

Fast Ray-Tracing for Field Strength Prediction in Cellular Mobile Network Planning

Kurt Tutschku and Kenji Leibnitz

Institute of Computer Science, University of Würzburg

Am Hubland, D-97074 Würzburg, Germany

Tel.: +49-931-8885513, Fax: +49-931-8884601

E-mail: [tutschku|leibnitz]@informatik.uni-wuerzburg.de

Abstract: In this paper we present a fast ray-tracing technique with scalable approximation accuracy for field strength prediction in cellular mobile network planning. Automatic network design methods, like the *Adaptive Base Station Positioning Algorithm (ABPA)*, [2, 3], perform a huge number of field strength estimations and require therefore a fast and accurate approximation. However, common prediction techniques either give only rough estimations or are too complex for fast evaluation, cf. [7]. The proposed new ray-tracing technique obtains its speed-up by taking advantage of the topological information inherent in the used triangulation data structure of the investigated terrain. By that, it is possible to apply simple mathematics and algorithms to trace individual rays. The applicability of the fast ray-tracing technique is demonstrated for both a single transmitter scenario and in conjunction with ABPA.

I. Introduction

The fast and careful planning of modern cellular mobile communication networks is an essential criterion for their commercial success. With the application of fast design methods, a network operator can reduce the planning costs and shorten the time of putting the system to operation. Additionally, careful network planning can keep the investments into required hardware on a minimal level while obtaining a high quality of service. The planning procedures for present mobile communication systems try to obey these essentials, cf. [4], but fulfill the objectives only partially. For example, commercial available mobile planning tools like *GRAND*, [6], help the system engineer to evaluate the radio coverage but they do not give a hint

where to place the transmitters. The engineer has to move them around in the virtual planning scenario of the tool until he finds a good configuration. Moreover, nowadays applied planning methods concentrate only on radio engineering aspects, like radio coverage and interference. They neglect the user behavior and the related teletraffic issues.

II. Integrated mobile planning

The shortcomings mentioned above lead to the need for an unsupervised and integrated design method. The approach is summarized in Figure 1. The core component is the *Adaptive Base Station Positioning Algorithm (ABPA)* and was first presented in [2]. The algorithm autonomously locates base stations in the investigated terrain. ABPA considers four different system engineering aspects: the radio wave propagation, the user behavior, the teletraffic allocation and the system architecture. So far, only the first two aspects are implemented and the latter two are in planning. The spatial behavior of mobile users is represented within ABPA by a distribution of discrete cluster points, cf. [3]. For obtaining an optimal mobile service supply the algorithm uses *competing base stations* which try to cover as many points as possible. The algorithm shifts around the transmitter in the virtual scenario until a considerable configuration is reached. The movement of the base station is conducted by a modified simulated annealing procedure. Since this is an iterative algorithm, it requires the evaluation of the radio coverage in each adaption step. However, the prediction methods used for cellular network design are either too complex and therefore computationally intensive [7] or not accurate enough for automatic planning [9]. Because of this fact, ABPA requires a fast and arbitrary accurate field strength prediction method, i.e. the engineer must be able to adjust the speed by choosing a certain accuracy.

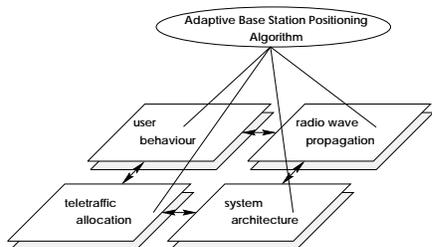


Figure 1: Integrated Network Planning Approach

III. Radio wave propagation models

The accuracy of the field strength prediction depends particularly on the applied *radio wave propagation model*. A lot of different propagation models have been proposed for

various environments, like *urban* or *rural* terrain. An efficient algorithm for the prediction has to regard the specific features of these models. In this section, we will not provide physical formulae, but an overview of some important field strength prediction methods and their capability. For a more detailed description of these methods, the interested reader is conferred to [1, 7, 9].

Free Space Propagation assumes that no obstacle interferes the radio wave propagation. Therefore the field strength depends only on the distance between the transmitter and the receiver. The approximation can be computed rapidly from one equation, but it estimates the actual value field only very roughly.

A more realistic propagation model was presented by *Okumura* [8]. It is essentially based on computing the free space path loss and then adding or subtracting correction factors to account for the different morphological features of urban and rural terrain as well as the antenna height. The model was made applicable by *Hata*, cf. [5]. He published empirical values for the correction factors. Since the *Okumura/Hata* model is an extension of the *Free Space Propagation* model, it is quite simple to compute, but limited in accuracy.

Complex propagation models like the methods described by *Kürner* et al. [7] are based on *ray-optical* approaches. The wave interactions are modeled in detail and described by the *uniform theory of diffraction* and *physical optics*. This approach is very accurate but it requires considerable computational effort, since conventional ray-tracing is applied.

To sum up, there are many field strength prediction methods with different complexity, but a method with arbitrary scalable speed and accuracy is not available yet.

IV. Fast Ray-Tracing

The field strength prediction method which we consider is based on the capability of *ray-tracing* to model optical effects very accurately. Ray-tracing was introduced by Whitted, [10], as a technique for creating photo-realistic pictures. Since ray-tracing for image generation and ray-tracing for field strength prediction have the same core idea but differ essentially, we first explain the image generation method and thereafter the field strength prediction approach.

A. Ray-Tracing for photo-realistic image generation

To generate high-quality pictures, photo-realistic ray-tracers consider the sum of all details within the picture, such as surface textures, shadows, reflections, etc. Therefore, it is necessary to describe the content of the environment in a computer processable format. Usually, this is achieved by specifying all objects as certain graphical primitives, e.g. rectangles, cubes or spheres. The image is generated with this information in the rendering step, see

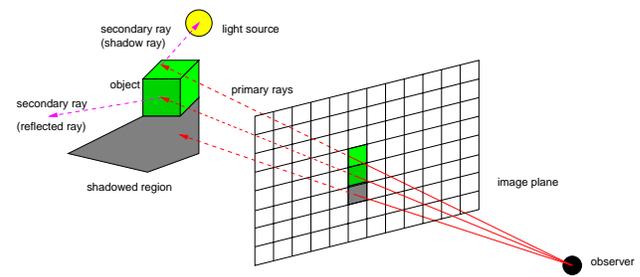


Figure 2: Schematics of ray tracing

Figure 2. This processing step assumes an observer which is looking at the modeled scenery from behind the image plane. The plane or window is segmented by a regular grid where each segment corresponds to a *pixel* of the generated picture. A ray originating at the observer is cast through the center of each pixel and traced until its first intersection with an object. The pixel color is set to the ambient color of this object. The color of the object is determined by recursively tracing and emitting *secondary* rays until a certain recursion depth is reached. By using *secondary* rays, this method can simulate optical effects like shadows, reflection, and refraction.

Unfortunately, *optical* ray-tracers are slow because the simulation of these effects requires high computational effort. Moreover, they generate only a two-dimensional picture of the visible objects of the three-dimensional scenery. However, for evaluating the radio coverage, the electric field strength on every surface element in the scenery has to be computed.

B. Ray-Tracing for field strength prediction

Ray-tracing for field strength prediction is based on the *light house* idea, see Figure 3. The ray-tracer emits ra-

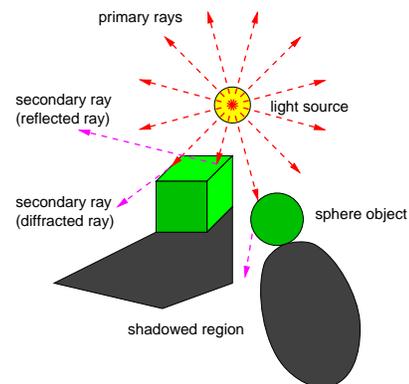


Figure 3: Ray-launching

dio beams from the transmitter in all directions and *illuminates* the scenery, i.e. the *supplying area*. At an intersection the appropriate physical laws for reflection, diffraction, and transmission of radio waves are applied. Because of this direct ray-tracing, often also referred to as *ray-launching*, the field strength is obtained on each surface element. Furthermore, this approach includes the

possibility of modeling multipath propagation. However, this method is still very slow, due to the lack of knowledge about the structure of the scenery.

C. Fast Ray-Tracing

Our proposed *fast ray-tracing* technique is based on the *ray-launching* approach and obtains a considerable speed up by using a sophisticated three-dimensional digital terrain model. The terrain is composed of triangles that interpolate the surface between the equidistant altitude samples, see Figure 4. By exploiting the topographical infor-

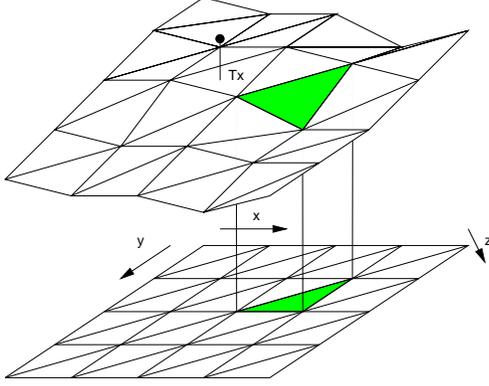


Figure 4: Three-dimensional terrain model

mation inherent in this data structure, the *fast ray-tracing* method avoids unnecessary intersection tests of the emitted ray with the triangles.

The projection of the triangles from the three-dimensional terrain model onto a two-dimensional (x, y) -plane takes on regular shapes (compare Figure 4). This facilitates a favorable numbering with three ordinates (x, y, z) , where x and y are the ordinates of usual two orthogonal axes of the projection plane and z is the index of the diagonals. In the (x, y) -projection an emitted beam traverses a sequence of adjacent triangles, see Figure 5. The points that lie on intersections of the projection of the ray and the projection of the triangle edges are called

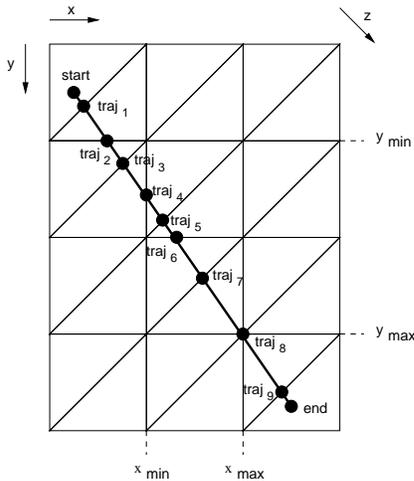


Figure 5: Trajectory points of a ray

trajectory points. They are stored as a sorted list. Once the list is obtained, an intersection of a ray with a surface element can be detected by comparing the height values Δ_i and Δ_{i+1} of two successive trajectory points i and $i+1$, see Figure 6. The height Δ_i of a trajectory point i is the alti-

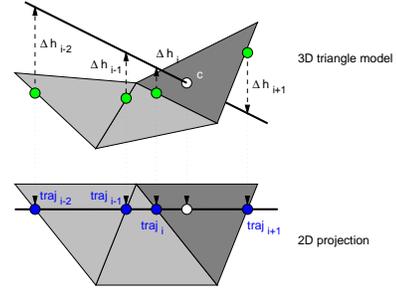


Figure 6: Calculation of a three-dimensional intersection point

tude of the ray at this point relative to the surface, marked by the arrows in Figure 6. A change of sign of these height values indicates an incident ray. Once a point of intersection is determined, the electric field strength on this surface element can be computed by modeling the proper physical effects [7].

The list of trajectory points can be obtained by applying Algorithm 1. The procedure *Generate Trajectory*

Algorithm 1 (Generate Trajectory)

variables:

x_{min}, x_{max}	range of x values
y_{min}, y_{max}	range of y values
line	traced ray
x, y, z	loop variables
num_p, num_q, num_d	index variables
p, q, d	indexed lists of points
traj	resulting list of trajectory points

algorithm:

```

1 funct gen_trajectory(start, end)  $\equiv$ 
2   begin
3      $x_{min} \leftarrow \text{Ceil}(\text{start}.x); \quad x_{max} \leftarrow \text{Floor}(\text{end}.x);$ 
4      $y_{min} \leftarrow \text{Ceil}(\text{start}.y); \quad y_{max} \leftarrow \text{Floor}(\text{end}.y);$ 
5     line  $\leftarrow \text{new\_line}(\text{start}, \text{end});$ 
6     for  $x \leftarrow x_{min}$  to  $x_{max}$  do
7        $p_{num_p} \leftarrow \text{point\_on\_line}(\text{line}, x);$ 
8        $num_p \leftarrow num_p + 1;$ 
9     od
10    for  $y \leftarrow y_{min}$  to  $y_{max}$  do
11       $q_{num_q} \leftarrow \text{point\_on\_line}(\text{line}, y);$ 
12       $num_q \leftarrow num_q + 1;$ 
13    od
14    for  $z \leftarrow x_{min} + y_{min}$  to  $x_{max} + y_{max}$  do
15       $d_{num_d} \leftarrow \text{point\_on\_line}(\text{line}, z);$ 
16       $num_d \leftarrow num_d + 1;$ 
17    od
18    traj  $\leftarrow \text{merge\_lists}(p, q, d)$ 
19    return traj.
20  end

```

intersects the projected ray with the parallels of the x -axis which have integer y -coordinates and stores the trajectory points. Then, the parallels of the y -axis and the z -axis are treated analogously. It is important to mention at this point, that intersections in a two-dimensional space are much easier to compute than in a three-dimensional environment. In Algorithm 1, `point_on_line()` calculates the point on the line for a certain integer x , y , or z coordinate. The function `new_line()` creates a new ray from `start` to `end`. The procedure `merge_lists()` consists of a `merge-sort` routine and combines the three lists of axes trajectory points.

A major drawback of common ray-launching methods is the appropriate setting of the stepwidth for the altitude and the latitude angle. For example, if the altitude angle Δ_{step} was selected too large, not every surface element is reached. If this angle was too small, too many superfluous beams are emitted, see Figure 7. This problem disappears

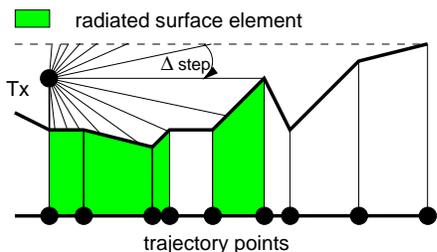


Figure 7: Fixed altitude angle stepwidth

if rays are aimed at the midpoints between every two successive trajectory points, see Figure 8.

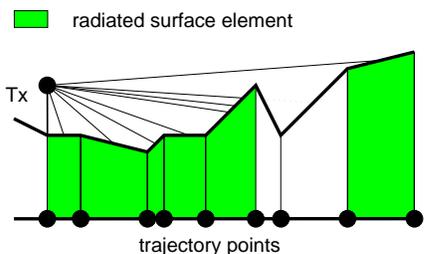


Figure 8: Variable altitude angle stepwidth

The complete *fast ray-tracing* algorithm is shown in Algorithm 2. The algorithm directs trajectory lines with latitude angle φ and computes their trajectory points. Then, the algorithm emits rays to the midpoints between two trajectory points, cf. Figure 8. An intersection of this ray with a surface element will occur in any case. The functions `collision()` computes the exact intersection point on the surface element. At this location the field strength is evaluated by the function `fsp()`, which models the proper physical effects, cf. [7]. The predicted field strength value is stored for each the surface element. Additionally, the intersection point serves as a starting point for secondary rays. The new beam is generated by function `create_secondary_ray()` and is recursively traced until a given depth is reached. The function `find_triangle()` relates the intersection point with its encompassing triangle. The function `max_range()` limits

Algorithm 2 (Fast Ray Tracing Algorithm)

variables:

<i>radius</i>	current maximum range of Tx
<i>dist</i>	Eucl. distance between Tx and sensors
<i>fs</i>	temp. variable for field strength calculation
<i>traj</i>	list of trajectory points
<i>c</i>	intersection point of ray with triangles
<i>t_i</i>	triangle of three dimensional terrain model
$\Delta\varphi, \varphi$	horizontal stepping and tracing angle

algorithm:

```

1 proc fast_ray_tracing(Tx)  $\equiv$ 
2   begin
3     radius  $\leftarrow$  max_range(Tx);
4     for all triangles  $t_i$  do
5        $t_i$ .field_strength  $\leftarrow$  0;
6     od
7     for  $\varphi := 0$  to  $2 \cdot \Pi$  step  $\Delta\varphi$  do
8       targettraj  $\leftarrow$  (Tx) + (sin( $\varphi$ ), cos( $\varphi$ ), 0);
9       traj  $\leftarrow$  gen_trajectory(Tx, targettraj);
10      for  $i := 1$  to ||traj|| do
11        targetdeci  $\leftarrow$   $\frac{1}{2} \cdot (traj_{j-1} + traj_j)$ ;
12        if distance(Tx, targetdeci) < radius
13          then
14            line  $\leftarrow$  new_line(Tx, targetdeci);
15             $\Delta h_{old}$   $\leftarrow$  antenna_height(Tx);
16            for  $j := 1$  to  $i$  do
17               $\Delta h_{new} \leftarrow h_j^{ray} - h_j^{triangle}$ ;
18              if sign( $\Delta h_{new}$ ) < 0
19                then
20                   $c \leftarrow$  collision
21                    (traj $j-1$ , traj $j$ );
22                fi
23                 $\Delta h_{old} \leftarrow \Delta h_{new}$ ;
24            od
25            dist  $\leftarrow$  distance(Tx, c);
26            fs  $\leftarrow$  fsp(Tx, dist);
27             $t_i \leftarrow$  find_triangle(c);
28             $t_i$ .field_strength  $\leftarrow$  fs;
29            create_secondary_ray(c);
30          fi
31        od
32      od
33    for all sensors  $s_i$  do
34       $t_i \leftarrow$  find_triangle(c);
35       $s_i$ .field_strength  $\leftarrow$   $t_i$ .field_strength;
36    od
37  end

```

the length of the trajectory to the maximum receiving distance and the function `distance()` computes the distance between two points in the three-dimensional model.

V. Results for a single transmitter

We applied our fast ray-tracing method on real terrain data of an area northwest of Würzburg, Germany. The extension of the area is 10km \times 10km. Figure 9 shows the

bird's eye perspective of the supplying area for a single transmitter located in the center. Points marked by 'x' represent cluster points (of the spatial user distribution, cf. section II.), which receive a field strength above a certain threshold. The points marked by '+' are cluster points where radio coverage is not given. The shape of

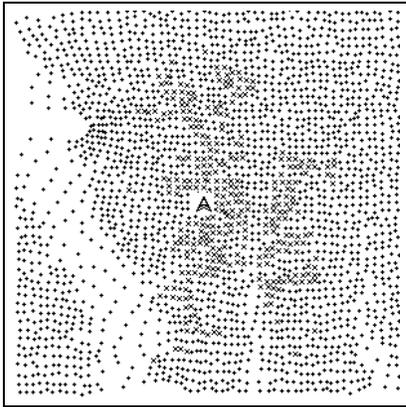


Figure 9: Supplying area of a single transmitter the supplying area has the expected “lacunarity” and is corroborated by results from real measurements.

VI. Applying Fast Ray-Tracing for locating multiple base stations

Figure 10 shows the final distribution of six base stations using ABPA in conjunction with fast ray-tracing. The

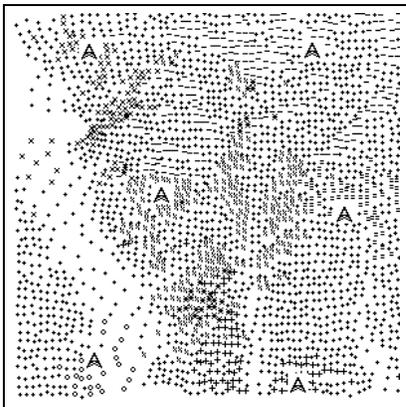


Figure 10: Result of fast ray-tracing with ABPA transmitters are marked by the radio towers and the different supplying areas for the individual transmitters are tagged by various symbols. The result was obtained on a SUN SparcStation 20/612 in less than two hours. The final placement of the transmitter was plausible to human experts.

VII. Conclusion

In this paper we presented a *fast ray-tracing* technique for field strength prediction. The proposed method obtains

its speed-up by taking advantage of the topological information inherent in the used triangulation data structure. The appealing feature is its scalability. By varying the triangle sizes, the stepwidth of the latitude angle, or the considered physical phenomena, it is possible to obtain the desired prediction accuracy or the intended speed. This *fast ray-tracing* methods permits the application of automatic cellular mobile network planning tools.

Acknowledgment: The authors like to thank Prof. P. Tran-Gia for supporting their work.

References

- [1] J. B. Andersen, T. S. Rappaport, and S. Yoshida. Propagation measurements and models for wireless communications channels. *IEEE Communications Magazine*, 33(1):42–49, 1995.
- [2] T. Fritsch and S. Hanshans. An integrated approach to cellular mobile communication planning using traffic data prestructured by a self-organizing feature map. In *Proceedings of the 1993 International Conference on Neural Networks*, pages 822D–822I. IEEE, IEEE Service Center, März 28 – April 1 1993.
- [3] T. Fritsch, K. Tutschku, and K. Leibnitz. Field strength prediction by ray-tracing for adaptive base station positioning in mobile communication networks. In *Proceedings of the 2nd ITG Conference on Mobile Communication '95, Neu Ulm*. VDE, Sep. 26 – Sep. 28 1995.
- [4] A. Gamst, E.-G. Zinn, R. Beck, and R. Simon. Cellular radio network planning. *IEEE Aerospace and Electronic Systems Magazine*, 1:8–11, Februar 1986.
- [5] M. Hata. Empirical formula for propagation loss in land mobile radio services. *IEEE Transactions on Vehicular Technology*, VT-29(3):317–325, 1980.
- [6] M. Krüger and R. Beck. Grand - Ein Programmsystem zur Funknetzplanung. *Philips (PKI) - Technische Mitteilungen* 2, 2:7–12, November 1990.
- [7] T. Kürner, D. Cichon, and W. Wiesbeck. Concepts and results for 3D digital terrain-base wave propagation models: an overview. *IEEE Journal on Selected Areas in Communications*, 11(7):1002–1012, September 1993.
- [8] Y. Okumura, E. Ohmori, T. Kawano, and K. Fukuda. Fieldstrength and its variability in VHF and UHF land mobile radio service. *Review of the Electrical Communication Laboratory*, 16(9-10):825–873, 1968.
- [9] D. Parsons. *The Mobile Radio Propagation Channel*. Pentech Press, London, 1992.
- [10] T. Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 23(6):343–349, 1980.