Building Footprint Simplification Techniques and Their Effects on Radio Propagation Predictions*

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

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

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

October 14, 2003

Abstract

Building footprint simplification is of critical importance to radio propagation predictions in wireless communication systems as the prediction time is closely related to the number of both buildings and vertices involved. Intuitively, if the complexity of footprints (i.e., the number of vertices in the footprints) is reduced, predictions can be generated more quickly. However, such reductions often affect the accuracy of results as the simplification error constrains the efficiency that can be achieved. To achieve a good vertex reduction rate for the footprints involved and at the same time preserve the shapes of footprints in terms of their areas, orientations, and centroids, we propose a number of efficient single-pass methods to simplify building footprints. To satisfy constraints on edges, areas, and centroid of simplified footprints, multi-pass methods are suggested. Hybrid methods take advantages of complementary property exhibited by different footprint simplification methods. We assess the baseline effectiveness of our proposed techniques, and carry out an extensive comparative evaluation with real GIS data from different municipalities. Through experimentation, we find that hybrid methods deliver the best performance in both vertex reduction rate and simplification error. We examine the effects that these footprint simplification methods have on the ray-tracing based radio propagation prediction systems in terms of processing time and prediction accuracy. Our experiments show that footprint simplification methods indeed reduce prediction time up to three-folds, and maintain prediction accuracy with high confidence as well. We also investigate the relationship between footprint simplification error and the prediction accuracy. We find that the prediction accuracy is sensitive to the distortion (i.e., change of shape) of building footprints. This helps us to better understand the trade-off between precision of building database and the accuracy of predictions generated by ray-tracing based radio propagation prediction systems.

Indexing Terms: Footprint simplification algorithms, simplified maps for radio propagation prediction, experimental evaluation of spatial reduction methods.

*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).

1 Introduction

The tremendous growth in mobile communications necessitates fast and accurate prediction of radio wave propagation for system deployment. Microcells are currently used to cover a small area, and thereby increase the capacity of the cellular communication systems [24, 22, 21, 23]. To achieve small area coverage, the height of antennas installed in a microcell base station may be lower than the average height of the surrounding buildings. In such modern radio wave propagation environments, conventional statistical models do not yield accurate propagation predictions [19]. On the other hand, ray-tracing based radio propagation prediction models have shown promise in microcell environments [52, 48, 49, 50, 51, 25, 26, 1, 27]. Such techniques take into account fine-level features of a geographic area including building shapes and orientations, electrical characteristics of building materials, terrain, locations of transmitters and receivers, heights and patterns of antennas used by both base and mobile stations, and can even incorporate weather and vegetation [53, 54]. Propagation predictions can play an important role in determining network parameters including coverage, transmitted-data rates, optimal base station locations, and antenna patterns. Whenever a network parameter is changed, the ray-tracing system has to be executed to evaluate the effect of the modification on the entire network. Such systems are frequently used in the design, analysis, and deployment of wireless networks.

For the elevated base station antennas of microcellular systems, raypaths involve combined propagation over and around the buildings. A database of buildings is required to make predictions in this case. When the base station antennas are below the surrounding buildings, as in microcellular systems, propagation appears to separate into two classes of raypaths [20]. One class goes around the sides of the buildings, essentially in the horizontal plane, while the other goes over the buildings in the vertical plane containing the base station antenna and the mobile [47, 20]. This study investigates a class of rays that go around the sides of buildings and rely on 2-D ray tracing methods.

For ray-tracing systems to work well, two key objectives have to be fulfilled: precision of the building database and reasonable prediction time. The prediction accuracy can be adversely affected by the precision of building features [47]. In this regard, many small details will contribute little to the predictions, but may increase the prediction time dramatically. Ray-tracing is computation intensive since it attempts to follow all possible routes (i.e., raypaths) from all source points (transmitters and diffraction corners acting as secondary transmitters) to all receivers [25, 51]. These raypaths may undergo multiple reflections, diffractions, and/or diffuse scattering effects incurring long processing time [20, 21]. The above two objectives are not orthogonal and require substantial trade-offs. The core issue is to speed up ray-tracing and at the same time maintain the prediction accuracy at an acceptable level. Consistent with ray tracing in 2-D for low base station antennas, we represent buildings by their footprints in this paper.

The most frequently used operation in ray-tracing systems is the ray-wall-intersection test which accounts for 90% of the processing time [3]. The ray-wall-intersection test is tightly coupled with the number of edges in the building database and a possible decrease in number of vertices may substantially reduce the processing time required. Diffraction corners act physically as secondary sources and are treated as additional transmitters. Reduction of the number of vertices in the building database also means smaller number of diffraction corners to be traced. Reduction of the amount of data needed to be processed allows us to design and use much simpler data structures. Such data structures enable more efficient operations (i.e., retrievals, insertions, and updates).

The above observations point out that simplification of the building footprints to a version that approximates the original but has fewer vertices will speed up the radio prediction process. Footprint simplification can be defined as an optimization problem whose objective function is to minimize the number of vertices in building footprints without unduly effecting prediction accuracy. Maintaining prediction accuracy suggests a number of constraints. First, all vertices in the simplified footprint should be a subset of vertices in the original footprint to avoid spurious secondary transmitters (diffraction corners). Second, long edges are much more important than short ones in a footprint since they may intercept and reflect more rays, and thus may illuminate more receivers and corners. Finally, change in area or translation of the footprint's centroid may affect the number of intercepted rays, the constituent segments of raypaths, and their unfolded lengths.

The objective to minimize the number of vertices in the building database can be evaluated by the vertex reduction rate defined as $(N_{old} - N_{new})/N_{old}$, where N_{old} and N_{new} are the number of vertices in the original and simplified building databases, respectively. The proposed constraints are quantified as follows: a) vertices in simplified footprints must be a subset of vertices in the original footprints; b) edges with length L have to be retained if $L \geq f_l L_{thd}$, where L_{thd} is the prespecified threshold, and f_l is the "length factor" ($f_l \geq 0$); c) area of the simplified footprint should satisfy $|(A_{new} - A_{old})/A_{old}| \leq f_a$, where A_{old} and A_{new} are the areas of the original and simplified footprints, and f_a is the "area factor" ($0 \leq f_a \leq 1$); d) the centroid of the simplified footprint should satisfy $D_{new,old} \leq f_c D_{thd}$, where $D_{new,old}$ is the distance between the centroids of the original and simplified footprints, D_{thd} is a given threshold, and f_c is the "centroid factor" ($0 \leq f_c \leq 1$); and e) simplification errors should distribute uniformly along the footprint boundary, instead of concentrating on only some portions of the footprint.

The organization of the paper is as follows: Section 2 discusses related work and Section 3 outlines performance metrics used. Section 4 proposes four families of single-pass simplification algorithms and Sections 5 and 6 discuss multi-pass and hybrid methods respectively. Section 7 presents an experimental comparison of the proposed algorithms and evaluates the impact of footprint simplifications in the prediction time and accuracy on radio propagation predictions. Our conclusions and future work can be found in Section 8.

2 Related Work

Building footprint simplification shares similarities and demonstrates differences with problems in diverse areas, including a) polygonalization in digital curve approximation, b) polygonal simplification in computer graphics, computer-aided design and virtual reality, and c) the generalization in geographic information systems (GIS).

Polygonalization methods are widely used in digital curve approximations [2, 5, 28]. The key objective is to approximate a curve with a polyline (connected straight line segments) so that the resulting polyline is within user-specified thresholds to the original curve [13, 7, 9, 10, 15]. There exists a wealth of polygonalization methods including the iterative end-point fit algorithm [17], the sequential fit algorithms [10, 11, 12], the area-based methods [7, 9], the error-norm optimal method [16], the computational-geometric methods [14], and the *minimax* method and its variations [13, 15]. The aforementioned methods are asymmetric in nature as the results generated could be different if started with different initial data points. Some such symmetric methods are proposed in [8]. Polygonalization and building footprint simplifications considered here demonstrate noteworthy differences. Building footprints are always polygons with vertices in order, whereas there is no such restriction in polygonalization. The number of vertices depicting a building is generally much less than the number of feature points in curve approximation. The objective functions and constraints are also different. In curve approximation, functions used include the number of edges, the length of perimeter, and the total area. In footprint simplification, we optimize by reducing the number of vertices while at the same time we maintain prediction accuracy of radio wave propagation. Symmetry is not required in building footprint simplification methods. Moreover, the asymmetric property can be exploited to further simplify footprints.

Polygonal simplification in computer graphics, computer-aided design, and virtual reality [33, 29] transforms a 3-D polygonal model into a simpler version by reducing the number of polygons. Adaptive subdivision algorithms commence with a simple, coarse base model, recursively subdivide it, adding more detail to local areas of the model at each step [34, 35]. Once the subdivided model approximates the original one to a user-specified degree, the algorithms terminate [35]. Geometry-removal algorithms start with the original model, simplify it by repeatedly removing faces or vertices [37, 38, 39]. Sampling-based algorithms sample the geometry of the original model by either considering a certain number of random points, or by overlaying the model with a grid and sampling each cell of the grid [43, 36]. Then, sampling-based algorithms attempt to create simplified models derived from the sampled data. Finally, hybrid-algorithms combine the above techniques to take advantage of their individual features to achieve a high quality approximation of the original model [42]. Although both polygonal and footprint simplification methods are approximations that strive to present simpler versions of initially "complex" models, they have substantial differences. In polygonal simplification, the objective function is to use as few tiles (polygons) as possible, while maintaining the approximation error within some tolerance. In our case, the respective objective function is to use as fewer edges as possible while tolerating some shape discrepancies. We can also impose constraints on centroid, area, and edges. In polygonal simplification, constraints are on curvatures, normals, and colors. Different metrics are used to quantify simplification error and define termination criteria.

Generalization in geographic information systems (GIS) deals with the creation of multi-scale representations of the same objects [44, 30]. The main constraints here are the preservation of perpendicularity, collinearity and "parallelity" [44]. These characteristics must be preserved, and sometimes enhanced or exaggerated to give observers the impression of seeing a specific object. Pertinent generalization algorithms are discussed in [45, 17, 57], least squares adjustment theory is applied for the same purpose in [44], and the use of specific rules for building facade characteristics is proposed in [46] and incorporated in modern GIS systems [40, 41]. GIS generalization and footprint simplification present key differences: a) In generalization, the objective is to present the user with a "well-formed" visual impression that inherently introduces artificial vertices to preserve or enhance the parallelity and rectangularity of a footprint. In our work, artificial vertices are not allowed. b) In footprint simplification, the constraints are imposed on area, centroid, and the length of edges whereas in generalization, the constraints are imposed on the edges in order to maintain parallelity.

Finally, an evaluation has been carried out on the influence of database accuracy on ray-tracing based radio propagation prediction in urban microcells [47]. Building databases for the same city but with different precisions are derived from different sources, including satellite maps (cadastre maps), transportation maps, and civil construction maps. In this paper, we consider data originating only from the same source; that is, we derive different maps with different accuracies from the same building database.

3 Simplification Error

The ratio of prediction times incurred by using the original and simplified maps, also termed as speedup, is used to evaluate the efficiency of a footprint simplification method. The speedup is defined as T_o/T_s, where T_o and T_s are the processing times when the original and simplified building databases are used in the ray-tracing system. In order to assess the effectiveness of footprint simplification algorithms, we correlate their work with obtained performance measurements that include mean (μ) and standard deviation (σ) of prediction errors. As radio propagation prediction time and accuracy are closely related to the nature and/or complexity of footprints, metrics on footprint simplification error can serve as reliable estimators for pertinent prediction accuracy.

The simplification error generated by a footprint simplification method can be described by the area difference map which is the symmetric difference between the original and the simplified footprints. The statistical features of area difference maps can be used to characterize the simplification errors, such as the mean and standard deviation, as well as the distribution of the area differences. Area difference map can be used to evaluate the performance of a given footprint simplification method since we can identify the following aspects: a) where the errors scatter, and what the size of the error scattering region is; b) whether or not the distribution of errors is uniform. If not, which part of the map contains most of simplification errors; and, c) what is the distribution of error sizes (granularities). It is desirable that the distribution of errors is uniform over the entire geographic area so that the distortion is similar for every footprint, which reduces the variance of simplification error. Also, it is desirable that granularity of errors is as small as possible so that distortion is not heavy at any part of footprints. Finally, footprint simplification should be "unbiased" in that no specific structural (regular) shapes appear in the area difference map. That is, it does not distorts certain types of footprints heavily, while changing others very little. Therefore, a footprint simplification method can be judged by the following conditions: 1) the distribution of errors (residuals) is random and uniform; 2) there is no structural (regular) shape in the difference map; and, 3) the granularity (size) of residuals is small, or the variance of sizes is small.

To better visualize and evaluate the difference between the centroids of simplified and original footprints, a centroid translation map can be constructed. The centroids of the original footprint and the simplified footprint are calculated for each building footprint. The displacement vectors of the centroids for all buildings constitutes the centroid translation map, from which statistics such as mean and variance of translations can be derived. From a centroid translation map, just as with the area difference map, we can calculate the distribution, granularity, orientation, mean, and standard deviation of the centroid translations. Therefore, centroid translation maps can be used as an alternative mechanism for the effectiveness of building footprint simplification methods.

4 Single-pass Footprint Simplification Algorithms

4.1 Inverse Midpoint Displacement Algorithms (IMD)

The midpoint displacement (MD) method is used in a wide variety of applications including fractal shape generation [3, 4]. Using a 2D example, MD works as follows: beginning with a triangle with vertices (x_1, y_1), (x_2, y_2) and (x_3, y_3) each edge is subdivided into two halves, moves the midpoint some distance in the plane, and yields a hexagon during the first iteration. The location for the new midpoint of the edge

(x_i, y_i) to (x_j, y_j) is calculated as $x_n = (x_i + x_j)/2$ and $y_n = (y_i + y_j)/2 + P(x_j - x_i)R(x_n)$, where $P(\cdot)$ is a perturbation function determining the extent of the midpoint movement, and $R(\cdot)$ is a random number generator. Should the above procedure be iterated a number of times, then a fractal object is formed [3]. The objects derived for three, four and seven iterations of the MD when applied to initial triangle of Figure 1 delineated by (0.0, 0.0), (10.0, 0.0), and (5.0, 5.0), are shown in Figures 2 to 4. Here, the perturbation function $P = x_j - x_i$, the random function $R = rand/len$, $rand$ is a random number in the range of $(0.0, len)$, and $len = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$. Figure 4 clearly demonstrates a spiky behavior that exposes the self-similarity property of the object.

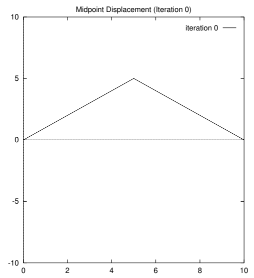

Figure 1: Midpoint Displacement (iteration 0)

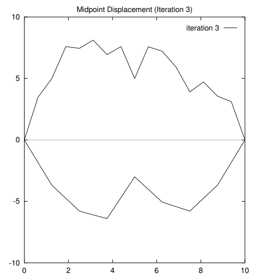

Figure 2: Midpoint Displacement (iteration 3)

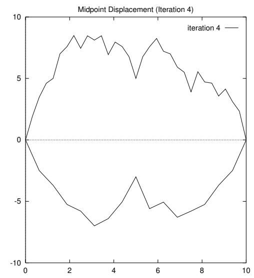

Figure 3: Midpoint Displacement (iteration 4)

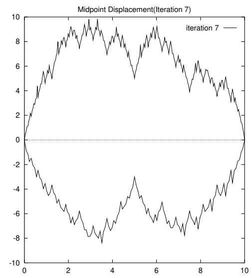

Figure 4: Midpoint Displacement (iteration 7)

In our proposed Inverse Midpoint Displacement algorithm (IMD), we take the inverse approach to the MD method; we remove points from the initial two-dimensional shape and transform an irregular object to a simplified and more regular one. Should we know one "original" point of the shape, the perturbation function as well as the random function used, we could easily derive the exact original shape. In general, such an option is not available. Therefore, we have to allow some approximation for the two functions (perturbation and random functions) and randomly select one point (that is assumed to be the anchor for the simplified shape). Although footprints do not possess the exact self-similarity property, it is reasonable to assume that they demonstrate statistical self-similarity property, that is, some parts of a footprint are statistically similar to the other parts of the same or different footprints in terms of orientation, length distribution, and slope distribution [55, 56].

To describe the IMD method, we define a basic concept termed *capture region* of a line segment l. Assume that the length of a line segment l is L. A strip is defined around l whose central axis is l and whose width W is a fraction of L, that is, $W = f_w L$, where f_w is the width factor. The area covered by this strip is termed the $capture\ region$ of l. The main idea of IMD algorithm is that vertices that happen to be located within a specific *capture region* can be eliminated. More specifically, for each vertex in the polygon-to-be-simplified, we construct the capture region for the line segment l formed by its prior vertex and successive vertex in the polygon[1]. If this vertex is within the capture region of l, then it is considered to be a "midpoint" of l and thus it can be removed. The latter offers the basic mechanism for the simplification of the shape in question. Algorithm 1 shows the steps of the IMD algorithm.

The running time of Algorithm 1 is linear in the number of vertices (n) in the initial shape (i.e., $O(n)$). There are two issues that affect the quality of the simplification process; they are: a) the selection of the

[1]These vertices can be easily determined as our shapes are defined with a counter-clockwise orientation for the outer boundary, and a clockwise orientation for the inner boundary (hole).

Algorithm 1 Inverse Midpoint Displacement Algorithm (IMD)

```
A starting vertex S is chosen from those vertices comprising the footprint F.
Insert all the vertices of F starting from S into a queue Q.
Let the set of vertices for the simplified version, V, to empty.
while (Q is not empty) do
   Remove vertex S from the head of Q.
   P ← predecessor_vertex(S); N ← successor_vertex(S); form the capture region for (P, N).
   if (S falls outside the capture region for (P, N)) then
      Insert vertex S into the set V.
   end if
end while
The polygon formed by the vertices in the ordered set V is the final simplified version of the polygon.
```

starting point and b) the designation of the width of the capture region. It is beneficial if the starting point is an actual midpoint in the object. We use two heuristics that attempt to designate midpoints as starting points with high probability. These heuristics are: 1) select a vertex from the shortest edge in the polygon and 2) select the vertex which has the maximum inner angle. Alternatively, the easiest way to select a starting point is by randomly picking one vertex in the given polygon. The width factor f_w of the the capture region is usually selected from the range [0.10, 1.00]. Anything higher than 1.00 would indicate a very aggressive simplification route. In this case, the shape of a footprint may change dramatically and may affect the accuracy of the radio propagation prediction. To avoid this, a limit can be imposed on the maximum number of vertices which can be removed consecutively from a footprint.

Figure 5 depicts a polygon obtained after applying our IMD algorithm in the polygon that appears in Figure 3. The width of capture region is set to 0.75. If we compare Figures 5 and 2, we can verify that the two polygons are very similar. In addition, the vertex reduction rate is 29% (or 34 edges down from 48 in the initial polygon of Figure 3).

4.1.1 Constrained Inverse Midpoint Displacement Algorithm (CIMD)

Our IMD algorithm above considers only one building at a time. In an urban terrain, one may need to simplify a number of buildings at the same time. The simplification of a footprint may affect one or more neighboring buildings. For instance, although all the vertices in a simplified polygon are also the vertices of the original one, some edges of the simplified footprint may very likely be different from those of the original polygon. This happens due to the introduction of artificial edges necessary for the simplification. The latter may intersect with the edges of neighboring footprints or polygons. In this case, it is imperative that simplified polygons still obey “natural” constraints, e.g., a building cannot overlap others.

In order to satisfy natural constraints, we maintain an outer as well as one (or more) inner boundaries for each building. All vertices and edges of a simplified footprint should lie inside the outer boundary and outside of inner boundary(ies). When simplifying, the algorithm should check that each artificial edge in the simplified building lies within the area formed between the inner and outer boundaries. The simplest and more accurate way to define the outer boundary is the area surrounded by the original polygon. In this case, we modify our IMD algorithm to account for the above constraint. The resulting algorithm, the constrained inverse midpoint displacement (CIMD), is depicted in Algorithm 2. The running time of CIMD depends on the shapes of the given outer and inner boundaries. If the outer contour of the original footprint is used

Algorithm 2 Constrained Inverse Midpoint Displacement Algorithm (CIMD)

```
1: A starting vertex S is chosen from those vertices comprising the footprint F.
2: Insert all the vertices of F starting from S into a queue Q.
3: Set the group of vertices for the simplified version, V, to be empty.
4: while (Q is not empty) do
5:     Remove vertex S from the head of Q.
6:     P ← predecessor_vertex(S); N ← successor_vertex(S); U is the last vertex in V; form the
       capture region for (P, N).
7:     if (Edge formed by vertices S and U is within specified outer and inner boundaries) ∧ (S falls outside
       the capture region of (P, N)) then
8:         Insert vertex S into V.
9:     end if
10: end while
11: The polygon formed by the vertices in the set V is the final simplified version of the polygon.
```

as the only outer boundary and no inner boundary is imposed, then the conditions of line 7 (Algorithm 2) can be evaluated in constant time rendering CIMD's complexity linear (i.e., $O(n)$ where n is the number of vertices in the initial footprint).

Figure 6 depicts the result of the CIMD algorithm for the polygon in Figure 3 with the original footprint as the outer boundary (i. e., all edges of the simplified polygon should be within the outer contour of the original footprint). We verify that the required constraint is satisfied by the simplified version. In this example, we do not consider any inner boundaries as the initial shape is solid. The vertex reduction rate achieved by CIMD algorithm is 22.92% and the number of edges has been reduced to 37 (down from 48). The constraint-based algorithm produces a larger number of edges if compared with the results of Algorithm 1. However, the resulting simplified footprints are more accurate in reference to the original polygon.

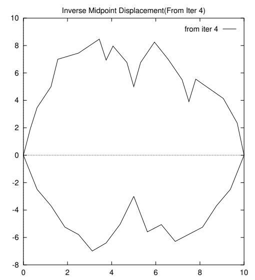

Figure 5: Simplified result by IMD for the polygon in Fig. 3

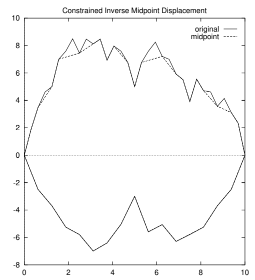

Figure 6: Simplified result by CIMD for the polygon in Fig. 3

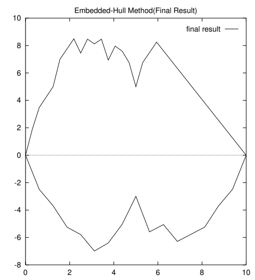

Figure 7: Simplified result by CHM method for polygon in Fig.3

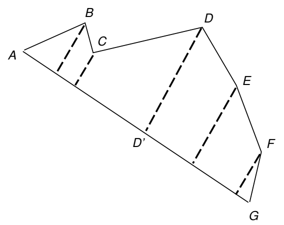

Figure 8: Recursive subdivision on the curve $ABCDEFG$

To satisfy additional constraints on area and centroid, backtracking should be used. First, a simplified footprint is found as usual. Then, the area and centroid of the simplified footprint are tested against the constraints. If constraints are not satisfied, the algorithm steps back one vertex and finds another feasible route. This procedure is repeated until a solution is found, or all possible paths have been tried.

4.2 Recursive Subdivision Methods (Res-FPS/Res-OCS)

In this section, we introduce algorithms based on the iterative end-point fit method or recursive subdivision method [17] with the help of Figure 8. Here the curve is represented by points from A to G, which are the input to the algorithm. The output of the algorithm is a group of connected line segments that collectively approximate the initial curve. We commence by connecting the two (provided) end-points of the set, which is the line (A, G) in our example. Then the point which has the largest perpendicular distance from the line segment (A, G) in the feature points is found (i.e., point D). If this largest distance $(DD^{'})$ is above a given threshold (set at run time by the user), then D is a vertex in the approximated curve. The new vertex D will form two new line segments, (A, D) and (D, G). These two new line segments replace our initial approximation line segment (A, G). The algorithm proceeds recursively with the new line segments (A, D) and (D, G) and the remaining feature points are partitioned into two pertinent sets. The algorithm terminates when the distance of each feature point to its nearest approximation line segment is below a given threshold. We can modify the above baseline algorithm to approximate polygons and offer simplification of building footprints. The key assumptions in [17] that curves are not of closed form and feature points are not necessarily provided in order do not apply here. These differences affect the selection of objective function, approximation error metrics, and the termination criteria for the approximation method.

In our algorithm, we first need to split the input polygon into two curves by selecting two footprint vertices. The core problem is how to select the above two vertices. One choice is to pick a pair of vertices that are located the farthest apart. The rationale for this choice is the resulting partition will yield two almost-equally populated sets of vertices and it will help maintain the shape of the original footprint. Once the two curves have been determined, the recursive subdivision algorithm can be individually applied to each one and compile the two simplified poly-lines. The synthesis of the two resulting poly-lines will offer the final simplified footprint. In order to locate the farthest located pair of points, we first derive the convex-hull of all given points of our polygon and then we use a divide-and-conquer algorithm [18] to identify the pair in question. To form the convex-hull we need $O(n)$ computational steps for a simple polygon (no self-intersection) with n vertices. The final selection of the farthest pair requires $O(k)$ steps where k is the number of the vertices in the convex-hull ($k \leq n$) [6, 18]. Therefore, the overall computation complexity to find the farthest-pair is $O(n)$.

Our proposed method (Res-FPS: recursive subdivision – farthest-pair split), is depicted in Algorithm 3. The termination criteria used in Res-FPS can be:

1. Distance-based: the perpendicular distance $D(V, L)$ of each vertex V to the nearest line segment L in the simplified curve should obey the inequality $D(V, L) \leq f_d L$ where f_d is the distance factor and has a value in the (0, 1) range.
2. Area-based: the total area of the closed contour formed by the curve and the line segment connecting the end-points of the curve should be less than the provided threshold.

The running times for worst-case and average-case of Res-FPS algorithm are $O(n^2)$ and $O(n \log(n))$, respectively. These results can be derived using a similar approach to that of Appendix A. Optimization techniques to speed up the worst-case running time to $O(n \log(n))$ are similar to those discussed in [57].

Figures 9-11 outline the application of the Res-FPS algorithm to the polygon of Figure 3. The utilized termination criterion is the distance-based criterion with its corresponding distance factor $f_d = 0.15$. The number of vertices in the simplified footprint is 32 (down from 48) and the vertex reduction rate is 33%.

Algorithm 3 Recursive Subdivision Algorithm-Farthest Pair Split (Res-FPS)

```
for (Each vertex V in the given footprint F) do
    V.flag ← false.
end for
Find the farthest-pair vertices, X and Y, among the vertices in F.
X, Y divide the polygon into two curves C_1 and C_2;    insert C_1 and C_2 into a queue Q.
X.flag ← true; Y.flag ← true.
while (Q is not empty) do
    Remove the curve C from Q.
    if (Termination Criteria for C not Satisfied) then
        C_s ← the first end-point of C; C_e ← the last end-point of C.
        Find the farthest point Z from the line C_sC_e
        Q ← Q ∪ {C_sZ} ∪ {ZC_e};    Z.flag ← true.
    end if
end while
All vertices with flag = true form the simplified version of F.
```

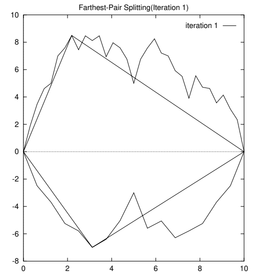

Figure 9: Result in iteration 1 by Res-FPS method for polygon in Fig. 3

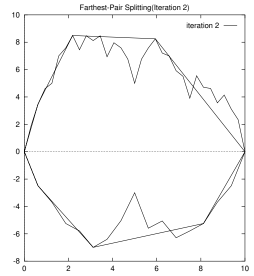

Figure 10: Result in iteration 2 by Res-FPS method for polygon in Fig. 3

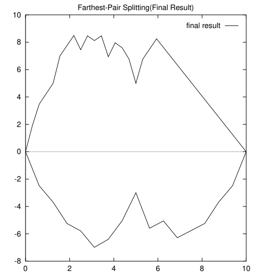

Figure 11: Final simplified result by Res-FPS method for polygon in Fig. 3

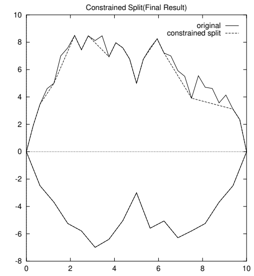

Figure 12: Simplified version by constrained Res-FPS for polygon in Fig. 3

Another way to split the polygon is to find that pair of vertices which are located at "opposite corners". In particular, we can first find four vertices $V_{ul}, V_{ur}, V_{ll}, V_{lr}$, where V_{ul} is the upper-left, V_{ur} is the upper-right, V_{ll} is the lower-left, and V_{lr} is the lower-right vertex. Then we can compare line segments $(V_{ul}V_{lr})$ and $(V_{ur}V_{ll})$ and use the longer to partition the curve, we name this method Res-OCS (Recursive subdivision – Opposite Corner Split). Similarly, we can construct the constrained recursive subdivision algorithms by requiring that all edges of simplified version must be between the given outer boundary and inner boundary. The result of such a constrained Res-FPS algorithm is shown in Figure 12 for simplification of the polygon in Figure 3. From this, we can establish that the simplified version satisfies the constraint, and the number of vertices in the final result is 34 with vertex reduction rate at 29.17% (which is less than the rate of the unconstrained version 33%).

4.3 Maximum Triangulation Area Method (MAT)

Tessellation uses polygonal meshes to describe surfaces and objects. As polygonal meshes demonstrate good connecting structure and can be manipulated efficiently, they are popular in object representation, image processing, and virtual reality applications [29, 33]. The core objective in handling meshes is to reduce their complexity by using simple and similar-type tile elements for mesh construction. This is done by reducing the number of tiles in the mesh while maintaining acceptable accuracy [31, 32]. Triangulation is the most popular tessellation method and can help us in simplifying building footprints. The rationale of our proposed algorithm here is based on two premises: a) triangulation of the building footprint, and b) for triangles adjacent to the perimeter of the footprint, check whether they satisfy imposed elimination criteria. If so, triangles are removed yielding a simpler footprint.

Conventional triangulation methods [29, 18] do not comply with a number of properties deemed necessary in footprint simplification. First, the number of triangles along the boundary should become potentially very large. In turn this will increase the probability that more triangles will be ultimately eliminated. Second, the shapes of triangles along the perimeter either should occupy small areas or should be long and skinny [33]. Finally, the aggregation of all internal triangles should maintain the original shape with high fidelity. While taking into consideration the above requirements, our proposed triangulation method works along the following lines. The triangulation process is conducted starting from the internal area of the polygon and moving toward the boundary of the footprint. The "internal" formed triangles should cover as large areas as possible. Our conjecture is that this will help maintain the shape of the original footprint. Before further triangulation of an area, it is beneficial to check whether this shape satisfies termination/elimination criteria.

There are two drawbacks if we proceed with a straightforward implementation of the above procedure: first, the running time is $O(n^3)$ where n is the number of vertices, and second such an implementation may generate (undesirable) "wedge" triangles[2]. This computational complexity is due to the fact that all possible triangles have to be examined in every iteration of the algorithm. Wedge triangles cannot be removed since they may partition a polygon into two others that have only one single common vertex. Should a large number of wedges be created (during triangulation), the vertex reduction rate would be adversely affected.

In order to overcome these problems, we modify our triangulation method as follows:

- **(a)** Find the longest edge L in the footprint.
- **(b)** Among all the triangles which contain L as one of their edges, find a triangle T that has the maximum area. We use T as the working simplified footprint, and denote the edges of T as L, L_1, and L_2.
- **(c)** The triangle T in discussion partitions the footprint into two distinct polygons containing L_1 and L_2 respectively. For each of these two polygons, we repeat step **(b)** until the termination/elimination criteria are satisfied.

The termination/elimination criteria in step **(c)** are that the area occupied by the polygon is less than a given threshold, and/or the aspect ratio α_r[3] of the polygon is very small (compared with a pre-specified threshold). In our algorithm, it is evident that no wedges are generated as there is no gap between triangles created in consecutive steps and triangles share a common edge. We term our proposed algorithm the Maximum-Area Triangulation (MAT) method (depicted in Algorithm 4). It can be shown that the worst-case and average-case running times of MAT are $O(n^2)$ and $O(n\log(n))$, respectively (see Appendix A for the derivation).

Figures 13 and 14 demonstrate the operation of the MAT method applied to the polygon of Figure 3 with aspect ratio α_r=0.15. In the final simplified footprint (Figure 15), there are 24 vertices (and with vertex reduction rate at 50%). In this algorithm, each triangle generated occupies as large an area as possible.

[2] A wedge is a triangle which is between two other "large" triangles.

[3] The aspect ratio α_r of a polygon is defined as the ratio of shortest over the longest side of the tightest rectangle that encloses the polygon.

Algorithm 4 Maximum Area Triangulation Method (MAT)

for (Each vertex V in the given footprint F) **do**
  $V.flag \leftarrow false$.
end for
Find the longest edge, (V_i, V_{i+1}), in F.
Find V_j such that triangle (V_i, V_{i+1}, V_j) has the maximum area among all triangles (V_i, V_{i+1}, V_k), for $k = 1, 2, ..., n$ (but $k \neq i, i+1$).
F is divided into two curves $C1$ and $C2$: $C1_s \leftarrow V_j, C1_e \leftarrow V_i$; $C2_s \leftarrow V_{i+1}, C2_e \leftarrow V_j$;
// subscripts s and e indicate start and end points respectively.
Insert tuples (V_i, V_j, C_1) and (V_j, V_{i+1}, C_2) into a queue Q.
$V_i.flag \leftarrow true$; $V_{i+1}.flag \leftarrow true$; $V_j.flag \leftarrow true$.
while (Q is not empty) **do**
  Remove the tuple (V_s, V_e, C) from Q
  if (Termination/elimination Criteria for C not Satisfied) **then**
    Find V such that triangle (V_s, V_e, V) has the maximum area among all triangles (V_s, V_e, V), where V is in C (but $V \neq V_s, V_e$).
    C is divided into two curves $C1$ and $C2$: $C1_s \leftarrow V$, $C1_e \leftarrow V_s$; $C2_s \leftarrow V_e$, $C2_e \leftarrow V$.
    Insert tuples (V_s, V, C_1) and (V, V_e, C_2) into Q.
    $V.flag \leftarrow true$.
  end if
end while
All vertices with $flag = true$ form the simplified version of F.

This helps in achieving higher vertex reduction rate and maintain the original shape with high probability. To corroborate the above claim, we generate each triangle randomly, with the result shown in Figure 16. In Figure 16, the number of vertices is 34 and the vertex reduction rate is 39%, much less than the one produced by the MAT method (50%). This observation points to the fact that the MAT algorithm is preferable to the randomized one.

4.4 Convex-Hull Based Method (CHM)

The convex hull, H, of a set of points, Q, is the smallest convex polygon for which each point in Q is either on the boundary of H or in its interior. A convex hull for a simple polygon features two properties that

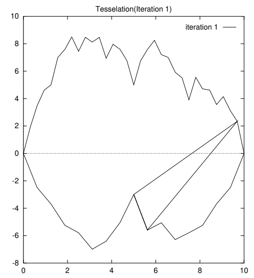

Figure 13: Result in iteration 1 by MAT for polygon in Fig. 3

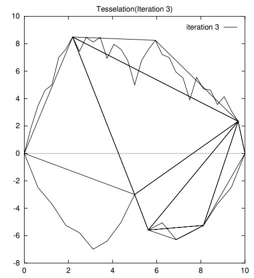

Figure 14: Result in iteration 3 by MAT for polygon in Fig. 3

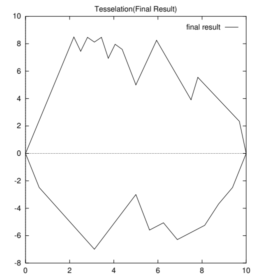

Figure 15: Simplified result by MAT method for polygon in Fig. 3

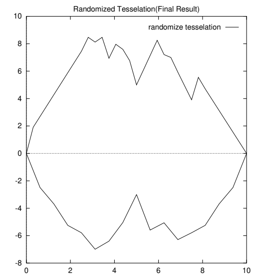

Figure 16: Simplified result by randomized triangulation for Fig.3

a footprint simplification algorithm might need: first, the vertices of H are a subset of the vertices of the original polygon; and second, all edges of the original polygon are within the convex hull H. A popular convex-hull based method used in digital curve approximation is the *minimax* algorithm [13]. *Minimax* tries to find a polygon, F, which not only has minimum number of edges but also features a maximum distance between the edges of the polygon and the data points being less than a given threshold. Subsequently, the *minimax* uses F to approximate the original digital curve. The above procedure cannot be directly used in footprint simplification as it introduces artificial vertices and the constraints may not suffice. For instance, besides the constraint used in minimax, we still need to keep long edges, limit the change of area, restrict the movement of centroid of the simplified footprint. Another convex-hull-based method is introduced in [15]. A set of points can be approximated by a line segment if a strip exists which contains all the points and is not wider than a given value. A convex hull is constructed each time a new point is added to the current set. When a strip satisfying the above condition no longer exists, the longest segment is found, that is, a line joining the extreme points of the last accepted strip. Although this method does not introduce artificial vertex in the simplification process, it may not retain "needed" long edges in the building footprint.

Our convex-hull-based simplification method (CHM), shown in Algorithm 5, is derived from [15]. However, some key changes are: 1) start from the longest edge in the given footprint; 2) keep all long edges in the original footprint; 3) construct the strip enclosing all points in the current set by using the line segment (V_s, V_e) as its axis, where V_s and V_e are the starting and end points for the curve in the current set (the width of the strip is set to $\alpha_s len(V_s, V_e)$, where α_s is the strip factor); and 4) use the axis of a strip when designating a line segment to approximate a curve in the current set instead of the line formed by extreme points. This convex-hull-based footprint simplification algorithm (CHM) has computational complexity of $O(n^2)$ in the worst case. The rationale is as follows. For each new vertex V to be added to the current set S, we first check whether the polygon formed by V and vertices in S is a simple polygon. This can be achieved by testing if edge (V, V_s) and edge (V, V_e) intersect any edge in S, where V_s and V_e are the starting and ending vertices in S. There are k - 1 edges in S, where k is the number of vertices in S, therefore, this step has complexity of $O(k)$. If P and S form a simple polygon, we then construct the convex hull containing P and S, which takes $O(k)$ computational steps. It is clear that the complexity of CHM is $O(\sum_{k=1}^{n} k) = O(n^2)$, where n is the total number of vertices in the original footprint. The result by applying CHM method to the footprint in Figure 3 is shown in Figure 7. The strip factor is α_s=0.15. The number of vertices in the simplified footprint is 32 and the vertex reduction rate is at 33.33%.

5 Multi-pass Simplification Algorithms with Multiple Constraints

The algorithms introduced thus far try to simplify footprints while attempting to satisfy constraints on edge, area, and centroid. However, single-pass methods may not be capable of considering all these constraints simultaneously if backtracking techniques are not used. For example, the centroid of a simplified footprint can only be computed once all its vertices are determined. If the centroid of the simplified polygon does not satisfy the given constraint, we need to backtrack and find a different route to proceed. Therefore, single-pass methods can only satisfy constraints on edges. Backtracking is in general time consuming. By taking advantage of the asymmetric nature of the methods presented so far, we are able to develop multiple pass simplification algorithms that not only make use of multiple constraints simultaneously but also avoid backtracking altogether. The rationale of multi-pass footprint simplification methods is rather straightforward. We first instantiate any of the proposed algorithms with a number of different starting

Algorithm 5 Convex-Hull Based Simplification Method (CHM)

Find longest edge (V_{s-1}, V_s) in given footprint F.
$V_{s-1}.flag \leftarrow true$, $V_s.flag \leftarrow true$, $S \leftarrow V_s$.
Put all other vertices of F starting at V_{s+1} into a queue Q and set their corresponding flags to $false$.
while (Q is not empty) **do**
    Remove the vertex V_i from Q and let V_s and V_e be the first and last vertices in S.
    if ((Edge (V_{i-1}, V_i) is a long edge) $\vee$ (edge (V_i, V_s) or (V_i, V_e) intersects any edge in S)) **then**
        $V_{i-1}.flag \leftarrow true$, $V_i.flag \leftarrow true$, $S \leftarrow V_i$.
    else
        $S_1 \leftarrow S \cup V_i$, form convex-hull, H, for S_1.
        if (H satisfies specified constraints) **then**
            $S \leftarrow S_1$.
        else
            $V_{i-1}.flag \leftarrow true$, $S \leftarrow \{V_{i-1}, V_i\}$.
        end if
    end if
end while
All vertices with $flag = true$ form the simplified footprint for F.

conditions (or values). For every such value, the resulting simplified footprint is checked against all the imposed constraints. If the latter are satisfied, the solution at hand becomes a candidate in the result set. The final result is the best candidate in the result set. Multi-pass algorithms avoid backtracking, may satisfy all constraints on edge, area, and centroid, and may improve vertex reduction rate as the final result is chosen from many candidates.

Constraint on area can help maintain the shape of a footprint. Intuitively, if large area differences exist between original and simplified building footprints, significant discrepancies in the numbers of intercepted rays that ultimately need to be investigated will appear. The latter will certainly affect the prediction accuracy. Area constraint is described as $(1-f_a)A_{old} \leq A_{new} \leq (1+f_a)A_{old}$, where A_{old} and A_{new} are areas of the initial and simplified footprints, and f_a is the "area factor". Another key feature for the shape of a footprint is its centroid point. Centroid change always implies changes in footprint's geometry. Consequently, directions of radio rays reaching the footprint in question as well as unfolded lengths of ray paths formed between transmitters and receivers via this footprint will be affected. This will influence the accuracy of radio prediction. The centroid constraint can be described as $\sqrt{(x_{new} - x_{old})^2 + (y_{new} - y_{old})^2} \leq f_c r$, where (x_{old}, y_{old}) is the x-coordinate and y-coordinate for the centroid of the original footprint, (x_{new}, y_{new}) is the x-coordinate and y-coordinate for the centroid of the simplified building footprint, r is a system parameter, and f_c is the "centroid factor". We can designate r to be a fraction of the minimum side of the bounding box for the original footprint. That is, $r = min(BB_{length}, BB_{width})$, where BB_{length} and BB_{width} are the length and width of the bounding box for the original footprint.

Single-pass algorithms may designate different initial points to commence their work. In multi-pass algorithms, there is a variety in selecting such initial points. For instance in a Multi-pass Inverse Midpoint Displacement (MIMD) algorithm, m initial vertices can be selected in one of the following ways ($m \leq n$, where n is the number of vertices in the given footprint F): a) Pick the first m vertices that are connected to the shortest edges; b) Choose the first m vertices that have the maximum inner angles; c) Find the endpoints (v_i, v_{i+1}) of the shortest edge, select those m vertices beginning from v_i; and d) Pick m vertices randomly.

The MIMD algorithm with multiple constraints is shown in Algorithm 6 and has complexity $O(nm)$.

Algorithm 6 Multi-pass Inverse Midpoint Displacement with Multiple Constraints (MIMD)

```
Select m number of vertices from the given footprint F, where m ≤ n, and n is the number of vertices
  in F.
Insert these m vertices into a queue Q.
Set S to be F, where S is the best solution so far.
while (Q is not empty) do
    Remove vertex V from queue Q.
    R ← Inverse_Midpoint_Displacement(V).
    if ((R satisfies all the constraints) ∧ (R is better than S in terms of vertex reduction rate)) then
        S ← R.
    end if
end while
Return S as the solution.
```

We demonstrate the operation of multi-pass footprint simplification algorithms MIMD with the help of footprint map of Rosslyn, VA which has 79 buildings and 412 vertices (shown in Figure 17). First, we run the single-pass IMD algorithm by keeping all long edges with a global threshold (i.e., average length of all edges for all footprints in the building database). Figure 18 depicts the resulting simplified map. We then run the MIMD algorithm with constraints on edges only. Figures 19 and 20 show the simplified footprints, respectively, by using global threshold and local threshold (i.e., average length of all edges for the footprint currently processed). Finally, the MIMD algorithm is run with additional constraints on area (f_a = 0.25) as well as on centroid (f_c = 0.10), while global threshold is used to keep long edges. The simplified footprints are shown in Figures 21 and 22.

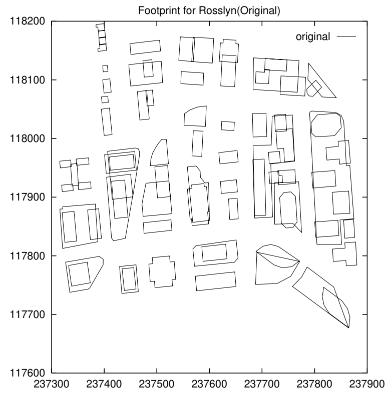

Figure 17: Original building Footprints for downtown Rosslyn, VA

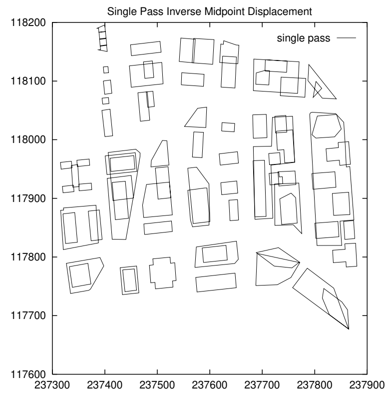

Figure 18: Result of single-pass IMD method for Rosslyn in Figure 17

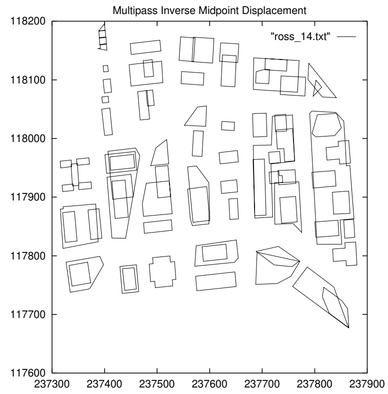

Figure 19: Result of MIMD with global constraint for Rosslyn in Figure 17

It can be seen that the single-pass IMD method reduces the number of vertices to 375 providing a 8.98% vertex reduction rate. There are 367 and 376 vertices in the simplified maps for global threshold and local threshold cases when the MIMD method is used. The corresponding vertex reduction rates are 10.92% and 8.74%. Evidently, MIMD with global threshold achieves the best vertex reduction rate. The simplification appears to be very aggressive when constraints are imposed only on edges. For example, some buildings at the upper right corner and in the middle are distorted heavily in Figure 18. Similarly, some buildings at the

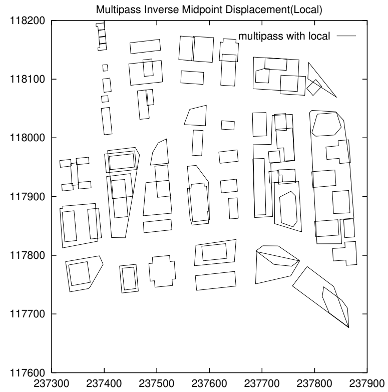

Figure 20: Result of MIMD with local constraint for Rosslyn in Figure 17

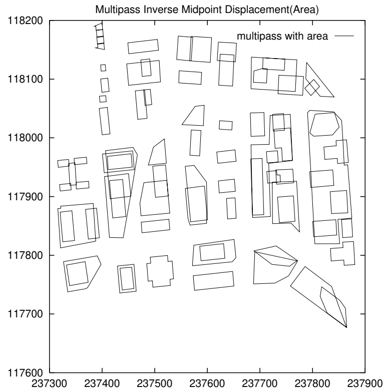

Figure 21: Result of MIMD with area constraint for Rosslyn in Figure 17

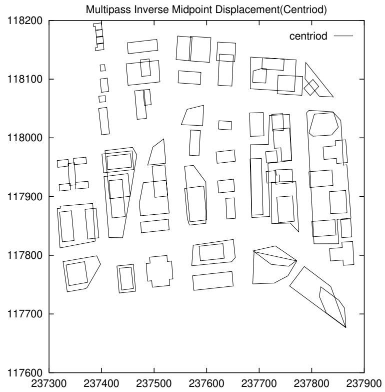

Figure 22: Result of MIMD with centroid constraint for Rosslyn in Figure 17

upper left, upper right corner, and in the middle are changed dramatically in Figures 19 and 20. The number of vertices in the simplified map is 370 (with vertex reduction rate of 10.19%) when area constraint is used, while it is 372 (with vertex reduction rate of 9.71%) when the centroid constraint is used. The resulting shapes of simplified footprints appear to be improved when multiple constraints are in use, although the vertex reduction rates are compromised.

Figure 23 shows the area difference map between Figure 17 and Figure 19 generated by MIMD method. The mean and standard deviation for area differences shown in Figure 23 are 24.69 m^2 and 71.83 m^2, respectively. The corresponding distribution of area difference is shown in Figure 26.

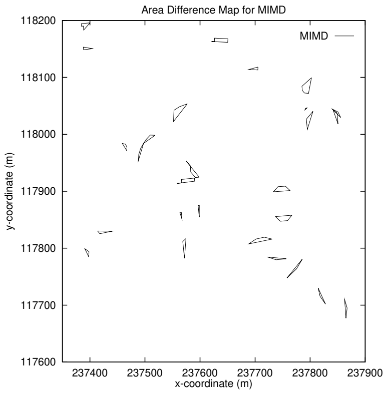

Figure 23: Area difference map by MIMD method with constraint on edges for Rosslyn

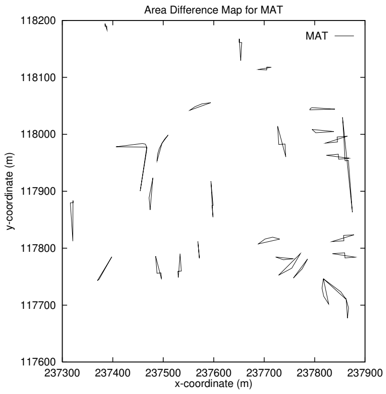

Figure 24: Area difference map by single-pass MAT method for Rosslyn

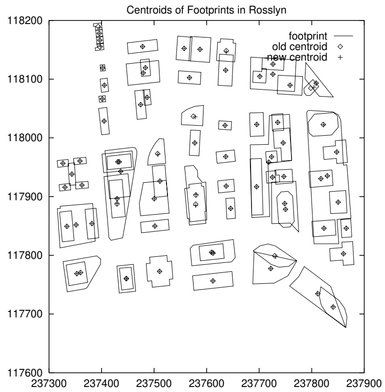

Figure 25: Centroid translation map by MIMD method for Rosslyn

To demonstrate the differences in the performance between the MIMD and MAT algorithms, we also create area difference maps for MAT method, as shown in Figure 24. Its distribution of area differences is depicted in Figure 27 and its mean and standard deviation are 23.86 m^2 and 67.40 m^2, respectively. Taking into account Figures 23, 24, 26, and 27, one can establish that although the error scattering regions for the MIMD and MAT algorithms is similar, the distributions of residuals are different. Errors are distributed

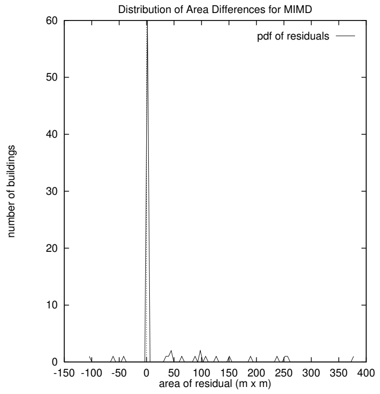

Figure 26: Distribution of area errors by MIMD with constraint on edges for Rosslyn

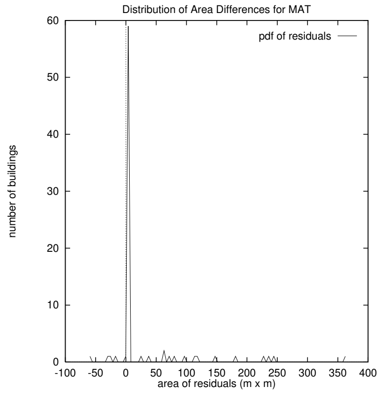

Figure 27: Distribution of area errors by single-pass MAT for Rosslyn

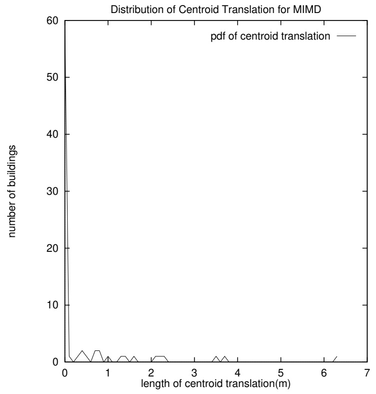

Figure 28: Distribution of centroid translations by MIMD for Rosslyn

more randomly in MAT than MIMD. The mean values of simplification errors for the above two algorithms are almost identical while the standard deviation for MAT is smaller than MIMD. All the above point into the fact that MAT's performance is superior to MIMD's for the building footprints in Rosslyn, VA.

Figure 25 and 28 respectively depict the centroid translation map and the distribution of incurred errors by MIMD for Rosslyn, VA. It is evident that the changes in the centroids for all footprints are very small, there is no large centroid translations (more than 75% of them are less than $1m$), and the orientations of centroid translations are random.

6 Hybrid Simplification Methods

By examining area difference and centroid translation maps, one can qualitatively establish that certain algorithms can better simplify different types of buildings than others. This observation indicates that two simplification methods can demonstrate a complementary relationship in which one performs better on some types of footprints while the second yields enhanced results in different types of polygons. Thus, combining two (or more) complementary footprint simplification methods is expected to offer better performance. However, if the selected algorithms are not complementary, the gains obtained by the resulting hybrid form will be negligible. It is therefore essential that we propose a method to determine whether two candidate algorithms make up complementary parties in a hybrid method. If we consider two simplification methods A and B, then such a process can function as follows:

- Construct area difference maps D_A and D_B for A and B methods respectively.
- Generate the area difference map M between D_A and D_B.
- If the sum of areas appearing in M is large comparing with the individual sum of areas in D_A and D_B, then algorithms A and B are complementary.

By combining two (or more) complementary simplification algorithms, the hybrid algorithm runs each individual algorithm on a building by building basis and selects the results which display the best vertex reduction rates.

We use the map for Rosslyn as our demonstration vehicle and consider all two-algorithm combinations out of three candidate algorithms: a) multi-pass inverse midpoint displacement (MIMD) with area constraint, b) multi-pass farthest-pair split (MFPS), and c) multi-pass maximum-area triangulation (MMAT). To determine whether or not these algorithms are complementary, we construct area difference maps for these three algorithms and show in Figures 29, 30, and 31 respectively. From these three area difference maps, we can

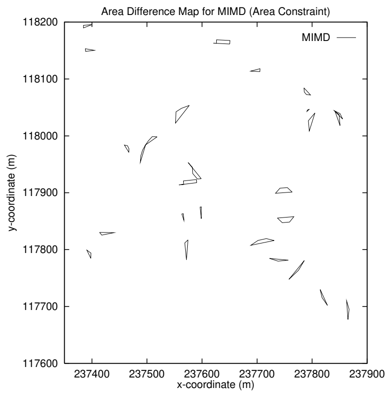

Figure 29: Area different map by MIMD method for Rosslyn

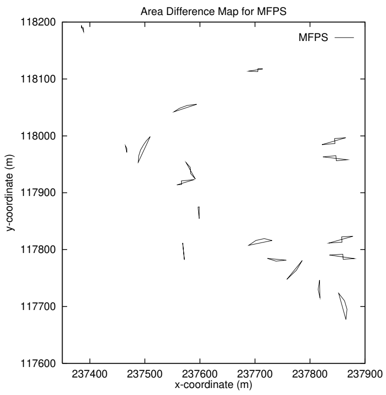

Figure 30: Area different map by MFPS method for Rosslyn

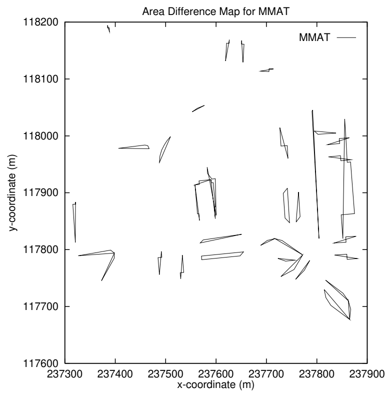

Figure 31: Area different map by MMAT method for Rosslyn

establish that there are overlaps among all three maps indicating that these methods have similar simplification performance on some types of building footprints. The distributions of simplification errors for MIMD and MFPS are quite different. In contrast, the area difference map for MFPS is a subset of that for MMAT. The granularity of the simplification errors for MMAT is quite different from those for MIMD and MFPS, and MMAT has the largest mean and standard deviation of simplification errors. These observations lead us to anticipate that performance improvement will be better facilitated if we combine MIMD and MFPS than MFPS and MMAT. Also, hybrid MIMD-MFPS will do better than its constituent components alone.

The simplified footprint maps generated by MIMD-MFPS, MIMD-MMAT, and MMAT-MFPS hybrid methods are shown in Figures 32, 33 and 34. The vertex reduction rates for these three multi-pass footprint

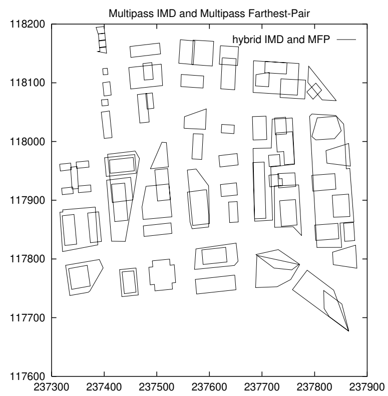

Figure 32: Simplification result by hybrid MIMD-MFPS for Rosslyn

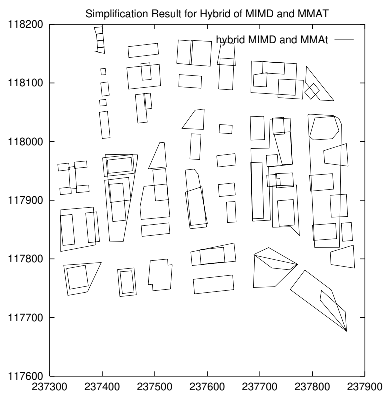

Figure 33: Simplification result by hybrid MIMD-MMAT for Rosslyn

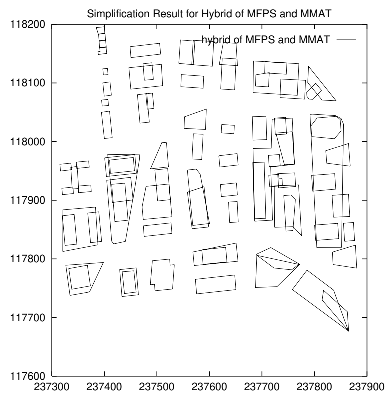

Figure 34: Simplification result by hybrid MFPS-MMAT for Rosslyn

simplification methods and their hybrid methods can be found in Table 2. The hybrid MIMD-MFPS gives the best improvement margin in terms of vertex reduction rate (from 10.19% achieved by MIMD alone to 12.86%) since these two algorithms demonstrate better complementary simplification preferences as discussed before. However, the hybrid MIMD-MMAT method gives the best absolute vertex reduction rate (17.72%). The MMAT and MFPS methods have similar instead of complementary simplification preferences, the combination of which does not offer much improvement (from 16.02% achieved by MMAT alone to 16.26%).

7 Performance Evaluation

In this section, we experimentally investigate the performance of the proposed building footprint simplification algorithms using a diversified set of measurements that include vertex reduction rate, simplification errors on edges, area, and centroid. More importantly, we assess the effects of utilized footprint simplification methods on the radio propagation prediction time and prediction error in the form of mean and standard deviation. The sensitivity of prediction accuracy to the distortion of building footprints is also examined in detail. All experiments are conducted on a single-CPU Sun Ultra 10 workstation with clock rate of 440 MHz, main memory of 384 MBytes, and swap area of 524 Mbytes. The operating system is Solaris 5.7. All algorithms are implemented by using C programming language and Perl script language.

7.1 Baseline Experiments on Footprint Simplification Methods

In our evaluation we use maps from three municipalities: Rosslyn, VA, Dupont Circle in Washington DC, and Turin, Italy. The city map of Rosslyn, VA has been simplified manually by eliminating very small buildings and unnecessary details making the overall building footprints appear "regular" (Figure 17). There are 79 building footprints that display 412 vertices. The buildings have from 4 to 13 vertices each, with an average of 5. The distribution of sizes in footprints is shown in Figure 35. The areas of footprints range from 94.06 m^2 to 15,052.55 m^2, with an average of 1,683.64 m^2, and the area distribution is depicted in Figure 36. The length of footprint edges are within the range of [2.17, 223.77] m, with an average of 29.83 m, and its distribution can be better viewed in Figure 37. Figure 38 shows the distribution of slopes for all edges in footprints, that is, the angle between the positive x-axis and the edge in question going counter-clockwise around the buildings. It is clear that most edges take horizontal or vertical orientations.

The second urban area shown in Figure 40 is Dupont Circle in Washington DC. There is 3,564 buildings featuring 23,181 vertices, 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 the footprints feature between 3 and 4 edges. The areas of footprints are also quite diverse; the minimum, maximum, and average areas are 0.01, 9,867.24, and 137.73 (in m^2), respectively. The lengths of edges in footprints distribute in a wide range, with minimum, maximum, and average of 0.11, 86.65, and 5.50 (in m), respectively. Most edges have horizontal and vertical orientations as shown in Figure 43. Compared to Rosslyn, the buildings in Dupont Circle have much smaller numbers of vertices, areas, and lengths of edges, and their distributions are more irregular.

The imprint of the last urban area, Turin, is shown in Figure 41. It has 31,249 buildings and 278,329 vertices in footprints. The number of edges for each footprint varies from 3 to 194, the average is 8, and most of the footprints have less than 12 edges. The minimum, maximum, and average in areas for footprints are

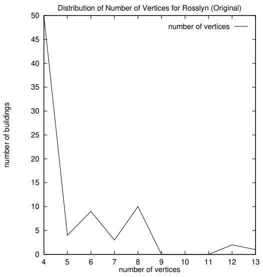

Figure 35: Size (number of vertices) distribution of footprints for Rosslyn in Fig. 39

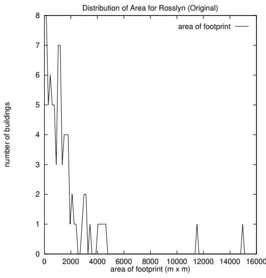

Figure 36: Area distribution of footprints (in m^2) for Rosslyn in Fig. 39

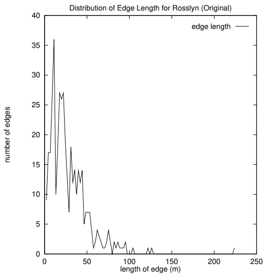

Figure 37: Edge length distribution of footprints (in m) for Rosslyn in Fig. 39

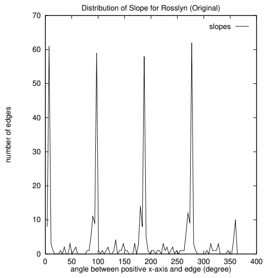

Figure 38: Slope distribution of footprints (in degree) for Rosslyn in Fig. 39

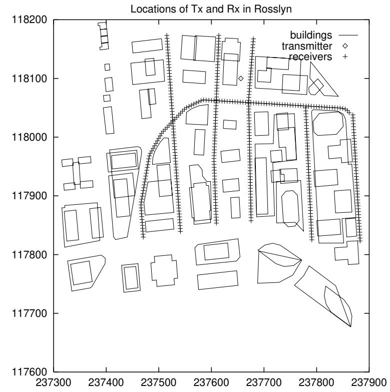

Figure 39: Locations of Tx and Rx in Rosslyn, VA

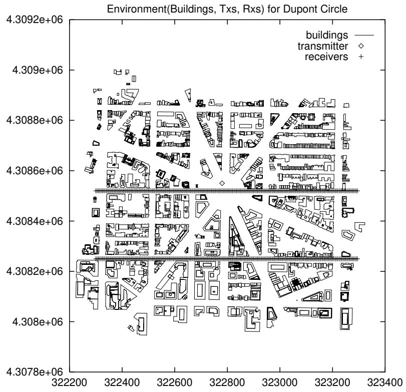

Figure 40: Locations of Tx and Rx in Dupont Circle

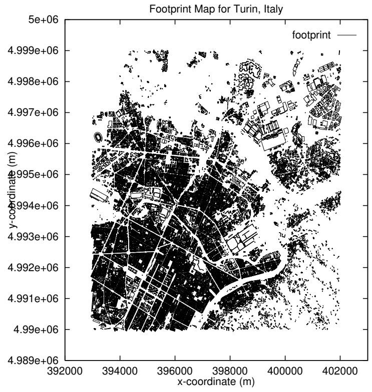

Figure 41: Building footprints in Turin, Italy

4.84, 325,699.19, and 731.69 m^2 respectively. The range for lengths of footprint edges is [0.03, 582.73] m, with average 10.934 m. Footprint edges take any orientations, but most of them lie in directions of 60, 160, 180, 240, and 340 degrees as shown in Figure 44. Because it encompasses an entire city, the Turin building database has less constraint in the number of walls, areas of footprints, orientations of buildings, and density of buildings.

The settings for experiments are as follows. No backtracking technique is used for all algorithms. The only constraint used in all single-pass methods is to keep all long edges based on local threshold with "length factor" f_l as 1.0. Constraints on edge, area, and centroid are imposed for all multi-pass and hybrid algorithms with both "area factor" f_a and "centroid factor" f_c being 0.15. The "width factor" f_w in IMD and MIMD, "distance factor" f_d in recursive subdivision methods, "strip width" α_s in CHM, and "aspect ratio" α_r in MAT and MMAT are all set to 0.15. We initially evaluate our single-pass building footprint simplification methods and derive statistics on vertex reduction rate, average length of edges and average area of footprints in the simplified maps. All relevant statistics for the Rosslyn, Dupont Circle, and Turin are provided in Table 1. For each footprint simplification method, we also calculate its running time, mean and standard deviation of simplification error in areas, and present them under columns "time", "mean err." and "dev. err." in Table 1. Figures 42 and 49 present the simplified maps generated by single-pass MAT method for Rosslyn and Dupont Circle respectively. The maps appear to be similar to the originals shown

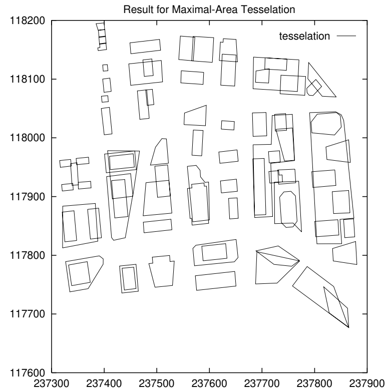

Figure 42: Simplified map by single-pass MAT method for Rosslyn, VA

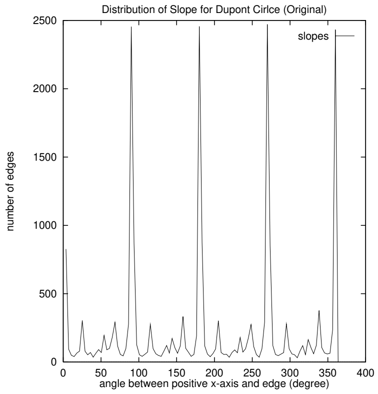

Figure 43: Slope distribution of original footprints for Dupont Circle in Fig. 40

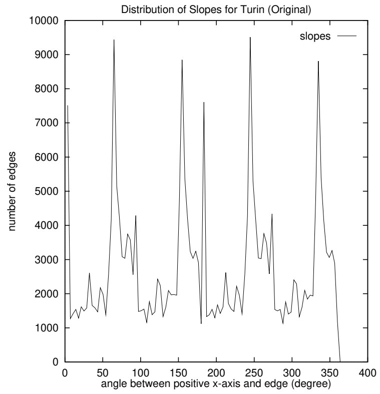

Figure 44: Slope distribution of original footprints for Turin in Fig. 41

in Figures 17 and 40. In addition to single-pass methods we have tested multi-pass and hybrid methods, the results are given in Table 2.

From Tables 1 and 2, we can observe that, among all single-pass methods, MAT method delivers the best overall performance in all cases. The MMAT produces the best vertex reduction rate among all the multiple-pass algorithms, while the hybrid MMAT-MFPS is the best performer among all methods. All footprint simplification methods increase the average lengths of edges implying that all methods indeed satisfy the dual constraints of keeping long and dropping short edges. The average length of edges for MAT is the largest among all single-pass methods, it therefore has the best vertex reduction rate. The same applies to both MMAT and hybrid MMAT-MFPS. The average areas of footprints are changed by all footprint simplification methods, but the changes can be positive (larger than original) or negative (less than original). The MAT method has the least average areas of footprints in almost all test cases which may contribute to its good performance. All methods perform better on Dupont Circle and Turin maps than Rosslyn map due to the fact that Rosslyn map was manually simplified before our algorithms were applied, while Dupont Circle and Turin maps are not preprocessed and contain many small and irregular footprints.

The simplification errors become larger if multi-pass algorithms are used instead of their single-pass counterparts, and they may be even larger when hybrid methods are employed. For instance, in Dupont Circle case, the mean and standard deviation pairs of simplification errors for MAT and MIMD are ($1.16\,\mathrm{m}^2$, $13.20\,\mathrm{m}^2$) and ($0.68\,\mathrm{m}^2$, $2.65\,\mathrm{m}^2$), while they are ($2.87\,\mathrm{m}^2$, $18.30\,\mathrm{m}^2$) for MMAT method and ($2.90\,\mathrm{m}^2$, $19.00\,\mathrm{m}^2$) for the hybrid MIMD-MMAT. The simplification error and vertex reduction rate is closely related but not linear. For instance, in Rosslyn case, the mean and standard deviation of simplification errors for IMD and MAT methods are similar, but the vertex reduction rate of MAT method is much better than that of IMD. Similar situations exist in Dupont Circle case. However, it is almost always the case that the worse the simplification error, the better the vertex reduction rate. For example, the MAT algorithm (single pass, multipass, or hybrid with other algorithm) delivers the best performance in terms of vertex reduction rate while it has worse simplification error than other methods.

The running times for all single-pass footprint simplification methods are similar, although their computational complexities are not the same (i.e., $O(n)$ for IMD, $O(n^2)$ for ReS-FPS, MAT, and CHM in worst-case). This is due to the fact that the average number of vertices for each footprint is relatively small.

method	*avg_e* (m)	*avg_a* (m^2)	*mean err.* (m^2)	*dev. err.* (m^2)	*vertices*	*red. rate%*	*time* (sec.)
			Rosslyn				
original	29.83	1683.64	0.00	0.00	412	0.00	0.00
ReS-FPS	31.78	1710.16	3.44	23.62	385	6.55	0.15
ReS-OCS	31.63	1677.49	6.97	39.29	387	6.07	0.14
IMD	32.53	1660.46	20.48	62.63	375	8.98	0.15
MAT	33.66	1618.74	23.86	67.40	362	12.14	0.17
random MAT	32.88	1608.02	23.00	66.01	371	9.95	0.16
CHM	30.39	1681.11	4.98	23.90	404	1.94	0.15
			Dupont Circle				
original	5.50	137.73	0.00	0.00	23181	0.00	0.00
ReS-FPS	6.83	143.37	0.61	9.35	18203	21.47	2.12
ReS-OCS	6.86	142.92	0.67	9.66	18398	20.63	2.12
IMD	6.65	133.72	0.60	2.57	18982	18.11	2.22
MAT	7.80	130.76	1.16	13.20	15993	31.01	2.23
random MAT	6.82	131.85	1.08	12.23	18400	20.62	2.23
CHM	6.26	136.24	0.26	3.67	20265	12.58	2.28
			Turin				
original	10.93	731.69	0.00	0.00	278329	0.00	0.00
ReS-FPS	14.57	751.96	0.31	2.33	204646	26.47	12.60
ReS-OCS	14.45	748.62	0.64	3.45	206365	25.86	12.60
IMD	13.13	718.82	1.87	5.59	228554	17.88	11.31
MAT	15.47	655.05	2.17	6.78	191472	31.21	12.43
random MAT	12.86	666.07	2.16	6.10	232996	16.29	12.41
CHM	12.61	717.98	0.44	2.22	239059	14.11	12.98

Table 1: Baseline experimental statistics(single pass algorithms) for Rosslyn, Dupont Circle, and Turin

It is true for both multi-pass and hybrid algorithms.

7.2 Impact of Footprint Simplifications on Radio Propagation Predictions

To assess the effects of footprint simplification methods on the radio propagation prediction time and accuracy, we use the city maps of Rosslyn and Dupont Circle (Figures 39 and 40). In our Rosslyn test environment, we use only one transmitter (denoted as Tx) located at the point having coordinates (237656.0, 118100.0) m. There are 350 receivers (denoted as Rx) scattered along several streets. Similarly, in our Dupont Circle test environment, one transmitter Tx located close to the epicenter of the map at (322780.0, 4308550.0) m is used. Also, there are 400 receivers scattered along two east-west (i.e., horizontal) direction streets. For the radio propagation component, we use 2D ray-tracing system developed at Polytechnic University [20] and the following key parameters:

- The carrier frequency is 900 MHz and isotropic antennas are used at both the base stations and mobile stations.
- The maximum number of reflections for each ray path is 4 for Rosslyn case, and 5 for Dupont Circle.
- The maximum order of diffractions for each ray path is 1. Diffracted rays are the main contributors to the received powers for those receivers in non-line of sight zone (NLOS).
- The antenna height of the transmitter is 10 m, while all receivers have height 2.5 m.
- All walls are assumed to be made of the same material (concrete and steel), and have dielectric constant ϵ_r = 6 for computing the reflection coefficient.

method	*avg_e* (m)	*avg_a* (m^2)	*mean err.* (m^2)	*dev. err.* (m^2)	*vertices*	*red. rate%*	*time* (sec.)
Rosslyn							
MFPS	32.18	1601.41	8.24	45.58	380	7.77	0.17
MIMD	33.17	1584.96	24.69	71.83	367	10.92	0.17
MMAT	35.01	1532.65	77.01	251.42	346	16.02	0.16
MIMD and MMAT	35.66	1530.03	79.62	246.47	339	17.72	0.17
MIMD and MFPS	33.89	1588.38	21.27	68.54	359	12.86	0.16
MMAT and MFPS	35.11	1531.70	77.96	251.34	345	16.26	0.17
Dupont Circle							
MFPS	7.21	142.73	1.43	15.39	17380	25.02	2.85
MIMD	6.76	134.27	0.68	2.65	18631	19.63	2.87
MMAT	8.05	128.67	2.87	18.30	15342	33.82	2.90
MIMD and MMAT	8.27	140.56	2.90	19.00	14925	35.62	3.62
MIMD and MFPS	7.36	135.39	1.48	15.07	17007	26.63	3.56
MMAT and MFPS	8.27	125.44	3.06	18.66	14922	35.63	3.67
Turin							
MFPS	15.33	750.69	0.80	4.60	192733	30.75	31.81
MIMD	13.34	728.30	5.99	7.70	225254	19.07	31.90
MMAT	16.92	650.50	2.25	20.01	171781	38.28	32.38
MIMD and MMAT	17.28	726.28	2.23	18.21	167958	39.65	43.54
MIMD and MFPS	15.62	740.33	1.11	7.58	188757	32.18	42.74
MMAT and MFPS	17.38	640.23	2.24	19.83	166977	40.01	43.87

Table 2: Baseline experimental statistics(multipass algorithms) for Rosslyn, Dupont Circle, and Turin

- Pincushion method is used to launch rays [20]. The angular separation (step size) is 0.5 degree in order to achieve reasonable accuracy.

The main reason that we have chosen Rosslyn as one of our test-beds is that we have real field measurements of received powers for all the receiver locations [25]. As compared to these measurements, the predictions of received powers using the original map for all mobile locations have errors whose mean and standard deviation are 1.48 dB and 10.27 dB (see Figures 45 and 46 for the comparisons). This is typical of ray tracing error where the mean error is about 1 dB and the standard deviation is 8-10 dB [26, 20, 25].

Table 3 outlines the mean and standard deviation of the difference between predicted received powers for all receivers obtained using the original and simplified maps for Rosslyn and Dupont Circle. This difference of predicted received powers is due to the simplification of building footprints. All footprint simplification methods invariably reduce the prediction time and achieve the goal of speeding up the prediction process (see column "*speedup*", which is the ratio of processing times by using the original and simplified maps). In Rosslyn, the speedup is about 1.36 on average for all simplification algorithms with the highest speedup being 1.42 achieved by MIMD-MMAT. For Dupont Circle, the average speedup for all simplification methods is 1.45 with the best speedup 1.68 delivered by the hybrid of MIMD-MMAT.

There are close relationships among the vertex reduction rate (column *"red. rate"*), number of diffraction corners (column *"df. corners"*), and speedup (column *"speedup"*). These relationships are derived in Appendix B which depicts that speedup, number of diffraction corners, and vertex reduction rate satisfy $speedup \approx CN_{d,o}/N_{d,s}$ and $speedup \approx C_1/(1 - reduction\_rate)$, where $N_{d,o}$ and $N_{d,s}$ are the number of diffraction corners generated by using the original and simplified maps respectively, C is the ratio of the average processing times per diffraction corner when the original and simplified maps are used, and C_1 is a constant multiplicity of C. When $C > 1$, the simplification method decreases the computational complexity

method	*vertices*	*red. rate%*	*time* (sec.)	*speedup*	*df. corners*	*rays*	*mean* (dB)	*dev.* (dB)
Rosslyn								
original	412	0.0	47.00	1.00	101	53366	0.00	0.00
ReS-OCS	387	6.07	36.20	1.30	89	47615	-0.06	1.80
IMD	375	8.98	35.66	1.32	87	46889	0.20	1.96
MAT	362	12.14	34.10	1.38	84	45389	0.04	1.34
random MAT	371	9.95	35.00	1.34	84	44786	-0.22	1.47
CHM	404	1.94	35.54	1.32	86	45739	-0.19	0.50
MIMD	372	9.71	34.57	1.36	84	45717	0.20	2.06
MFPS	380	7.77	33.46	1.41	81	43562	-0.54	1.37
MMAT	346	16.02	35.61	1.32	88	47493	-0.31	1.57
MIMD and MFPS	359	12.86	34.58	1.36	85	46207	-0.54	1.86
MIMD and MMAT	339	17.72	33.10	1.42	85	46158	0.59	2.32
MFPS and MMAT	345	16.26	34.67	1.36	86	46593	-0.06	2.40
Dupont Circle								
original	23181	0.0	1558.54	1.00	1569	815049	0.00	0.00
ReS-FPS	18203	21.47	1166.67	1.34	1306	709769	-0.94	2.23
ReS-OCS	18398	20.63	1145.00	1.36	1289	698866	-1.03	2.41
IMD	18982	18.11	1200.03	1.30	1343	722322	0.19	2.11
MAT	15993	31.01	966.46	1.61	1162	632835	0.24	2.13
CHM	20265	12.58	1301.70	1.20	1392	737682	0.20	1.35
MIMD	18631	19.63	1200.87	1.30	1337	718960	-0.85	2.13
MFPS	17380	25.02	1085.03	1.44	1254	686763	-0.03	2.65
MMAT	15342	33.82	969.82	1.61	1176	644172	0.30	3.47
MIMD and MFPS	17007	26.63	1075.54	1.45	1262	691220	-0.28	3.65
MIMD and MMAT	14925	35.62	926.10	1.68	1149	631606	1.45	4.01
MFPS and MMAT	14922	35.63	936.03	1.67	1138	627039	0.31	3.53

Table 3: Statistics of prediction results for Rosslyn and Dupont Circle

of each diffraction corner, thereby accelerates the prediction process. The constant C and C_1 can be obtained empirically for each method. Based on the experimental results in Table 3, C and C_1 can be calculated as 1.50 and 1.31, respectively, for Dupont Circle.

When the statistics of the prediction errors, as compared to actual measurements, is known for the original database, we can find the error statistics using the simplified database. Suppose that μ_o and σ_o are the mean and standard deviation of prediction errors compared with real measurements when the original map is used, while μ_s and σ_s are the mean and standard deviation of prediction errors compared with real measurements when the simplified map is used. Furthermore, let μ_{os} and σ_{os} be the mean and standard of prediction errors when we compare prediction results obtained using the simplified map with those generated using the original map. Then, we derive in Appendix C that $\mu_s = \mu_o + \mu_{os}$ and $\sigma_s^2 = \sigma_o^2 + \sigma_{os}^2 + \epsilon$. Here ϵ is typically very small compared to σ_o^2. When μ_{os} and σ_{os} are small, μ_s and σ_s are very close to μ_o and σ_o. For example, when the MIMD method is applied to Rosslyn, μ_o = 1.48 dB, σ_o = 10.27 dB, and μ_{os} = 0.20 dB, σ_{os} = 2.06 dB. Thus, μ_s = 1.68 dB and σ_s = 10.47 dB, which are very close to μ_o and σ_o. These results indicate that our simplification methods will not significantly affect the prediction errors if $\mu_{os} \leq 1$ dB and $\sigma_{os} \leq 4$ dB.

In Table 3, the prediction error statistics μ_{os} and σ_{os} are listed in the columns *"mean"* and *"dev."*. All footprint simplification methods have a very small mean prediction error, but their standard deviations vary dramatically. For example, in the Rosslyn case, the difference between the predictions made using the original and the simplified databases has a mean less than or equal to 0.59 dB, but the standard deviation ranges from 0.70 to 2.40 dB. For Dupont Circle, the mean of prediction error is typically less than 1 dB and the standard deviation is less than 4 dB. It can be noticed that σ_{os} is roughly proportional to the speedup, as well as to the vertex reduction rate. In Dupont Circle, the speedup for ReS-FPS and hybrid MIMD-MMAT are 1.34 and 1.68, their vertex reduction rates are 21.47% and 35.63%, while their standard deviations of prediction errors are 2.23 dB and 4.01 dB respectively. Therefore, there is a trade-off between speedup and vertex reduction rate, and between speedup and prediction accuracy. Among all single-pass algorithms, MAT delivers the best speedup in both test cases. The MFPS and MMAT have the best performance in Rosslyn and Dupont Circle respectively among all multi-pass methods. The hybrid MIMD-MMAT method presents the best overall speedup rates in both Rosslyn and Dupont Circle.

The simplification methods also affect the number of raypaths (column *"rays"*). This essentially determines the prediction time and accuracy. As we trace more raypaths we not only improve the prediction accuracy but also we need longer time for the prediction result. For instance, in Dupont Circle case, the number of traced raypaths is 722,322 for the IMD method, with corresponding speedup 1.30, while for the MAT algorithm the number of processed raypaths is 632,835 and speedup at 1.61.

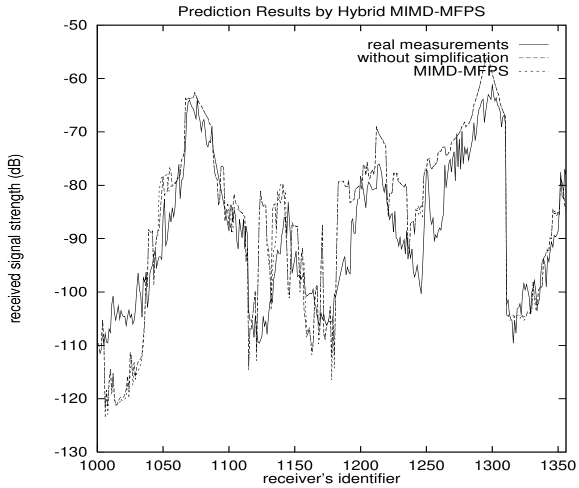

Figure 45: Comparison of predictions by MIMD-MFPS with real measurements for Rosslyn

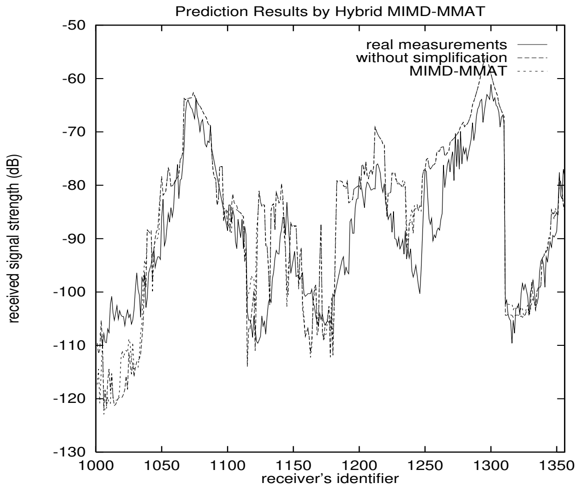

Figure 46: Comparison of predictions by MIMD-MMAT with real measurements for Rosslyn

Figures 45 and 46 show our prediction results generated by both hybrid MIMD-MFPS and MIMD-MMAT for Rosslyn. The x-axis shows all 350 receivers (with identifiers from 1000 to 1350) used in the Rosslyn area and the y-axis depicts the predicted/measured powers received. The graph features three curves: prediction results obtained using the original map (with no simplifications), prediction results by using the simplified maps generated by either MIMD-MFPS or MIMD-MMAT simplification algorithm, and finally, the real field measurements. Figures 45 and 46 show that the prediction error generated by the hybrid of MIMD-MFPS is evenly distributed among all receivers, while the hybrid MIMD-MMAT yields a distribution mostly located in the receivers' range of [1000, 1050], [1120, 1125], and [1310, 1320]. Thus, the standard deviation of the prediction errors for hybrid MIMD-MMAT method is worse than hybrid MIMD-MFPS method.

Figures 47 and 48 show the prediction results generated by the MIMD and hybrid MFPS-MMAT, respectively, for Dupont Circle. The x-axes depict the identifiers of the receivers and the y-axes the corresponding predicted power received by each receiver. These figures clearly point out that the prediction error generated by MIMD is scattered more uniformly among all receivers than that of MFPS-MMAT. If the latter is used, the prediction error generated mostly concentrates on receivers in the range of [0, 25], [140, 155], and [160, 170]. Therefore, the standard deviation of prediction error for hybrid MFPS-MMAT method is larger than MIMD method (3.53 dB vs. 2.13 dB). However, the speedup for hybrid MFPS-MMAT method is better (1.67 vs. 1.30). Evidently, there is a trade-off between prediction accuracy and prediction time.

Prediction error has close relationship with the complexity of building databases. The Dupont Circle map contains many more buildings (footprints and vertices) than the Rosslyn map. Prediction errors generated by our simplification methods for Dupont Circle are also worse than those for Rosslyn. To avoid deterioration of prediction accuracy, single-pass or multi-pass algorithms (e.g., MAT or MFPS) can be chosen for large coverage areas, while hybrid algorithms (e.g., MIMD-MMAT) can be used in sites with small or medium areas.

Due to the fact that we used 2-D ray-tracing method in the foregoing studies, only building corners near the transmitters are illuminated. Should we employ 3-D ray-tracing method, we anticipate that the resulting speedup rates will be greatly improved since rays from the transmitter can pass over the nearby buildings to directly illuminate the corners of more distant buildings. In order to support the validity of the above

method	*red. rate%*	*time* (sec.)	*speedup*	*df. corners*	*rays*	*pass-rays*	*total rays*
original	0.0	12709.18	1.00	2071	1077055	793855	1870910
ReS-FPS	21.47	6597.05	1.93	1739	943289	704796	1648085
ReS-OCS	20.63	6721.12	1.89	1752	945480	709788	1655268
IMD	18.11	7455.51	1.70	1765	948317	703039	1651356
MAT	31.01	4367.46	2.91	1525	830176	627545	1457721
CHM	12.58	7987.07	1.59	1811	960270	719737	1680007
MIMD	19.63	7321.60	1.74	1765	948132	706428	1654560
MFPS	25.02	5730.91	2.22	1690	922583	687272	1609855
MMAT	33.82	4136.94	3.07	1539	845049	649263	1494312
MIMD and MFPS	26.63	5600.30	2.67	1708	934279	693855	1628134
MIMD and MMAT	35.62	3913.03	3.25	1517	833874	639635	1473509
MFPS and MMAT	35.63	3901.43	3.26	1492	823276	634090	1457366

Table 4: Prediction results for Dupont Circle while allowing "pass-rays" over buildings

conjecture, we have run experiments in which rays are allowed to pass over buildings. In this way, we force more illuminated corners (i.e., diffraction corners) to be generated, which act as secondary sources of rays that must be traced. Table 4 depicts the results for Dupont Circle when rays are allowed to "pass" over at most one building. The same constraints and parameters as those discussed in this section are used for all footprint simplification algorithms (see column "*red. rate%*" in Table 4). It is evident that the speedup rates improve considerably. For instance, the MFPS-MMAT method achieves the best overall speedup of 3.26 while all methods achieve rates higher than 1.59. The column entitled "*df. corners*" shows the number of diffraction corners that are illuminated. Clearly, these numbers are significantly larger than their counterparts in Table 3. In this experiment, the running time has increased nearly ten-fold (column "*time*") due to the larger number of the traced raypaths (columns "*rays*", "*pass-rays*", and "*total rays*").

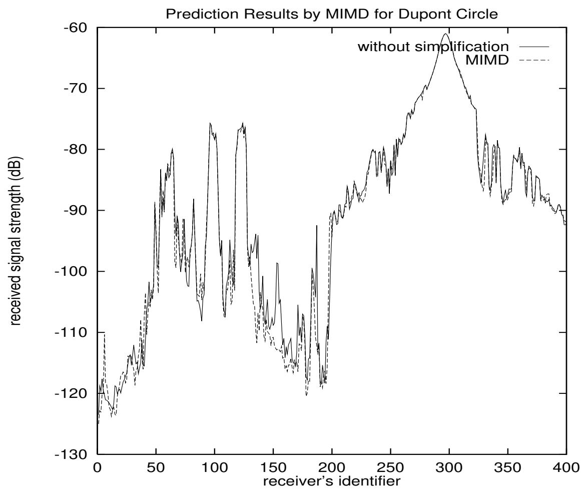

Figure 47: Predictions with original map and simplified map generated by MIMD method for Dupont Circle

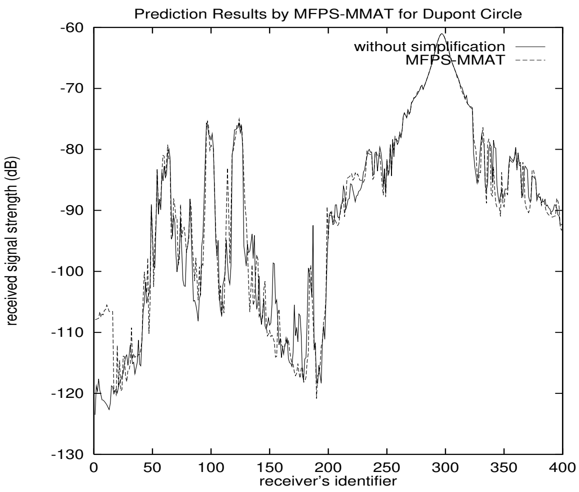

Figure 48: Predictions with original map and simplified map generated by hybrid MFPS-MMAT for Dupont Circle

7.3 Sensitivity to Simplification Error

In this part, we further investigate the trade-off between prediction error and vertex reduction rate of the simplification methods. As it is impossible to express the relationship between prediction error and precision of building databases in a closed-form, we carry out a sensitivity analysis and in this manner we attempt to obtain insights into this relationship. The footprints of Rosslyn are again used as our test setting. The hybrid method of MIMD-MMAT is chosen as our show-case because of its excellent performance in terms of vertex reduction rate and speed-up. The maximum numbers of reflections and diffractions are 10 and 2, respectively. The length factor f_l is 1.0 and local threshold (i.e., the average length of all edges in the footprint under processing) is used. No constraints are imposed on area and centroid to achieve as high vertex reduction rate as possible. The width factor f_w used in MIMD is fixed to be 0.15, while the aspect ratio α_r in MMAT method varies in [0.10, 0.90]. It is intuitive that the larger α_r is, the higher the vertex reduction rate. For each α_r, the footprint simplification algorithm is applied to the original map, the generated simplified map is used as the input of the ray-tracing radio propagation prediction system to generate prediction results. The prediction errors are computed for the prediction results based on the prediction results generated by using the original map. To evaluate the correlation between α_r and the constraints on area and centroid, we calculate the percentages of footprints which satisfy constraints on area and centroid by setting both the area factor f_a and centroid factor f_c to 0.15.

Table 5 presents the results derived with the hybrid MIMD-MMAT process for the various values of the α_r parameter. Simplified Rosslyn city map and its area difference map are shown in Figures 50 and 51, respectively, for α_r = 0.20. Some key observations can be drawn from the above Table and Figures. First, the speedup for the prediction procedure (column "speed") is not a monotonic function of the vertex reduction rate (column "red.rate%"). For instance, the vertex reduction rate is 33.98% when the aspect ratio is α_r=0.50, its speedup is 1.46; while the vertex reduction rate is 28.64% when α_r= 0.40, but its speedup is 1.47. Next, the larger the α_r, the higher the vertex reduction rate, however, the lower the percentages of footprints which satisfy the pre-specified constraints on areas and centroids (columns "cnst. area" and "cnst.

α_r	*vertices*	*vert. diff.*	*red. rate%*	*time* (sec.)	*speed*	*mean* (dB)	*deviation* (dB)	*cnst. area* (%)	*cnst. centroid* (%)
0.00	412	0	0.0	279.00	1.00	0.00	0.00	100	100
0.10	363	50	12.14	200.72	1.39	0.22	2.48	100	100
0.20	330	82	19.90	193.75	1.44	0.66	3.57	96	99
0.30	316	96	23.30	192.42	1.45	0.95	4.01	84	86
0.40	294	118	28.64	189.80	1.47	1.44	5.23	58	66
0.50	272	140	33.98	191.12	1.46	2.54	6.35	29	37
0.90	265	147	35.68	180.01	1.55	4.57	8.79	24	28

Table 5: Mean and standard deviation of prediction errors for Rosslyn, VA (Hybrid MIMD-MMAT)

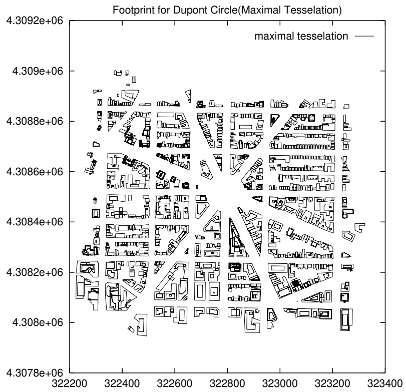

Figure 49: Simplified map by single-pass MAT method for Dupont Circle, DC

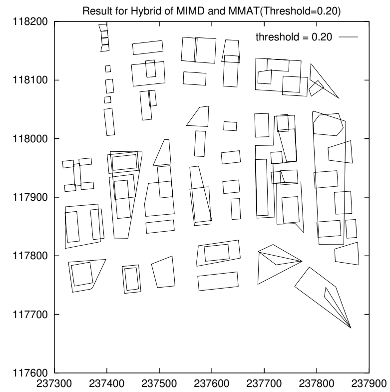

Figure 50: Simplified map for Rosslyn, VA – Hybrid MIMD-MMAT with α_r = 0.20

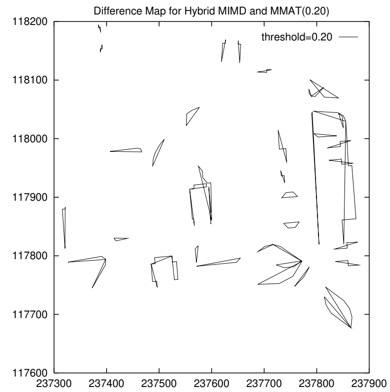

Figure 51: Difference map for Rosslyn, VA – Hybrid MIMD–MMAT with α_r = 0.20

centroid"). Finally, the prediction accuracy (mean and standard deviation in columns "mean" and "deviation") decreases as α_r increases. That is, the accuracy of the predictions is sensitive to the simplification error generated by the footprint simplification algorithm. For instance, if the number of vertices removed is about 20% (using α_r = 0.20), the mean and standard deviation of the prediction errors are about 0.66 dB and 3.57 dB, respectively. However, if the vertex reduction rate is about 30% (using α_r = 0.40), the mean and standard deviation are 1.44 dB and 5.23 dB, which are larger than the acceptable thresholds (i.e., 1 dB and 4 dB). Therefore, the feasible value for parameter α_r is [0.00, 0.30].

One common observation from all these experiments is that the speedup for the prediction process is closely related to the vertex reduction rate. This is because simplification methods reduce the number of corners in footprints, thereby speed up operations on corner level. Reduction of corners also means reduction in the number of diffraction corners as well as processing time for diffraction corners. However, the relationship between the speedup and the vertex reduction rate is non monotonic. One reason is that the number of buildings does not change as a result of simplification, and operations on building level (retrieving a building, traversing all buildings within a grid) require similar processing time for both the original map and simplified map. Another reason is that simplification methods also change the geometry of the test area, so that the number of segments per ray may change due to the distortion of building shapes. Therefore, the processing time for a raypath may be different when using different maps.

8 Conclusions and Future Work

Today's wide-spread mobile communications call for effective and accurate radio wave propagation prediction systems. Among the parameters that greatly affect the function of such systems is the complexity of the building database. In this paper, we deal with the problem of simplifying the building footprint layout in urban areas so that the prediction process becomes faster without undue loss of accuracy. We propose four families of single-pass footprint simplification algorithms, namely: the inverse midpoint displacement (IMD), recursive subdivisions (ReS-FPS/ReS-OCS), maximum-area triangulation (MAT), and convex-hull-based (CHM) methods. We also suggest multi-pass methods that take advantage of the asymmetric property that single-pass algorithms demonstrate. The idea behind multi-pass methods is to run single-pass algorithms for all possible starting points in the footprint and then select the best result. Hybrid methods exploit the complementary property exhibited by various simplification algorithms. We propose the notions of area difference map and centroid translation map as the core mechanisms for evaluating simplification error that may be introduced. Statistics based on area difference and centroid translation maps are also used as metrics to assess the performance of our methods.

Footprint simplification methods are critical to the reduction of the prediction time and the maintenance of acceptable prediction accuracy in radio wave propagation systems based on ray-tracing. However, the relationships among the errors generated by the utilized simplification algorithms, the achieved prediction times, and the attained accuracy levels are not straightforward. In general, the preservation of footprint shape is closely related to the prediction accuracy of the radio wave propagation model. In order to maintain the shapes of footprints, the characteristics of the shape of a footprint, such as its vertices, long edges, area, centroid, should be maintained within an acceptable level. That means multiple constraints on vertices, edges, area, and centroid should be used during the footprint simplification.

We have experimented with our proposed algorithms in the context of three representative urban areas and have assessed the impact that our techniques have on both speed and accuracy of radio propagation. Our proposed simplification methods provide a flexible way to reduce the complexity of the underlying building footprint (GIS) databases with algorithmic parameters that can be used to tune the simplification error. Multi-pass methods overall outperform their single-pass counterparts, while hybrid methods can achieve higher vertex reduction rate than their constituent methods. On large databases, our methods decrease the computation time for the radio propagation prediction process by up to a factor of three without introducing undue prediction error. The radio propagation prediction accuracy is very sensitive to the distortion of the GIS databases. Our experimental results finally indicate that sizable perturbation of the simplification algorithm tunable parameters leads to unacceptable radio propagation prediction results.

We plan to extend our work in a number of promising directions: a) establish analytical model to investigate the correlations among different parameters used in footprint simplification methods; b) develop empirical rules to guide the selections of suitable footprint simplification methods for different types of footprints; c) simplify the buildings database further by merging adjacent or combining nearby buildings to form "larger" GIS entities; d) use progressive and approximate techniques in the propagation prediction models as it is likely that different ray paths may contribute differently to the ultimate outcome; e) employ parallelization techniques on a network of workstations to improve the accuracy of radio propagation in light of minimally modified building footprints.

Acknowledgments
The authors are grateful to the reviewers for their detailed comments that helped us significantly improve

the presentation of our work. We are also very much indebted to Boris Aronov for providing comments on earlier versions of the paper.

References

[1] Kim, S. C. et al. (1999) Radio Propagation Measurements and Prediction Using Three-Dimensional Ray Tracing in Urban Environments at 908MHz and 1.9GHz, *IEEE Transactions on Vehicular Technology*, May, **48.3**, 931–946.

[2] Castleman, K. R. (1996) Digital Image Processing, Prentice Hall, Englewood Cliffs, New Jersey 07632.

[3] Foley, J. D., van Dam, A., Feiner, S. K., and Hughes, J. F. (1990) Computer Graphics – Principles and Practice, Addison-Wesley & Benjamin Cummings, Reading, MA.

[4] Hearn, D. and Baker, M. P. (1996) Computer Graphics, C Version, Prentice Hall PTR, Upper Saddle River, NJ.

[5] Jain, R., Kasturi, R., and Schunck, B. G. (1995) Machine Vision, McGraw-Hill, New York, NY.

[6] Cormen, T. H., Leiserson, C. E., and Rivest, R. L. (1997) Introduction to Algorithms, MIT Press, Cambridge, MA.

[7] Wall, K. and Danielsson, P. E. (1984) A Fast Sequential Method for Polygonal Approximation of Digitized Curves, *Computer Vision, Graphics, and Image processing*, **28**, 220–227.

[8] Leu, J. G. and Chen, L. (1988) Polygonal Approximation of 2D Shapes Through Boundary Merging, *Pattern Recognition Letters*, **7**, 231–238.

[9] Wall, K. (1986) Curve Fitting Based on Polygonal Approximation, *Proceedings of the 8th IEEE International Conference on Pattern Recognition*, Los Alamitos, CA.

[10] Sklansky, J. and Gonzalez, V. (1980) Fast Polygonal Approximation of Digitized Curves, *Pattern Recognition*, **12.5**, 327–331.

[11] Williams, C. M. (1978) An Efficient Algorithm for the Piece-wise Linear Approximation of Planar Curves, *Computer Graphics and Image Processing*, **8.2**, 286–293.

[12] Williams, C. M. (1981) Bounded Straight Line Approximation of Digitized Planar Curves and Lines, *Computer Graphics and Image Processing*, **16**, 370–381.

[13] Kurozumi, Y. and Davis, W. A. (1982) Polygonal Approximation by the Minimax Method, *Computer Graphics and Image Processing*, **19**, 248–264.

[14] Imai, H. and Iri, M. (1986) Computational-Geometric Methods for Polygonal Approximations of a Curve, *Computer Vision, Graphics, and Image Processing*, **36.1**, 31–41.

[15] Dettori, G. (1982) An On-line Algorithm for Polygonal Approximation of Digitized Plane Curves, *Proceedings of the 6th International Conference in Pattern Recognition*, **2**, 840–842.

[16] Pavlidis, T. and Horowitz, S. L. (1974) Segmentation of Planar Curves, *IEEE Transactions on Computers*, August, **C-23.8**, 860–870.

[17] Ramer, U. E. (1972) An Iterative Procedure for the Polygonal Approximation of Plane Curves, *Computer Graphics and Image Processing*, **1**, 244–256.

[18] Preparata, F. and Shamos, M. (1985) Computational Geometry: An Introduction, Springer Verlag, New York, NY.

[19] Loew, K. (1992) Comparison of Urban Propagation Models with CW-measurements, *Proceedings of Vehicular Technology Conference*, May, **1**, 936–942.

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

[21] Lee, W. C. Y. (1997) Mobile Communications Engineering, McGraw-Hill, Singapore.

[22] Garg, V. K. (2000) IS-95 CDMA and cdma2000, Prentice-Hall PTR, Upper Saddle River, NJ.

[23] Garg, V. K. and Wilkes, J. E. (1996) Wireless and Personal Communications Systems, Prentice-Hall PTR, Upper Saddle River, NJ.

[24] Pahlavan, K. and Levesque, A. H. (1995) Wireless Information Networks, John Wiley & Sons, Inc., New York, NY.

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

[26] Piazzi, L. and Bertoni, H. L. (1999) Achievable Accuracy of Site-Specific Path-Loss Predictions in Residential Environments, *IEEE Transactions on Vehicular Technology*, May, **48.3**, 922–930.

[27] O'brien, W. M., Kenny, E. M., and Cullen, P. J. (2000) An Efficient Implementation of a Three-Dimensional Microcell Propagation Tool for Indoor and Outdoor Urban Environments, *IEEE Transactions on Vehicular Technology*, March, **49.2**, 622–630.

[28] Mori, S., Suen, C. Y., and Yamamoto, K. (1992) Historical Review of OCR Research and Development, *Proceedings of the IEEE*, July, **80.7**, 1029–1058.

[29] Puppo, E. and Scopigno, R. (1997) Simplification, LOD and Multiresolution – Principles and Applications, *Eurographics*, **16.3**,

[30] Dettori, G. and Puppo, E. (1996) How Generalization Interacts with the Topological and Metric Structure of Maps, *Proceedings 7th International Symposium on Spatial Data Handling*, August, Delft, The Netherlands.

[31] Hoppe, H. (1996) Progressive Meshes, *Proceedings of the ACM SIGGRAPH Conference*, August, pp 99–108, New Orleans, LA.

[32] Klein, R. and Gumhold, S. (1998) Data Compression of Multiresolution Surfaces, *Visualization in Scientific Computing*, pp 13–24, Springer.

[33] Erikson, C., TR96-016 (1996) Polygonal Simplification: An Overview, Department of Computer Science, Univ. of North Carolina, Chapel Hill, NC.

[34] Dehaemer, M. J. and Zyda, M. J. (1991) Simplification of Objects Rendered by Polygonal Approximations, *Computer and Graphics*, **15.2**, 175–184.

[35] Kalvin, A. D. and Taylor, R. H. (1996) Superfaces: Polygonal Mesh Simplification with Bounded Error, *IEEE Computer Graphics*, **16.3**, 64–77.

[36] Rossignac, J. R. and Borrel, P., Technical Report RC 17697 (no.77951) (1992) Multi-resolution 3D Approximations for Rendering Complex Scenes, IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY10958.

[37] Paul, H., and Hansen, C. (1993) Geometric Optimization, *Proceedings of Visualization*, October, pp. 189–195.

[38] Hamann, B. (1994) A Data Reduction Scheme for Triangulated Surfaces, *Computer Aided Geometric Design*, **11.2**, 197–214.

[39] Schroeder, W. J., Zarge, J. A., and Lorensen, W. E. (1992) Decimation of Triangle Meshes, *Computer Graphics (SIGGRAPH 92 Proceedings)*, February, **26**, 65–70.

[40] Lee, D. (1999) New Cartographic Generalization Tools, *Proceedings of the 19th ICA International Cartographic Conference*, Ottawa, Canada.

[41] Lamy, S., Ruas, A., Demazeau, Y., Jackson, M., Mackaness, W., and Weibel, R. (1999) The Application of Agents in Automated Map Generalization, *Proceedings of the 19th ICA International Cartographic Conference*, Ottawa, Canada.

[42] Turk, G. (1992) Re-Tiling Polygonal Surfaces, *Computer Graphics (SIGGRAPH 92 Proceedings)*, July, **26.2**, 55–64.

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

[44] Sester, M. (2000) Generalization Based on Least Squares Adjustment, *International Archives of Photogrammetry and Remote Sensing*, **XXXIII.B4**, 931–938.

[45] Douglas, D., and Peucker, T. (1973) Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature, *The Canadian Cartographer*, **10.2**, 112–122.

[46] Powitz, B. (1992) Computer-assisted Generalization - an Important Software Tool for GIS, *International Archives of Photogrammetry and Remote Sensing, ISPRS Commission IV Symposium, Washington, USA*, **30.4**, 664–673.

[47] Rizk, K., Wagen, J. F., and Gardiol, F. (2000) Influence of Database Accuracy on Two-Dimensional Ray-Tracing-Based Predictions in Urban Microcells, *IEEE Transactions on Vehicular Technology*, **49.2**, 631-642.

[48] Lawton, M. C., and McGeehan, J. P. (1994) The Application of a Deterministic Ray Launching Algorithm for the Prediction of Radio Channel Characteristics in Small-Cell Environments, *IEEE Transactions on Vehicular Technology*, **43**, 955-969.

[49] Tan, S. Y., and Tan, H. S. (1995) Propagation Model for Microcellular Communications Applied to Path Loss Measurements in Ottawa City Streets, *IEEE Transactions on Vehicular Technology*, **44.2**, 313–317.

[50] Tan, S. Y., and Tan, H. S. (1996) 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.

[51] Erceg, V., Fortune, S. J., Ling, J., Rustako, A. J., and Valenzuela, R. A. (1997) Comparisons of a Computer-Based Propagation Prediction Tool with Experimental Data Collected in Urban Microcellular Environments, *IEEE Journal on Selected Areas in Communications*, **15.4**, 677-684.

[52] T. Kurner, T., Cichon, D. J., and Wiesbeck, W. (1993) Concepts and Results for 3D Digital Terrain-Based Wave Propagation Models: An Overview, *IEEE Journal of Selected Areas in Communications*, **11.7**, 1002–1012.

[53] Piazzi, L. and Bertoni, H. L. (1998) Effect of Terrain on Path Loss in Urban Environments for Wireless Applications, *IEEE Transactions on Antennas and Propagation*, **46.8**, 1138–1147.

[54] Torrico, S. A., H.L. Bertoni, H. L., and Lang, R. H. (1998) Modeling Tree Effects on Path Loss in a Residential Environment, *IEEE Transactions on Antennas and Propagation*, **46.6**, 872–880.

[55] Mandelbrot, B. (1982) The Fractal Geometry of Nature, W. H. Freeman & Co., San Francisco.

[56] Barnsley, M. (1988) Fractals Everywhere, Academic Press, San Diego.

[57] Hershberger, J. and Snoeyink, J. (1992) Speeding up the Douglas–Peucker Line Simplification Algorithm, *Proceedings of 5th International Symposium on Spatial Data Handling*, Charleston, South Carolina, August, pp 134–143.

A Running Time for Maximum-Area Triangulation Method

First, we analyze the worst-case running time for the maximum-area triangulation algorithm. Let $T(n)$ be the worst-case running time for this method on an n-vertex polygon. Obviously, $T(n) = \Theta(1)$ for $n \leq 3$. Without lost of generality, assume that edge (1,2) is the longest one among all the edges of the polygon, and triangle (1, 2, p) has the maximum area among all triangles (1, 2, i), where $i = 3, 4, \ldots, n$ and $3 \leq p \leq n$. The vertices are separated by p into two sets, $R_1 = \{2, 3, \ldots, p\}$ and $R_2 = \{1, p, p+1, \ldots, n\}$. The size q of set R_1 is $q = p - 1$, and the size of set R_2 is $n + 1 - q$. It is clear that $2 \leq q \leq (n-1)$. The worst-case running time occurs when the vertices are always partitioned into two parts and one of them has a constant number q of vertices at each step. We have the recurrence:

$$T(n) = \max_{2 \leq q \leq n-1} \{T(q) + T(n+1-q)\} + \Theta(n) \tag{1}$$

By guessing that $T(n) \leq cn^2$ for some constant c and substituting it into (1), we obtain,

$$T(n) \leq \max_{2 \leq q \leq n-1} \left\{cq^2 + c(n+1-q)^2\right\} + \Theta(n) = c \max_{2 \leq q \leq n-1} \left\{q^2 + (n+1-q)^2\right\} + \Theta(n) \tag{2}$$

Let $f(q) = q^2 + (n+1-q)^2$, we differentiate $f(q)$ with respect to q twice, and get $f^{(1)}(q) = 4q - 2(n+1)$ and $f^{(2)}(q) = 4 > 0$. Therefore, $f(q)$ achieves its maximum over the range of $2 \leq q \leq n-1$ at one of its endpoints. Since $f(2) = f(n-1) = n^2 - 2n + 5$, we obtain,

$$\begin{aligned} T(n) &\leq c \max_{2 \leq q \leq n-1} \left\{cq^2 + c(n+1-q)^2\right\} + \Theta(n) \leq c\left[n^2 - (2n-5)\right] + \Theta(n) &(3)\\ &= cn^2 - [c(2n-5) - \Theta(n)] \leq cn^2 &(4) \end{aligned}$$

Since we can pick the constant c large enough so that the term $c(2n-5)$ dominates the $\Theta(n)$ term. Thus, the worst-case running time of the maximum-area triangulation algorithm is $O(n^2)$.

To analyze the average-case running time for MAT algorithm, let $T(n)$ be the average time to process an n-vertex polygon. At each recursive step, MAT partitions the vertices into two parts with sizes of q and $n+1-q$. By assuming that each partition occurs with equal probability, $T(n)$ can be expressed as:

$$T(n) = \frac{1}{n-2} \sum_{q=2}^{n-1} [T(q) + T(n+1-q)] + \Theta(n) = \frac{2}{n-2} \sum_{q=2}^{n-1} T(q) + \Theta(n) \tag{5}$$

We guess that $T(n) \leq an\log(n) + b$ for some constants $a > 0$ and $b > 0$ to be determined (we use base 2 for log function here). We can pick a and b sufficiently large so that $an\log(n) + b > T(3)$. Then for $n > 3$, by substituting $T(n) \leq an\log(n) + b$ into (5), we have

$$T(n) \leq \frac{2}{n-2} \sum_{q=2}^{n-1} [aq\log(q) + b] + \Theta(n) = \frac{2a}{n-2} \sum_{q=2}^{n-1} q\log(q) + 2b + \Theta(n) \tag{6}$$

Since

$$
\begin{aligned}
\sum_{q=2}^{n-1} q\log(q) &= \sum_{q=2}^{\lceil n/2\rceil-1} q\log(q) + \sum_{q=\lceil n/2\rceil}^{n-1} q\log(q) \le \log(n/2)\sum_{q=2}^{\lceil n/2\rceil-1} q + \log(n)\sum_{q=\lceil n/2\rceil}^{n-1} q && (7)\\
&= \log(n)\sum_{q=2}^{n-1} q - \sum_{q=2}^{\lceil n/2\rceil-1} q = \left[\frac{n(n-1)}{2} - 1\right]\log(n) - \left[\frac{1}{2}(n/2-1)(n/2)-1\right] && (8)\\
&= \frac{1}{2}(n+1)(n-2)\log(n) - \frac{1}{8}\left[n(n-2)-1\right] && (9)
\end{aligned}
$$

By plugging (9) into (6), we obtain,

$$
\begin{aligned}
T(n) &\le \frac{2a}{n-2}\left\{\frac{1}{2}(n+1)(n-2)\log(n) - \frac{1}{8}\left[n(n-2)-1\right]\right\} + 2b + \Theta(n) && (10)\\
&\le an\log(n) + b - \left[an/4 - (\Theta(n)+b)\right] && (11)\\
&\le an\log(n) + b && (12)
\end{aligned}
$$

As long as $n \ge 2$, inequality in (11) holds. We can choose a large enough so that $an/4$ term dominates $\Theta(n) + b$ in (12). We conclude that MAT algorithm has average running time of $O(n\log(n))$.

B Relationship between Speedup and Vertex Reduction Rate

Let N_t be the number of transmitters in the area under study, $N_{d,o}$ and $N_{d,s}$ be the number of diffraction corners generated in the original and the simplified footprint respectively, $T_{t,o}$ and $T_{t,s}$ be the average processing times for a transmitter in the original and simplified footprint respectively, $T_{d,o}$ and $T_{d,s}$ be the average processing times for a diffraction corner in the original and the simplified footprint, respectively.

First, we derive the relationship between speedup and number of diffraction corners. The total processing times T_o and T_s by using the original and simplified footprints are

$$T_o = N_t T_{t,o} + N_{d,o} T_{d,o}; \qquad T_s = N_t T_{t,s} + N_{d,s} T_{d,s}$$

The speedup achieved by the simplification method is

$$speedup = \frac{T_o}{T_s} = \frac{N_t T_{t,o} + N_{d,o} T_{d,o}}{N_t T_{t,s} + N_{d,s} T_{d,s}}$$

When $N_{d,o} >> N_t$, $N_{d,s} >> N_t$, $T_{t,o} \approx T_{d,o}$, and $T_{t,s} \approx T_{d,s}$, we have

$$speedup \approx \frac{N_{d,o} T_{d,o}}{N_{d,s} T_{d,s}} \approx C\frac{N_{d,o}}{N_{d,s}}$$

where $C = T_{d,o}/T_{d,s}$ is the ratio of the average processing time per diffraction corner for original map to the average processing time per diffraction corner for simplified map. When C is larger than 1, the simplification method decreases the computational complexity of each diffraction corner.

Next, we derive the relationship between speedup and the vertex reduction rate. If we assume that the number of diffraction corners is proportional to the total number of corners in the footprints, we get

$$N_{d,o} = p_o N_{c,o}; \qquad N_{d,s} = p_s N_{c,s}$$

where p_o and p_s are constants with values in $[0, 1]$, $N_{c,o}$ and $N_{c,s}$ are the total numbers of corners in the original footprints and simplified footprints, respectively. Then

$$reduction\_rate = \frac{N_{c,o} - N_{c,s}}{N_{c,o}} = 1 - \frac{N_{c,s}}{N_{c,o}}$$

$$speedup = C\frac{p_o N_{c,o}}{p_s N_{c,s}} = \frac{C_1 N_{c,o}}{N_{c,s}} = \frac{C_1}{1 - reduction\_rate}$$

where $C_1 = Cp_o/p_s$.

C Derivation of Prediction Error

The estimation of the mean and standard deviation of the prediction error when the simplified map is used is based on the following parameters:

- μ_o and σ_o are the mean and standard deviation when the original map is used and prediction results are compared with real measurements,
- μ_s and σ_s are the mean and standard deviation when the simplified map is used and prediction results are compared with real measurements,
- μ_{os} and σ_{os} are the mean and standard deviation of prediction errors when we compare prediction results by using the simplified map with those by using the original map,
- M_i is the measurement of received power for receiver i, $P_i(o)$ and $P_i(s)$ are the predictions of received power for receiver i by using the original and simplified maps, respectively,
- N is the number of receivers.

More specifically we have the following formulas:

$$\begin{aligned}
\mu_o &= \frac{1}{N}\sum_{i=1}^{N}[M_i - P_i(o)]; \qquad \mu_s = \frac{1}{N}\sum_{i=1}^{N}[M_i - P_i(s)]; \qquad \mu_{os} = \frac{1}{N}\sum_{i=1}^{N}[P_i(o) - P_i(s)] \\
\sigma_o^2 &= \frac{1}{N}\sum_{i=1}^{N}[M_i - P_i(o)]^2 - \mu_o^2; \qquad \sigma_s^2 = \frac{1}{N}\sum_{i=1}^{N}[M_i - P_i(s)]^2 - \mu_s^2; \qquad \sigma_{os}^2 = \frac{1}{N}\sum_{i=1}^{N}[P_i(o) - P_i(s)]^2 - \mu_{os}^2
\end{aligned}$$

Using the above equations, we can have the following derivations:

$$\begin{aligned}
\mu_s &= \frac{1}{N}\sum_{i=1}^{N}[M_i - P_i(s)] = \frac{1}{N}\sum_{i=1}^{N}[M_i - P_i(o)] + \frac{1}{N}\sum_{i=1}^{N}[P_i(o) - P_i(s)] = \mu_o + \mu_{os} \\
\sigma_s^2 &= \frac{1}{N}\sum_{i=1}^{N}[M_i - P_i(s)]^2 - \mu_s^2 \\
&= \frac{1}{N}\sum_{i=1}^{N}([M_i - P_i(o)]^2 + [P_i(o) - P_i(s)]^2 + 2[M_i - P_i(o)][P_i(o) - P_i(s)]) - (\mu_o^2 + \mu_{os}^2 + 2\mu_o\mu_{os}) \\
&= \sigma_o^2 + \mu_o^2 + \sigma_{os}^2 + \mu_{os}^2 + \frac{2}{N}\sum_{i=1}^{N}[M_i - P_i(o)][P_i(o) - P_i(s)] - (\mu_o^2 + \mu_{os}^2 + 2\mu_o\mu_{os}) \\
&= \sigma_o^2 + \sigma_{os}^2 + \epsilon
\end{aligned}$$

where $\epsilon = \frac{2}{N}\sum_{i=1}^{N}[M_i - P_i(o)][P_i(o) - P_i(s)] - 2\mu_o\mu_{os}$.