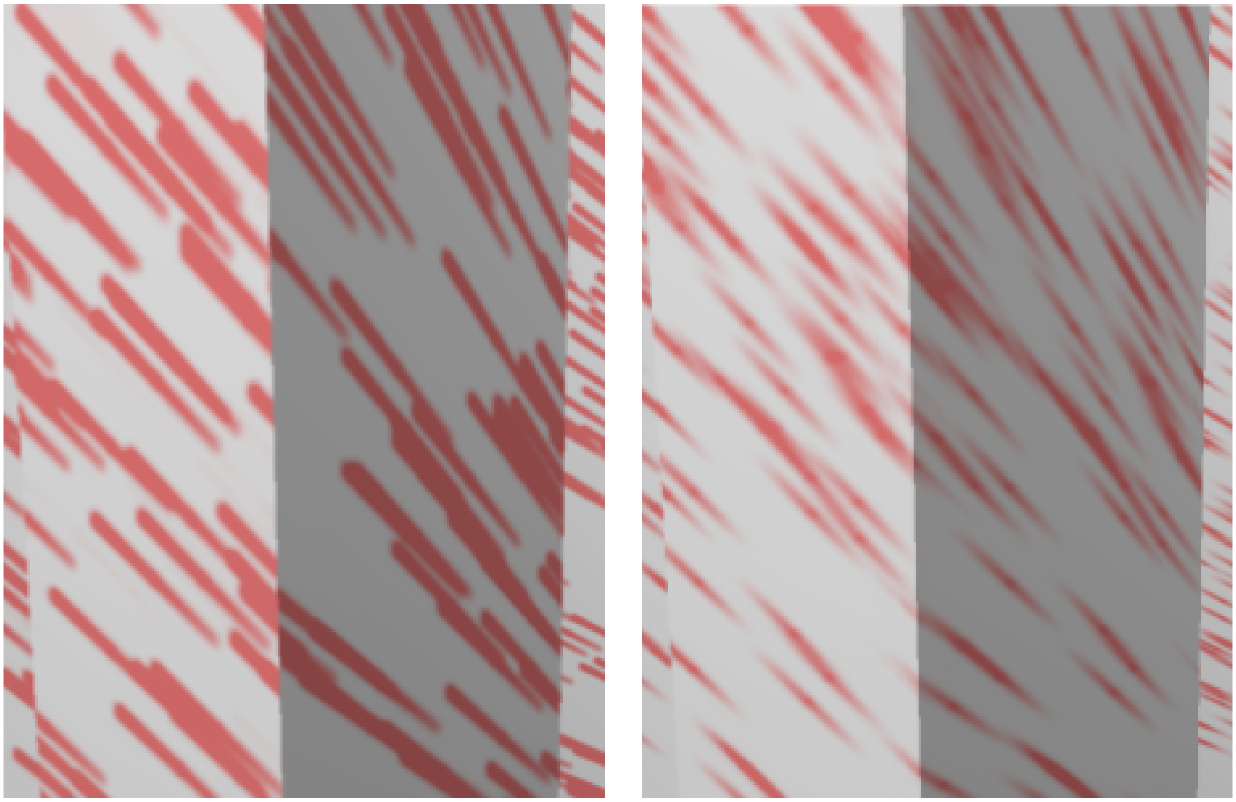


Figure 3.7: *Left:* Line Integral Convolution (LIC) using only the *uv-coordinates*. This LIC technique causes a discontinuity of streamlines where the walls meet. *Right:* LIC using *uv-coordinates* for placing the noise and adjusting the noise advection using screen space coordinates. This allows streamline continuity with a stable parameterization of the surfaces.

3.4.4 Max LIC

Table 3.3: Visual Encoding - Max LIC

Information	Visual Encodings
Strongest Router ID	Color
Strongest Signal Strength	Opacity
Direction to strongest signal source	Streamline orientation

We designed Max LIC to highlight the area where two routers have equal signal strength and reduce visual clutter. Whereas contour lines can estimate where signals are identical, Max LIC can directly show the user where signals are similar by creating a color boundary. We im-

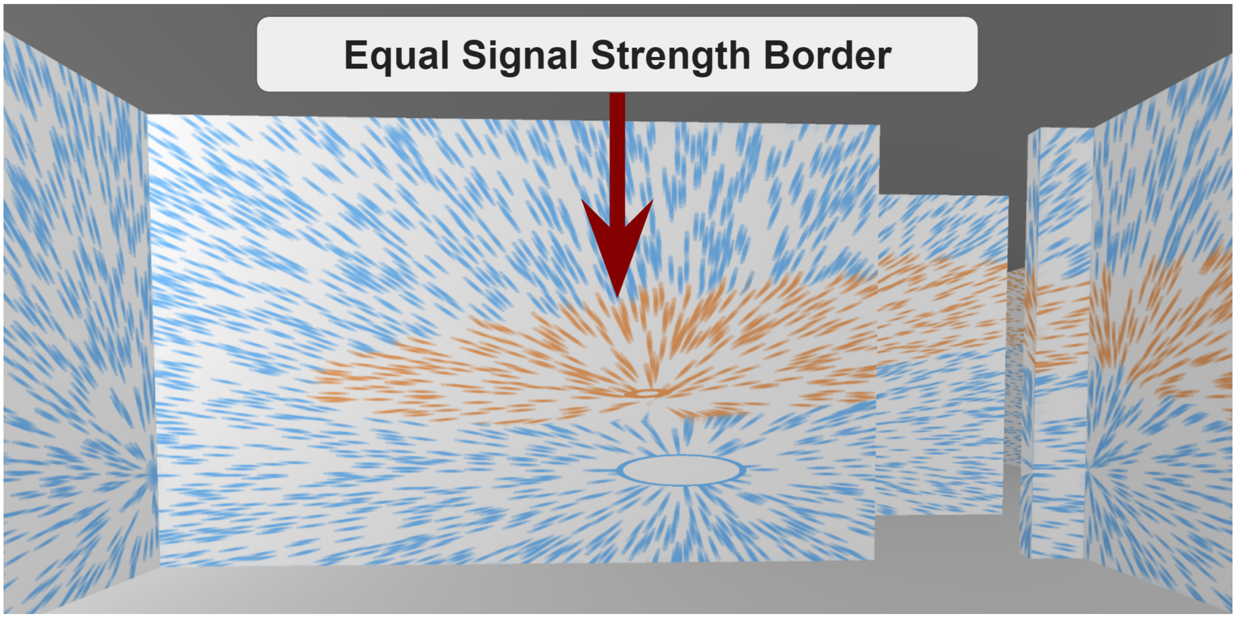


Figure 3.8: Max LIC visualization of two routers. This technique reduces visual clutter by only showing the information for the strongest router. It also shows the border where the signal strengths are equal through color changes.

plemented Max LIC using the same LIC technique from the layered LIC case 3.4.3, but here, we only view the router with the highest signal strength at any given point. A streamline's color, orientation, and opacity correspond to the router with the highest signal strength at that location. This visualization creates prominent borders where routers have the same signal strength to understand the interference potential in that area. An example of such a case appears in Figure 3.8. The most significant trade-off of this visualization is that we cannot see every router at all positions. This shortcoming can become an issue when one wants to find a specific router from far away as it may be dominated at that point and thus not visible. We can offset this issue by providing further information on the floor in our texton visualization.

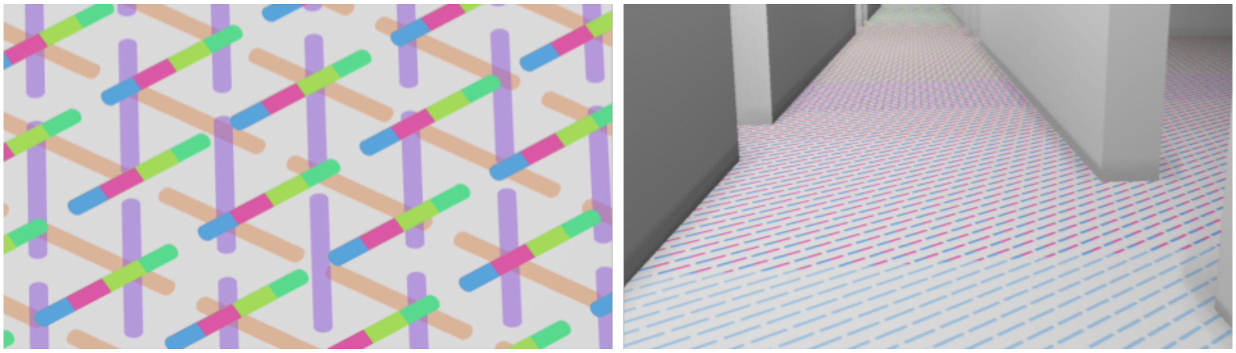


Figure 3.9: *Left:* Each router is given its own color and orientation based on its frequency channel. If the orientations are the same, they are binned into a single multi-router texton. *Right:* View of textons looking down the hallway showing strong borders where routers fall off, and frequencies change.

3.4.5 Frequency Textons

We developed a novel visual encoding to tackle the frequency interference use case. We realized that our visualizations could encode all three dimensions by using just the walls. Therefore, we could remove the visualizations from the floor and ceiling without losing any information. This decision gave us new surfaces to place data. We used textons to display our channel information on the building floor. In this way, it can be added to our other three visualizations to supplement their use. Textons are textures designed to convey features using the highly-parallel pre-attentive vision. This property of textons allows them to have a user identify boundaries between differing textons effortlessly. Specifically, we used a texton called Oriented Slivers [78]. The idea is that each router is assigned a color as before (usually the same color as the wall visualization to link the two representations) and orientation to represent the channel on which it communicates. We orient each texton to make the angle between successive frequencies as large as possible. We only need to assign an orientation to the rendered channels. Whenever routers share the same orientation, we group them into a single texton to reduce visual clutter. This visualization makes the border between frequencies quite apparent, even from afar.

We designed our textons to emphasize co-channel interference areas and reduce the search space for the adjacent channel case. However, interference is more prevalent where routers share similar signal strength. Since textons do not encode signal strength, this visualization is best used with another visualization.

Table 3.4: Visual Encoding - Textons

Information	Visual Encodings
Router ID	Color
Frequency/Channel	Orientation
Number of routers in channel	Number of Segments

3.4.6 Virtual Reality Implementation

We designed specialized fragment shaders to visualize our data on surfaces for each method. These functions are written in GLSL and run in real-time. We utilize a deferred rendering pipeline to reduce the number of per-frame computations for performance reasons. For our wall-based visualizations, we require two functions for any signal model. Both functions take the world position as input; one must return the distance to the router center, and the other must return the direction to the router center. The floor only requires the distance function. Since we are using ellipsoids, these functions come directly from the equation of an ellipsoid.

It is important to note that when visualizing three-dimensional data on surfaces, the lack of surfaces also implies a lack of information. This dearth of information could be an issue indoors when routers are far from the walls. However, a user can still see all three dimensions at any given time by looking at perpendicular walls, floor, and ceiling. Thus, as long as a signal emitter's coverage exceeds the size of a room, a user should easily localize where a signal emitter is

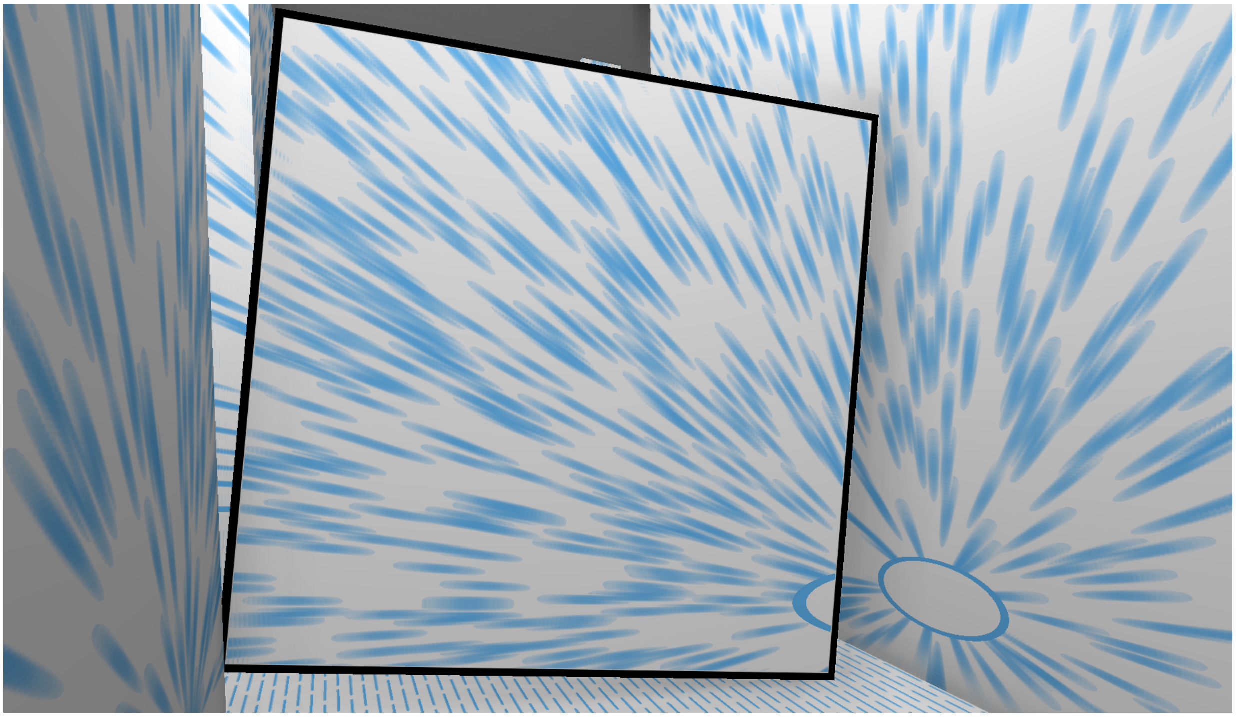


Figure 3.10: The Inspector Tool in use. A user can position the quad anywhere in the scene. This positioning freedom facilitates visualizing the WiFi signal data on the quad to assess the signal strength anywhere without requiring a fully three-dimensional data visualization.

without any additional information. Additionally, we developed a simple Inspector Tool (shown in Figure 3.10) to allow the user to place an artificial wall segment anywhere in the scene. This functionality allows the user to see the data at any point in the environment. The user can then adjust the Inspector Tool in real-time to find the exact position of the router inside a room.

3.4.7 Augmented Reality Prototype

After developing our visualizations in VR, we wanted to see what they would look like in a real-world environment. Prior work has shown success with in-situ visualizations in AR [79]. One challenge that could arise from an AR implementation would be the lack of pristine white walls as our visualization's background. The noisier real-world walls may cause finer details, such as the LIC streamlines, to be lost. With this in mind, we developed an AR experience using

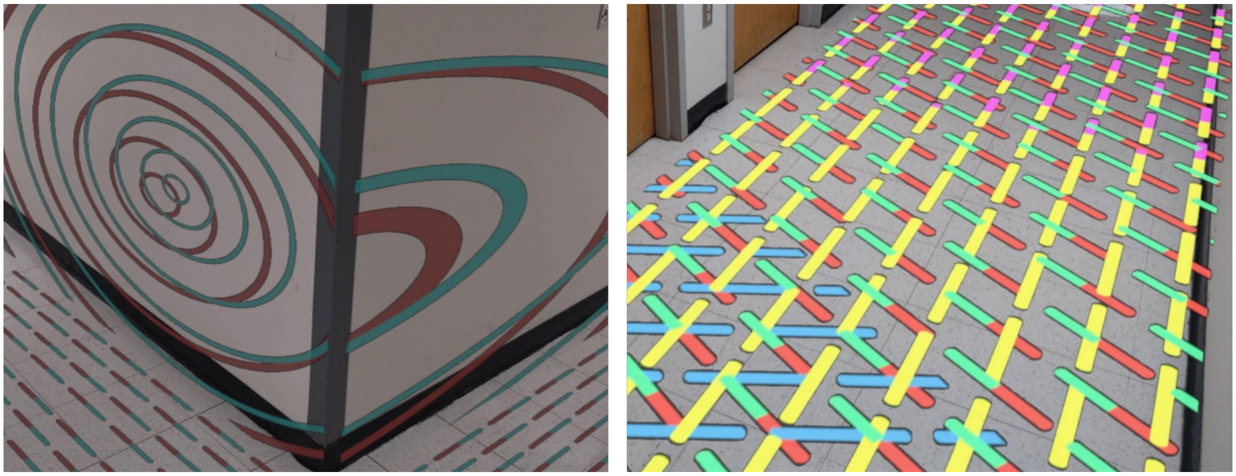


Figure 3.11: Pilot visualization of WaveRider. *Left:* Contour lines and textons in AR on the Hololens 2. The contour visualization on the wall shows that the two routers are close together and thus have similar signal strength. The textons on the floor are grouped, meaning the routers communicate on the same frequency channel. Taken together, these two features provide strong evidence of co-channel interference. *Right:* Texton visualization on the ground provides frequency information to the user. Each router has a color and an orientation based on its frequency channel. When the multiple routers communicate on the same channel, they are binned into a single texton (as with the red/green texton). This image shows strong borders where the frequency layers change. For example, the blue router drops out in the bottom left corner.

the HoloLens 2. Our prototype AR application, seen in Figure 3.11, shows how WaveRider's visualizations can facilitate the diagnosis of real-world network issues. In this prototype application, we manually registered our visualization with the real world.

In Figure 3.11 (left), the proximity of the blue and red contour lines shows that their corresponding routers are placed close together. Using the texton visualization on the floor, the user can determine that these routers communicate on the same channel since they are grouped in a single texton. Taken together, this means it is likely that this configuration will lead to co-channel interference. Figure 3.11 (right) shows how the frequency distribution of the routers varies across the corridor. Since each router is given a unique color and is oriented based on its frequency, a clear border is discernible whenever a router's coverage ends or begins. This easily comprehensible segmentation facilitates intuitive visual analysis of the WiFi network signals. We plan to extend our pilot AR efforts to augment real-world spaces and more fully assess how the situated signals information meshes with cluttered backgrounds.

3.4.8 Data Collection

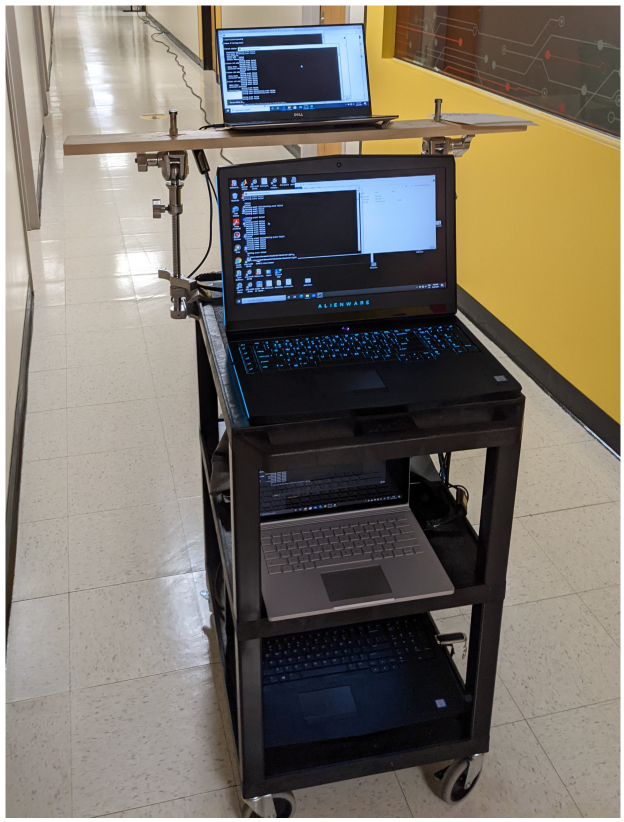


Figure 3.12: The setup used for data collection. We used four networked laptops to simultaneously sample the network along every corridor of the building.

We collected the data via multiple laptops (see Figure 3.12) using Window's Native WiFi API. We gathered our data in the hallways of a building on our university campus. The data was localized by hand since GPS is unreliable indoors and each router was fit to an ellipsoid using Principle Component Analysis (PCA) to model signal falloff. Our dataset consists of 692 samples representing 616 separate routers. Finally, we display this data on a physically accurate building model. It is worth mentioning that improvements to this data collection system can be made such as utilizing modern indoor localization techniques to remove the human-in-the-loop to speed up collection. In a future AR system, collection could also be done at the same time as analysis to improve collection density and allow analytical decisions to be made in the environment.

3.5 Expert Feedback

To evaluate WaveRider, we recruited conducted a series of expert feedback sessions. The goal for WaveRider was to create a tool appropriate for a professional, skilled audience thus, to evaluate our use cases, design requirements, and visualization designs, we recruited five experts in signal analysis and met with each of them one-on-one. Due to the skill set needed to understand both our data and tasks, a general user study would not have been a reasonable path, as teaching the necessary background and skills required to complete the analysis tasks to a general population would have been infeasible under time constraints. We modeled our evaluation after similar works [79, 80], both of which recruited a small group of field experts for their evaluations as proposed by [81]. The experts we recruited were professional telecommunications researchers and had the requisite understanding of signal propagation to complete our tasks. In our feedback evaluation, we introduced our experts to our visualizations and asked them to complete four fixed tasks inside our VR prototype application, and encouraged them to voice their opinions as they interacted with the visualizations. These tasks were designed to simulate the workflow of a network planner. We asked the users to localize a router using the contour visualization and the layered LIC approaches outlined in Section 3.4 and then asked them to identify whether there was a risk of interchannel or adjacent-channel interference when presented with multiple routers using contour lines and Max LIC both with textons on the ground. After the users completed their tasks, we gave them additional time to explore the environment and look at various router sets while asking questions. We then discussed each visualization's design and utility in a scripted interview to get their feedback. We did not collect time to completion or accuracy statistics during the tasks. This interview aimed to determine whether our visualizations could solve real-world tasks for network planners and administrators and to get their opinion on which visualizations appeared more promising. Each feedback session took about one hour and was conducted between an individual expert and the research team.

3.5.1 Immersive Design Impact

We asked the expert users whether immersion aided their understanding of the scene or their ability to process the data. Two experts reported no VR experience, one reported a casual level of experience, and two reported a significant background. The consensus among users was that being immersed in the scene is helpful because it allows for more natural movement and facilitates intuitive analysis. We also observed that the users with significant VR experience completed the localization tasks much quicker than the others and were more likely to utilize the Inspector tool. These indicators suggest that analytical VR performance may be a learned skill.

An expert pointed out that our tool could help during the phasing in of 5G (a promising direction discussed in Section 3.7) as small-cell coverage is low, their signals attenuate quickly through surfaces, and they require many access points that must run on non-overlapping frequency bands. Another expert highlighted the Inspector Tool as *"... very useful because it allows the user to customize the information they receive."* They described the tool as *"a new wall."*

3.5.2 Localization Visualizations

We first introduced the experts to Contour Lines and Layered LIC as they were both designed for localization and signal coverage. Next, we had them use the VR headset to localize specific routers using each technique. One user described the contour lines visualization as a *"good intuitive measure of signal strength"* and explained how it would be helpful for propagation analysis. Propagation analysis is the study of how signals interact in an environment. A user could use this to learn about the shielding properties of the environment and guide future router placement. The most common concern amongst the users of contour lines is that the contours straighten out and move farther apart as they get distant from the signal source. Our original design used logarithmic contour lines to deemphasize routers of low signal strength and declutter the visualization at scale. Thus, the contours cannot provide as much directional context when further away from the signal source. Closer to the router, flat regions of the contour lines can rely on neighboring areas of higher curvature to give context to the user. From our user feedback, we have decided to represent our signal strength linearly instead, giving users more closely-spaced contour lines even when away from the signal source.

All five experts preferred the LIC technique, citing difficulty following the contours when far away. One user initially described LIC as less intuitive than contours because it required them to draw information from two perpendicular walls to localize the signal source. However, after spending more time with LIC, they switched their preference. One user mentioned that *"LIC is growing on me... it is pointing me directly toward the signal source."* This quote suggests that the effective use of LIC may be a learned skill and that the direct encoding of signal source direction removes the cognitive burden of localization from the user. A concern expressed about LIC was that the streamlines are symmetrical, which means that it is difficult to identify the correct direction to the source when the lines are near parallel. When combined with masking from Max LIC, finding specific router sources becomes quite challenging.

3.5.3 Frequency Visualizations

After getting the users familiar with contour lines and layered LIC, we present the concept of frequency interference and textons. Most of our experts interacted with the textons as we had anticipated. First, they used it to augment the data of the other visualizations to determine if two routers were on the same channel. An expert went so far as to say that *"this solves a genuine problem that I encounter that is not trivial."* They then described how the current tool they use – the waterfall readout on a Spectrum Analyzer – does not allow the user to distinguish between routers on the same channel. With this amount of information, they theorized that a network planner could maximize coverage while minimizing the number of overlapping channels. However, one of the users focused on how the textons blended together when viewed from a distance producing a new color. They then stated that they could use these colors as a *"natural segmentation."* This segmentation would allow a network administrator to break the environment into regions with the same channel configurations for easier workflow.

We also showed users our Max LIC technique, and they expressed interest in it for reasons different from what we had expected. We believed that the users would be interested in the region where the two signals are equal to evaluate interference potential. However, two users did not express interest in interference potential. They explained that they were more interested in finding rogue signals – identifying emitters or receivers that are not supposed to be present –

and therefore, interference information was not as pertinent to them. Another user stated their specific interest in this feature. They noted how the color segmentation tells the user which routers a device would most likely connect to in a given area. This suggestion provided an exciting new use case that we had not anticipated. However, all our users liked how clear Max LIC made the visualization after layering.

3.6 Limitations and Future Work

Our expert feedback sessions indicate that we are on the right track with our visualizations. Each expert verified that the visualizations were compelling, and they could understand and use them within a small time-frame. However, we would like to in the future verify the usability of these visualizations more generally in a user study.

As shown in Section 3.4.7, WaveRider can be adapted to function in AR. We are interested in the technical aspects of making a general WaveRider system in AR regarding performance and tracking requirements. We believe that an interesting follow-up to this study would investigate the challenges of bridging the gap between a clean VR prototype and a real-world AR system. Such a system would come with several additional design considerations beyond the scope of this work. Namely, an AR application would have to handle real-world textures on the walls where the data is rendered. An AR application would have to adjust the visualization to have the data stand out. Further considerations include potentially hazardous environments where uniform sampling may be infeasible and real-time data collection.

Finally, we intend to revisit our system's use of the ellipsoid modeling described in Section 3.4.1. The use of ellipsoids was chosen as a reasonable baseline. However, the attenuation of

RF signals is complicated. Though more accurate modeling of these signals is outside the scope of this publication, we would like to point out how such a method would increase the usability of WaveRider even further by allowing users to identify areas of significant attenuation and identify gaps in router coverage caused by the building environment. We designed our visualizations to be model-agnostic, so future work could explore how these visualizations handle novel modeling techniques.

3.7 Conclusion

We have introduced WaveRider as a solution for visualizing indoor WiFi networks. WaveRider can help a user localize routers, detect gaps in signal coverage, and identify areas with high interference potential. Current state-of-the-art methods can not accomplish these tasks in an environmental context with as many routers as WaveRider. We recruited signal analysis experts to provide feedback to validate our mixed reality tool. These experts showed genuine interest in our work and expressed that our visualizations solve many challenges they encounter better than their current state-of-the-art techniques.

WaveRider's comprehensibility and abundance of information make it well-suited for several real-world signal analysis tasks. For instance, in the field of rogue signal detection, WaveRider could assist in the identification of anomalous signals. WaveRider's visualizations are also helpful for modeling and deploying new 5G networks. Compared with other methods, the more complex network models and frequency spaces of 5G networks should greatly benefit from WaveRider's immersive visual capabilities.

With the evolution of sophisticated network architectures and ever-increasing demand for high-speed network connections, our work provides a novel way of looking at WiFi signals, which brings signal analytics into three dimensions and enables simultaneous visualization of multiple routers in a way that has not yet been possible. We have found that expert analysts can use WaveRider to make informed decisions about channel interference and router placement. WaveRider's ability to simultaneously show signal distribution from multiple routers enables network planners to minimize interference and maximize coverage, allowing them to operate their networks at peak efficiency.

Chapter 4: Exploring Effective Immersive Approaches to Visualizing WiFi

4.1 Introduction

The world is filled with invisible communications, which have become the backbone of our society. These signals transmit essential medical data to hospitals, help secure our homes, and allow the delivery of food and other resources across the world through the internet. Many interactions with the internet are done wirelessly using Bluetooth, Cellular, or WiFi connections. These protocols utilize the radio frequency (RF) spectrum to transmit data over the air. However, RF communications are invisible to the human eye, making it difficult to diagnose issues with their operation. This paper proposes two novel ways to visualize RF networks to allow experts to analyze their systems and troubleshoot any problem they encounter. These visualization techniques are evaluated via user study against a state-of-the-art WiFi visualization technique called contour lines. We will focus on visualizing WiFi data as it is broadly used and easily collected. Future work will determine the efficacy of our visualizations on other signal models.

RF signals have been studied since 1885 when Hertz showed electronic signals could be transmitted wirelessly. Ever since, researchers have developed ways of visualizing these signals. We will go over a summary of these works in section 4.2.3. RF signal strength visualizations can be thought of in two broad categories. The first is area-based, where collected data is fit to a mathematical function that approximates the signal's propagation through an environment or is

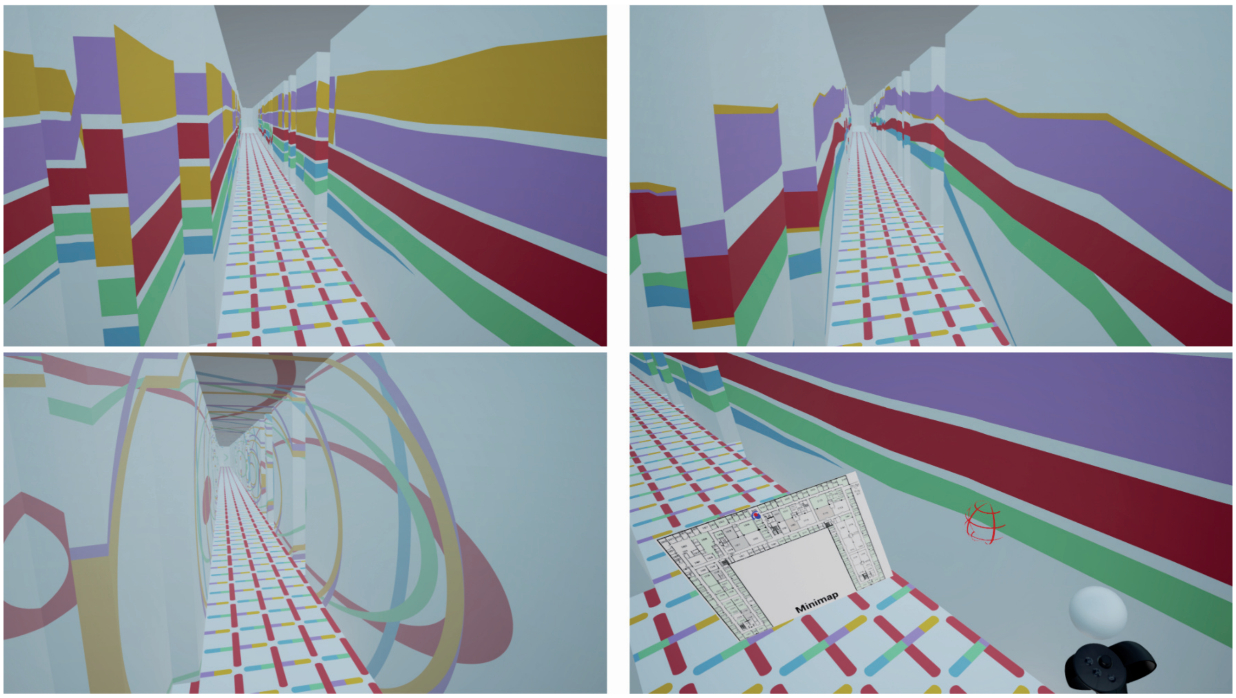


Figure 4.1: Our two novel visualization techniques and the state-of-the-art technology we use for comparison. All three visualizations are paired with Oriented and Segmented glyphs, which provide per-router frequency information. The Oriented Segmented Glyphs can be found on the floor. *Top left*: The Stacked Bars visualization. Routers are ordered on the wall based on their signal strength ranking at that location. The line's thickness represents the router's signal strength. *Top Right*: Wavelines. Both the line height and its thickness encode signal strength. *Bottom Left*: Contour lines. The baseline that we used as a comparison point for our analysis. Each router is surrounded by concentric ellipsoids whose center represents the signal source's location. *Bottom Right*: Application features available to participants in our study, including a minimap with positional and directional cues, 3D marker (red sphere on the wall), and selection preview (sphere above controller).

interpolated. This method allows the user to see the strength of a router everywhere in the environment; however, it has a few significant drawbacks. Firstly, any analysis decisions made from these visualizations depend on the model's accuracy. Since WiFi propagation is complex and data collection is inherently noisy, a model's efficacy will always be questionable. Additionally, RF signals can be obstructed by a number of materials, and thus, it is not always reliable to interpolate adjacent samples. Next, when the user first begins data collection, the collected data will start out as insufficient for the model to achieve an accurate representation. The computation and scale of the data needed for accurate modeling are significant hurdles for any area-based visualization that relies on modeling. Additionally, area-based visualizations obscure the underlying data and, by doing so, make it difficult to make decisions about the reliability of one's analysis and where to conduct further data collection.

The second visualization type is glyph-based. With glyph-based visualizations, users are shown the data without any modeling or interpolation via placing virtual objects in the environment. This allows users to see the underlying structure of the data and introduces its own complications. Firstly, since data is unmodelled, the status of a network may be unknown at a given location, as no data sample may have been collected. An absence of a data sample does not imply the absence of a signal, and thus an analyst must make coverage decisions by analyzing adjacent samples. Further, glyph-based visualizations tend to obstruct their environments by placing glyphs where the samples were gathered. These glyphs can cause occlusion with the environment and with other glyphs. It is important to note that glyph-based models can be employed to visualize modeled data by sampling the model function and visualizing the output.

Both visualization types we have analyzed here do not satisfactorily solve the problem of indoor WiFi visualization. We, therefore, designed our own visualizations to tackle this problem by using the best attributes of both area-based and glyph-based visualizations. We evaluated these novel visualizations along with a state-of-the-art visualization to analyze the benefits of each to further the field of RF analysis. Due to the complexity of RF data and its inherent connection to its environment, we decided to focus our visualization in virtual reality (VR). We chose VR over AR as the AR technology currently suffers from tracking errors over large areas and struggles in largely featureless environments such as ours and other typical office buildings. A VR application created a more stable environment for analyzing our visualizations. This allows us to gain the benefits of immersion and situated visualization and paves the way for a future augmented reality (AR) application.

In this chapter, we present our work from ISMAR 2023 [].

4.1.1 Research Questions

When we began our work, we formed the following questions to structure our research: What visualization techniques are best at localizing routers? What visualizations are best at determining if a router has coverage in an area? What visualization gives users the best understanding of the signal strengths of routers in an area? What visualization is best at allowing users to determine if two routers may interfere? What visualization do users most trust? What visualization do users find most intuitive? To address these questions, we designed a user study to evaluate our visualizations and build a better understanding of what visualizations to use for a given analysis task.

4.1.2 Contributions

Our work has made several contributions to the field of signal visualization:

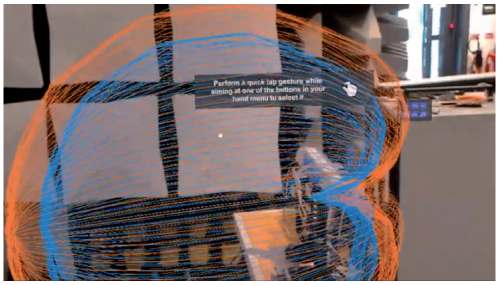
- We designed and implemented two novel visualizations discussed in section 4.4.
- We also conducted a user study to examine the impact of our novel visualizations on user confidence, task accuracy, and time-to-completion. The user study is described in section 4.5, and the results are in section 4.6.

4.2 Related Works

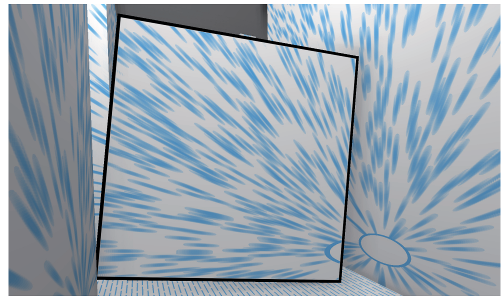
4.2.1 Situated Visualization and Immersive Analytics

Immersive analytics is the field of data analysis that centers around immersive environments. It utilizes modern display technologies such as Head Mounted Displays and Cave Automatic Virtual Environments (CAVE)s to allow users to interact intuitively with and explore data [47]. Immersive analytics has been shown to aid in data analysis tasks such as searching [86], situational awareness [87] and recall [88]. It has also been shown to support novel interaction techniques [48, 89]. Immersive analytics has developed over the years and has grown with technological development. For example, the advent of affordable augmented reality has spurred the creation of a specialized form of immersive analytics called situated visualization.

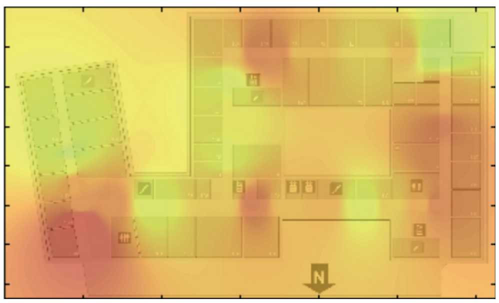
The act of visualizing data in its real-world context is often called situated visualization [50]. Prominent examples of situated visualizations include Activity Wallpaper [90] and LeafView [91]. Situated representations allow the user to gather helpful data independent of the dataset, such as environmental context. Additionally, visualizing data on top of their physical ref-



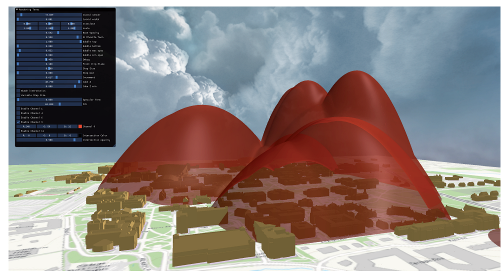
(a) 3D Field Topologies



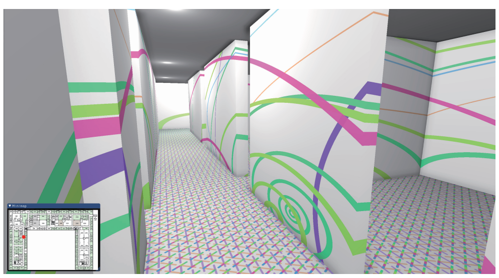
(b) Line Integral Convolution



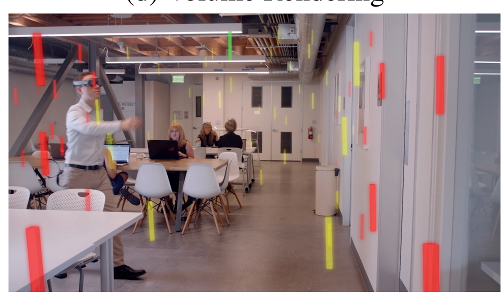
(c) Pazl



(d) Volume Rendering



(e) Contour Lines with textons



(f) SeeSignal

Figure 4.2: State of the Art WiFi visualizations. a: a visualization for Electromagnetic Compatability Testing by Guarese et al. [82] which visualizes RF signals as surfaces via mesh extraction. b: a WiFi visualization strategy by Rowden et al. [2] which uses the natural surfaces of an indoor environment to display router information using streamlines which point in the direction of the router. c: Radu et al. used a heatmap to visualize their WiFi networks in their WiFi monitoring application Pazl [83]. d: An outdoor visualization of WiFi developed by Rowden et al. [84] using volume rendering to represent signal isosurfaces. e: Another visualization technique proposed by Rowden et al. in [2] which introduced the use of contour lines for WiFi network visualization. They also used a technique called textons to visualize each router's channel configuration on the ground. f: BadVR's commercial application for indoor network modeling, which shows router signal strength via color-coded glyphs [85].

erents is known as embedded visualization [50]. For example, in RFIG lamps, Raskar *et. al* [92] use video-pass-through devices to render information about RF tags over their real-world locations and SiteLens [51] and other works [93] use AR-enabled devices to render information about underground infrastructure in its real-world location to allow construction workers to work more efficiently, in a process known as Virtual Utility Marking. Our work applies this principle to the world of Indoor WiFi signal propagation. Embedded representations enable users to visualize their networks in their environmental context and draw conclusions as they explore their spaces.

4.2.2 Overdraw and Occlusion

Visualizing point samples comes with challenges. First, when visualized, the data will be too dense or spread out. To see all of the data in a reasonable area, the visualization will need to be scaled, which would result in a dense visualization. Dense data leads to overdraw, the overlap of glyphs in a visualization, and occlusion when glyphs block the user's line of sight. There are several ways a visualization can get around these problems. One method is to summarize the data points. Several visualizations have been designed to summarize large datasets while still giving an accurate picture of their distributions. Some popular strategies are Phoenixmaps [94], Splatterplots [95], and Contour lines.

Contour lines are a proven visualization strategy that allows the viewer to understand two and three-dimensional scalar fields. Contour lines are particularly useful because they allow the user to look at multiple fields simultaneously without harmful blending effects [96]. Contour lines have been used in a variety of fields, including analysis of predictions with uncertainty [97], Computational Fluid Dynamics [98], weather prediction [99], and contours have even been used to visualize WiFi data [2]. In this work, signal analysts looked at modeled network data visualized using contour lines and attested that the visualization helped them answer real analysis questions about the networks.

4.2.3 WiFi Visualization

There are three main forms WiFi visualizations have taken recently. Firstly, there are area-based visualizations. These visualizations use techniques like volume rendering [84], heatmaps [83, 100–103] and three-dimensional field topologies [82] to describe the full shape of a router's coverage in either two or three dimensions. These visualizations excel at giving analysts a notion of coverage of a single router or network but struggle to visualize multiple routers because of color mixing or visual clutter [95]. The second visualization kind is glyph-based. In these visualizations, small objects or shapes are used to represent a sample point. Data attributes are then expressed through any number of channels like area, color, and shape [104]. These visualizations [85, 103, 105, 106] may be able to represent any number of routers but will struggle greatly with visual clutter and occlusion. The final visualization type is frequency visualizations [2, 38, 60], where the frequency channels are visualized to counteract interference. These visualizations fail to provide environmental context. Examples of these visualization types can be seen in fig. 4.2. In Rowden et al.'s work WaveRider [2], they were able to form an area-based visualization in three dimensions for an indoor environment that did not obstruct the user's view or cause color mixing. They were able to achieve such a visualization by representing the three-dimensional data on the surfaces of the indoor environment. Our novel visualizations will borrow this approach while changing the data encoding to achieve a new visual result. WaveRider included two basic visualizations of signal strength, contour lines, and line integral convolution (LIC). While the experts who evaluated their visualizations praised both visualizations, LIC was indicated as the favorite. We, however, will compare our novel visualizations to contour lines for two key reasons. Rowden et al.'s feedback suggests that their contour lines visualization is more intuitive for users. Additionally, LIC is not scalable. When visualizing more than two routers, the visualization quickly becomes cluttered, and the frame rates drop substantially. In addition to their area-based visualization, Rowden et al. also introduced a visualization technique called textons, which visualizes the routers' channel configurations on the floor to supplement their visualizations. We will adapt this visualization for our purposes as described in section 4.4.1.

4.2.4 Ranking Visualization

Visualizing ordered classes over time requires special tools. Bump charts were made to visualize bump races, a form of rowing race, in England [107]. These visualizations work by giving each item in a set a line that is equally spaced apart along the y-axis. The x-axis generally defines time either as a continuous variable or a series of discrete events. Whenever the ranking changes, the order of the lines changes. The lines move in straight lines across rankings. These visualizations were then used to visualize other data like census data [108]. Bump charts are still used today in academic circles [109]. The Sankey Diagram is a similar but more generalized visualization [110]. These visualizations show the flow of one set of values to another. We utilize a similar strategy in our Stacked Bars visualization, which we define in section 4.4.2.2

4.3 Analysis Tasks

To design visualizations for network analysis, we first needed to determine the essential tasks a user would need to perform with these visualizations. Through a thorough review of the literature listed in section 4.2.3, we identified the following essential analysis tasks: Localization, Ranking, Coverage, and Interference.

4.3.1 Localization

One must know its location to reconfigure a network or move a router. Therefore, a visualization must allow the user to identify where the router lies. A visualization can localize if a user can identify changes in signal strength. Due to attenuation, the received power will go down as you go further from the signal source. Therefore, if a user can determine the direction of the signal fall-off, they can go in the opposite direction to find the signal source. It is worth mentioning that in the real world, the attenuation of a signal is not uniform in all directions. Different objects have different attenuation parameters, and thus signal fall-off is complex.

4.3.2 Ranking

To establish the quality of a network, an admin needs to understand how strong the signals are. This will help determine if users will be able to connect as well as help in determining if frequency interference is likely. To determine if a participant understands the signal strength of the routers, we will ask them to rank the routers from highest signal strength to lowest signal strength.

4.3.3 Coverage

A network is of no value to a user if they are outside of its coverage, meaning the network has no signal strength at the user's position. Thus, an important task for our users is to determine whether a router has coverage in an area. A network engineer can use this information to determine where additional routers may be needed in the future or determine if signal leakage, when a router's coverage extends beyond where it is intended, is occurring. Signal leakage is a large security vulnerability as it exposes a network to outside threats.

4.3.4 Interference

Our final task for the users is determining the likelihood of interference. Our WiFi routers communicate on segments of the RF spectrum called frequency channels. These channels may overlap each other and cause adjacent channel interference. However, this type of interference is easily avoided by properly configuring your networks to non-overlapping channels. Cochannel interference, however, is when routers communicate on the same frequency. Since limited non-overlapping channels exist for WiFi communication (only three in 2.4GHz WiFi in the US), multiple routers will need to communicate on the same channel. However, the challenge occurs when the routers have similar signal strengths - defined as about 20dB. In this case, the routers' communications get congested and slow user communications.

4.4 Visualization Design

With the criterion for our visualizations laid out, we began the designing phase of our work. In this work, we compare our two novel visualizations to a state-of-the-art technique, contour lines. The decision for why we chose contour lines for our baseline is described in detail in section 4.4.3. Each visualization utilizes Rowden et al.'s technique [2] of presenting the signal strength visualizations on the natural surfaces of the environment, like the floor and walls, to avoid visual clutter and occlusion.

4.4.1 Segmented and Oriented Glyphs

We utilize the multi-frequency textons from Rowden et al. [2] where each frequency is assigned an orientation, and each router is represented by a pill-shaped glyph as seen on the floor of fig. 4.1 and fig. 4.5. Thus, two orientations on the ground mean the router set includes two unique frequencies. To emphasize routers that communicate on the same channel, glyphs with the same orientation are grouped together into a single segmented glyph. We use these glyphs to convey frequency information and allow users to make determinations regarding frequency interference. However, we scale the glyphs to make it easier for the users to comprehend the frequency data. Once we increase the size of these glyphs, labeling them as textons is inappropriate, as they lose their precognitive attributes. Thus, we refer to our scaled version as segmented and oriented glyphs. This visualization is used to supplement our other visualizations. For a systematic use of textons in WiFi visualizations, see [2].

4.4.2 Novel Visualizations

Our novel visualizations were designed to allow the users to see the signal strength of every router at each point without requiring that the data be modeled. It is important to note that these visualizations can also represent modeled data, but they do not require a model. We also want to limit the amount of extrapolation that is done to ensure the visualization makes no assumption for the analyst.

4.4.2.1 Wavelines

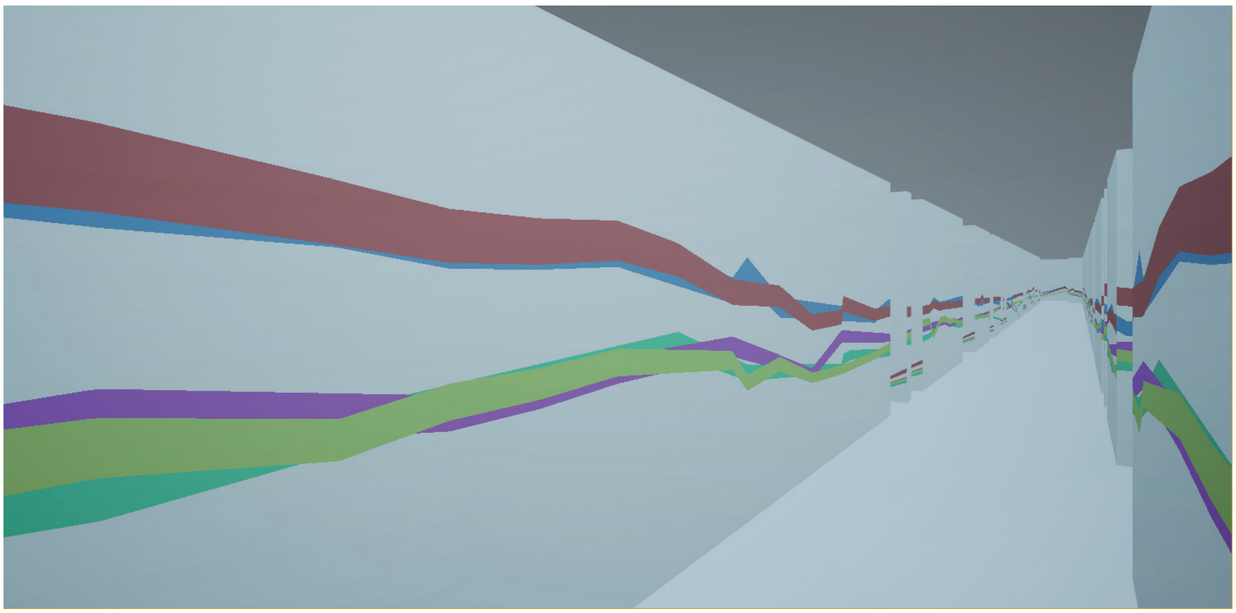


Figure 4.3: Waveline visualization showing five routers. In this visualization, the routers are represented by lines on the wall whose height and thickness encode signal strength. In this way, Wavelines can be thought of as a line chart.

The first novel visualization is inspired by traditional line graphs. Since we are visualizing electromagnetic waves, we decided to call this visualization Wavelines (See fig. 4.3). In Wavelines, unlike line graphs, the router's signal strength is dual encoded in both a line's height on the wall and its thickness. Wavelines allow users to localize by finding where the router is highest on the wall, which makes it easy to find as you look down a hallway. You can also determine the strongest signals by their order on the wall. Coverage is simply determined by whether a router's color can be found on a given wall segment. This visualization, however, suffers from occlusion when routers have similar strengths. In our study, we fixed the occlusion order throughout the environment. If the occluded router extends above the occluder, the occluded router has a higher signal strength. If the occluded router extends below the occluder, the occluder is stronger. If the occluder completely overlaps the occluded router, these routers will have the same signal strength. We explained this process to each participant in the tutorial before the user study. Since the signal strengths of the routers vary spatially, the third case (of complete occlusion) typically does not occur over large regions allowing the user to discern the relative strength. Still, it is a concern and was one of our motivations for designing the Stacked visualization, which we discuss next.

4.4.2.2 Stacked Bars

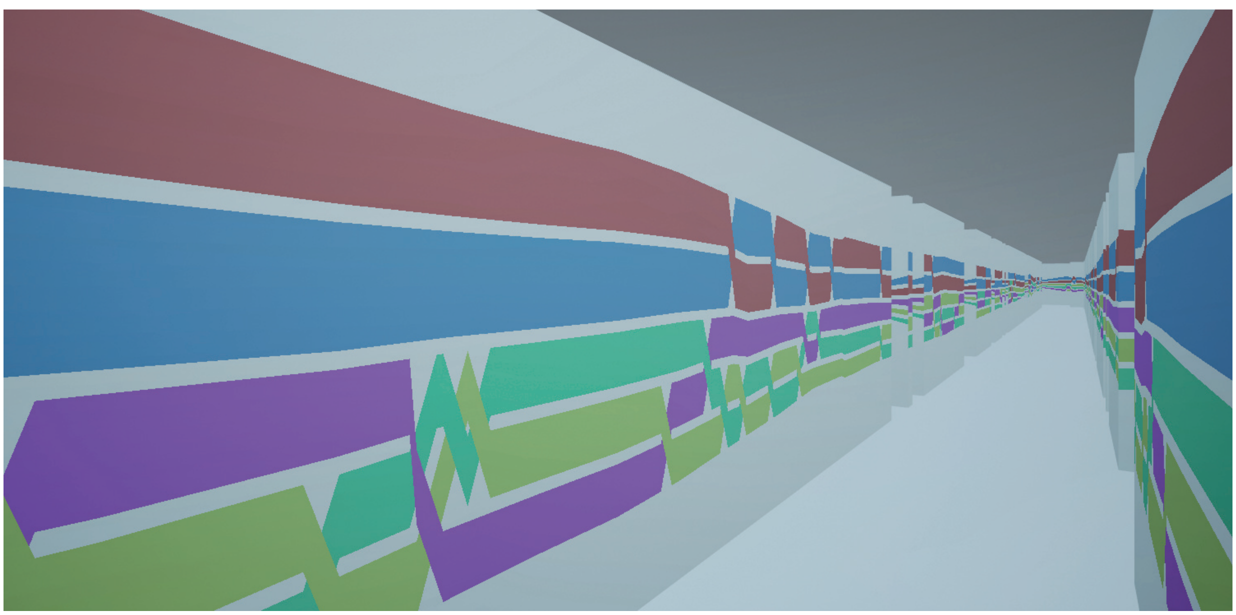


Figure 4.4: Stacked visualization showing five routers. In Stacked, routers are ordered on the wall by their signal strength, with the strongest router on the top. The line thickness encodes signal strength. When the router order changes, we draw a smooth transition line to give the user a line to follow.

Stacked Bars is a visualization based on bump charts, Sankey diagrams, and slope graphs. In this paper, we also refer to this visualization as Stacked for brevity. In Stacked, we primarily visualize the order of the routers in terms of signal strength by arranging them on the wall based on their ranking, thus removing the problem of overlap. We then encode signal strength in line thickness. Stacked can be found in fig. 4.4. This makes it much easier to determine the relative signal strengths of the routers and see where they are similar. Still, finding the router location becomes more challenging in this visualization. The height on the wall no longer indicates the maximum signal strength for a router, but rather it tells the analyst which router has the highest signal strength at that point. Once again, coverage is determined by whether the router is represented on the wall at all. The total thickness of the wall limits the maximum thickness of each line. When the number of routers increases, the change in thickness becomes more difficult to perceive. An adaption could be made to this visualization to reclaim white-space when overall power drops. Reclaiming that space, however, means line thickness cannot be easily compared across different areas of the environment. This leads to the necessity of balancing varying widths and stable user perception. For our study, we chose to evaluate the non-space-reclaiming visualization and defer a comparative evaluation of space-reclaiming methods for future work.

4.4.3 Contour Lines

Since contour lines have been used to visualize WiFi data before, we decided to use them as a comparison point for our novel visualizations. As in Rowden et al.'s work [2], we modeled the WiFi data using Principal Component Analysis and drew contour lines extending from the router's location. We also decided to use linear distance for our contour lines rather than logarithmic, as their expert feedback suggests the distance between contours becomes too great when far from the router. An example of the Contour lines found in our application can be seen in fig. 4.5.

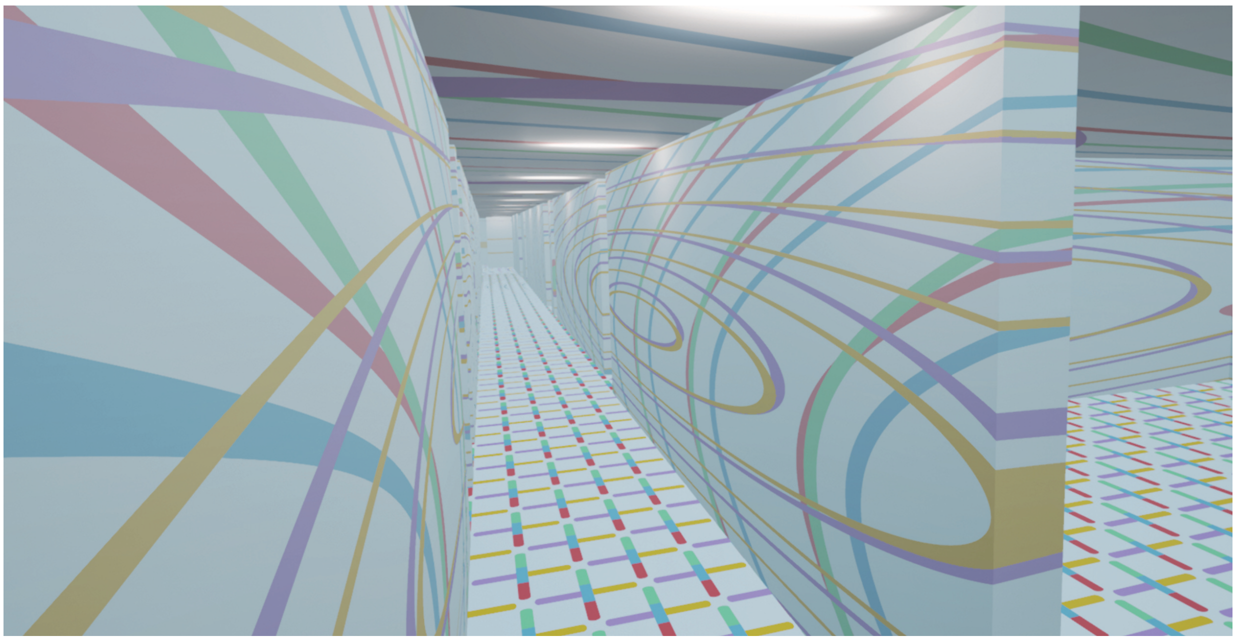


Figure 4.5: Contour line visualization showing five routers which we use to compare our novel visualizations to the state-of-the-art. Each contour represents a constant distance to a router. This visualization takes advantage of a model. We modeled the routers as ellipsoids in our case, but other models are possible. One needs to follow the contours inward to find a router's source. For ellipsoids, you need to find the center. The oriented segmented glyphs on the floor are used to represent the routers' frequencies.

Contour lines allow the user to localize by following the curvature to the center of the ellipses. When an ellipse appears on a wall, the user has localized in one dimension. A perpendicular wall is also needed to localize fully. Determining signal strength is done by contour number, which allows users to complete both the ranking and interference tasks. A router can be said to have coverage if it lies inside any ellipse.

Additionally, in their paper, Rowden et al. developed a visualization technique that used line integral convolution. While they reported a user preference for LIC, the technique does not allow users to visualize more than two routers and comes with severe performance limitations. For these reasons, we only included contour lines in our analysis.

4.5 User Study

To evaluate our visualizations, we designed a user study where we compared user confidence and collected performance metrics. This user study asked participants to complete a series of tasks in our baseline visualization contour lines and the two novel visualizations - Wavelines and Stacked. Contours were chosen as a baseline as it is a visual standard, has been used to visualize RF data in the past, and is already understood by many users. In section 4.5.1, we will outline the complete design of the user study and our metrics.

4.5.1 Study Design

For our study, we asked participants to complete a series of analysis tasks using each of our signal-strength visualizations. Segmented and Oriented glyphs accompany each visualization to provide channel information. The user study takes place in an office building environment using both hallways and large rooms. The tasks are described in section 4.3, and in section 4.5.2, we show how users submitted their responses for each analysis task. All participants completed the same tasks with the same configurations, but the order of the visualizations was randomized by Latin Square to minimize learning effects. We used the Meta Quest Pro for this experiment. Each participant underwent a guided tutorial where they completed one iteration of the task-visualization pairs with the study proctor. During that time, the participant could ask any question regarding the visualizations, and common mistakes like incomplete localization for contour lines, as described in 4.4.3, were highlighted. After the tutorial, the participants could no longer ask questions and were instructed to complete three additional iterations of the tasks. Once finished, the participants were given an exit questionnaire (see section 4.5.3), where they were asked to give their subjective feedback on the visualizations. Since our user study was long, we collected demographic information at the end to limit user fatigue. This could cause a reporting bias. Each participant spent about one hour in the headset and half an hour completing other aspects of the user study.

4.5.2 User Interaction

For localization, we allowed the user to place a marker in the environment by pointing at any wall. This marker can be seen in fig. 4.1. When the marker was placed, they could submit their answer using the confirm button in the localization menu. Accuracy was determined via two-dimensional distance to the router as the user was constrained to a single floor of the building during the study. In the ranking use case, the user is asked to rank all five routers based on their signal strength at a given position. The position is indicated by the same marking mechanism as from the localization case. To input their response, the user is presented with a menu with five numbered empty buttons. The user is asked to rank each router by assigning the router's color to the button with that ranking. For example, the button with the number one must be assigned the color of the router with the highest signal strength. One of the ranking buttons is slightly higher than the others. This elevation indicates which button is currently editable. The next button to the left will become editable when the user selects a router, changing the color of the currently editable button. We give the user two methods for inputting their responses. In the first method, the user will interact with the color palette in the task menu. This palette contains a button for each router. When a color button is clicked, that color is placed in the editable button. The second method allows the user to select the router from the wall directly. The user points at a router on the wall and presses a key to select it. We preview the hovered-over router's color on a sphere above the user's controller to make it clear which router they are pointing at. The preview mechanism can be seen in the Supplementary Material.

The next two menus are simple. For the coverage case, the user is asked to identify if a router has coverage at a marked position on the wall. We use the same marker mechanism as before. The router is identified by its color, and the user is asked to input their response via two buttons - yes or no. Similarly, for the interference case, the user is asked to decide if two routers are likely to suffer from co-channel interference. For our purposes, we instruct the user to select yes only if the two routers communicate on the same frequency and have similar signal strengths. Both routers are identified by color, and the user can respond yes or no by selecting the appropriate button.

After each task, we asked users to rate their confidence on a seven-point Likert scale using a simple menu. In section 4.6.4, we show how this data supports **H1**. Task timings were not collected while this menu was shown to the user. Users were instructed to take any needed breaks while this screen was shown.

4.5.3 Questionnaire

After the user completed all the study trials, they were asked to complete an exit survey. In the survey, we collected basic demographic information and asked them to share how much experience they have had in VR. These responses will be discussed in section 4.6.1. We then asked the users to rate each of our visualizations based on their experiences using some subjective assessment standards. For this study, we decided to have the users complete the NASA TLX (Task Load Index), the System Usability Scale (SUS), and a scaled-down version of AttrakDiff. We use these standards as they are widely used and easily understandable for our non-expert participants. We present our survey results in section 4.6.5.

4.5.4 Hypotheses

When we designed our user study, we developed a set of hypotheses to provide structure to our analyses. These hypotheses are,

H1. Users will report greater confidence in their decision-making when using our novel visualizations.

H2. Our novel visualizations will be at least as accurate as the contour-line method, which is currently used to visualize WiFi data.

H3. Users will perform their analyses with similar speeds regardless of visualization type.

4.6 Results

4.6.1 Demographics

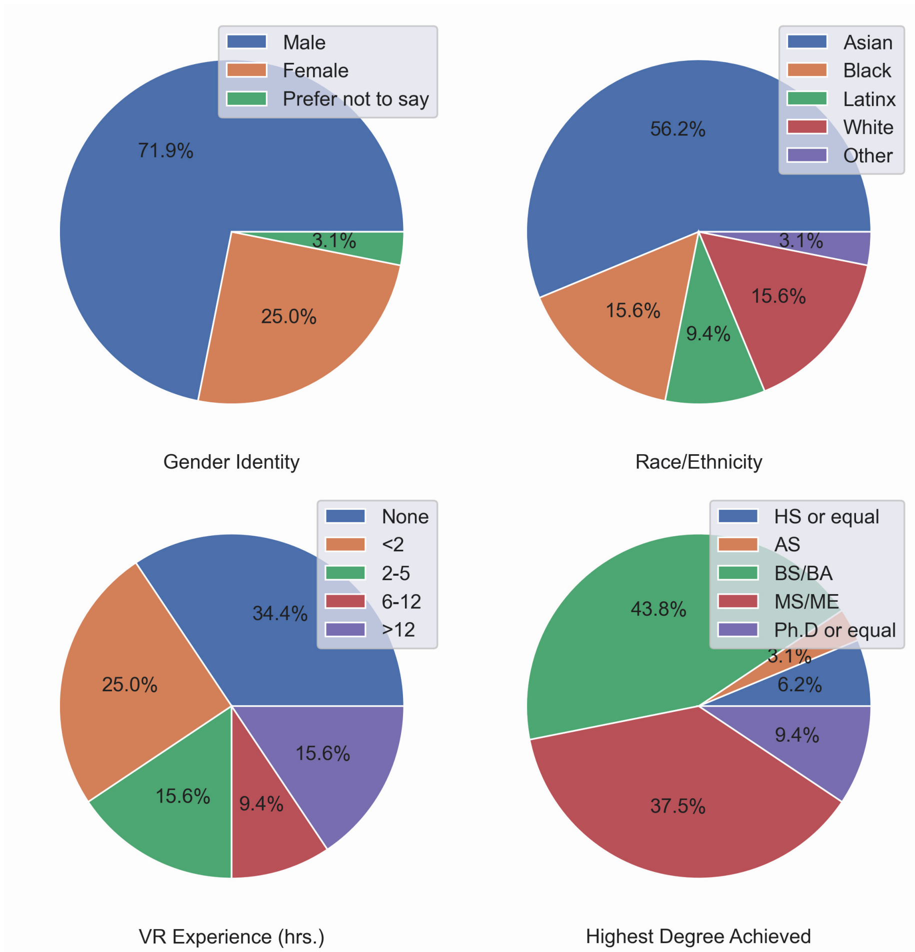


Figure 4.6: Pie charts representing the user population. *Top Left:* Gender Identity. *Top Right:* Race\Ethnicity. *Bottom Left:* VR experience in hours. *Bottom Right:* Highest degree achieved.

Our study attracted 32 participants. As seen in fig. 4.6, our paper has a good distribution of educational background and VR Experience; however, we have a skew toward men in our sampling. This is likely since we recruited largely from our university's Computer Science and Engineering departments to ensure a sufficient understanding of RF propagation.

During the study, we recorded user responses, their reported confidence, and time spent completing each task. It is important to note that we did not record the time the user spent recording their confidence. The following sections detail how we analyzed each metric and how they affect our hypotheses. Significance values for our analyses can be found in fig. 4.7.

Metrics	Visualization	Tasks			
		Localization	Ranking	Coverage	Interference
Timings	Contour lines	155.461	88.300	18.389	19.460
	Stacked Bars	116.547	**33.808**	**16.144**	**17.697**
	Wavelines	**93.442**	52.718	17.642	18.247
Accuracy/Distance	Contour lines	2.085	.496	**.979**	.712
	Stacked Bars	1.930	**.981**	**.979**	**.854**
	Wavelines	**1.078**	.812	.958	.802
Confidence	Contour lines	4.167	3.844	5.531	5.260
	Stacked Bars	4.573	**5.635**	**5.823**	**5.531**
	Wavelines	**4.875**	5.042	5.750	5.396

Table 4.1: Average values for each visualization for each metric. Times are reported in seconds, and distances are reported in meters. Bold values represent the best score achieved.

4.6.2 Accuracy

First, we address user accuracy. Each task had its own accuracy definition, thus the analysis must be specialized for each case. The simplest accuracy measurement comes from localization, where the accuracy is the two-dimensional distance to the router. Since the user can only answer with a point on the wall, we moved the ellipsoids so they would share the same router location as the novel visualizations. In this way, all visualizations will have the same answer, and user responses will contain no bias from the distance of the answer to the nearest wall. The average distance to the nearest router can be seen in fig. 4.8 and section 4.6.1. Users are more proficient at localizing while using Wavelines than the other visualizations (p ¡ 0.01), and this difference is

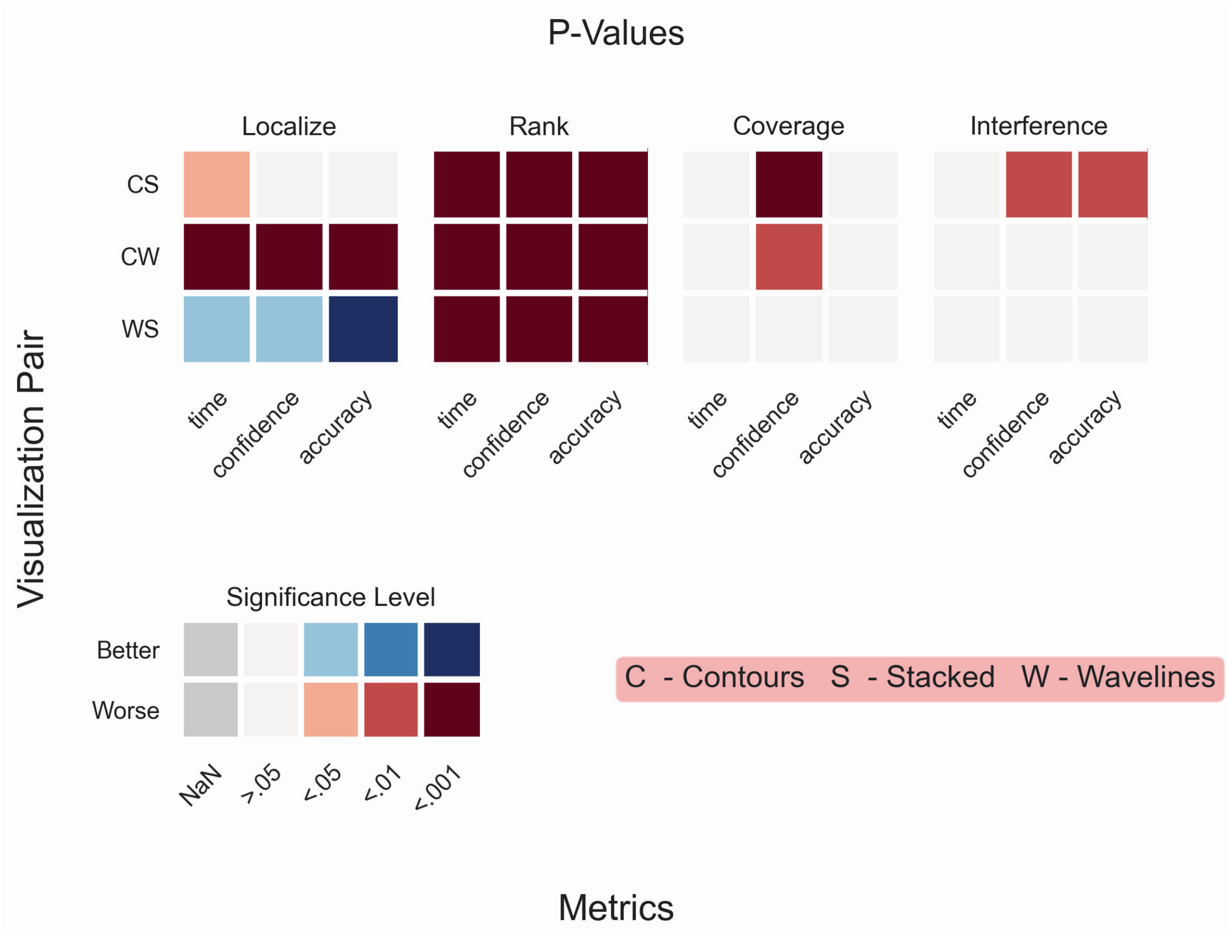


Figure 4.7: Visualization of the p-values received from our significance testing. Each subgraph represents a different analysis task. The rows represent a comparison between two visualizations: C for Contour lines, S for Stacked, and W for Wavelines. Thus, the comparisons between Contours and Stacked are found in the row labeled CS. The columns represent the different metrics. The colors represent the different significance levels. If the first visualization performs better, the color will be blue.

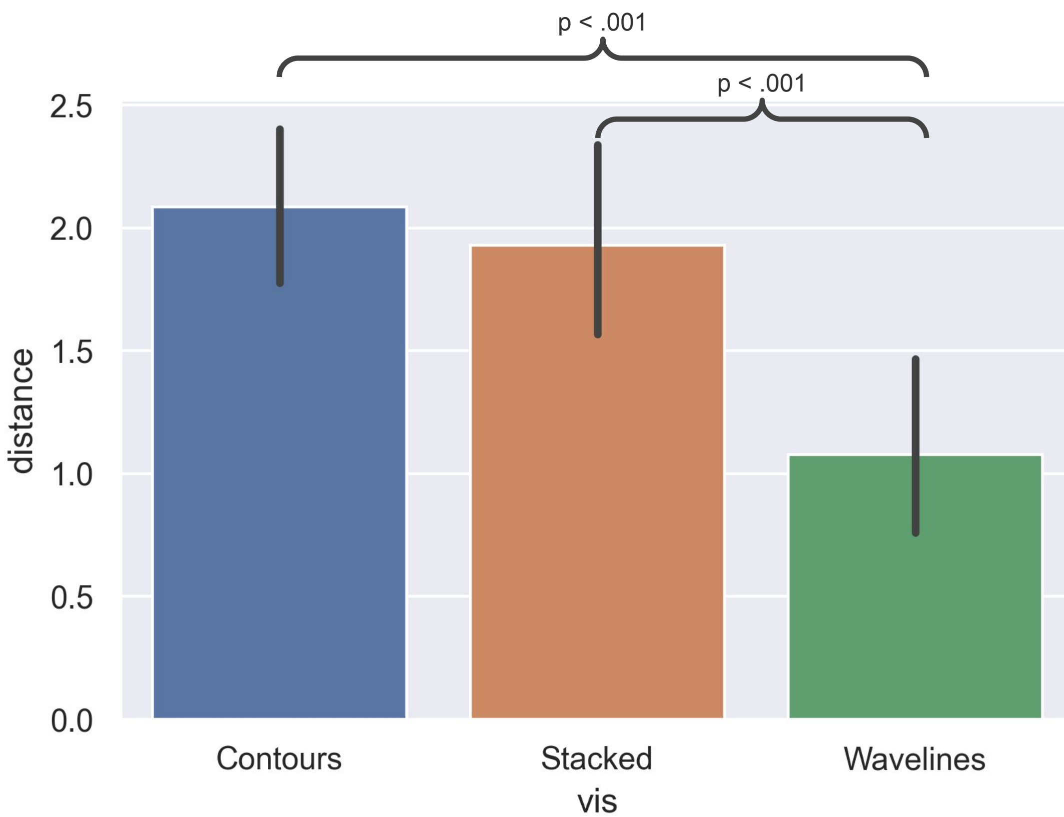


Figure 4.8: Visualization of the distances (in meters) to the correct position for the localization task broken down by visualization. Error bars represent a 95% confidence interval.

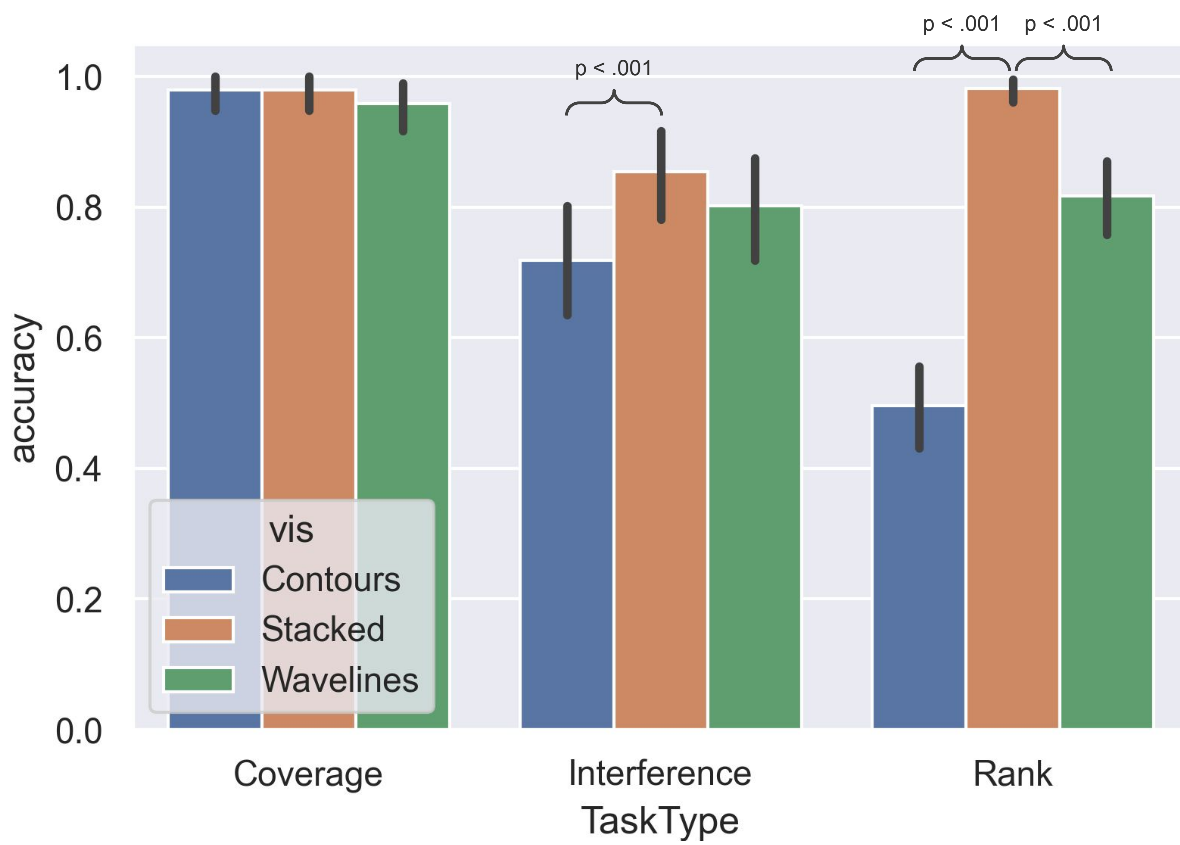


Figure 4.9: Visualization of accuracy broken down by task and visualization. Error bars represent a 95% confidence interval. For a full description of each task's accuracy metric see section 4.6.2.

significant according to Kruskal-Wallis significance tests. However, there is no statistical significance between Stacked and Contours. Both of these conclusions support **H2**.

The remaining accuracies for each analysis task can be seen in fig. 4.9 as well as section 4.6.1. For ranking, our accuracy metric is the ratio of correctly ranked routers to the total number of routers. However, it is important to account for routers with extremely similar signal strengths. If two routers are within .1dBm (decibel-milliwatt) of each other, we allow their rankings to be swapped. This gives us a more stable and conservative estimate of user accuracy. Users were substantially worse at ranking routers while using Contour lines (p ¡ 0.01). All three visualizations had statistically significant accuracy differences for the ranking task, with Stacked performing the best (p ¡ 0.01). Once again, we used Kruskal-Wallis for our significance testing. These results also support **H2**.

The accuracy metrics for interference and coverage worked differently. Since both analyses required a yes or no response, the user accuracy was binary. Since this forms a binomial distribution, we used Barndard's Exact test. Coverage did not show any statistical difference between the visualizations, but Contour lines showed significantly worse performance than the Stacked visualization for interference (p ¡ 0.05). Once again, these results support **H2**. Coverage was simply determined by whether there was greater than zero signal strength at the marked position. Interference worked by checking if the routers communicated on the same router, and testing to see if they were within 10dBm of each other, the sweet spot of interference.

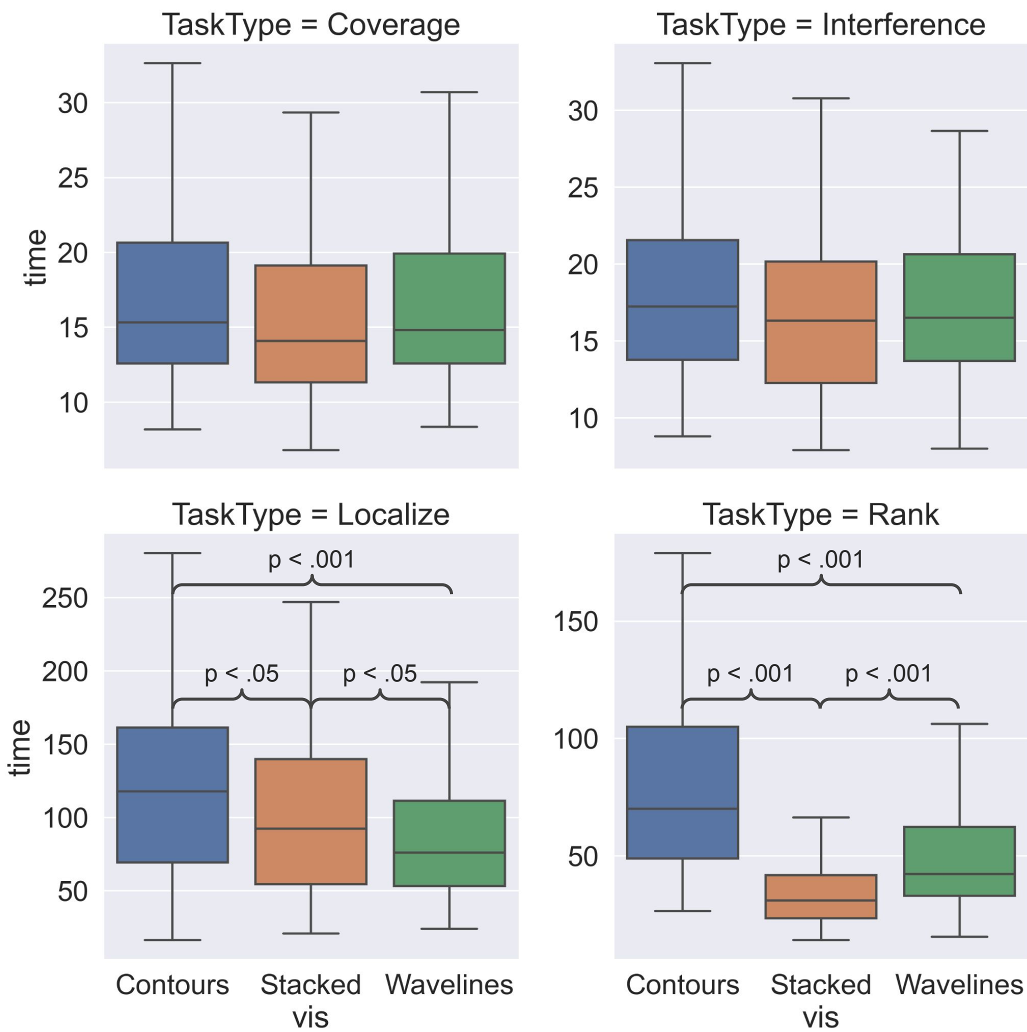


Figure 4.10: The distribution of time-to-completion for each task visualization pair in seconds. The individual Quartile Plots represent the analysis tasks, while the colors represent the visualizations. Ranking had large differences, with Stacked being the fastest of the three visualizations. Here we highlight the statistical significances found in the Localization and Ranking tasks. No significance was found in coverage and interference.

4.6.3 Time

Next, we analyze the task completion time. The analysis of our collected time-to-completion data was the same for each task. Each user was timed from the beginning of the trial until they confirmed their answer through menu interaction. The time the user spent inputting their confidence was not included. The timing distributions per task can be seen in fig. 4.10 and section 4.6.1. Task completion time for coverage and interference did not vary much between the visualizations. Kruskal-Wallis showed that users take significantly longer to localize to a router with Contour lines (Stacked: p ¡ 0.1; Wavelines: p ¡ 0.01). Wavelines also significantly outperformed the Stacked visualization (p ¡ 0.01). Similarly, for Ranking, we found that Contour lines were significantly slower than both Wavelines (p ¡ 0.01) and Stacked (p ¡ 0.01). However, this time the fastest visualization was Stacked, which outperformed Wavelines (p ¡ 0.01). Thus we show that **H3** holds up as well.

4.6.4 Confidence

We also examine user-reported confidence. After each task, the user was asked to rate their confidence in their analysis on a seven-point Likert scale. We report the distribution of user responses in fig. 4.11 and section 4.6.1. We use Kruskal-Wallis tests to determine significance, and as reported in fig. 4.7, users reported significantly more confidence when using the Wavelines visualization for localization than using the other two visualizations (Contours: p ¡ 0.01; Stacked: p ¡ 0.1). Wavelines also outperformed Contour lines in both ranking and coverage (p ¡ 0.01 and p ¡ 0.05, respectively), as did the Stacked Visualization (both with p ¡ 0.01). Stacked bars, however, outperformed Wavelines in ranking (p ¡ 0.01). Wavelines and Stacked inspired similar confidence

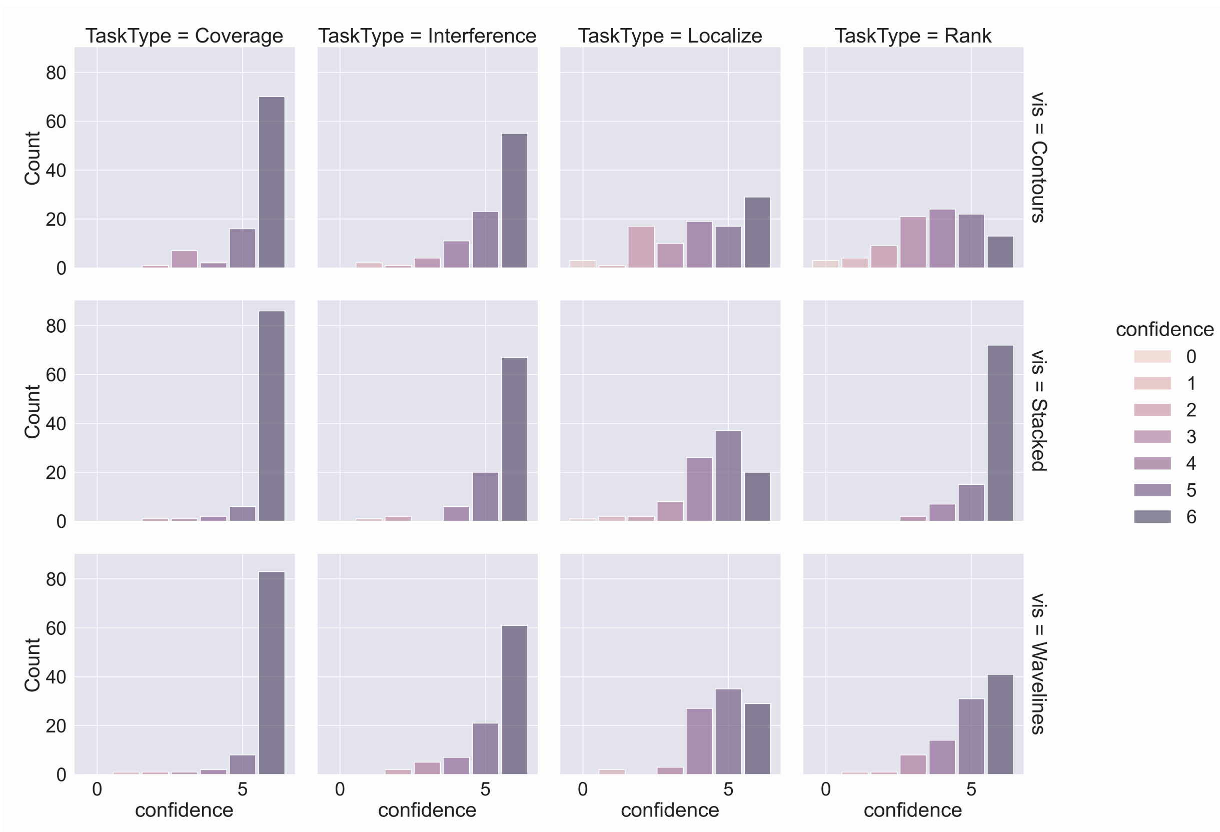


Figure 4.11: Our visualization of the aggregated user confidence levels. Each row represents a different visualization, while each column shows a different analysis task. A confidence score of six implies the user is extremely confident in their response, while a response of zero implies little-to-no confidence.

when users were analyzing coverage and interference. Finally, the Stacked visualization inspired more confidence than Contours for interference analysis (p ¡ 0.05). Thus, **H1** is largely supported but not for every visualization-task pair. Notably, our novel visualizations never inspired less confidence than Contour lines.

4.6.5 Questionnaire Results

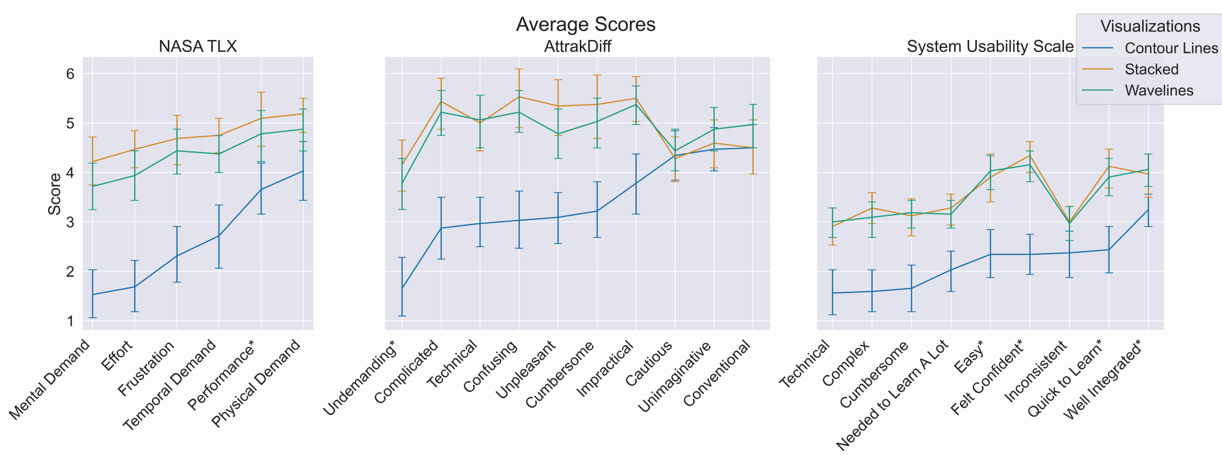


Figure 4.12: Plot of the average responses to the subjective response surveys. Each subplot represents a different standard survey. The lines include 95% confidence intervals. The x-axis shows each question in the survey. Questions that end with an asterisk (*) were flipped, so a higher score implied a more favorable response.

Users were asked to complete an exit questionnaire when they completed their tasks. This survey included three standard assessments designed to extract user opinion. For our purposes, we chose to use: NASA TLX to assess the workload each task imposed, AttrakDiff to assess user perception, and SUS to assess overall usability. The average responses are reported in fig. 4.12 and show that users generally had a worse response to Contour lines, indicating that it is confusing and unpleasant as well as the most frustrating. Our two novel visualizations scored similarly throughout the survey.

4.7 Conclusion and Future Work

In this paper, we have presented two new visualizations for indoor signal propagation analysis, which do not require an RF propagation model. Further, we have shown through a user study that these new visualizations perform just as well, if not better, than a state-of-the-art visualization both in time-to-completion and accuracy for a number of relevant analysis tasks. We have also shown that the novel visualizations improved the confidence of the users in their decision-making and their subjective perception of the visualizations.

During the design of these visualizations, we noticed that they cause much less visual clutter and obstruction of the visual field. Therefore, we experimented with how the visualization would look with more routers, and the results were promising. In future work, we would like to formally evaluate how the visualizations handle more routers. Additionally, the evaluation of this work was carried out on non-experts. Network analysis and maintenance are skilled domains that require expertise. Presenting experts with these visualizations and extracting their opinions would benefit the long-term study of network visualization. This visualization featured a pre-collected dataset and static visualizations. In an analysis scenario, users may collect their data alongside their analysis. More work should be done analyzing how users interact with their visualizations during collection, and what additional challenges collection will present.

Standard area-based visualizations often struggle to visualize multiple routers due to color blending and obscure the user's visual field when presenting three-dimensional data. We have removed environmental occlusion by displaying the data on the environment itself. This allows us to gain the benefits of area-based visualizations in an immersive environment. Our visualization designs do not use color blending and can handle many routers at once. Our new visualizations were designed to work with modeled and unmodeled data. Thus, our new visualizations get the best of both visualization styles. Additionally, we have shown how our novel visualizations improve the realm of WiFi visualization with a user study. In this study, our two new visualizations were compared to contour lines - a state-of-the-art technique. Our visualizations increased user performance and user experience. Our visualizations could be extended to other domains. For instance, RF signals, in general, could benefit from these visualizations with very few changes. Our work could easily benefit the deployment of more complex, next-era networks like 5G and 6G.

Chapter 5: Conclusion and Future Work

Signal visualization has revealed itself to be a fascinating research area with many facets. During my Ph.D., we have explored many different ways for representing RF signal data. My work has focused on WiFi visualization as an easily collectible and prolific source of signal data, but the visualizations we have developed are easily transferable to other RF signals.

When we first started this project, we focused on large outdoor spaces where we developed Programmable Transfer Functions to visualize the isosurfaces of large-scale networks. These transfer functions allowed volume renderings to be customized on the fly, allowing for a more in-depth understanding. Our work outdoors had very few occluders and large open spaces to build our data volumes. When we pivoted to indoor data visualization, we lost this advantage. Inside buildings, walls and ceilings abound and occlude much of the visual field. In this way, volume rendering became an inefficient data visualization, adding even more occlusion and visual clutter to the environment. Thus, we needed to develop further visualizations.

When designing an indoor visualization, we took inspiration from our previous work using volume rendering. Similar to isosurfaces, we began by modeling the iso-contours of the signals. As mentioned, rendering full volumetric surfaces caused too much occlusion, even when using translucency. Instead, We decided to make the natural features of the environment work for us, using it as a canvas to render our data. Thus, our first indoor visualization rendered contour lines on the ceiling and floor of the environment. This visualization required users to extrapolate the direction to the signal center by estimating the normal of the contour. We also decided to implement a visualization that did this calculation for the user called Line Integral Convolution. These visualizations required the data to be modeled before it could be rendered, leading to our next work.

In the final work of this thesis, we developed novel visualizations that do not require a propagation model for the RF sources. The visualizations were designed to create minimal visual clutter by allowing only one line to represent each router at each point on the wall. These visualizations, Wavelines and Stacked Bars were modeled after line graphs and bump charts, respectively. We evaluated these visualizations using a 32-participant user study containing four analysis tasks. This user study showed that our novel visualizations perform at least as well as contour lines in every task and outperform them in certain tasks.

5.0.1 Future Work

Our last work showed that our novel visualizations outperform contour lines in certain conditions. Namely, only five routers were visualized at a time throughout the user study. An interesting future work would analyze user performance with differing router numbers. These visualizations were designed for scale; we hope they would outperform the state of the art when representing many signal sources. Additionally, our user study was conducted with non-experts. These visualizations primarily target professionals with expertise, namely signal analysis and network administrators. Thus, it would be largely valuable to get the feedback of those groups.

More work exists in this realm beyond future evaluation of our visualization strategies.

Firstly, further exploration into how the user of Augmented Reality technology would affect user analysis could be significant. Additionally, in a real-world application, data may need to be updated in real time to aid in the collection process. This update could cause perceptual issues and increase the likelihood of simulator sickness. Exploration of both these areas could aid the development of future immersive applications beyond just those meant to aid network administrators.

Further, it would be of great significance to use these visualizations we have developed here beyond WiFi. Our works were designed to be generalizability. Requiring either no propagation model or designed to easily replace propagation models. With that in mind, future work should explore visualization of other RF signals, especially cellular and Bluetooth, as these are particularly prolific communication technologies.

Bibliography

[1] Alexander Rowden, Eric Krokos, Kirsten Whitley, and Amitabh Varshney. Visualization of wifi signals using programmable transfer functions. *Information*, 13(5), 2022.

[2] Alexander Rowden, Süleyman Aslan, Eric Krokos, Kirsten Whitley, and Amitabh Varshney. Waverider: Immersive visualization of indoor signal propagation. In *Proceedings of the 2022 ACM Symposium on Spatial User Interaction*, SUI '22, New York, NY, USA, 2022. Association for Computing Machinery.

[3] Alexander Rowden, Eric Krokos, Kirsten Whitley, and Amitabh Varshney. Exploring effective immersive approaches to visualizing wifi. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2023.

[4] Vasileios Kokkinos, Kostas Stamos, Nikolaos Kanakis, Kurt Baumann, Anna Wilson, and James Healy. Wireless crowdsourced performance monitoring and verification: WiFi performance measurement using end-user mobile device feedback. In *2016 8th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 432–437, Lisbon, Portugal, October 2016. IEEE.

[5] Wanchai Sangkusolwong and Anya Apavatirut. Indoor WIFI Signal Prediction Using Modelized Heatmap Generator Tool. In *2017 21st International Computer Science and Engineering Conference (ICSEC)*, pages 1–5, Bangkok, November 2017. IEEE.

[6] Valentin Radu, Lito Kriara, and Mahesh K. Marina. Pazl: A mobile crowdsensing based indoor WiFi monitoring system. In *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013)*, pages 75–83, October 2013. ISSN: 2165-963X.

[7] Thor S. Prentow, Antonio J. Ruiz-Ruiz, Henrik Blunck, Allan Stisen, and Mikkel B. Kjærgaard. Spatio-temporal facility utilization analysis from exhaustive WiFi monitoring. *Pervasive and Mobile Computing*, 16:305–316, January 2015.

[8] Jouni Tervonen, Markku Hartikainen, Marjo Heikkila, and Marjut Koskela. Applying and Comparing Two Measurement Approaches for the Estimation of Indoor WiFi Coverage. In *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–4, Larnaca, Cyprus, November 2016. IEEE.

[9] A. Selamat, H. Fujita, and H. Haron. *New Trends in Software Methodologies, Tools and Techniques: Proceedings of the Thirteenth SoMeT_14*. IOS Press, August 2014. Google-Books-ID: oN3YBAAAQBAJ.

[10] Hao Hu, Steven Myers, Vittoria Colizza, and Alessandro Vespignani. WiFi networks and malware epidemiology. *Proceedings of the National Academy of Sciences*, 106(5):1318–1323, February 2009.

[11] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, May 1988.

[12] Robert A. Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. *ACM SIGGRAPH Computer Graphics*, 22, June 1988.

[13] Wei Li, K. Mueller, and A. Kaufman. Empty space skipping and occlusion clipping for texture-based volume rendering. In *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, pages 317–324, Seattle, WA, USA, 2003. IEEE.

[14] Taosong He, Lichan Hong, Arie Kaufman, and Hanspeter Pfister. *Generation of Transfer Functions with Stochastic Search Techniques*. IEEE Computer Society Press, 1996.

[15] J Kruger and R Westermann. Acceleration Techniques for GPU-based Volume Rendering. *IEEE Visualization*, Vis'03, October 2003.

[16] E. LaMar, B. Hamann, and K.I. Joy. Multiresolution techniques for interactive texture-based volume visualization. In *Proceedings Visualization '99 (Cat. No.99CB37067)*, pages 355–543, October 1999. ISSN: 1070-2385.

[17] Stefan Roettger, Stefan Guthe, Daniel Weiskopf, Thomas Ertl, and Wolfgang Strasser. Smart Hardware-Accelerated Volume Rendering. *VisSym03 Joint Eurographics - IEEE TCVG Symposium on Visualization*, page 8 pages, 2003.

[18] Sujal Bista, Jiachen Zhuo, Rao P. Gullapalli, and Amitabh Varshney. Visual Knowledge Discovery for Diffusion Kurtosis Datasets of the Human Brain. In Ingrid Hotz and Thomas Schultz, editors, *Visualization and Processing of Higher Order Descriptors for Multi-Valued Data*, pages 213–234. Springer International Publishing, Cham, 2015.

[19] Hsueh-Chien Cheng, Antonio Cardone, Somay Jain, Eric Krokos, Kedar Narayan, Sriram Subramaniam, and Amitabh Varshney. Deep-Learning-Assisted Volume Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 25(2):1378–1391, February 2019.

[20] O. Sharma, T. Arora, and A. Khattar. Graph-Based Transfer Function for Volume Rendering. *Computer Graphics Forum*, n/a(n/a).

[21] B. Pflesser, U. Tiede, and K. H. Höhne. Towards Realistic Visualization for Surgery Rehearsal. In Nicholas Ayache, editor, *Computer Vision, Virtual Reality and Robotics in Medicine*, Lecture Notes in Computer Science, pages 487–491, Berlin, Heidelberg, 1995. Springer.

[22] M. Sheelagh T. Carpendale, D.J. Cowperthwaite, and F.D. Fracchia. Distortion viewing techniques for 3-dimensional data. In *Proceedings IEEE Symposium on Information Visualization '96*, pages 46–53, October 1996. ISSN: null.

[23] Cheuk Yiu Ip, Amitabh Varshney, and Joseph JaJa. Hierarchical Exploration of Volumes Using Multilevel Segmentation of the Intensity-Gradient Histograms. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2355–2363, December 2012.

[24] J. Kniss, G. Kindlmann, and C. Hansen. Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):270–285, July 2002.

[25] I. Viola, A. Kanitsar, and M.E. Groller. Importance-driven volume rendering. In *IEEE Visualization 2004*, pages 139–145, October 2004. ISSN: null.

[26] S.M.F. Treavett and M. Chen. Pen-and-ink rendering in volume visualisation. In *Proceedings Visualization 2000. VIS 2000 (Cat. No.00CH37145)*, pages 203–210,, Salt Lake City, UT, USA, 2000. IEEE.

[27] Balázs Csébfalvi, Lukas Mroz, Helwig Hauser, Andreas König, and Eduard Gröller. Fast Visualization of Object Contours by Non-Photorealistic Volume Rendering. *Computer Graphics Forum*, 20(3):452–460, 2001.

[28] L. Linsen, T. Van Long, P. Rosenthal, and S. Rosswog. Surface Extraction from Multi-field Particle Volume Data Using Multi-dimensional Cluster Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1483–1490, November 2008.

[29] Joe Kniss and Charles Hansen. *Volume Rendering Multivariate Data to Visualize Meteorological Simulations: A Case Study*. The Eurographics Association, 2002.

[30] J. Kniss, S. premoze, M. Ikits, A. Lefohn, C. Hansen, and E. Praun. Gaussian transfer functions for multi-field volume visualization. In *IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, pages 497–504, Seattle, WA, USA, 2003. IEEE.

[31] Yun Jang, Ralf P. Botchen, Andreas Lauser, David S. Ebert, Kelly P. Gaither, and Thomas Ertl. Enhancing the Interactive Visualization of Procedurally Encoded Multifield Data with Ellipsoidal Basis Functions. *Computer Graphics Forum*, 25(3):587–596, 2006.

[32] N. Sauber, H. Theisel, and H.-P. Seidel. Multifield-Graphs: An Approach to Visualizing Correlations in Multifield Scalar Data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):917–924, September 2006.

[33] Ismail Demir, Johannes Kehrer, and Rüdiger Westermann. Screen-space silhouettes for visualizing ensembles of 3d isosurfaces. In *2016 IEEE Pacific Visualization Symposium (PacificVis)*, pages 204–208, 2016.

[34] Randima Fernando. *GPU Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics*. Pearson Higher Education, 2004.

[35] Jochen Jankowai and Ingrid Hotz. Feature level-sets: Generalizing iso-surfaces to multi-variate data. *IEEE Transactions on Visualization and Computer Graphics*, 26(2):1308–1319, 2020.

[36] Olgor.com. WiFi analyzer. Cairo, Egypt, August 2020, url:`https://play.google.com/store/apps/details?id=abdelrahman.wifianalyzerpro&hl=en_US&gl=US`, 2020.

[37] Tarlogic Research. Acrylic WiFi heatmaps. Version 1.1.3, Spain, 2022, url: `https://www.acrylicwifi.com/en/wlan-wifi-wireless-network-software-tools/wifi-site-survey-software-acrylic-heat-maps/`, 2022.

[38] Takashi Nakamura, Mondher Bouazizi, Kohei Yamamoto, and Tomoaki Ohtsuki. Wi-Fi-CSI-based fall detection by spectrogram analysis with CNN. In *IEEE Global Communications Conference, GLOBECOM 2020, Virtual Event, Taiwan, December 7-11, 2020*, pages 1–6. IEEE, 2020.

[39] Robert A. Drebin, Loren Carpenter, and Pat Hanrahan. Volume rendering. *SIGGRAPH Comput. Graph.*, 22(4):65–74, June 1988.

[40] Jean Rioult, Samuel Delgrande, Nicolas Bremard, Gregoire Copin, and Virginie Deniau. Autonomous electromagnetic mapping system in augmented reality. In *2019 International Symposium on Electromagnetic Compatibility - EMC EUROPE*, pages 138–143. IEEE, 2019.

[41] Wi-Fi Solutions. WiFi AR. Version 5.6.1, Kiev, Ukraine, January 2022. url: `https://play.google.com/store/apps/details?id=ua.com.wifisolutions.wifivr&hl=en`,, 2022.

[42] Ekahau. Ekahau survey. Helsinki, Finland, 2022, url:`https://www.ekahau.com/products/ekahau-connect/survey/`, 2022.

[43] Ken Kawamoto. AR sensor. Version 1.1.3, January 2019, url: `https://play.google.com/store/apps/details?id=com.ken.arsensor`, 2019.

[44] Ivan E. Sutherland. A head-mounted three dimensional display. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, AFIPS '68 (Fall, part I), page 757–764, New York, NY, USA, Dec 1968. Association for Computing Machinery.

[45] Ian E. McDowall, Mark T. Bolas, Steven D. Pieper, Scott S. Fisher, and Jim Humphries. Implementation and integration of a counterbalanced crt-based stereoscopic display for interactive viewpoint control in virtual-environment applications. page 136, Santa Clara, CA, United States, Sep 1990.

[46] Steve Bryson and Creon Levit. The virtual windtunnel: an environment for the exploration of three-dimensional unsteady flows. In *Proceedings of the 2nd conference on Visualization '91*, VIS '91, page 17–24, Washington, DC, USA, Oct 1991. IEEE Computer Society Press.

[47] Adrien Fonnet and Yannick Prie. Survey of immersive analytics. *IEEE Transactions on Visualization and Computer Graphics*, 27(3):2101–2122, Mar 2021.

[48] R. van Teylingen, W. Ribarsky, and C. van der Mast. Virtual data visualizer. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):65–74, Jan 1997.

[49] Elif Hilal Korkut and Elif Surer. *Visualization in virtual reality: a systematic review*. Number arXiv:2203.07616. Mar 2022. arXiv:2203.07616 [cs] type: article.

[50] Wesley Willett, Yvonne Jansen, and Pierre Dragicevic. Embedded data representations. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):461–470, Jan 2017.

[51] Sean White and Steven Feiner. Sitelens: situated visualization techniques for urban site visits. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, page 1117–1120, New York, NY, USA, Apr 2009. Association for Computing Machinery.

[52] Wolfgang Büschel, Anke Lehmann, and Raimund Dachselt. Miria: A mixed reality toolkit for the in-situ visualization and analysis of spatio-temporal interaction data. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, page 1–15, New York, NY, USA, May 2021. Association for Computing Machinery.

[53] Stefanie Zollmann, Tobias Langlotz, Raphael Grasset, Wei Hong Lo, Shohei Mori, and Holger Regenbrecht. Visualization techniques in augmented reality: A taxonomy, methods and patterns. *IEEE Transactions on Visualization and Computer Graphics*, 27(9):3808–3825, Sep 2021.

[54] Y. Shreyas, Shivam Seth, and Rajat Agarwal. Wireless Network Visualization and Indoor Empirical Propagation Model for a Campus WI-FI Network. *World Academy of Science, Engineering and Technology*, August 2008.

[55] Richard A. Becker, William S. Cleveland, and Ming-Jen Shyu. The Visual Design and Control of Trellis Display. *Journal of Computational and Graphical Statistics*, 5(2):123–155, June 1996.

[56] Xiangtong Chu, Xiao Xie, Shuainan Ye, Haolin Lu, Hongguang Xiao, Zeqing Yuan, Zhutian Chen, Hui Zhang, and Yingcai Wu. Tivee: Visual exploration and explanation of badminton tactics in immersive visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):118–128, 2022.

[57] Nobuaki Ohno and Akira Kageyama. Scientific visualization of geophysical simulation data by the CAVE VR system with volume rendering. *Physics of the Earth and Planetary Interiors*, 163(1):305–311, 2007. Computational Challenges in the Earth Sciences.

[58] Joe Kniss, Charles D Hansen, Michel Grenier, and Tom Robinson. Volume rendering multivariate data to visualize meteorological simulations: a case study. In *VisSym*, pages 189–194. Citeseer, 2002.

[59] IEEE standard for information technology—telecommunications and information exchange between systems local and metropolitan area networks—specific requirements - part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. Standard, Institute of Electical and Electronics Engineers, December 2016.

[60] Wei Liu, Daan Pareit, Eli De Poorter, and Ingrid Moerman. Advanced spectrum sensing with parallel processing based on software-defined radio. *EURASIP Journal on Wireless Communications and Networking*, 2013:228, 12 2013.

[61] Yuhong Zhou, Thi Hong Diep Dao, Jean-Claude Thill, and Eric Delmelle. Enhanced 3D visualization techniques in support of indoor location planning. *Computers, Environment and Urban Systems*, 50:15–29, March 2015.

[62] Tim Sodergren, Jessica Hair, Jeff M. Phillips, and Bei Wang. Visualizing Sensor Network Coverage with Location Uncertainty. In *2017 IEEE Visualization in Data Science (VDS)*, pages 52–59, October 2017.

[63] Chi Yoon Jeong, Beom Hwan Chang, and Jung Chan Na. A Survey on Visualization for Wireless Security. In *2008 Fourth International Conference on Networked Computing and Advanced Information Management*, volume 1, pages 129–132, September 2008.

[64] Ryan Custer, Erik Lee, and Jamie Van Randwyk. Using 3D visualization to enhance understanding of computer network state. In *ACM SIGGRAPH 2005 Posters*, SIGGRAPH '05, pages 130–es, Los Angeles, California, July 2005. Association for Computing Machinery.

[65] George Koutitas, Varun Kumar Siddaraju, and Vangelis Metsis. In Situ Wireless Channel Visualization Using Augmented Reality and Ray Tracing. *Sensors*, 20(3):690, January 2020.

[66] J. Gray, A. Bosworth, A. Lyaman, and H. Pirahesh. Data cube: a relational aggregation operator generalizing GROUP-BY, CROSS-TAB, and SUB-TOTALS. In *Proceedings of the Twelfth International Conference on Data Engineering*, pages 152–159, 1996. ISSN: 1063-6382.

[67] Brian Cabral and Leith Casey Leedom. Imaging vector fields using line integral convolution. In Mary C. Whitton, editor, *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1993, Anaheim, CA, USA, August 2-6, 1993*, pages 263–270. ACM, 1993.

[68] Hans-Christian Hege and Detlev Stalling. Fast LIC with Piecewise Polynomial Filter Kernels. In Hans-Christian Hege and Konrad Polthier, editors, *Mathematical Visualization*, pages 295–314. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.

[69] Henrik Battke, Detlev Stalling, and Hans-Christian Hege. Fast Line Integral Convolution for Arbitrary Surfaces in 3D. In Hans-Christian Hege and Konrad Polthier, editors, *Visualization and Mathematics: Experiments, Simulations and Environments*, pages 181–195. Springer, Berlin, Heidelberg, 1997.

[70] Detlev Stalling and Hans-Christian Hege. Fast and resolution independent line integral convolution. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 249–256, New York, NY, USA, September 1995. Association for Computing Machinery.

[71] D. Stalling. LIC on Surfaces Detlev Stalling, 1997.

[72] L. K. Forssell. Visualizing flow over curvilinear grid surfaces using line integral convolution. In *Proceedings Visualization '94*, pages 240–247, 1994.

[73] Xiaoyang Mao, Makoto Kikukawa, Noboru Fujita, and Atsumi Imamiya. Line Integral Convolution for 3D Surfaces. In Wilfrid Lefer and Michel Grave, editors, *Visualization in Scientific Computing '97*, Eurographics, pages 57–69, Vienna, 1997. Springer.

[74] T. Toledo and W. Celes. Visualizing 3d flow of black-oil reservoir models on arbitrary surfaces using projected 2d line integral convolution. In *2011 24th SIBGRAPI Conference on Graphics, Patterns and Images*, pages 133–140, 2011. ISSN: 2377-5416.

[75] Renan Guarese, Pererik Andreasson, Emil Nilsson, and Anderson Maciel. Augmented situated visualization methods towards electromagnetic compatibility testing. *Computers & Graphics*, 94:1–10, 2021.

[76] Bela Julesz. Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97, March 1981.

[77] Haleh. H. Shenas and Victoria Interrante. Compositing color with texture for multi-variate visualization. In *Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, GRAPHITE '05, pages 443–446, Dunedin, New Zealand, November 2005. Association for Computing Machinery.

[78] Christopher Weigle, William G. Emigh, Geniva Liu, Russell M. Taylor, James T. Enns, and Christopher G. Healey. Oriented sliver textures: A technique for local value estimation of multiple scalar fields. In *Proceedings of the Graphics Interface 2000 Conference, May 15-17, 2000, Montr'eal, Qu'ebec, Canada*, pages 163–170, May 2000.

[79] Lasse H. Hansen, Philipp Fleck, Marco Stranner, Dieter Schmalstieg, and Clemens Arth. Augmented reality for subsurface utility engineering, revisited. *IEEE Transactions on Visualization and Computer Graphics*, 27(11):4119–4128, 2021.

[80] Eric Krokos, Alexander Rowden, Kirsten Whitley, and Amitabh Varshney. Visual analytics for root dns data. In *2018 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pages 1–8, 2018.

[81] Adam Perer and Ben Shneiderman. Integrating statistics and visualization: Case studies of gaining clarity during exploratory data analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, page 265–274, New York, NY, USA, 2008. Association for Computing Machinery.

[82] Renan Guarese, Pererik Andreasson, Emil Nilsson, and Anderson Maciel. Augmented situated visualization methods towards electromagnetic compatibility testing. *Computers & Graphics*, 94:1–10, Feb 2021.

[83] Valentin Radu, Lito Kriara, and Mahesh K. Marina. Pazl: A mobile crowdsensing based indoor wifi monitoring system. In *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013)*, page 75–83, Oct 2013.

[84] Alexander Rowden, Eric Krokos, Kirsten Whitley, and Amitabh Varshney. Visualization of wifi signals using programmable transfer functions. *Information*, 13(55):224, May 2022.

[85] BadVR. Seesignal, 2019.

[86] Ahmed Bayyari and M. Eduard Tudoreanu. The impact of immersive virtual reality displays on the understanding of data visualization. In *Proceedings of the ACM symposium on Virtual reality software and technology - VRST '06*, page 368, Limassol, Cyprus, 2006. ACM Press.

[87] Ciro Donalek, S. G. Djorgovski, Scott Davidoff, Alex Cioc, Anwell Wang, Giuseppe Longo, Jeffrey S. Norris, Jerry Zhang, Elizabeth Lawler, Stacy Yeh, Ashish Mahabal, Matthew Graham, and Andrew Drake. Immersive and collaborative data visualization using virtual reality platforms. 2014.

[88] Eric Krokos, Catherine Plaisant, and Amitabh Varshney. Virtual memory palaces: immersion aids recall. *Virtual Reality*, 23(1):1–15, Mar 2019.

[89] Payod Panda, Byungsoo Kim, Sana Behnam-Asl, and Elif Sener. Immersive analytics: A user-centered perspective. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*, page 1–3, Osaka Japan, Dec 2021. ACM.

[90] Tobias Skog. Activity wallpaper: ambient visualization of activity information. In *Proceedings of the 2004 conference on Designing interactive systems processes, practices, methods, and techniques - DIS '04*, page 325, Cambridge, MA, USA, 2004. ACM Press.

[91] Sean Michael White. *Interaction and presentation techniques for situated visualization*. phd, Columbia University, USA, 2009. AAI3373578 ISBN-13: 9781109341492.

[92] Ramesh Raskar, Paul Beardsley, Jeroen van Baar, Yao Wang, Paul Dietz, Johnny Lee, Darren Leigh, and Thomas Willwacher. Rfig lamps: interacting with a self-describing world via photosensing wireless tags and projectors. *ACM Transactions on Graphics*, 23(3):406–415, Aug 2004.

[93] Lasse H. Hansen, Philipp Fleck, Marco Stranner, Dieter Schmalstieg, and Clemens Arth. Augmented reality for subsurface utility engineering, revisited. *IEEE Transactions on Visualization and Computer Graphics*, 27(11):4119–4128, 2021.

[94] Junhan Zhao, Xiang Liu, Chen Guo, Zhenyu Cheryl Qian, and Yingjie Victor Chen. Phoenixmap: An abstract approach to visualize 2d spatial distributions. *IEEE Transactions on Visualization and Computer Graphics*, 27(3):2000–2014, Mar 2021.

[95] Adrian Mayorga and Michael Gleicher. Splatterplots: Overcoming overdraw in scatter plots. *IEEE Transactions on Visualization and Computer Graphics*, 19(9):1526–1538, Sep 2013.

[96] Sihang Li, Jiacheng Yu, Mingxuan Li, Le Liu, Xiaolong Zhang, and Xiaoru Yuan. A framework for multiclass contour visualization. *IEEE Transactions on Visualization and Computer Graphics*, page 1–10, 2022.

[97] Roeland Scheepens, Huub van de Wetering, and Jarke J. van Wijk. Contour based visualization of vessel movement predictions. *International Journal of Geographical Information Science*, 28(5):891–909, May 2014.

[98] John David Anderson and John Wendt. *Computational Fluid Dynamics*, volume 206. Springer, 1995.

[99] Páll Bergthörsson and Bo R. DÖÖs. Numerical weather map analysis. *Tellus*, 7(3):329–340, 1955.

[100] Jean Rioult, Samuel Delgrande, Nicolas Bremard, Gregoire Copin, and Virginie Deniau. Autonomous electromagnetic mapping system in augmented reality. In *2019 International Symposium on Electromagnetic Compatibility - EMC EUROPE*, page 138–143, Barcelona, Spain, Sep 2019. IEEE.

[101] Jouni Tervonen, Markku Hartikainen, Marjo Heikkila, and Marjut Koskela. Applying and comparing two measurement approaches for the estimation of indoor wifi coverage. In *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, page 1–4, Nov 2016.

[102] Y. Shreyas, Shivam Seth, and Rajat Agarwal. *Wireless Network Visualization and Indoor Empirical Propagation Model for a Campus WI-FI Network*.

[103] Yuhong Zhou, Thi Hong Diep Dao, Jean-Claude Thill, and Eric Delmelle. Enhanced 3d visualization techniques in support of indoor location planning. *Computers, Environment and Urban Systems*, 50:15–29, Mar 2015.

[104] Tamara Munzner. *Visualization Analysis and Design*. CRC Press, 2014.

[105] George Koutitas, Varun Kumar Siddaraju, and Vangelis Metsis. In situ wireless channel visualization using augmented reality and ray tracing. *Sensors*, 20(3):690, Jan 2020.

[106] Ryan Custer, Erik Lee, and Jamie Van Randwyk. Using 3d visualization to enhance understanding of computer network state. In *ACM SIGGRAPH 2005 Posters*, SIGGRAPH '05, pages 130–es, Los Angeles, California, Jul 2005. Association for Computing Machinery.

[107] W.E Sherwood. *Oxford rowing; a history of boat-racing at Oxford from the earliest times, with a record of the races; compiled principally from official sources*. Oxford, Frowde, 1900.

[108] Henry Gannett. Statistical atlas of the united states, based upon the results of the eleventh census, 1898.

[109] Isabella Cingolani, Pietro Panzarasa, and Lucia Tajoli. Countries' positions in the international global value networks: Centrality and economic performance. *Applied Network Science*, 2(1):21, Dec 2017.

[110] Ethan Otto, Eva Culakova, Sixu Meng, Zhihong Zhang, Huiwen Xu, Supriya Mohile, and Marie A. Flannery. Overview of sankey flow diagrams: Focusing on symptom trajectories in older adults with advanced cancer. *Journal of Geriatric Oncology*, 13(5):742–746, Jun 2022.