Progressive and Approximate Techniques in Ray-Tracing Based Radio Wave Propagation Prediction Models*

Zhongqiang Chen
Department of Computer & Information Science
Polytechnic University
Brooklyn, NY 11201
zchen@milos.poly.edu

Henry L. Bertoni
Department of Electrical & Computer Engineering
Polytechnic University
Brooklyn, NY 11201
hbertoni@duke.poly.edu

Alex Delis †
Department of Informatics & Telecommunications
University of Athens
Athens, 15771, Greece
ad@di.uoa.gr

June 2003

Abstract

Progressive and approximate techniques are proposed here for ray-tracing systems used to predict radio propagation. In a progressive prediction system, intermediate prediction results are fed back to users continuously. As more raypaths are processed, the accuracy of prediction results improves progressively. We consider how to construct a progressive system that satisfies the requirements of continuous observability and controllability as well as faithfulness and fairness. Adding a workload estimator to such a progressive prediction system allows termination of the computation when a desired accuracy (mean and standard deviation of the error) is achieved *without knowing the final result* that would be obtained if the prediction system runs to completion.

The sample generator is at the core of the progressive prediction system and serves to cluster and prioritize raypaths according to their expected contributions to prediction results. Two types of progressive approaches, source-group-raypath-permute and raypath-interleave, are proposed. The workload estimator determines the number of raypaths to be processed to achieve the specified requirement on prediction accuracy. Two approximate models are described that adjust the workload dynamically during the prediction process. Our experiments show that the proposed progressive and approximate methods provide flexible mechanisms to trade prediction accuracy for prediction time in a relatively fine granularity.

Indexing Terms: progressive and approximate prediction model, observability and controllability, faithfulness and fairness of prediction results, sample generator and workload estimator, prediction accuracy and prediction error

*This work was partially supported by NSF under grant IRI-9733642, the U.S. Department of Commerce under grant 4000186, and the New York Center for Advanced Technology in Telecommunications (CATT).

†Work while at Polytechnic University.

1 Introduction

Ray-tracing based radio wave propagation prediction models have become popular and important in modern wireless system designs [16, 28, 13, 17, 25]. As wireless systems become more complex, and requirements for quality of services (QOS) delivered by wireless networks become diverse and sophisticated [2, 16], there is a growing need for versatile features of propagation prediction models. For instance, during the design of wireless networks, it may be desirable to know if the received signal strengths at some locations are above a threshold. If prediction models feed back prediction results continuously, then designers can get the required information quickly and terminate the prediction process before completion. Currently, prediction models operate in batch mode [21, 17] so that designers need to submit specifications and requirements to the prediction model before prediction procedure begins. Based on these parameters (specifications and requirements), the prediction model will process a large volume of ray trace data eventually returning complete prediction results. While the program is running, designers have no control on the prediction procedure.

Progressive prediction models are attractive as they can provide successive refinements and give quick feedback during the prediction process. They also give users flexible control over the whole prediction process and convenient mechanisms to trade prediction accuracy for prediction time. Users can change parameters and therefore the prediction process on the fly based on changing or unquantifiable human factors, such as time constraints, accuracy needs, and priority of other tasks. Since the user can observe the ongoing process, there is no need to specify these factors in advance. Traditional prediction models are usually optimized to finish the entire prediction task with minimum processing time, but there is no requirement for or constraint on intermediate prediction results. If the prediction procedure terminates prematurely, its outputs may be bias and misleading.

Progressive and approximate techniques have been applied in many fields, such as engineering mathematics [15], dynamical systems [4], fractal geometry [18, 1], database systems [23, 11], information retrieval systems [14], computer graphics [12, 26, 6], and data compression and visualization [7, 8, 5]. In engineering mathematics [15], progressive and approximate techniques are employed to determine roots of an equation, such as Newton's recursive method, regula falsi method, and iterative method. Similar techniques are also used to find approximation for eigenvalues of matrices. In computer graphics, progressive and approximate techniques are integrated into the ray-tracing systems for time-critical or interactive applications [10, 9, 24, 27]. An approximate and low-quality image is delivered to the user by generating and tracing only a small set of samples. As more samples are generated and processed, the quality of the resulting image will improve. Such interactive and progressive ray-tracing systems can be found in [22, 19, 20].

While progressive and approximation techniques have been used for some time, they are not currently used in computer software systems for predicting UHF radio wave propagation in cities, which are based on geometrical optics and the geometrical theory of diffraction. This approach to propagation prediction can be thought of as tracing rays radiated by a base station transmitter as they intersect buildings where they are reflected, or illuminate building corners where they are diffracted. The building corners act as secondary sources of rays that can again be traced as they interact with buildings. It is usual in ray programs to find the rays from a single base station that illuminate many receiver locations along the streets (one-to-many). Ray programs usually account for rays that undergo five or more reflections and one or two diffractions at vertical building edges. For the study reported here, we have carried out simulations using a 2D ray tracing package, which is appropriate for low base station antennas in a high rise building environment [2]. However, the ideas reported here can be applied to codes that trace rays over and around buildings in 3D [13, 17, 2].

Generally speaking, progressive and approximate prediction models for propagation prediction should satisfy the following requirements: 1) the prediction procedure should be continuously observable during the entire prediction process; 2) the prediction procedure should let the user change parameters to control the prediction procedure at any time; 3) intermediate prediction results should be faithful approximations in that it is the best (tightest) approximation of the actual final prediction results under given time constraints; 4) intermediate and final prediction results should be fair and unbiased for all receivers are updated at the same rate; 5) mechanisms should be provided to trade prediction accuracy for prediction time; and 6) the prediction model should allow integration of other techniques into the model to further improve its performance.

Although there are differences in the progressive and approximate techniques used in different fields, they all share the same fundamental structure: 1) the original computation is decomposed into many small tasks and are prioritized according to their (estimated) contributions to the final results. The high-priority tasks are grouped into base sets, while the low-priority tasks are clustered into enhancement (refinement) sets; 2) the result generated by processing all tasks in the base sets delivers a rough approximation to the real final result; 3) as more tasks in the enhancement sets are processed, the quality of the result is continuously improved. By monitoring the output of the process as it progresses, it is possible to terminate the computations knowing that desired accuracy conditions (mean and standard deviation of the error) will be achieved without knowing the final result.

The remaining of the paper is organized as follows. Section 2 presents two theorems for idealized conditions that indicate approaches to satisfying faithfulness and fairness requirements. Mechanism to trade prediction accuracy for prediction time is suggested by the theorem developed in Section 2 for idealized conditions. Based on this analysis, two types of progressive prediction models, source-group-raypath-permute and raypath-interleave, are proposed in Section 3. In Section 4, two approximate prediction models, source-group-raypath-permute approximate model and raypath-interleave approximate model, are proposed to provide flexible mechanisms to trade prediction accuracy for prediction time. All these methods (algorithms) are implemented and their performance are evaluated by using real Geometric Information System (GIS) databases in Section 5. We also compare and analyze advantages and disadvantages of different prediction models. Conclusion can be found in Section 6.

2 Considerations on Progressive and Approximate Predictions

Faithfulness requires that the prediction system always delivers the best prediction results (the least mean of prediction errors for all transmitters and receivers) no matter when the user terminates the prediction procedure. If we assume that the processing time for each raypath is the same, and the maximum number of raypaths which can be traced is t under the given constraints on time and computation resources, then, the prediction model should trace the t raypaths that minimize the prediction errors. The most important design index in a wireless system is the received signal strengths for all receivers. We therefore use the mean of the received signal strengths at all receivers as the faithfulness metric to evaluate the quality of predictions. Assuming that the processing time for each raypaths is the same, the following theorem (Theorem 1) guarantees that the faithfulness requirement is always satisfied no matter when the prediction procedure is terminated.

Theorem 1 (Faithfulness) *Suppose that the processing time for each raypath is the same, and prediction* *errors are described by the difference between the generated received signal strengths (in dB) and final received signal strengths (in dB) for all receivers. The contribution ratio of a raypath is defined as the ratio of its contribution (in Watt) to a receiver R and the sum of contributions (in Watt) of all unprocessed raypaths for R. If raypaths are always traced according to their nonincreasing contribution ratios to the received signal strengths no matter which receivers they illuminate, then the mean prediction error (in dB) is minimized, however, its variance (in dB^2) may fluctuate (increase or decrease).*

The proof of Theorem 1 is similar to that given for Theorem 2, and can be found in [3]. To apply Theorem 1, the prediction model needs to be aware of the contributions of all the raypaths to the prediction results and sort them in nonincreasing order before the raypaths in discussion are actually traced. For this theorem, the prediction accuracy is characterized only by the mean prediction error and not by its variance. Obviously, it is unrealistic to assume that the raypaths can be sorted in advance. However, methods to approximate this sorting process are discussed in Section 4.

For wireless system design, it is also desirable that the distribution of prediction errors should spread evenly among all receivers to minimize the variance of the errors. In this case, the processing strategy based on Theorem 1 is not enough since it may concentrate errors on some receivers. Fairness requires that the mean and variance of prediction errors should both decrease monotonically during the whole prediction process. Theorem 2 addresses this requirement.

Theorem 2 (Fairness) *Suppose that there are n receivers R_i ($i = 0, 1, \ldots, n-1$ and $n > 1$), and R_i is illuminated by m_i raypaths, $r_{i,j}$, $j = 0, 1, \ldots, m_i - 1$. The contribution to the received power by $r_{i,j}$ is $P_{i,j}$ (in $Watt$), and all $P_{i,j}$ for R_i form a geometric progression, that is, $P_{i,j+1} = r_i P_{i,j}$ ($0 < r_i < 1$). By clustering the g-th raypath for all receivers together to form a group $G(g)$, we get a sequence of groups of raypaths, $G(g)$, ($g = 0, 1, \ldots$). If we trace $G(g)$ group by group in that order, then a) the mean (in dB) of prediction errors decreases monotonically with the number of processed groups; b) the variance (in dB^2) of prediction errors decreases monotonically with the number of processed groups if it is further assumed that $m_i = \infty$ for $i = 0, 1, \ldots, n$.*

The proof of Theorem 2 can be found in Appendix A. Similar to Theorem 1, application of Theorem2 also requires that the contribution to final prediction results by each raypath should be known before it is traced. The condition that contributions of raypaths illuminating the same receiver satisfy the geometric progression relation will, at best, be only approximately true. In moderately complex environments, there will be an infinite number of raypaths illuminating each receiver. In reality, most raypaths may contribute so little to the final prediction results that they can be safely dropped off.

2.1 Tradeoff between Prediction Accuracy and Prediction Time

An important requirement for a progressive prediction model is that it should provide a flexible and convenient mechanism with users to trade prediction accuracy for prediction time. For example, network designers only want to find general trends or gross patterns of the prediction results, instead of the exact results with very high precision. This is in fact an approximation problem that may be phrased two ways. One is, given a time constraint, what is the best prediction accuracy the model can deliver. The other is, given the worst tolerable prediction error, what is the minimum processing time. An approximate prediction model based on Theorems 1 and 2 can solve the first problem as long as the conditions in the Theorems are satisfied.

To solve the second problem, we define the prediction error as the difference between the final prediction results and the prediction results generated so far (both in dB).

Note that the ray tracing procedures simplify the buildings and the physical processes involved. As a result, there will always be differences between the measurements and final prediction using all of the rays. Because of these simplifications, models run for receivers distributed over kilometer distances exhibit differences between prediction and measurements (in dB) that typically have an average of about 1 dB and a standard deviation of about 8 dB [17, 2]. Now suppose that we stop the prediction process at the point when the difference between the interrupted predictions and the final predictions using all the rays has an average of 1 dB or less, and a standard deviation of 4 dB or less. In this case, the errors of the interrupted predictions as compared to measurements will have an average less than 3 dB and standard deviation approximately 9 dB (i.e., $\sqrt{4^2 + 8^2}$) [3]. This example suggests the conditions under which we may interrupt the prediction without significantly impacting the achievable accuracy as compared to measurements. In using the definition of prediction error as the difference between the final predictions and the intermediate results, we would need to trace all the raypaths and finish the entire prediction process, which contradicts the main purpose of the approximation procedure. Theorem 3 is designed to solve this dilemma.

Theorem 3 (Workload) *Suppose that there are n receivers R_i, $i = 0, 1, \ldots, n-1$, each of them is illuminated by an infinite number of raypaths. The contribution to the received power of receiver R_i by raypath $r_{i,j}$ is $P_{i,j}$ (in $Watt$, and $P_{i,j} \geq 0$), and all $P_{i,j}$ for R_i form a geometric progression, that is, $P_{i,j+1} = r_i P_{i,j}$ ($0 < r_i < 1$ for $j = 0, 1, 2, \ldots$). Raypaths $r_{i,j}$ are clustered into a group $G(j)$, ($i = 0, 1, 2, \ldots, n-1$ and $j = 0, 1, \ldots$). Groups $G(j)$ are traced group by group in that order. Given the tolerable mean and variance of prediction errors as μ (in dB) and σ^2 (in dB^2), then after tracing k groups of raypaths, the mean $\mu(k)$ and variance $\sigma^2(k)$ of prediction errors generated by the prediction model will satisfy $\mu(k) \leq \mu$ and $\sigma^2(k) \leq \sigma^2$ as long as k satisfies:*

$$k = \frac{\log\left[1 - \frac{1}{10^{\max(\mu,\sigma)/10}}\right]}{\log(r_{max})}; \qquad r_{max} = \max_{i=0}^{n-1} r_i$$

The proof of Theorem 3 can be found in Appendix B. Theorem 3 shows that prediction errors can be found even though the final prediction results are unknown, provided that all the conditions described in Theorem 3 are satisfied.

Even though the conditions (mentioned or implied) in Theorems 1, 2, and 3 cannot be satisfied in real systems, the following guidelines derived from these theorems can be used for designing progressive and approximate prediction models. First, trace raypaths according to their expected or estimated contributions to the final prediction results. This strategy helps to minimize the mean of prediction errors at any time during the whole prediction process. The contributions to the final prediction results by raypaths can be estimated by some heuristic methods. Next, when the variance of prediction errors is important, raypaths should be clustered and processed group by group. Raypaths in the same group should cover (illuminate) all or most of the receivers. Those groups contributing most to the final prediction results should be assigned higher priorities. Furthermore, the contributions of different groups should form a nondecreasing relation (e.g., geometric progression). The relationship among different raypaths can be exploited to cluster raypaths to approximate the above properties. Finally, when prediction accuracy is traded for prediction time, workload should be estimated to deliver prediction results as quickly as possible based on Theorem 2. To make the estimate of workload accurate, dynamic adjustment of workload is necessary.

2.2 Generic Progressive and Approximate Models

In the radio wave propagation prediction problem, if each raypath from real transmitters to receivers is considered as a sample, then the raypaths between all transmitters and receivers will form a huge sample space. Different samples contain different path components (reflections, diffractions), require different processing time, illuminate different set of receivers and building corners, and contribute differently to the final prediction results. Comparing with traditional prediction models, a progressive prediction model needs to cluster raypaths into different orders before processing them. To do this, it is convenient to divide the geometric data processing into two parts, a sample generator and a ray-tracing engine. The sample generator clusters raypaths into groups and assigns different priorities to them based on some criteria, then hands these groups to the ray-tracing engine according to their priorities. It is clear that at the core of a progressive prediction model is the sample generator [24]. The design of sample generators should be based on the guidelines derived from Theorems 1, 2, and 3, and also should take into account the characteristics of radio wave propagations. For example, we know that transmitters carry much higher energy than secondary transmitters (diffraction corners). Also, raypaths representing the first diffraction by a corner may carry much higher energy than those representing a second diffraction by a corner. Different raypaths may illuminate different sets of receivers, and some may illuminate a large set of receivers, while others may illuminate only a small or empty set of receivers. Finally, raypaths emitted by different source points that are well separated spatially may illuminate different sets of receivers.

Based on these observations as well as the guidelines derived from Theorems 1, 2, and 3, we design a generic sample generator for a progressive or approximate prediction model. All raypaths emitted from real transmitters are put into the high-priority base sets, while those raypaths generated by diffraction corners are put into the enhancement sets. Raypaths in the same set may be further partitioned into different priority groups based on other criteria. In Sections 3 and 4, we present heuristic methods for grouping rays, according to the guidelines suggested by Theorems 1, 2, and 3. Raypaths are handed to the ray-tracing engine in a group-by-group fashion. Based on the feedback from the ray-tracing engine, it is desirable to adjust the grouping criteria, regroup raypaths, and change priorities of different groups if necessary. When termination conditions are satisfied, proceed to electromagnetic indices computation part, and generate the final prediction results.

The design of the ray-tracing engine for a progressive prediction model is more complicated than a traditional prediction model since it needs to collect statistics information for the sample generator. Raypaths from the sample generator may not have any particular order. They may come from different source points, or from the same source point but with random order. To avoid the time-consuming operation of ray-wall-intersection check, the ray-tracing engine should use information accumulated in the processing of previous groups. The ray-tracing engine should organize this accumulated information efficiently for retrieval, update, and removal. Furthermore, the ray-tracing engine needs to interact with the electromagnetic indices computation part, which will generate the intermediate results, and display them through the user interface.

3 Progressive Prediction Models

The pincushion ray-tracing method [2] is used in our prediction models. For a 2D ray trace, the pincushion method launches rays for each source point (real transmitter or diffraction corner) at some small angular separation δ in the horizontal plane. The intersection of a ray with all the walls in the database is computed, and the one closest to the source point is chosen as the reflecting wall, since the other intersections are shadowed by the first wall. The resulting ray segment is then tested to see if it illuminates any of the receiver locations and building corners. Subsequently, a reflected ray starting at the point of intersection with the wall and traveling in the direction of specular reflection is traced to the next wall intersection. This reflected segment is tested to see if it illuminates receivers or building corners. The foregoing process is repeated until the ray passes out of the computational domain, or the number of reflections exceeds some preset threshold.

3.1 The Source-Group-Raypath-Permute (SGRP) Model

In a traditional prediction model, all real transmitters are processed sequentially, and all diffraction corners are processed in the order they are generated. For each source point (real transmitter or diffraction corner), rays are launched sequentially around the source point. It is clear that this tracing strategy does not satisfy the requirements for the progressive prediction models, especially the faithfulness and fairness of prediction results. To find a better grouping and tracing strategy, we analyze the raypath-launching procedure in more detail. When δ is very small (less than 0.01 radian, or $0.6°$), raypaths emitted from the source are likely to illuminate neighbors among a set of receivers. If we trace raypaths sequentially from a source point, their contributions to the final prediction results may concentrate on a small group of receivers. Also, if the maximum numbers of reflections and diffractions encountered by a raypath are relatively small, then a source point may illuminate only the nearby receivers and building corners. Therefore, if we trace all raypaths from the same source point together, they only affect a small portion of receivers. Raypaths emitted from different source points, close to each other, will illuminate very similar sets of receivers. On the contrary, if the source points are very far away from each other, then their raypaths will illuminate different sets of receivers. If these raypaths are put into the same group and processed together, then almost all receivers are affected which will help to reduce variance of prediction errors.

Based on the above analysis, we propose the source-group-raypath-permute progressive (SGRP) model, which overcomes the shortcomings of the traditional models. Its main ideas are as follows. 1) All real transmitters are partitioned into N portions according to their geometric locations (e.g., by grids or wedges). Each portion has n_i transmitters, $i = 1, 2, \ldots, N$. Then the jth group of transmitters is formed as $G_j = \cup_{k=1}^{N} T_{k,j}$, ($j = 1, 2, \ldots, \max_{i=1}^{N} n_i$), $T_{k,j}$ is the jth transmitter in the kth portion. All these groups are put into the high-priority base sets, and processed group by group based on their priorities. 2) To further improve the variance of prediction errors at any time of the prediction process, raypaths from the same transmitter are permuted before being traced. The permutation method can be random shuffle, hopping with different angular separations, or partitioning and round robin sampling. 3) Each order of diffraction corners (typically 1 or 2 orders) is processed similar to the real transmitters except that diffraction corners are partitioned into M portions with $M \gg N$ since the number of diffraction corners is much larger than the number of real transmitters. Each diffraction corner emits raypaths only within a wedge (typically $270°$), thereby its processing time is shorter. For large M, the processing time for each group is nearly the same, and the update rate for prediction results is almost the same during each time interval.

3.2 The Raypath-Interleave (RI) Models

It is reasonably straightforward to implement the SGRP model. However, when the building database is very large, or when the number of receiver locations is large, the performance of this model may not be satisfactory. The main reason is that source points are processed sequentially, and the processing time for each source point will be relatively long. This may not be desirable and can be improved based on the following observations. First, due to the limits on the maximum numbers of reflections and diffractions a raypath can undergo, a source point (real transmitter or diffraction corner) may illuminate a very small set of receivers. Even though all raypaths from a source point are processed, only a small portion of receivers can update their predicted received powers. This will affect the variance of prediction errors. Second, raypaths from different source points which are far away from each other may illuminate totally different sets of receivers.

It will be beneficial to interleave raypaths from different source points together and put them into same groups to cover a large set of receivers. For two nearby source points, their raypaths may still illuminate quite different sets of receivers if orientations of raypaths are quite different (for example, in the opposite directions). Therefore, interleaving raypaths from different source points is still a good clustering strategy to improve the mean and variance of prediction errors. The above observations and analysis lead to the raypath-interleave (RI) prediction model, which involves the following three steps. First, all transmitters T_i, $i = 1, 2, \ldots, N$ are put into the high-priority base set. Suppose that T_i has n_i raypaths, $r_{i,j}$, $i = 1, 2, \ldots, N$, and $j = 1, 2, \ldots, n_i$. Let $S_{i,j}$ be the angle between the raypath $r_{i,j}$ and the positive x-axis, S_{thd} be a given parameter, U be the set of interleaving raypaths for all T_i, $i = 1, 2, \ldots, N$, and u_k be the kth raypath in U. Then, U is constructed by picking raypaths from different transmitters in a round robin manner. The choice of raypath within each transmitter is determined by the following rule: if $u_k = r_{i,j}$ is the previous raypath from T_i, then the current raypath should be $u_{k+1} = r_{s,t}$, where $s = (i+1)\%N$, and $S_{s,t} = (S_{i,j}+S_{thd})\%360$ (where $\%$ is the modulo operation). Second, the raypaths in U are partitioned into groups with group size M. These groups are assigned different priorities and processed group by group according to their priorities. Third, for each level of diffraction corners, the procedure for the real transmitters is used except that raypaths from diffraction corners are put into enhancement sets, and assigned lower priorities. Also, the parameters such as N, M, and n_i ($i = 1, 2, \ldots, N$) should be changed to make sure that the processing time for each group is the about the same, and the update rate of prediction results is approximately the same at each time interval.

The RI model has the same computational complexity as the SGRP model but requires more main memory since it should store information about the status for each source point. When the numbers of transmitters or diffraction corners are very large, after their raypaths are interleaved and partitioned into groups, the raypaths within the same group may come only from a small portions of source points. Thus, the number of receivers illuminated by a group of raypaths may be small. To handle this situation, we can combine SGRP and RI models to form the source-group-raypath-interleave (SGRI) model which partitions the source points into different geometric regions based on their geometric locations, interleaves raypaths within the same regions, and clusters raypaths from different regions into groups.

4 Approximate Prediction Models

One of the requirements for a progressive or approximate prediction model is to provide users with a mechanism to trade the prediction accuracy for the prediction time. The progressive prediction models introduced in Section 3 indeed provide a basis for such a mechanism. However, if tolerable prediction errors are specified, the prediction model needs to determine the number of raypaths it should process to achieve the given constraints.

Typically, the number of real transmitters is much smaller than the number of building corners, which act as secondary sources of diffracted waves. Hence the processing time spent on real transmitters is very small compared to that for diffraction corners and the prediction time will not decrease significantly if we drop some or all raypaths from the real transmitters. Also, the strength of the secondary sources at the corners is derived from the incident rays due to the real transmitters. One result of this is that the ray fields traced from the secondary sources are much smaller than the ray fields traced directly from the real transmitters. Thus dropping some raypaths from real transmitters, the prediction accuracy will be dramatically affected.

An upper limit for relative accuracy of predictions can be found in terms of the achieved accuracy compared to measurements when many rays are included. Compared to measurements, predictions have average error of about 1 dB and standard deviation of about 8 dB. As discussed previously, the overall accuracy compared to measurements will not be appreciably degraded if relative accuracy of predictions with a limited set of rays has mean and standard deviation less than 2 dB and 4 dB, respectively [3]. Generally speaking, processing of real transmitters alone cannot achieve the above tolerances, so some raypaths from diffraction corners are needed.

To estimate the workload needed to deliver predictions having a desired accuracy in minimum time, we make use of Theorem 3. According to Theorem 3, workload can be determined based on the given tolerant mean μ and variance σ^2 of prediction errors, and the geometric ratio r_{max}. To find r_{max}, we need to know r_i ($i = 0, 1, \ldots, n-1$ and n is the number of receivers). The simplest way to estimate r_i for receiver R_i is as follows. For receiver R_i, record the number, m_i, of illuminating raypaths traced so far, and find the minimum received power $P_{i,min}$ and maximum received power $P_{i,max}$. Then, r_i can be estimated as $r_i = (P_{i,max}/P_{i,min})^{1/m_i}$. The number of groups k can be calculated based on Theorem 3, and the total number of raypaths to be traced is $N_0 = kn$, where n is the number of receivers. During the ray-tracing process, some processed raypaths may not illuminate any receiver at all, or their contributions to the received powers are not significant and dropped. To take these raypaths into account, we still need to estimate the ratio of the retained significant raypaths to total processed raypaths $f_{s,t}$. We may then estimate the workload N_1 (i.e., total number of raypaths to be processed to achieve the given tolerances) as $N_1 = kn/f_{s,t}$.

The quality of the estimated workload N_0 or N_1 depends on the accuracies of the estimated parameters r_{max} and $f_{s,t}$. It may not be tight due to the complex characteristics of raypaths. An alternative method to the estimate of workload is based on the fact that for a finite incremental separation angle δ between the traced rays, there will be a finite number N of rays traced from the real transmitters and the diffracting corners. The method assumes that the prediction accuracy is a monotonic function of number of processed raypaths, and assumes that the relationship between the prediction errors and number of processed raypaths is linear. Suppose that the mean and standard deviation of the prediction results are μ_0 and σ_0, respectively, after we trace all rays from all of the real transmitters. Let μ_i and σ_i be the mean and standard deviation of prediction errors after we process the ith raypath starting at the diffraction corners. Then, $\mu_i = (1 - i/N)\mu_0$, and $\sigma_i = (1 - i/N)\sigma_0$. In other words, given maximum allowable mean and standard deviation of prediction errors, μ and σ respectively, the number of raypaths N_2 we should process is $N_2 = \max\left((1 - \mu/\mu_0)N, (1 - \sigma/\sigma_0)N\right)$. While μ_0 and σ_0 are not known in advance of a complete ray trace, it is reasonable and accurate enough to set μ_0 = 10 dB and σ_0 = 10 dB. In summary, if the maximum allowed mean and standard deviation of prediction errors are μ and σ, respectively, then the estimated workload (the total raypaths to process) W should be $W = \min\left(N_1, N_2 + N_{thd}, N\right)$, where N_{thd} is a tunable parameter.

One implementation for limiting on the total number of rays W to be traced is the source-group-raypath-permute approximate prediction model (SGRP-APP). This model is based on its counterpart in the progressive prediction model and integrates with a new element, the workload estimator. The SGRP-APP model also adjusts the workload estimate W dynamically based on information collected by the sample generator and the ray-tracing engine as follows. First, all transmitters are processed. Necessary information is collected to estimate the ratio of significant raypaths to total processed raypath $f_{s,t}$. The parameter N_1, N_2, and W can be calculated as discussed above, and the mean and standard deviation of prediction errors μ_0 and σ_0 can be estimated empirically. Then, the next group of source points (with size M) is fetched from the sample generator, and is handed to the ray-tracing engine. At the same time, all necessary feedback is collected from the sample generator and the ray-tracing engine, for example, $P_{i,max}$, $P_{i,min}$, m_i for receiver R_i, the number of significant raypaths, and the number of total processed raypaths. Finally, the estimated workload is readjusted based on the above feedback. If all the estimated workload has been processed, then the prediction process terminates. Otherwise, the previous and current steps are repeated. As its counterpart in progressive prediction model, the different diffracting corners are processed sequentially, which may affect the accuracy of workload estimation.

To overcome the limitation of processing the diffraction corners sequentially, we proposed the raypath-interleave approximate prediction (RI-APP) model. Similar to the SGRP-APP model, a workload estimator module is added into the RI-APP. However, due to the fact that raypaths from different diffraction corners are interleaved together and processed in a mixed manner, the methods to estimate and adjust the workload are quite different. At the beginning of the prediction process, the number of raypaths illuminating each receiver is relatively small, it is very difficult to estimate $P_{i,min}$ and $P_{i,max}$ accurately for receiver R_i based on these small number of raypaths. As a result, the workload N_1 based on those parameters is not stable and accurate. If the workload estimator uses this information only, the prediction process may terminates prematurely due to the fluctuation of N_1. Therefore, the estimation of the workload should mainly depend on N_2 at the beginning of the process. As the prediction process proceeds, the accumulated raypaths for each receiver increase continuously, and the changes of the received powers for receivers are relatively small, all parameters required for calculating N_1 can be estimated with higher accuracy.

Based on the above analysis, the estimate of workload W for the RI-APP model is defined as

$$W = \begin{cases} N_2 & \text{when } N_c < N_t, \\ \min(N_1, N_2) & \text{when } N_c \geq N_t. \end{cases} \tag{1}$$

Here, N_c is the total number of raypaths having been processed currently, N_t is a tunable parameter. The workload estimator for the RI-APP model works as follows. First, raypaths from transmitters are processed. The workload estimator collects all necessary information to calculate the following parameters: ratio of significant raypaths to total processed raypath $f_{s,t}$; the mean and standard deviation of prediction errors μ_0 and σ_0; and N_2. The estimated workload is set as $W = N_2$, and $N_t = W/2$. Then, the next group of raypaths (with size M) is fetched from the sample generator and handed to the ray-tracing engine. The parameter N_c is computed based on the feedback from the sample generator and the ray-tracing engine. Information is collected for $P_{i,max}$, $P_{i,min}$, m_i for receiver R_i, the number of significant raypaths, and the number of total processed raypaths. Finally, the estimated workload is readjusted and re-estimated. Parameters $f_{s,t}$, N_1, and N_2 are recalculated, the workload W is updated based on (1). If all the estimated workload has been finished, then the prediction process terminates. Otherwise, the previous and current steps are repeated.

In the RI-APP method, the workload and related parameters are readjusted and re-estimated after every group of raypaths. The size of group can be adjusted dynamically to control the frequency of recalculation and the accuracy of estimations. Since raypaths from different source points are interleaved together, it is more likely that after tracing M groups of raypaths, all receivers are illuminated and their received powers are updated. Therefore, it is expected that the RI-APP model may decrease the mean and variance of prediction errors more uniformly and steadily than the SGRP-APP method.

5 Implementations and Experiments

We have implemented two progressive and two approximate prediction methods, have tested their performance by using two GIS databases, one of Rosslyn, VA, and the other of Dupont Circle, Washington DC. All experiments were performed on a Sun Ultra 10 machine, with CPU clock rate of 440 MHz, main memory of 384 MB, and Solaris 5.7 operating system.

The building footprint shown in Figure 1 represents the core part of Rosslyn, VA, and consists of 79 buildings with 412 walls. The buildings have from 4 to 13 vertices each, with an average of 5. We use only one transmitter Tx that is located at the point having coordinates (237656.0, 118100.0)m. There are 400 receivers Rx which are placed along several streets. Dupont Circle, Washington DC, shown in Figure 2 has streets that run radially from Dupont Circle, as well as on a rectangular grid. There are 3,564 buildings featuring 23181 walls, with each building having from 3 to 86 vertices, with an average of 6. It is evident that the number of vertices in footprints varies dramatically, but most of the footprints have less than 8 edges, and more than half of them feature between 3 and 4 edges. Only one Tx is installed, which is located close to the epicenter of the map at (322780.0, 4308550.0)m. There are 400 receivers (Rx) which are located along the two East-West (horizontal) streets.

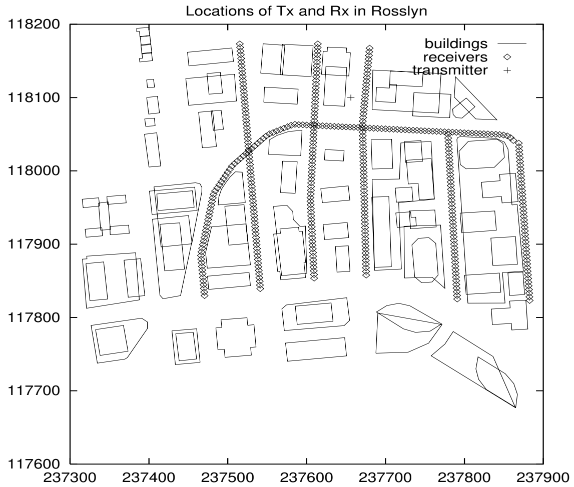

Figure 1: Locations of buildings, Tx and Rx in Rosslyn, VA

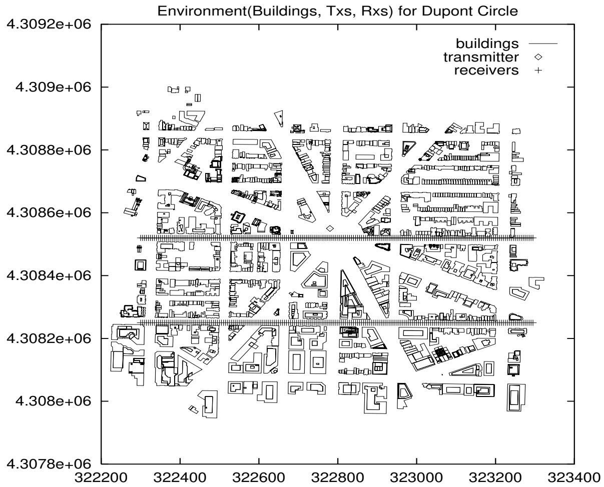

Figure 2: Locations of buildings, Tx and Rx in Dupont Circle, Washington DC

The 2-D ray-tracing method is used for a frequency of 900 MHz. The maximum numbers of reflections and diffractions for each raypath are 8 and 1, respectively. Diffracted rays are the main contributors to the received powers for those receivers in non-line of sight (NLOS) zones. The antenna height of the transmitter is 10m, while all receivers have height 1.5m. All walls are assumed to be described by the Fresnel reflection coefficient for a dielectric constant ϵ_r = 6. The pincushion method is used to launch rays[2] with an angular separation (step size) of $0.5°$.

5.1 Results of Progressive Prediction Models

In this section we evaluate the faithfulness and fairness of the progressive prediction models described in Section 3. The intermediate and final prediction results of all the proposed progressive models are compared with those generated by the traditional prediction model. In a traditional prediction model, all the real transmitters are processed in the order specified by the user, while all diffraction corners are processed in the orders they are generated. We name this traditional model "sequential method". In each experiment, we collect and calculate at various times: the number of illuminated receivers, the number of diffraction corners being processed, the number of raypaths having been traced, as well as the mean and standard deviation of the prediction errors. Notice that when we calculate the mean and standard deviation of the prediction errors, we only count those receivers that are illuminated by some traced raypaths instead of all receivers, therefore, the mean and standard deviation may fluctuate during the prediction process. The results are presented in Tables 1 and 2, respectively, for Rosslyn and Dupont Circle.

time (sec.)	il.rx	df.cor.	rays	mean (dB)	dev. (dB)	time (sec.)	il.rx	df.cor.	rays	mean (dB)	dev. (dB)
sequential method						source-group-raypath-permute					
1.83	251	0	720	2.69	4.44	1.83	251	0	720	2.69	4.44
5.88	318	10	6132	7.74	14.10	6.40	326	10	6123	10.70	18.33
11.45	327	20	11530	8.03	15.03	11.09	332	20	11443	7.49	11.46
16.49	334	30	16934	5.17	10.92	17.55	344	30	16718	4.49	6.75
22.05	335	40	21676	4.12	9.71	21.68	346	40	22037	3.78	6.11
35.34	348	60	32059	2.60	8.50	33.05	350	60	31998	3.50	8.10
48.19	353	80	42599	2.12	7.31	46.43	361	80	42727	1.88	6.92
57.77	356	100	53114	1.91	6.95	57.92	362	100	52592	0.98	6.41
70.52	363	120	63315	0.70	6.32	69.76	364	120	63219	0.04	0.20
73.31	365	127	66925	0.00	0.00	74.02	365	127	66925	0.00	0.00
raypath-interleave						source-group-raypath-interleave					
1.83	251	0	720	2.69	4.44	1.80	251	0	720	2.69	4.44
6.71	333	84	5169	6.84	11.66	6.54	341	84	5156	6.78	11.33
12.18	345	127	10312	4.25	6.53	11.80	342	127	10325	4.70	7.28
17.51	349	127	15452	3.45	5.39	17.49	349	127	15481	3.46	5.46
23.16	350	127	20609	2.79	4.01	22.99	350	127	20613	2.79	4.02
34.51	358	127	30901	1.82	2.96	34.27	358	127	30901	1.81	2.96
45.66	359	127	41205	1.19	2.23	45.19	359	127	41212	1.23	2.26
57.09	362	127	51530	0.70	1.56	56.58	362	127	51527	0.70	1.57
68.42	365	127	61877	0.26	1.28	67.68	365	127	61861	0.16	1.20
73.75	365	127	66925	0.00	0.00	73.26	365	127	66925	0.00	0.00

Table 1: Rosslyn: statistics for progressive methods

From these tables, we can observe that all progressive methods illuminate more receivers than the sequential method (i.e, the traditional method) at any given time during the entire prediction process. The raypath-interleave (RI) method and source-group- raypath-interleave (SGRI) method perform better than source-group-raypath-permute (SGRP) method, and the SGRI method is the best. It is obvious that the RI and SGRI methods illuminate a large portion of receivers at very early time which may help improve the prediction accuracy. Also, the RI and SGRI methods traverse more diffraction corners than sequential method and SGRP method at any given point during the entire prediction procedure which helps to illuminate more receivers. They usually visit all diffraction corners within 10% of the processing time.

It can be seen that the mean and standard deviation of prediction errors delivered by all progressive methods are much better than those given by the sequential method at any given time prior to the end of the entire prediction process. When the RI or SGRI method is used, it takes only about half of the total processing time to drive both the mean and standard deviation of prediction errors within 3 dB, and only two-third of the total processing time to reach 1 dB. In contrast, it takes almost all the processing time to bring the mean and standard deviation down to 1 dB and 3 dB, respectively, in the sequential method. Within the same time constraint, all methods trace about the same number of raypaths. However, the means and standard deviations of prediction errors delivered by the progressive methods are much lower (more than 4 dB at most of the time) than those by the sequential method. It is clear that the sample generators and ray-tracing engines in our proposed progressive methods indeed tend to assign high priority to raypaths contributing significantly to predictions and trace them first.

time (sec.)	il.rx	df.cor.	rays	mean (dB)	dev. (dB)	time (sec.)	il.rx	df.cor.	rays	mean (dB)	dev. (dB)
sequential method						source-group-raypath-permute					
10.40	180	0	720	5.37	9.11	10.40	180	0	720	5.37	9.11
121.69	308	100	52449	21.15	23.05	111.65	340	100	53266	16.61	18.92
236.28	335	200	101961	19.96	19.90	214.46	368	200	106594	12.91	12.82
448.96	338	400	207143	17.90	18.01	444.54	379	400	211981	9.88	10.20
659.09	351	600	311235	17.71	17.74	652.60	385	600	315622	5.97	7.42
1057.05	362	900	465692	13.85	14.07	1066.93	385	950	495354	5.44	6.85
1237.91	376	1050	545665	10.60	11.83	1248.18	385	1100	572235	4.19	5.72
1666.19	387	1400	728221	3.82	5.73	1641.30	388	1450	754217	2.20	4.99
1931.59	388	1650	858578	2.14	4.65	1928.43	394	1700	883000	0.32	0.92
2236.12	395	1956	1014773	0.00	0.00	2232.71	395	1956	1014773	0.00	0.00
raypath-interleave						source-group-raypath-interleave					
10.40	180	0	720	5.37	9.11	10.40	180	0	720	5.37	9.11
124.57	308	958	50761	18.81	20.50	118.31	363	952	50743	15.00	15.47
286.47	381	1956	126885	8.93	8.19	288.77	384	1956	126902	7.96	7.41
455.14	389	1956	203008	6.28	5.72	455.65	389	1956	203001	6.25	5.72
625.91	390	1956	279078	4.71	4.31	623.30	390	1956	279089	4.49	4.26
1012.36	394	1956	456683	2.91	3.11	1012.27	394	1956	456698	2.91	3.11
1292.40	395	1956	583537	2.02	2.31	1295.17	395	1956	583526	1.97	2.30
1686.02	395	1956	761119	1.15	1.69	1684.17	395	1956	761160	1.09	1.68
1903.49	395	1956	862646	0.99	1.64	1896.16	395	1956	862631	0.59	1.10
2239.96	395	1956	1014773	0.00	0.00	2238.18	395	1956	1014773	0.00	0.00

Table 2: Dupont Circle: statistics for progressive methods

Finally, the relationship between the prediction errors (mean and standard deviation) and the processing time is almost linear for the RI and SGRI methods. Therefore, they update the received powers of all receivers continuously, and with almost the same rate. Similarly, the relationship between the prediction errors and the number of processed raypaths is nearly linear for both the RI method and the SGRI method

city map	target errors (μ,σ)(dB)	time (sec.)	speedup	raypaths	il.rx	mean (dB)	dev (dB)
Rosslyn	(0.0, 0.0)	74.00	1.00	66925	365	0.00	0.00
	(1.0, 1.0)	66.75	1.11	60125	364	0.30	1.00
	(1.0, 2.0)	62.21	1.19	57096	363	0.95	2.50
	(2.0, 3.0)	58.89	1.26	52992	362	1.90	3.44
Dupont Circle	(0.0, 0.0)	2232.00	1.00	1014773	395	0.00	0.00
	(1.0, 1.0)	1872.46	1.19	866723	395	0.40	0.95
	(1.0, 2.0)	1782.53	1.25	817432	394	0.88	1.96
	(2.0, 3.0)	1699.19	1.31	788332	393	1.36	3.10

Table 3: speedup, prediction errors by the SGRP-APP method

indicating that their grouping strategy satisfies the fairness requirement.

Based on the above observations, we can conclude that the RI and SGRI methods are better than traditional models and the SGRP method and satisfy all the requirements for progressive models, especially the faithfulness and fairness requirements.

5.2 Results of Approximate Prediction Models

First, we evaluate the performance of the source-group-raypath-permute approximate (SGRP-APP) prediction model, especially its ability of workload adjustment and its sensitivity to the input parameters. We execute the SGRP-APP method for various targeted values of the mean and standard deviation of the prediction errors. The ranges for the mean and standard deviation of prediction errors are [0.0, 2.0] dB and [0.0, 3.0] dB, respectively, and the step size is 1 dB. For each experiment, we collect and calculate the following parameters: the processing time, the speedup, the number of processed raypaths, the number of illuminated receivers, the actual mean and standard deviation of prediction errors it delivered. These parameters are shown in Table 3 for Rosslyn and Dupont Circle.

From Table 3, it can be seen that the mean and standard deviation of prediction errors delivered by this model are quite close to the targeted prediction errors. Therefore, the workload estimator can adjust its workload estimates based on the inputs. The estimator indeed provides a mechanism to trade the prediction accuracy for the prediction time. When the tolerance for prediction errors is relaxed, the prediction time can be reduced. Controllability of prediction errors is relatively coarse. The speedup does not change with the same rate as the targeted mean and standard deviation of prediction errors. When the targeted mean and standard deviation of prediction errors are relaxed (become larger) from (1.0, 1.0) dB to (2.0, 3.0) dB, the speedups only increase from 1.19 to 1.31 for Dupont Circle. Prediction accuracy fluctuates around the specified tolerance, it can be better or worse than the given tolerance. The distribution of prediction errors among all receivers is not uniform. The prediction errors tend to concentrate on a subset of receivers. The main reason for this uneven distribution of prediction errors is that all source points are processed sequentially and some source points are not processed at all when the prediction procedure terminates.

In Section 5.1, it was found that the raypath-interleave (RI) method and the source-group-raypath-interleave (SGRI) method are the best performers. It is expected that the approximate prediction models based on these progressive methods may also have the best performance. We use the RI-APP method here as an example to evaluate its performance, especially the fine-tuning capability and sensitivity to inputs of its workload estimator. Experiments similar to those for the SGRP-APP method are performed, except that we use a finer step size of 0.5 dB instead of 1.0 dB. Table 4 presents the experiment results for Rosslyn and Dupont Circle.

city map	target errors (μ,σ)(dB)	time (sec.)	speedup	raypaths	il.rx	mean (dB)	dev (dB)
Rosslyn	(0.0, 0.0)	74.00	1.00	66925	365	0.00	0.00
	(0.5, 0.5)	69.41	1.07	62879	365	0.16	0.99
	(1.0, 1.0)	58.12	1.27	21533	362	0.55	1.50
	(1.0, 2.0)	48.63	1.52	43406	360	1.09	1.99
	(2.0, 2.0)	41.52	1.78	37125	359	1.40	2.30
	(2.0, 3.0)	30.10	2.46	26890	358	2.04	3.05
Dupont Circle	(0.0, 0.0)	2232.00	1.00	1014773	395	0.00	0.00
	(0.5, 0.5)	2139.91	1.04	964107	395	0.30	0.78
	(1.0, 1.0)	1792.41	1.25	811907	395	1.00	1.64
	(1.0, 2.0)	1631.28	1.37	735787	395	1.48	2.04
	(2.0, 2.0)	1300.40	1.72	589537	395	1.95	2.29
	(2.0, 3.0)	1087.55	2.05	487038	395	2.60	2.90

Table 4: speedup, prediction errors by RI-APP method

From Table 4, we can observe that the RI-APP method provides a better mechanism to balance the prediction accuracy and prediction time than the SGRP-APP method. For example, in Dupont Circle case, the speedup can be as high as 2.05 when the targeted mean and standard deviation of prediction errors are (2.0, 3.0) dB, which is much higher than that given by the SGRP-APP method (which is 1.31). Therefore, the workload estimator in the RI-APP model performs much better than that in the SGRP-APP method in its adjustability, controllability and accuracy of estimation. There is a very close relationship between the prediction accuracy and the number of processed raypaths. Under the same tolerances for the prediction errors, the RI-APP method delivers much better prediction results and speedups than the SGRP-APP method. For instance, for Dupont circle case, when the specified mean and standard deviation of prediction errors are 2.0 dB and 3.0 dB, respectively, the actual mean and standard deviation generated by the SGRP-APP model are 1.36 dB and 3.10 dB, respectively, and the speedup is 1.31. while in the RI-APP method, the actual mean and standard deviation are 2.60 dB and 2.90 dB, respectively, but the speedup is 2.05. If the same processing time is used (i.e., about 1699.19 seconds), then the RI-APP method can deliver mean and standard deviation less than 1.48 dB and 2.04 dB, with speedup larger than 1.37. In addition, the distribution of prediction errors is more uniform than that in the SGRP-APP method. Thus, the RI-APP method has better prediction fairness than the SGRP-APP method.

6 Conclusions

Progressive and approximate prediction models are attractive since they can provide continuous feedback to users during the entire radio wave propagation prediction process. They offer users more flexible and fine-scale controls over the prediction processing, including effective mechanisms to trade prediction accuracy for prediction time. Moreover, they integrate easily with other techniques to further improve the performance of the system. We have presented experiment results for the proposed progressive prediction methods and approximate prediction methods. From these results, it is evident that the proposed progressive prediction methods (the source-group-raypath-permute method, the raypath-interleave method and the source-group-raypath-interleave method) deliver better prediction results than traditional models continuously and progressively at any time during the entire prediction process. In most experiments, the prediction errors were reduced by more than 5 dB for both the mean and standard deviation of the prediction errors. In the proposed progressive prediction methods, the SGRI method is the best in terms of prediction accuracy, number of illuminated receivers, and number of processed diffraction corners. Both approximate prediction methods, the SGRP-APP method and the RI-APP method, provide flexible mechanisms to trade prediction accuracy. The actual mean and standard deviation of the prediction errors are very close to the targeted ones. The RI-APP method has better performance than the SGRP-APP method. Under the same time constraint, the RI-APP method delivers much better prediction results than the SGRP-APP method in terms of prediction accuracy. Similarly, under the same targeted prediction errors, the RI-APP method generates prediction results with less time, thereby has higher speedup rate. The RI-APP method also has better capabilities to estimate and adjust the workload, and can manipulate the relation between the prediction accuracy and the number of processed raypaths in a much finer granularity.

Acknowledgments

The authors are grateful to the reviewers for their comments that helped us significantly improve the presentation of our work. We are also very much indebted to Dr. Boris Aronov and Dr. Yi-Jen Chiang for reviewing earlier versions of the paper.

References

[1] M. Barnsley. *Fractals Everywhere*. Academic Press, San Diego, 1988.

[2] H. L. Bertoni. *Radio Propagation for Modern Wireless Systems*. Prentice-Hall PTR, Upper Saddle River, NJ, 2000.

[3] Z. Chen. Acceleration Techniques for Ray-Tracing Systems in Wireless Radio Wave Propagation Prediction. *Ph.D. dissertation, Polytechnic University*, 2003.

[4] P. Chossat and M. Golubitsky. Symmetry Increasing Bifurcations of Chaotic Attractors. *Physica*, D(32):423–436, 1988.

[5] C. Christopoulos, A. Skodras, and T. Ebrahimi. The JPEG2000 Still Image Coding System: An Overview. *IEEE Transactions on Consumer Electronics*, 46(4):1103–1127, November 2000.

[6] D. Cohen-Or, D. Levin, and O. Remez. Progressive Compression of Arbitrary Triangular Meshes. In *IEEE Symposium on Information Visualization*, San Francisco, California, October 1999.

[7] JBIG Committee. 14492 FCD, Information Technology – Coded Representation of Picture and Audio Information – Lossy/Lossless Coding of Bi-Level Images. July 1999.

[8] Joint Technical Committee. Information Technology – JPEG 2000 Image Coding System – Part I: Core Coding System. December 2000.

[9] B. Freisleben, D. Hartmann, and T. Kielmann. Parallel Raytracing: A Case Study on Partitioning and Scheduling on Workstation Clusters. In *Proceedings of Thirtieth International Conference on System Sciences*, pages 596–605, Hawaii, 1997.

[10] B. Freisleben, D. Hartmann, and T. Kielmann. Parallel Incremented Raytracing of Animations on a Network of Workstations. In *Proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications*, pages 1305–1312, July 1998.

[11] J. M. Hellerstein, P. J. Haas, and H. J. Wang. Online Aggregation. In *ACM SIGMOD International Conference on Management of Data*, Tucson, Arizona, May 1997.

[12] H. Hoppe, T. Derose, T. Duchamp, J. Mcdonald, and W. Stuetzle. Mesh Optimization. In *Proceedings of ACM SIGGRAPH Proceedings*, pages 19–26, Anaheim, CA, August 1993.

[13] S.-C. Kim, B.J. Carino, T.M. Willis, V. Erceg, S.J. Fortune, R.A. Valenzuela, L.W. Thomas, J. Ling, and J.D. Moore. Radio Propagation Measurements and Prediction Using Three-Dimensional Ray Tracing in Urban Environments at 908MHz and 1.9GHz. *IEEE Transactions on Vehicular Technology*, 48(3):931–946, May 1999.

[14] J. M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[15] E. Kreyszig. *Advanced Engineering Mathematics*. John Wiley and Sons, Inc., 1962.

[16] T. Kurner and A. Meier. Prediction of Outdoor and Outdoor-to-Indoor Coverage in Urban Areas at 1.8 GHz. *IEEE Journal on Selected Areas in Communications*, 20(3):496–506, April 2002.

[17] G. Liang and H.L. Bertoni. A New Approach to 3-D Ray Tracing for Propagation Prediction in Cities. *IEEE Transactions on Antennas and Propagation*, 46:853–863, 1998.

[18] B. Mandelbrot. *The Fractal Geometry of Nature*. W. H. Freeman & Co., San Francisco, 1982.

[19] M. J. Muuss. Towards Real-Time Ray-Tracing of Combinatorial Solid Geometric Models. In *Proc. BRL-CAD Symposium'95*, Aberdeen Proving Ground, MD, June 1995.

[20] M. J. Muuss and M. Lorenzo. High Resolution Interactive Multispectral Missile Sensor Simulation for ATR and DIS. In *Proc. BRL-CAD Symposium'95*, Aberdeen Proving Ground, MD, June 1995.

[21] W.M. Obrien, E.M. Kenny, and P.J. Cullen. An Efficient Implementation of a Three-Dimensional Microcell Propagation Tool for Indoor and Outdoor Urban Environments. *IEEE Transactions on Vehicular Technology*, 49(2):622–630, March 2000.

[22] S. Parker, P. Shirley, Y. Livnat, C. Hansen, and P. P. Sloan. Interactive Ray Tracing. *Symposium on Interactive 3D Graphics (I3D)*, pages 119–126, April 1999.

[23] R. Ramakrishnan and J. Gehrke. *Database Management Systems*. McGraw Hill, New York, 2000.

[24] A. Reisman, C. Gotsman, and A. Schuster. Interactive-Rate Animation Generation by Parallel Progressive Ray-Tracing on Distributed Memory Machines. *Journal of Parallel and Distributed Computing*, 60:1074–1102, 2000.

[25] S. Y. Tan and H. S. Tan. A Microcellular Communications Propagation Model Based on the Uniform Theory of Diffraction and Multiple Image Theory. *IEEE Transactions on Antennas and Propagation*, 44:1317–1325, 1996.

[26] G. Taubin, A. Gueziec, W. Horn, and F. Lazarus. Progressive Forest Split Compression. In *SIGGRAPH'98*, Orlando, FL, July 1998.

[27] I. Wald and P. Slusallek. State of the Art in Interactive Ray Tracing. *Eurographics*, pages 21–42, September 2001.

[28] X. Zhao, J. Kivinen, P. Vainikainen, and K. Skog. Propagation Characteristics for Wideband Outdoor Mobile Communications at 5.3 GHz. *IEEE Journal on Selected Areas in Communications*, 20(3):507–514, April 2002.

A Proof of Fairness Theorem

Proof a) Let $E_i(g)$ (in dB) be the prediction error for receiver R_i after the gth group of raypaths is processed, and its mean (in dB) be $\mu(g)$ ($g = 0, 1, \ldots$). Since $P_{i,j+1} = r_i P_{i,j}$ for $i = 0, 1, \ldots, n$ and $j = 0, 1, \ldots, m_i - 1$, we denote $P_i = P_{i,0}$, then

$$E_i(g) = \left[\sum_{l=0}^{m_i-1} P_{i,l}\right]_{dB} - \left[\sum_{l=0}^{g-1} P_{i,l}\right]_{dB} = \left[\sum_{l=0}^{m_i-1} P_i r_i^l\right]_{dB} - \left[\sum_{l=0}^{g-1} P_i r_i^l\right]_{dB} = \left[\frac{1 - r_i^{m_i}}{1 - r_i^g}\right]_{dB} \tag{2}$$

$$\mu(g) = \frac{1}{n}\sum_{i=0}^{n-1} E_i(g) = \frac{1}{n}\sum_{i=0}^{n-1}\left[\frac{1 - r_i^{m_i}}{1 - r_i^g}\right]_{dB}; \quad \mu(g+1) = \frac{1}{n}\sum_{i=0}^{n-1}\left[\frac{1 - r_i^{m_i}}{1 - r_i^{g+1}}\right]_{dB} \tag{3}$$

$$\mu(g) - \mu(g+1) = \frac{1}{n}\sum_{i=0}^{n-1}\left\{\left[\frac{1 - r_i^{m_i}}{1 - r_i^g}\right]_{dB} - \left[\frac{1 - r_i^{m_i}}{1 - r_i^{g+1}}\right]_{dB}\right\} = \frac{1}{n}\sum_{i=0}^{n-1}\left[\frac{1 - r_i^{g+1}}{1 - r_i^g}\right]_{dB} \tag{4}$$

since $0 < r_i < 1$, it is easy to show that $(1 - r_i^{g+1})/(1 - r_i^g) > 1$ and $\left[(1 - r_i^{g+1})/(1 - r_i^g)\right]_{dB} > 0$ for $i = 0, 1, ..., (n-1)$, therefore, $\mu(g) > \mu(g+1)$.

b) Let $\sigma^2(g)$ be the variance (in dB^2) of the prediction errors after the gth group of raypaths has been processed ($g = 0, 1, \ldots$), then

$$\sigma^2(g) = \frac{1}{n}\sum_{i=0}^{n-1} E_i^2(g) - \mu^2(g); \quad \sigma^2(g+1) = \frac{1}{n}\sum_{i=0}^{n-1} E_i^2(g+1) - \mu^2(g+1) \tag{5}$$

$$\sigma^2(g) - \sigma^2(g+1) = \frac{1}{n}\left[\sum_{i=0}^{n-1}(E_i^2(g) - E_i^2(g+1)\right] - \left[\mu^2(g) - \mu^2(g+1)\right] \tag{6}$$

$$= \frac{1}{n}\sum_{i=0}^{n-1}\{[E_i(g) + E_i(g+1)][E_i(g) - E_i(g+1)]\} - [\mu(g) + \mu(g+1)][\mu(g) - \mu(g+1)] \tag{7}$$

Let $E_i(g) + E_i(g+1) = a_i$ and $E_i(g) - E_i(g+1) = b_i$. Since each receiver is illuminated by infinite numbers of raypaths, it is clear that

$$E_i(g) = \left[\sum_{l=0}^{\infty} P_i r_i^l\right]_{dB} - \left[\sum_{l=0}^{g-1} P_i r_i^l\right]_{dB} = \left[\frac{P_i}{1 - r_i}\right]_{dB} - \left[\frac{P_i(1 - r_i^g)}{1 - r_i}\right]_{dB} = \left[\frac{1}{1 - r_i^g}\right]_{dB} \tag{8}$$

$$a_i = E_i(g) + E_i(g+1) = \left[\frac{1}{(1 - r_i^g)(1 - r_i^{g+1})}\right]_{dB}; \quad b_i = E_i(g) - E_i(g+1) = \left[\frac{1 - r_i^{g+1}}{1 - r_i^g}\right]_{dB} \tag{9}$$

$$\mu(g) + \mu(g+1) = \frac{1}{n}\sum_{i=0}^{n-1}[E_i(g) + E_i(g+1)] = \frac{1}{n}\sum_{i=0}^{n-1} a_i; \quad \mu(g) - \mu(g+1) = \frac{1}{n}\sum_{i=0}^{n-1} b_i \tag{10}$$

$$\sigma^2(g) - \sigma^2(g+1) = \frac{1}{n}\sum_{i=0}^{n-1}(a_i b_i) - (\frac{1}{n}\sum_{i=0}^{n-1} a_i)(\frac{1}{n}\sum_{i=0}^{n-1} b_i) \tag{11}$$

It is easy to show that $a_i \geq 0$ and $b_i \geq 0$ for $i = 0, 1, \ldots, n-1$. Also, for any pair of a_i and a_j and any pair of b_i and bj, we can get

$$a_i - a_j = \left[(\frac{1 - r_j^g}{1 - r_i^g})(\frac{1 - r_j^{g+1}}{1 - r_i^{g+1}})\right]_{dB}; \quad b_i - b_j = \left[\frac{\frac{1 - r_i^{g+1}}{1 - r_i^g}}{\frac{1 - r_j^{g+1}}{1 - r_j^g}}\right]_{dB} \tag{12}$$

By assuming that $0 < r_j < r_i < 1$, we can show that $(1 - r_j^g)/(1 - r_i^g) > 1$, $(1 - r_j^{g+1})/(1 - r_i^{g+1}) > 1$, and $(\frac{1-r_j^g}{1-r_i^g})(\frac{1-r_j^{g+1}}{1-r_i^{g+1}}) > 1$, so, $\left[(\frac{1-r_j^g}{1-r_i^g})(\frac{1-r_j^{g+1}}{1-r_i^{g+1}})\right]_{dB} > 0$, therefore, $a_i > a_j$. By letting $f(r) = (1 - r^{g+1})/(1 - r^g)$, $(0 < r < 1)$, we can find its derivative with respect to r as

$$\frac{df}{dr} = \frac{r^{g-1}(1 - r)\left[g - r(1 + r + \cdots + r^{g-1})\right]}{(1 - r^g)^2} > 0 \tag{13}$$

Therefore, $f(r)$ is a monotonically increasing function. For $0 < r_j \leq r_i < 1$, we have $(1 - r_i^{g+1})/(1 - r_i^g) \geq (1 - r_j^{g+1})/(1 - r_j^g)$, thereby $b_i \geq b_j$.

In summary, if we assume that $r_i \geq r_{i+1}$ for $i = 0, 1, \ldots, n-2$, then, $a_i > 0$, $b_i > 0$, $a_i \geq a_{i+1}$, $b_i \geq b_{i+1}$. Similarly, if we assume that $r_i \leq r_{i+1}$ for $i = 0, 1, \ldots, n-2$, then, $a_i > 0$, $b_i > 0$, $a_i \leq a_{i+1}$, $b_i \leq b_{i+1}$. According to Chebychev's inequality which states that if $a_i > 0$, $b_i > 0$, $i = 0, 1, ..., (n-1)$, $a_i \geq a_{i+1}$ and $b_i \geq b_{i+1}$, or $a_i \leq a_{i+1}$ and $b_i \leq b_{i+1}$, then, $(\frac{1}{n}\sum_{i=0}^{n-1} a_i)(\frac{1}{n}\sum_{i=0}^{n-1} b_i) \leq \frac{1}{n}\sum_{i=0}^{n-1}(a_i b_i)$; we derive that

$$\sigma^2(g) - \sigma^2(g+1) = \frac{1}{n}\sum_{i=0}^{n-1}(a_i b_i) - (\frac{1}{n}\sum_{i=0}^{n-1} a_i)(\frac{1}{n}\sum_{i=0}^{n-1} b_i) \geq 0 \tag{14}$$

Therefore, $\sigma^2(g) \geq \sigma^2(g+1)$ indicating that the variance of prediction errors is a decreasing function of g. **QED**

B Proof of Workload Estimation Theorem

Proof In Theorem 2, we have already shown that the mean and variance of prediction errors monotonically decrease with the number of traced groups of raypaths. Therefore, we need only to find the minimum number of groups k to be traced such that the mean and variance of prediction errors are less than or equal to the given μ and σ^2. Let $E_i(k)$ be the prediction error (in dB) for receiver R_i after tracing group $G(k)$, $\mu(k)$ and $\sigma^2(k)$ be the mean and variance (in dB and dB^2, respectively) of prediction errors at the end of processing of group $G(k)$, $r_{max} = \max_{i=0}^{n-1} r_i$ and $r_{min} = \min_{i=0}^{n-1} r_i$. It is easy to show that $1/(1 - r_i^g) \leq 1/(1 - r_{max}^g)$, $1/(1 - r_i^g) \geq 1/(1 - r_{min}^g)$, and $1/(1 - r_{min}) \leq 1/(1 - r_{max})$, then

$$\mu(k) = \frac{1}{n}\sum_{i=0}^{n-1} E_i(k) = \frac{1}{n}\sum_{i=0}^{n-1}\left[\frac{1}{1-r_i^k}\right]_{dB} \leq \frac{1}{n}\sum_{i=0}^{n-1}\left[\frac{1}{1-r_{max}^k}\right]_{dB} = \left[\frac{1}{1-r_{max}^k}\right]_{dB} \tag{15}$$

$$\begin{aligned}\sigma^2(k) &= \frac{1}{n}\sum_{i=0}^{n-1} E_i^2(k) - \mu^2(k) = \frac{1}{n}\sum_{i=0}^{n-1}\left[\frac{1}{1-r_i^k}\right]_{dB}^2 - \left\{\frac{1}{n}\sum_{i=0}^{n-1}\left[\frac{1}{1-r_i^k}\right]_{dB}\right\}^2 \\ &\leq \frac{1}{n}\sum_{i=0}^{n-1}\left[\frac{1}{1-r_{max}^k}\right]_{dB}^2 - \left\{\frac{1}{n}\sum_{i=0}^{n-1}\left[\frac{1}{1-r_{min}^k}\right]_{dB}\right\}^2 \leq \left[\frac{1}{1-r_{max}^k}\right]_{dB}^2\end{aligned} \tag{16}$$

Thus, $\mu(k) \leq \mu$ and $\sigma_k^2 \leq \sigma^2$ when $k = \max(k_1, k_2)$ and

$$k_1 \geq \frac{\log\left[1 - \frac{1}{10^{\frac{\mu}{10}}}\right]}{\log(r_{max})}; \qquad k_2 \geq \frac{\log\left[1 - \frac{1}{10^{\frac{\sigma}{10}}}\right]}{\log(r_{max})} \tag{17}$$

QED