A Virtual Reality System for the Simulation and Manipulation of Wireless Communication Networks

Tobias Rick* Anette von Kapri† Torsten Kuhlen‡

Virtual Reality Group
JARA, RWTH-Aachen University

ABSTRACT

The knowledge of the propagation behavior of radio waves is a fundamental prerequisite for planning and optimizing mobile radio networks. Propagation effects are usually simulated numerically, since real-world measurement campaigns are time-consuming and expensive. Automatic planning algorithms can explore a vast amount of network configurations to find good deployment schemes. However, complex urban scenarios demand for a great emphasis on site-specific details in the propagation environment which are often not covered by automatic approaches. Therefore, we have combined the simulation of radio waves with an interactive exploration and modification of the propagation environment in a virtual reality prototype application. By coupling real-time simulation and manipulation tasks we can provide an uninterrupted user-centered workflow.

Index Terms: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality I.6.3 [Simulation and Modeling]: Applications—

1 INTRODUCTION

The propagation simulation of radio waves is a fundamental prerequisite for planning and optimization of radio networks. For instance, coverage analysis, interference estimation or channel and power allocation all rely on propagation predictions. In wireless communication networks optimal antenna sites are determined by either conducting a series of extensive propagation measurements or by estimating field strengths numerically. In order to cope with the vast amount of different configurations to select the best candidate from and to avoid expensive measurement campaigns, numerical predictions have to be both accurate and fast.

Dense urban areas demand for a great emphasis on site-specific details in the propagation environment which are often not covered by automatic approaches. City models may not be as up-to-date as recent satellite images of the same area and often geographical databases that serve as input to the propagation simulations are incomplete or missing information. There may be critical sites or clinical areas (e.g. hospitals) where exposure to certain power levels may be hazardous to medical equipment and a minimum safe distance for radio transmitter sites has to be maintained.

Since real-time algorithms for the propagation simulation have become available recently, we propose an interactive manipulation and modeling of the propagation environment which is directly coupled to the simulation thereof. We present a prototype application that integrates real-time simulations of propagation predictions with the interactive exploration and analysis in a Virtual Environment (VE).

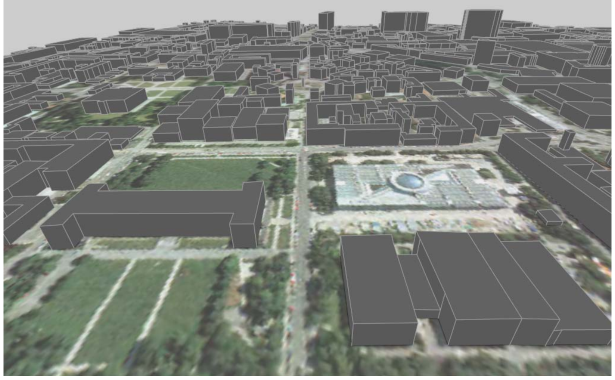

Figure 1: Typical urban propagation environment with simplistic city model and satellite image for geographical reference.

We consider the simulation in dense urban environments with the frequency range of common mobile communication systems, i.e., several hundred MHz up to few GHz. The basic propagation phenomena in this frequency range are reflection, diffraction and scattering. All effects contribute to the radio signal distortions and give rise to signal fluctuations and additional signal propagation losses. One important aspect in radio wave propagation is the prediction of the mean received signal strength which can be simulated by taking complex interactions between radio waves and the propagation environment into account. The signal strength of multiple antennas can then be used to compute derived quantities for deterministic network statistics. We use the parallel many-core architecture of modern Graphics Processing Units (GPUs) for the implementation of a real-time simulation algorithm. All user input and manipulation from within the VE is directly communicated to the simulation algorithm which immediately updates all propagation predictions such that the effect is instantly visible to the user.

The remainder of this paper is organized as follows. After briefly discussing the technical background and reviewing previous work in Section 2, we give an overview of the general application layout and features in Section 3. Then, we discuss the technical realization of our Virtual Reality (VR) prototype application in Section 4 and conclude the paper in Section 5.

2 BACKGROUND & RELATED WORK

In wireless communication networks a widely adapted model (cf. [8]) for describing the path loss between transmitter and receiver at a distance d is

$$P^{\mathrm{dB}}(d) = c_f + \gamma \cdot 10 \cdot \log_{10}(d)$$

at a frequency f where $c_f = 20 \cdot \log_{10}\left(\frac{4\pi f}{c}\right)$ is the frequency dependent loss, c denotes the speed of light. The path loss coefficient γ depends on the land cover and usually ranges between 2 (free space) and 3.5 (urban environment).

---

*e-mail: tobias.rick@rwth-aachen.de
†e-mail: anette.kapri@rwth-aachen.de
‡e-mail: kuhlen@vr.rwth-aachen.de

IEEE Virtual Reality 2011
19 - 23 March, Singapore
978-1-4577-0038-5/11/$26.00 ©2011 IEEE

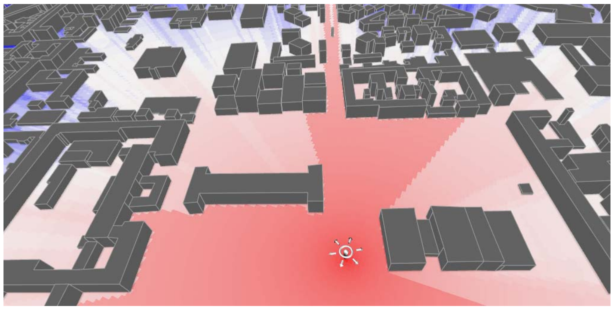

Figure 2: Simulation of radio wave propagation with visualization of mean received field strength, colors ranging from red (strong) to white (medium) and blue (weak).

Propagation models that approximate the actual path loss by a parametrized function like above can be categorized as empirical models. Well-known empirical models are the work of Hata [4] and Ikegami [5]. They determined their parameter values by conducting extensive measurement campaigns and analyzed the dependence of field strength in urban propagation environments with respect to height gain, dependence on street width, propagation distance and radio frequency. However, empirical models are prone to prediction errors if their original model assumptions contradict the physical reality of the supplying area. Especially in Europe with its heterogeneous propagation environments of historically grown cities these models provide only limited value.

More sophisticated approaches rely on the computation of actual propagation paths due to wave guiding effects like reflection, diffraction and scattering. Typical algorithms are often based on ray tracing which was originally introduced by Whitted in [12] to compute global illumination effects based on geometric optics. Although, global illumination as formulated by Kajiya [6] and radio wave propagation are similar problem statements, different propagation effects like diffraction or interference become dominant when shifting from visible light to radio waves due to the different size of wavelengths. GPU implementations of radio wave propagations are presented by Rick and Kuhlen [9] who trace propagation paths in a discrete fashion by repeated rasterization of shadow volumes and Schmitz et. al [10] extending classical beam tracing. Recent VR applications of simulations include but are not limited to the reconstruction of traffic flows in [11] or the interactive simulation of nanoparticle manipulation in [1].

3 APPLICATION

We formulate the conceptual requirements of our VR application based on informal discussions with domain experts as follows: (1) visualization of the radio network within a geographical reference frame, (2) in situ simulation of propagation effects and (3) manipulation of site-specific details.

Furthermore, a fundamental prerequisite is the real-time requirement of the overall system in order to provide an interactive VR experience. We derived the main tasks from the above formulated requirements: adjustment of visualization parameters, configuration of transmitter sites and manipulation of the city model. We coupled our VR application directly to the simulation such that every user input triggers a real-time update of the simulation input parameters and thereby a subsequent computation of the propagation predictions.

3.1 Geographical Reference & Visualization

Geographical reference is usually provided by a specification of the supplying area in a global coordinate system (e.g. WGS84) whereas context is provided in form of satellite images and a city model (cf. Figure 1) which for instance can be acquired from LIDAR data as demonstrated for the creation of virtual cities in [7]. Most of the time satellite images are up-to-date and very accurate, whereas city models are usually very expensive and of rather coarse resolution (two to five meters). Often, we observe a gap between the information of recent satellite images and the corresponding building data of the city model. Building information may be incomplete or do not reflect the latest building development. Sometimes recent buildings are missing completely. We address this issue in Section 3.3.

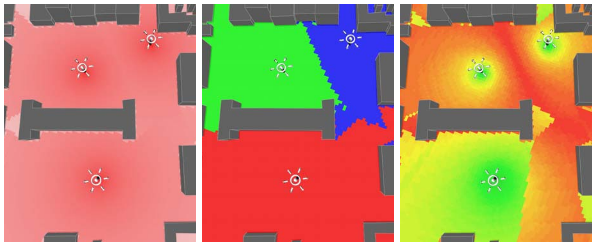

Figure 3: The relationship of multiple antennas are displayed as network properties that reveal different statistics. The left image shows the coverage of field strength whereas the middle picture indicates from which antenna the strongest signal arrives. The right image depicts the ratio of interference from other signals, green stands for low interference, red for high interference. For good reception high coverage must be combined with low interference.

As illustrated in Figure 2 transmitter sites are visualized on top of the geographical data and may be subject to further manipulation. The simulation results in terms of predicted signal strength or interference are displayed as a colored pixel image parallel to the ground plane. Initially, it depicts the propagation effects at 1.5 meters above ground which is common for analyzing cell phone reception.

3.2 Simulation of Propagation Effects

Ray tracing approaches are an established technique for radio wave propagation simulations, however, such approaches need to be extended to include diffraction, which is a predominant effect for common mobile radio frequencies. Diffraction along edges is usually modeled by shooting a multitude of rays into the shadow cone of the diffracting edge which usually results in a large computational overhead.

The key idea for implementing diffraction on the GPU is to utilize the concept of shadow volumes to mark regions which are in shadow. For the propagation of radio waves, only those propagation paths are of interest which pass through a certain height level above ground where cell phone reception is required. By applying a modified shadow volume technique, all pixels that are inside a diffraction cone are identified on an image plane, which is setup such as to correspond to the receiver plane. Hence, a GPU algorithm can implement the problem of finding diffraction rays as repeated shadow computations, which can be done extremely fast on recent graphics cards. We will not go into further specifics about the algorithm, details of our GPU implementation can be found in [9].

Here, we want to discuss what our VR prototype has to know about the propagation algorithm, hence what is the input and the output of the radio wave simulation that the system and thereby the user will be aware of. We start with the data requirements of the simulation which basically is just a simplified city model. Shape of rooftops are usually omitted in common propagation algorithms. Thus, a building is described by its polygonal outline and one height value. The propagation loss is calculated as a function of city model, radiation parameters and location of the transmitting antenna. If multiple antennas are present in a network, the simulation has to be done separately for each given transmitter location.

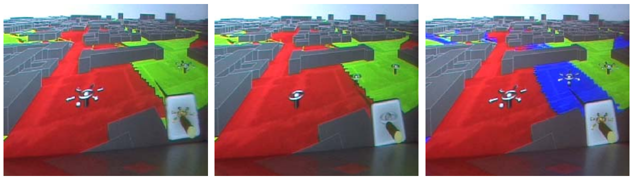

Figure 4: The creation of a new transmitter site is depicted in the sector view where the receiver plane is colored according to the base station with maximum strength. When the mode for adding a transmitter site is entered, a wireframe model of a transmitter is rendered at the target location (middle image). The antenna is added to network by a button press and the propagation simulation immediately updates the sector view (blue for the new antenna in the right image).

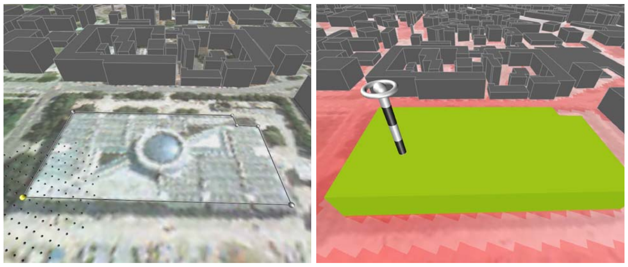

Figure 5: A missing building is sketched directly on the satellite image. A grid raster is displayed around the cursor to assist the user in case of jitter. The effect of the new building becomes instantly visible in the simulation results.

The simulation result is a simple pixel map where each pixel corresponds to a grid location within the supplying area that describes the mean signal attenuation.

Changes to a particular antenna lead to a complete recalculation of network properties. Currently, we provide the following statistics for network analysis: (1) Maximum intensity (MI), (2) Best server (BS) and (3) Carrier-to-Interference (CI). The MI view displays the attenuation of the strongest received signal strength among all antennas at each location. For the BS view each antenna is assigned a unique color and each receiving location is colored with the color of the strongest antenna, it provides a so-called sector view. The CI view shows the signal to interference ratio and is a major indicator of network capacity in interference limited networks which basically are all current (3G and 4G) standards for mobile telecommunications. Figure 3 gives an impression of the three statistic views.

3.3 Manipulation of Site Specific Details

We will first describe the manipulation options for transmitter sites and then those for the city model.

Transmitter Sites An initial network setup with the description of transmitter site locations can for instance be computed by automatic cell planning algorithms. In our prototype application the user can change the location of antenna sites at runtime. If required, additional sites can be added to the network by pointing with a hand-held 6DOF device at the desired position and pressing the corresponding button. Figure 4 depicts the creation of a new transmitter site within the VR prototype. During all antenna operations, visual feedback and simulation updates are performed instantly in real-time.

City Model To account for missing or incorrect information in the city model database, we let the user manipulate the city model directly in the simulation environment. Sketching has been an established method for generating content on-the-fly, a recent example is content authoring in AR games as described in [3]. Since satellite images are already provided as geographical context information we let the user correct or create missing buildings by drawing their footprint on top of the satellite image, see Figure 5. A 3D model is created automatically by extruding the floor plan along the axis pointing upwards. For fine-tuning building heights the rooftop level of each building can be adjusted separately. All changes to the building database are directly communicated to the simulation algorithm which immediately updates all propagation predictions such that the effect is instantly visible to the user.

4 REALIZATION

So far we have described the conceptual features of our prototype application from the view of the user. In this section we want to discuss issues that arose in the concrete implementation of certain features and what challenges had to be faced to maintain interactive frame rates for simulation and manipulation tasks. We will focus on the differences and difficulties we faced when providing a virtual reality interfaces as opposed to common 2D desktop metaphors.

Our CAVE system of five walls each with passive stereo (1600 × 1200 pixels) is run by 10 rendering slaves (one for each eye per wall) and a master node which handles user inputs, tracking, data distribution and synchronization. Each node is equipped with an Nvidia Quadro FX 5600 graphics card. Latency of our infrared optical tracking system was between 80–120ms at an update rate of 60 Hz. In our standard setup every slave runs its own instance of the VR application performing all computations on its own to minimize data transfer between nodes.

In order to realize the above described interaction we need the following basic building blocks: real-time *simulation* of propagation effects, *navigation* through the virtual environment, *selection* of objects that can be subject to manipulation, object *manipulation* based on 6DOF tracking data and *system control*.

Simulation The technical details for achieving a real-time simulation of a single transmitter site are described in [9]. Here, we focus on how to couple the simulation with a changing propagation environment. In order to offer a real-time manipulation we had to update the city model database directly on the GPU. For computation efficiency the city model was initially transformed into a vertex buffer object in GPU memory. Additional information was attached to the vertex geometry (e.g. material properties and computational flags). For the manipulation of existing buildings (change of roof height) we introduced an additional dependent texture lookup in the vertex shader that would transform the building geometry. The texture lookup maps unique building identifiers to a height value. To minimize data transfers between GPU (device) and CPU (host) we kept identical copies of the lookup table in host and device memory. Upon changes to building heights it is sufficient to upload only the modified part of the lookup table to texture memory. A subsequent simulation of the propagation prediction will then automatically use the updated texture for the dependent lookup. The remainder of the simulation code was left unchanged. Newly created buildings were registered to the simulation by introducing new vertex buffer objects which were created dynamically at runtime.

Navigation Since the user is equipped with a hand-held 6DOF device, navigation is achieved by a simple travel-by-pointing metaphor. Furthermore, the user can change the overall scale between being in the city and a world-in-miniature view to provide overview of the whole scene while providing details on demand.

Selection We introduce selection as an extra paragraph here because the number of objects that the user can interact with proved to be a performance issue. Our test scenario of approximately 7 km^2 consists of 2,086 distinct buildings with approximately 82,000 triangles in total. First tests showed heavy drops in performance when computing the intersections between selection ray and objects naively on every render node (which is default for the used VR toolkit). We then switched the computation to a designated node (the master) which would not do any renderings and used an asynchronous connection that would broadcast the results from the selection process on a multicast address where every rendering node could listen and react accordingly. A refresh rate of 30 Hz to update object selection did not seem to introduce a noticeable lag to the system which corresponds to half the tracking update rate. By decoupling the selection process from the main body of the application and using an asynchronous broadcast for scattering the information, the frame rate did not drop below 40 frames per second for a network of eight transmitters and 2,086 buildings.

Manipulation The main challenge was how to benefit from direct 3D interaction and avoid jitter or tracking inaccuracies without imposing restrictions on user interaction. For all manipulation tasks, extremely small hand movements were compensated similar to the PRISM [2] technique. Furthermore, we split the manipulation of transmitter sites into two subtasks: changing site location and changing antenna height. This effectively reduced the 3D manipulation task into a two-dimensional (changing location) and a one-dimensional task (changing height). Though this might seem like a restriction at first, it turned out to be much more precise and would naturally stick to the planning process: first find out where your antenna site should be located and then adjust the height of the antenna tower for maximum coverage and minimum interference.

Sketching a new building on the satellite image also suffered heavily from jitter in hand movements. We assisted the creation process by depicting discrete grid points over the image and building edges would snap onto the grid. A new building is created by first entering the edit mode, then one button adds the current grid point to the building outline whereas another button allowed to undo the last operation. A building is finished by placing the last building corner in the vicinity of the first corner. Building walls are extruded to generate a 3D model and visualization and simulation databases are updated accordingly.

System Control We needed an interface that would make the control of visualization and network parameters accessible directly from within the VE. Common approaches are the use of 3D menus or small hand-held computer devices (e.g. Tablet-PC). However, we felt not comfortable with either of them. The 3D menus would clutter the visual field of the VE and are sometimes hard to use due to jitter. An additional hand-held device would encumber the user and would require him to switch context between immersive 3D and a 2D screen. We tried to combine both approaches by tracking the non-dominant hand and attached a virtual screen to the hand position as soon as the user raises the hand above his waist. This idea is inspired from the use of a mobile smart phone that the user takes out of his pocket. The virtual screen depicts icons for controlling the application which can be selected by pointing and clicking with the device in the dominant hand. Figure 6 shows a picture of a user in the VE with his left hand raised and selecting a network statistic with his right hand.

5 CONCLUSIONS

We presented a VR prototype application for planning support of wireless communication networks that combines the real-time simulation with an interactive manipulation of the propagation environment. We identified three major interaction tasks: Adjustment of visualization and simulation parameters, setup and modification of transmitter sites and the manipulation of the city model which provides the computational basis for the simulation algorithm. All computational intensive tasks were performed on the GPU to achieve a real-time response directly in the VE; for a network of eight transmitter sites, the frame rate of our application did not drop below 40 frames per second.

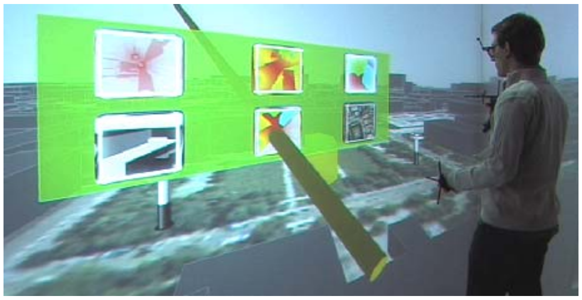

Figure 6: User stands in the VE with a tracking device attached to his non-dominant hand and a tracked pointing device in his dominant hand. His left hand is slightly raised to look at a panel for system control. The selection of the panel icon is done with the pointing device in his right hand. Upon completion the panel is hidden by lowering the non-dominant hand.

We performed a primary questioning of two domain experts (communication theory) regarding usability and features. Both found the potential of our application interesting. When asked for the worst feature they stated that they would like to have more control over the propagation simulation settings (e.g. antenna radiation pattern). They liked the overview in the VE, especially the possibility to create missing buildings and stated that it would be great for debugging propagation algorithms. They assessed the overall application as good for finding weak spots in an initial planning phase. We hope that by coupling simulation and manipulation with a VR interface we contributed in the understanding of the underlying mathematical models and algorithms required for planning and optimizing radio networks.

REFERENCES

[1] K. Bhavaraju. Interactive virtual reality simulation for nanoparticle manipulation and nanoassembly using optical tweezers. In *IEEE Virtual Reality Conference*, pages 251–252, 2009.

[2] S. Frees, G. D. Kessler, and E. Kay. PRISM interaction for enhancing control in immersive virtual environments. *ACM Trans. Comput.-Hum. Interact.*, 14(1):2, 2007.

[3] N. Hagbi, R. Grasset, O. Bergig, M. Billinghurst, and J. El-Sana. In-place sketching for content authoring in augmented reality games. In *IEEE Virtual Reality Conference*, pages 91–94, 2010.

[4] M. Hata. Empirical formula for propagation loss in land mobile radio services. *IEEE Trans. Veh. Technol.*, 29:317–325, 1980.

[5] F. Ikegami, S. Yoshida, T. Takeuchi, and M. Umehira. Propagation factors controlling mean field strength on urban streets. *IEEE Trans. Antennas Propag.*, 32:822–829, 1984.

[6] J. T. Kajiya. The rendering equation. In *SIGGRAPH ’86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 143–150. ACM Press, 1986.

[7] C. Poullis and S. You. Automatic creation of massive virtual cities. In *IEEE Virtual Reality Conference*, pages 199–202, 2009.

[8] T. S. Rappaport. *Wireless Communications: Principles and Practice*. Prentice-Hall, Inc., 1995.

[9] T. Rick and T. Kuhlen. Accelerating radio wave propagation algorithms by implementation on graphics hardware. In *Wave Propagation in Materials for Modern Applications*, pages 103–122. 2010.

[10] A. Schmitz, T. Rick, T. Karolski, T. Kuhlen, and L. Kobbelt. Efficient rasterization for outdoor radio wave propagation. *IEEE Trans. Vis. Comput. Graphics*, 17:159–170, 2011.

[11] J. van den Berg, J. Sewall, M. Lin, and D. Manocha. Virtualized traffic: Reconstructing traffic flows from discrete spatio-temporal data. In *IEEE Virtual Reality Conference*, pages 183–190, 2009.

[12] T. Whitted. An improved illumination model for shaded display. *Commun. ACM*, 23(6):343–349, 1980.