

Convex Space Building Discretisation for Ray-tracing

Eamonn Kenny¹, Eamonn O Nuallain²

¹ADAPT Centre, School of Computer Science and Statistics, Trinity College Dublin, Dublin, Ireland

²School of Computer Science and Statistics, Trinity College Dublin, Dublin, Ireland

This paper describes a point to multipoint three-dimensional convex space-based ray-tracing technique. This visibility list is calculated and stored and can be reused as needed. What distinguishes our method is that the visibility list is transmitter location independent, is a three dimensional implementation and is highly computationally efficient. The division of the building into free and filled convex spaces leads to an efficient Method of Images reflection and diffraction path generation algorithm. This technique can be used to optimise the locations of base transceivers in a highly efficient manner. The first step in producing this tool is the generation of efficient ray-tracing algorithms. The ray-tracing algorithm was specifically designed for later incorporation into a transmitter optimisation algorithm. This requires a fast ray-tracing method because of its computationally intensive needs - running multiple times over a point-to-multipoint grid. Our algorithm is executed for sample building environments and then for a real building and compared with measurements to confirm its validity. It is clear that the results are in good agreement but do indicate that a highly accurate spatial modeling of the building is required.

Index Terms—ray-tracing, convex space, propagation, measurements.

I. INTRODUCTION

THE use of ray-optical methods to generate ray-paths in radio planning tools for indoor and outdoor scenarios is well documented in the literature of which some recent application examples are [1], [12], [18]. Most ray-tracing solutions use some sort of visibility or spatial decomposition algorithm which is needed to locate reflection, transmission and diffraction points. Ray tracing algorithms establish RF connectivity using these visibility algorithms. The visibility algorithm can be precomputed or form part of the main ray tracing algorithm. The visibility algorithm is one of the most important aspects of any ray-tracing tool. The execution time of a visibility algorithm largely determines the computational speed of any program that uses it for the optimal placement of wireless transmitters. This paper focuses on the creation of a transmitter location independent spatial partitioning algorithm for an indoor scenario using convex spaces to model the building. A sufficiently complex problem uses simplified convex spaces described as regular boxes to demonstrate the technique in this paper. From this point on we describe the convex spaces as “boxes”. The spatial decomposition is done by precomputing and storing a connectivity map of adjoining boxes including the adjoining boundary information. In so doing, the visibility information is also precomputed for every box making up the building. Because boxes have the property that rays passing through them enter at one point and leave at another point only, the visibility search space is greatly reduced by the choice of spatial decomposition. In short, the visibility information is contained in the convex space connectivity map. Although examined for the indoor case, the method can be also applied in outdoor propagation.

Many indoor ray-tracing algorithms use spatial decomposition to group wall/object facets in a bucket algorithm such as 2D quad-tree or 3D oct-tree algorithms [17] to allow the

search path algorithm to find the ray destinations in a more efficient manner. The spatial decomposition is used to reduce the time in the search path space by localising the search to the vicinity of the ray in question. Many outdoor ray-tracing algorithms perform polar sweeps [4] around the transmitter in 2D and 3D to find surrounding visible objects. These objects are not always close to the transmitter but they immediately reduce the search space of the first order ray-paths to a small region about the transmitter. The algorithm in this paper uses a spatial decomposition of the building to provide a natural subdivision of the building into storeys, rooms, corridors and unfilled spaces to quickly determine transmitter and receiver locations. It avoids the use of a conventional visibility algorithm altogether. The visibility space of a transmitter/receiver in a room is dictated by the space the rays must travel through. If these spaces are chosen to be convex in structure with known connectivity to adjoining spaces the method presented here is similar to the tetrahedron ray-tracing technique of Yun[19] except that it also includes the walls, doors and windows and takes into account a more natural subdivision of these objects. Also, the visibility from the antenna or images of the antenna is not actually required since the ray leaving a convex space can only leave through a small number of adjoining boundaries. If the connections to all other adjoining convex spaces are known, the search space in which the ray can travel is fully known before any ray-path search algorithm is implemented. This means that the building gives a natural set of connective paths along which a ray can travel. Also, it is clear that this spatial algorithm leads to a transmitter location independent algorithm for spatial decomposition since every point in a convex space can see every point on its boundary. The visibility algorithm is therefore completely built into the building model and needs to be determined once only before a ray-path search is performed. Usually visibility algorithms must be implemented along the ray-path from the antenna and its images each time a new ray is being defined. However our algorithm does not have this restriction. No additional

visibility information must be computed while the ray-path search is being performed leading to large computational efficiencies in particular when multiple receiver points are involved.

There are three main varieties of ray-optical methods. These are ray-tracing, ray-launching and beam-forming techniques. Ray-tracing is the most commonly implemented form of the ray-optical approach because it generates the exact ray-paths between the source and receiver. Its main advantage is its simplicity. Ray-launching on the other hand uses multiple rays leaving the transmitter and arriving close to or at the receiver and so better accounts for multipath. The technique usually involves some form of geodesic sphere theory. Beam forming is commonly implemented because the user of the planning tool often wishes to calculate point-to-multipoint signal power values. For example in O'Brien[11] such a point-to-multipoint ray-tracing code using a 2D beam-forming technique is given.

In addition a large number of papers exist on the planning and design of indoor wireless networks such as in CINDOOR[16], Stola[14], Inanoglu[5] to name but a few. Though there exist a large number of ray-tracing models, to the best of our knowledge our approach has not been documented in the available literature though a simpler 2D convex-space based approach (using triangles to model terrain) was employed by Yun et al[18] for outdoor propagation. Yun et. al. [19] also applied their technique using 3D tetrahedrons in a rudimentary environment consisting of one floor and applied their algorithm directly to the ray-path search algorithm. They did not take into account the additional computational saving achievable by decomposing the building a-priori into connected convex spaces which speeds up the computation even further and allows the point-to-multipoint algorithm to reuse the algorithm repeatedly again. The tetrahedron method could be extended to handle more sophisticated problems by noting for instance that a dome can be modelled as a single convex space whilst still retaining a transmitter independent visibility/spatial connectivity algorithm.

The way in which a building geometry is stored determines how quickly one can extract visibility information. Many planning tools use a drawing exchange format to store building geometries generated by AutodeskTM. However, recently people have begun to use sophisticated building information modelling (BIM) to describe the layout of a building and so compute its expected energy consumption McGlenn[9]. The method described in this paper uses the type of polygonal layout achievable with recent BIM layouts. However, in this paper we do not describe how the polygons/boxes are extracted since this will be specific to each tool and would require a lot of object oriented procedures to nest polygons within polygons in so doing hiding complex internal wall structures.

Given the regular geometry of most buildings it is natural to model the building as a set of boxes (six faced polyhedra or four faced pyramidal polyhedra). In this paper the algorithm will be demonstrated for the six sided polyhedra with planes existing in the x, y and z coordinate planes. This gives us enough flexibility to test the system in a real and fictitious environment whilst showing that the proposed layout is computationally efficient.

We begin by describing the data structures required to describe the building and the information that can be stored for reuse making the method computationally more efficient than current methods. The algorithm that uses this data is then described. The implementation is then run through a comprehensive set of tests to show that it is algorithmically correct and is tested against canonical solutions and real building measurement data.

II. BUILDING DESCRIPTION AND ALGORITHM

A. Building Storage Using Convex Spaces

A convex space/box has the property that a line drawn through it will intersect at most two points on its boundary. A six faced polyhedron (see Figure 1) is thus a convex space. If a building is stored as a set of boxes, it is then possible to navigate from one point to another across many boxes very quickly. This will form the basis of the spatial decomposition and provides a natural visibility algorithm. Assuming the building can be completely described using a set of boxes, the connections between each box need to be determined only once. This information is then stored for reuse. We should point out that storing visibility information for reuse is itself not a new idea (see for example [3]). However the algorithm presented here is transmitter location independent at the visibility level although it still uses a method of images approach for each transmitter which itself is not transmitter location independent.

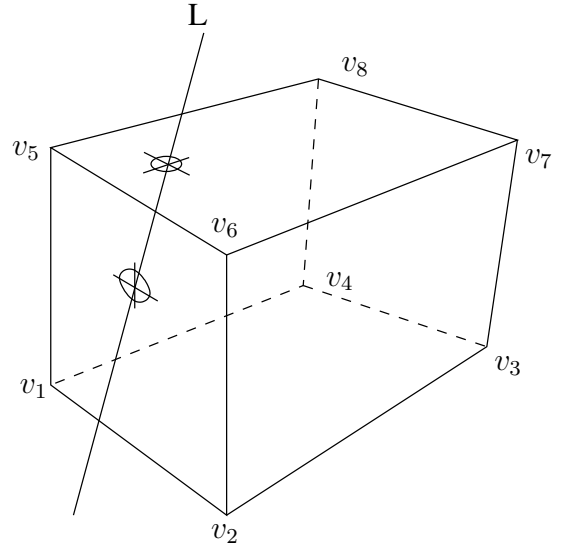


Fig. 1. An example of a convex space/box intersected by a ray/line. All rays enter and exit at one point.

Complicated building geometries including L-shaped rooms and torus shapes must be split into a discrete number of boxes (see Figure 2) otherwise the algorithm described in this work will omit ray paths in the generation process.

Each box has six boundaries $\{B_j\}_{j=1}^6$, each of which is described by four vertices $\{v_i\}_{i=1}^4$ or by the equation of a plane. A useful numbering system for the boundaries is defined such that for two boxes C_i and C_j touching one another, if C_i is connected to C_j by boundary B_k , then C_j is connected

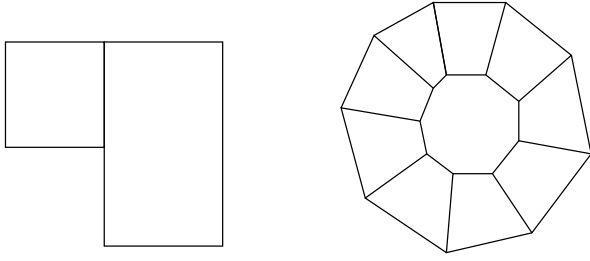


Fig. 2. An L -shaped region seen from above and torus shaped region split into boxes. Failure to split results in lost rays.

to box C_i by boundary B_{7-k} (see Figure 3). By way of example, an internal wall with a specific boundary B_2 can touch the $B_{7-2} = B_5$ boundary of an unfilled box adjoining it. Similarly, B_1 with outward normal pointing down from the ceiling will link to B_6 of the open room and boundary B_6 with outward normal pointing up from the floor facet links to B_1 in the open room.

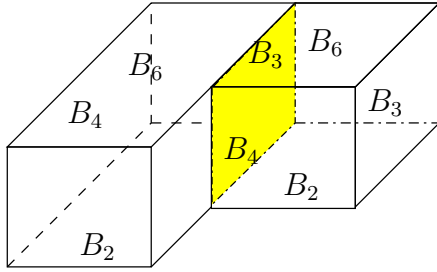


Fig. 3. Two boxes connected at boundaries B_3 and B_4 . Common boundaries have information about adjoining box numbers.

A building consists of filled or empty boxes. By empty boxes we mean that the box contains empty space. Solid boxes can be lossless or lossy dielectrics. If we want them to be impenetrable to EM radiation then we define them to be comprised of PEC material.

A building consists of stories containing filled or empty boxes. Stories in turn are comprised of rooms, and rooms themselves are empty boxes (see Figure 4). A diagram showing the connections between the boxes is given in Figure 5. Floors, walls and doors are all solid boxes.

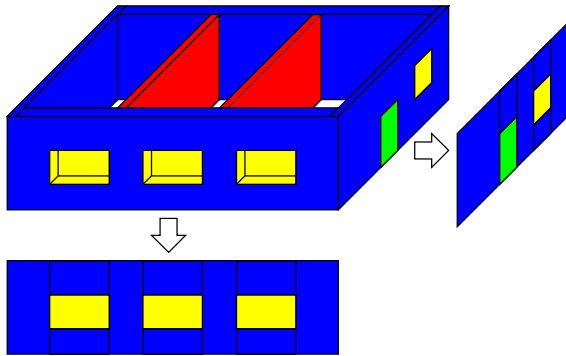


Fig. 4. A typical building example comprised of filled and empty boxes. These include interior/exterior walls, windows, doors, ceilings, floors and stories, but most importantly the unfilled space between them (visibility space).

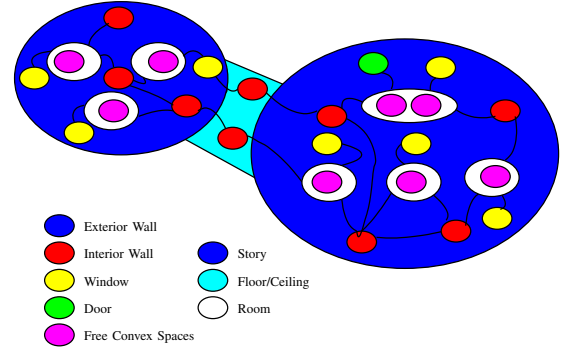


Fig. 5. A schematic representation of a hierarchical description of a building consisting of stories, floors, walls, windows, doors and empty spaces.

A floor may contain apertures to model lift shafts or stairwells which are empty boxes. Doors and windows are given different depths when compared with the surrounding walls to give more realistic modelling.

The way we obtain spatial connectivity information is first by finding the equation of the planes at the boundaries of each filled and unfilled box in the building. We then calculate the distance of each box from every other box using a dot product distance formula for points to planes. If the dot product is zero between a plane and another plane they are possibly connected but a further check is required to determine if any of the four points of a potentially adjacent boundary are within the boundary of interest. This calculation is performed for all planes until the connectivity for the whole building is obtained. Doing this for the whole building will give us all the connections we need. This information needs only be calculated once and stored. If for some reason a new box is added to the building, the software can automatically recalculate the inter-connectivity information to allow for the change.

B. Computational Savings and Efficiency

Using the method proposed here there are a number of computational efficiencies some of which are not so obvious without explanation by way of examples. Many ray-tracing methods in the literature report savings of 25-36% in the ray-tracing by employing non-conventional spatial methods (see Yun[19]), but such methods do not generate linear improvements in their efficiency. Any ray-trace should be split into three categories To arrive at the total compute time the ray-tracing should be determined by summing the compute times of the following three components:

- 1) The time to implement the Method of Images for the transmitter and all diffraction points.
- 2) The actual search algorithm to define all ray-paths, such as reflections, transmissions and diffractions.
- 3) The time to calculate the signal power at every receiver point.

It is very difficult to remove any images in the Method of Images without losing ray-paths. The signal power is straightforward to calculate so this paper concentrates on improving the search path for finding rays.

We give a simple example to illustrate the computational saving that can be made using a connectivity of convex spaces to describe the building (see Figure 6) and we compare with existing methods.

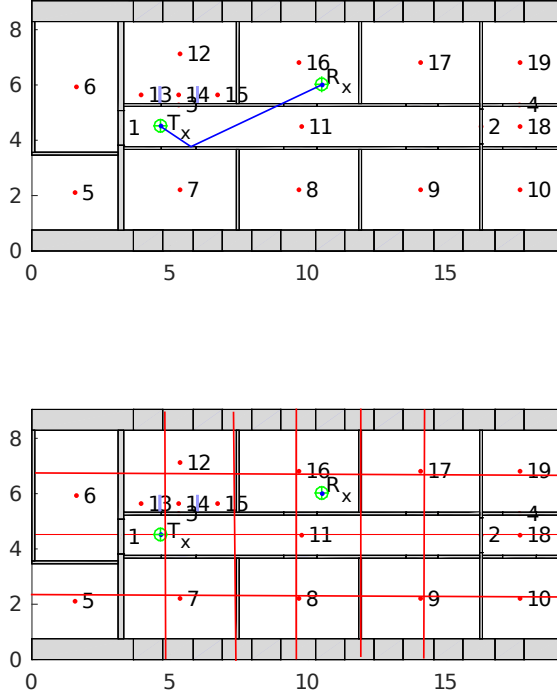


Fig. 6. A sample building used to explain where computational savings can be made when obtaining a single reflection from a transmitter T_x to a receiver R_x . And below, some of the quad-tree splitting that can produce faster ray-tracing.

The transmitter T_x is located in the corridor which is contained in Room/Box 11 and a receiver point R_x is located in room which is contained in Room/Box 16. As stated already, the building is discretised into a set of boxes rather than planes. This means that the error in the positioning of obstacles is reduced when compared to other methods that present the walls as infinitely thin 2D partitions as in Yun[19]. This means also that the location of the transmitter in Room 11 will be more accurate. Assuming we know that there is a reflection off the wall joining Room 11 to Room 7 we know there are a number of checks that need to be performed. Different reflection/transmission point finding algorithms perform these checks in different ways as detailed below:

- 1) The **oct-tree** or **quad-tree** method will recursively divide the building into volumes to map out the vicinity of the 2D planes or 3D volumes representing obstacles (as indicated by the red lines in Figure 6). Note that some objects will appear in multiple quadrants making the algorithm a little slower. Usually finer meshes are used to circumvent objects being contained in multiple quadrants. In the example in Figure 6, the building is

divided into 4 quadrants and each of these quadrants are subsequently subdivided into quadrants. Therefore the transmitter is quickly located in the 2^{nd} sublevel of the quadtree but this level does not contain the reflection point. The path searching algorithm must move in a south-easterly direction to another rectangular boundary to a quadrant containing some obstacles. The reflection point on a neighbouring wall is found by moving towards that reflection point and finding the boundary containing it. This boundary is a 2D plane containing a wall, a door and the space above the door connecting to rooms 7. The reflection point on the door is found and a path is made from the reflection point to the receiver R_x by travelling through a number of bounding boxes and arriving at a transmission point at the wall. The transmission coefficient is calculated and further obstacles must be looked for to arrive at the receiver point.

- 2) A **non-spatial visibility algorithm** can be used to perform a polar sweep about the transmitter T_x and images of the transmitter to determine the set of possible walls from which the reflection can occur. The walls, within the correct angular range, to determine the correct reflector. Therefore this requires a complete sweep of box number 11 which will result in 8 door cavities and over 16 visible edges in a 2D environment and many more in a 3D environment. The reflection points and transmission points are found by finding the angle of arrival of the ray at the reflection point and transmission point in question. This requires a large number of ray traversals. Also the visibility algorithm would change significantly if the transmitter were placed in a different location.
- 3) Using the **tetrahedron method** of Yun[19], any room represented by a box can be split into tetrahedron spaces. A rectangular room can be composed of a minimum of 5 tetrahedrons and usually consists of 6. This means that the ray traversal is relatively computationally expensive compared to the case where boxes are used. Also, the interior wall thicknesses are not taken into account as in Yun[19] so there will be reflection point inaccuracies and problems in properly calculating the transmission coefficients. Using this approach there are 1-2 tetrahedrons traversals to arrive at the reflecting wall. Then there are 3-4 traversals needed to arrive at the wall on the other side. Because of the wall position inaccuracies, wall thickness will have to be associated with the 2D planes to arrive at the correct reflection and transmission coefficients.
- 4) Using the **convex space approach** all convex spaces are predetermined and reusable when the transmitter is moved to a new location. Also, using the correct type of convex space (in this case polygon with 8 vertices), the search space is reduced compared with that of the tetrahedron method. The ray travels towards the reflection point on the boundary of the box containing the transmitter which is in box number 11. There is still a check run through the list of bounding boxes on the

wall that contains the 3 doors connecting to rooms 7, 8 and 9 in order to find the reflection point. However these are comparable to the quad-tree bounding box checking method. When the reflection point is found the reflection coefficient is obtained and the transmission point on the opposite boundary is found by determining the objects on the boundary of the adjoining wall. Note that there is no need to search for intermediate objects and once the transmission coefficient is found there are also no checks needed in connecting the transmitted ray to the receiver point.

Here we propose that a building be split into a natural subdivision that accommodates the different shapes of the rooms, doors, windows, lifts, corridors and stories. All convex spaces should be determined and these will provide a natural visibility algorithm for the whole building. It should be noted that convex spaces can be placed in oct-tree structures if there are a large number of them in a small region. Also, in the example shown above, there is a list of 2D planes to traverse on a boundary of a convex space to arrive at the correct reflection point on an obstacle. This process can also be speeded up by implementing a quad-tree inside the boundary elements of a convex space. Additionally, this method can achieve extra computational gains by combining the existing oct-tree algorithms with this convex space algorithm to find the originating convex space for the transmitter or receiver (this will result in a hybrid method, that is not covered in this paper).

If complex convex spaces, such as a dome with a large number of boundary facets are to be modelled, it would also be beneficial to employ a oct-tree within the convex space to quickly determine the connecting facets in the reflection/transmission algorithms.

In the results section we give a brief overview of the breakdown of overall computation. It is important to note that the Method of Images calculation includes no inherent speed up from this method. The main computational savings are in the determination of the reflected and transmission points and the paths along which the rays travel to arrive at these reflectors and transmission points.

C. Determining the Reflection Points in a Ray-Path

There are a number of algorithms that need to be implemented before finding the reflection points in a building. One important algorithm is finding the box containing the receiver or transmitter points of interest. We know that the observation points must lie in empty boxes. In Figure 5 we see that the empty box are contained inside stories so it is easy to determine in which story the point is. It is then a simple matter of finding in which empty box the point lies by traversing each empty box. It should be noted that a quad-tree or oct-tree can be employed to speed up this operation inside building stories.

The algorithm for determining the reflection points in a ray-path of order n for a specific branch of the image tree is described as follows:

- 1) First determine the empty boxes where the receiver and transmitter points lie. This is achieved by first finding

in which storey the transmitter/receiver point is located. The room boundaries are then analysed to find these locations. The boxes within this room are then traversed to find which one contains the point of interest. It should be noted that this example has rooms consisting of one box only but can be generalised to include rooms of more complex geometries consisting any number of boxes. The receiver is pushed onto a stack of point data which will form a ray-path list. The receiver is also given a key (see Figure 7).

- 2) Create a line segment from the receiver point to the n^{th} order image in the image tree.
- 3) We extend the line through the current box to give a point of contact P with the next box. If the point of contact P is on boundary b_i then we know that it is on the boundary of b_{6-i} in the adjacent box. We have already stated that all boundaries are connected to adjacent boundaries so it is easy to determine which adjacent box contains the point of contact on its boundary.
- 4) If the point of contact P is on a filled box then we need to update our transmission count. If we pass through too many filled spaces, that is, a number greater than the maximum that we wish to allow, then that ray-path is discounted.
- 5) If the point of contact P is on the positive side of the line, that is, $d = (P - v) \cdot \hat{n} > 0$ where v is some point on the reflective plane and \hat{n} is the outward normal from the plane, we know that we must continue until $d = 0$. If we find that $d < 0$ then we have not found a point on the reflective plane and therefore the ray-path is invalid.
- 6) The line segment can intersect at most two points on a box, that is at P and Q . We already know the location of P so it is easy to determine Q on the opposite side of the box. We then continue with our procedure as given in the previous two steps until we determine the point of reflection. Once we obtain this point of reflection we push it onto the ray-path stack and this becomes the new starting point for the algorithm. From here we repeat the algorithm except that we use the previous image point to determine our line segment.
- 7) Repeat the process until no image points remain.

D. Determining Reflection and Transmission Points

As is customary, the ray-trace is restricted to a preassigned maximum order of reflection and diffraction points. In addition, the UTD diffraction coefficients are generated based on the work of Kouyoumjian, Pathak[8] for single diffraction scenarios. We first address the creation of the image tree and calculating the reflection and transmission points. The remainder of this section will then describe how the incident and diffraction angles are found.

The steps required to determine the reflection and transmission points in a building are:

- 1) Determine all images of a transmitter $T_x \equiv T_x(x, y, z)$ using the Method of Images.
- 2) Determine the valid reflection points by tracing back from the receiver point to the image of a transmitter.

- 3) Determine the transmission points in the process and store this information.

We now present each step of the algorithm.

E. Point Method of Images

The Method of Images [2] is used to determine the locations of the transmitter images in the building. This is a method that is widely used because it can be used to determine ray paths up to a given order. Because we have now described the building as a set of boxes it is possible to use our compact storage of the building to speed up the Method of Images process as will be seen below. It should be noted that we do not expect to make any computational savings from this part of the algorithm but because the building is stored as convex spaces rather than flat 2D planes we need to adjust the algorithm accordingly. We must ensure that there is no extra computational expense in dealing with these convex spaces.

The algorithm is defined as follows:

- 1) A set of reflective planes about the transmitter must be determined. We note that reflection points lie on the boundaries of filled boxes so we need not check the connections with empty boxes inside rooms and apertures. From each of the six boundaries of these filled boxes we determine a unique set of equations for reflective planes in the form $Ax + By + Cz = D$, oriented so that the point $\hat{n} = (A, B, C)$ defines an outward normal to the plane.
- 2) Once the reflection planes are obtained, the transmitter T_x is defined to be the 0^{th} order reflection point $T_x^{(0)}$ in what can be described as a 'tree' of images (see Figure 8 below).
- 3) We test the condition $d = (T_x^{(0)} - v) \cdot \hat{n} > 0$ where v is some point on the plane to determine whether the transmitter $T_x^{(0)}$ is on the positive side of each reflective plane. If $d > 0$ then trace the image point $T_x^{(0)}$ in the plane to obtain a new image point $T_{x_1}^{(1)}$. This operation is performed for all reflective planes until all first order images are created producing a list $\{T_{x_i}^{(1)}\}_{i=1}^{N_0}$ with a pointer back to the index of the originating image that created it. N_0 denotes the number of images generated from the base transmitter.
- 4) From the j^{th} first order image $T_{x_j}^{(1)}$ we generate second order images of the form $\{T_{x_i}^{(2)}\}_{i=1}^{N_j}$ and higher order images are obtained by repeating the procedure in the previous step for all transmitter images at the current level. N_j denotes the number of images generated using the current transmitter index j . Each node in the tree need only store the transmitter location and a number reference to the generating 2D-plane.

The image tree is created in such a way that a node in the $(n+1)^{th}$ level of the tree can connect with the node in the n^{th} level of the tree. We do not need to store the information about the reflective planes that generated the image because the node in each level contains a reference to the generating plane indicated by the index j or k in level 1 and m or n

in level 2 (see Figure 8). It is a simple matter to obtain the midpoint between the n^{th} and $(n+1)^{th}$ points and then obtain the reflective plane.

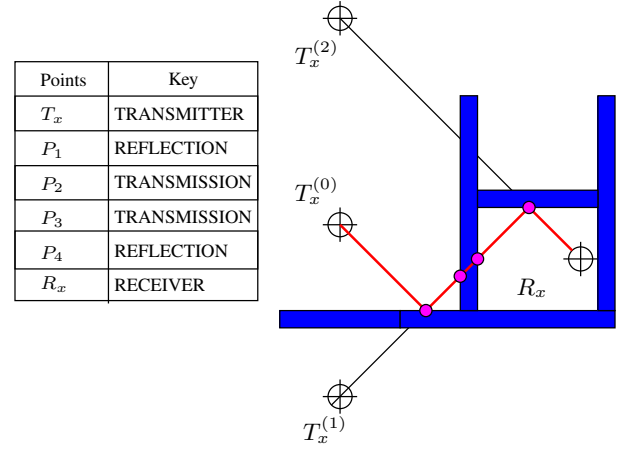


Fig. 7. The reflection and transmission points generated by the Method of Images along with a set of keys that inform the propagation model about which parameters to apply.

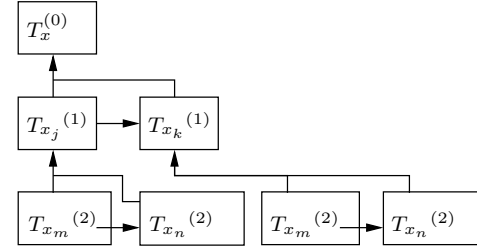


Fig. 8. The Method of Images tree including lists of images of each image in the hierarchy. In this illustration, the transmitter T_x is reflected in the j^{th} and k^{th} plane and then reflected again in the m^{th} and n^{th} planes.

F. Finding Points of Diffraction

The algorithm for finding diffraction points, or in the case of corner diffraction, finding virtual diffraction points is described as follows:

- 1) Determine all diffraction edges in the building.
- 2) Sort the diffraction edges and images of the diffraction edges in a diffraction tree created using the Method of Images.
- 3) Find the n^{th} order ray containing one diffraction using the reflection and diffraction trees.

Each of the steps in the algorithm are covered in the next sections.

G. Determining Diffraction Edges of Filled Convex Spaces

The algorithm for finding all valid diffracting edges in an indoor environment incorporating the convex space methodology is defined as follows:

- 1) When using diffraction algorithms within a building we must first define all valid diffraction edges. Only filled boxes can generate a valid diffraction edge. An edge is

contained on two boundaries B_i and B_j of a box and so we obtain 12 permutations of the couple (i, j) :

$$\begin{aligned} &(1, 2), (1, 3), (1, 4), (1, 5) \\ &(2, 3), (3, 4), (4, 5), (5, 2) \\ &(6, 2), (6, 3), (6, 4), (6, 5) \end{aligned}$$

- 2) If we extract a list A_i of the empty boxes adjacent to B_i and a list A_j of the empty boxes adjacent to B_j then for each box C_m in A_i and each box C_n in A_j we check to see if an adjacent boundary to C_m is in A_j or a adjacent boundary to C_n is in A_i . We say that this boundary is part of box C_k . Assuming there isn't a boundary then we have no valid diffraction edge, but assuming there is, we need to do some further checks.
- 3) By intersecting the edge of C_k between B_i and B_j with boundary B_{6-j} of C_m and B_{6-j} of C_n we form a diffracting edge.
- 4) If B_{6-i} in C_m and B_{6-j} in C_n are coplanar then we have no valid diffracting edge, otherwise we do have a valid diffracting edge.

H. Creating a Method of Images Tree for Diffractions

If we describe the diffraction edge by its endpoints, then the method is identical to the Method of Images for a point except that both points are reflected in reflective planes instead of one point.

The tree will contain many diffraction edges at the 0th order layer and will have images of the diffraction edges when reflected in planes at the 1st layer (see Figure 9).

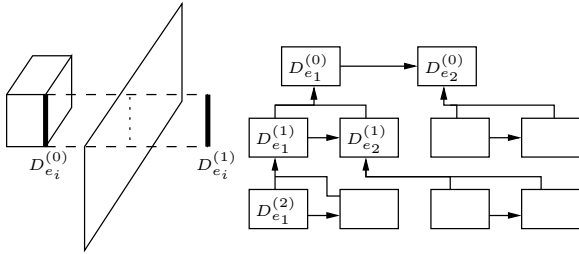


Fig. 9. An example of the image point lists generated by Method of Images for diffraction points obtained by diffracting edges in filled convex space boundaries.

I. Determining the Diffraction Point and Ray-Path

We will see that for an m^{th} order ray with one diffraction point which is the n^{th} order term of the list, that there are $m-n-1$ reflections preceding the diffraction point, and there are n ray-paths between the diffraction point and the receiver.

Figure 10 shows how a 4th order ray-path is decomposed into a path from $T_x^{(m-n-1)} = T_x^{(2)}$ to the diffraction point $P_{m-n} = P_{4-1} = P_3$ on the diffraction edge $D_e = [D_e^1, D_e^2]$. There is then a path generated from $D_e^{(1)} = [D_e^{1(1)}, D_e^{2(1)}]$ to the receiver point R_x .

Before calculating the ray-path we need to describe in more detail how the point of diffraction P is found given that the ray-path contains $(m-n-1)$ reflections, one diffraction and a subsequent n reflections:

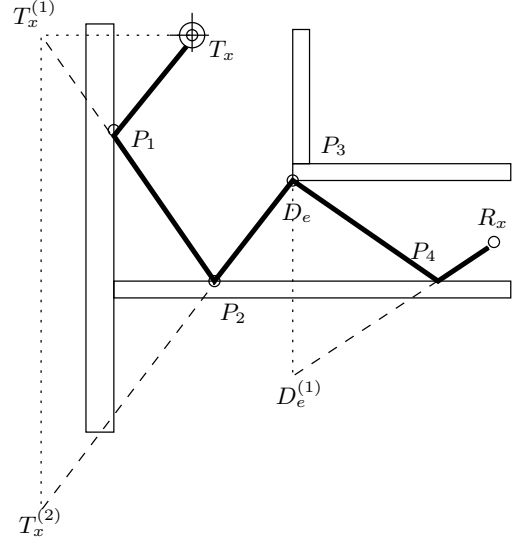


Fig. 10. An example of a ray path for a 4th order ray including one diffraction point at P_3 . Note that no image tree is used for any receiver point.

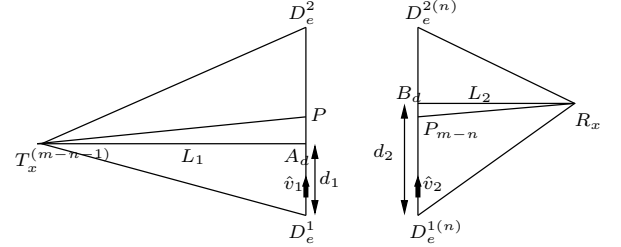


Fig. 11. The ray-path of order m with a diffraction point at the $(m-n)^{th}$ point.

- 1) Calculate the lengths $L_1 = (T_x^{(m-n-1)} - D_e^1) \times \hat{v}_1$ and $L_2 = (R_x^{(m-n-1)} - D_e^{2(n)}) \times \hat{v}_2$ using shortest distance formula to D_e and $D_e^{(n)}$ respectively (see Figure 11).
- 2) Construct the distances $d_1 = (T_x^{(m-n-1)} - D_e^1) \cdot \hat{v}_1$ and $d_2 = (R_x - D_e^{2(n)}) \cdot \hat{v}_2$ which in turn gives the points of contact $A_d = D_e^1 + \hat{v}_1 d_1$ and $B_d = D_e^{1(n)} + \hat{v}_2 d_2$.
- 3) Define the length $d = d_2 - d_1$.
- 4) Using the ratio $L_1 : L_2$ it is possible to locate P using the formula: $P = A_d + \hat{v}_1 \frac{L_1 d}{L_1 + L_2}$.

Once the diffraction point is found we are in a position to describe the method for finding the ray path. This algorithm will again be computationally efficient as a result of the box description of the building. Another reason this algorithm is computationally efficient is the continuous reuse of the same diffraction tree. It is never a good idea to build the diffraction tree into the point image tree. If the diffraction tree is inserted into the point image tree then we end up with multiple copies of the tree which is very inefficient. Instead it is better to know exactly what branch of the diffraction tree (n^{th}) needs to be extracted given that we know the order of reflection $(m-n-1)$ preceding the diffraction point. This efficient method was described by Schettino[13] and many other authors.

The method for extracting ray-paths is described as follows:

- 1) Let the receiver R_x be the $(m+1)^{th}$ image point.
- 2) Looping over all n^{th} order diffraction edges of the

diffraction tree, extract a single term $D_e^{(n)}$ and also determine its generating edge D_e . The generating edge is found by working back up to the 0^{th} order level of the diffraction tree (see Figure 9).

- 3) Extract a transmitter image of order $(m-n-1)$ and then construct the diffraction point P_{m-n} using the method described earlier in this section.
- 4) If P_{m-n} is contained in D_e then it is a valid diffraction point. If it is not contained in D_e then it is still needed for the corner diffraction algorithms and is denoted as a virtual diffraction point. Each reflection point P_{m-n+j} is constructed for all $j = 1, 2, \dots, n$ by working back from the receiver point R_x to the diffraction point P_{m-n} using the algorithm described in Section II-C.
- 5) The reflection points P_j for all $j = 1, 2, \dots, m-n-1$ are generated by working back from the diffraction point to the transmitter using the same algorithm described in Section II-C.

The construction of the double diffraction points is not as straightforward. In the past the method of finding the diffraction points was easily described for coplanar edges but in practice the edges may be oriented in any direction, especially in an indoor environment. In this case the Newton-Raphson formula is used to define an iterative solution which converges to the valid diffraction points.

III. IMPLEMENTATION DESCRIPTION

When a building and room connectivity description are first analysed the software will produce a connectivity map much like that shown in Figure 5. All of the empty boxes are collected together to define an efficient way to traverse each individual room later on. The connectivity within the building is flagged as being either of type room, internal wall, external wall, door or window. The stories for instance consist only of internal wall types. The six boundaries of any convex space/box are then tagged with coplanar boundaries that reside partially or fully inside them from surrounding boxes.

All of this information is computed once only and then stored. It is important to state that the Method of Images technique for reflections and diffractions is dependent on the transmitter location but the visibility and transmission algorithms are not and this is what gives rise to a computationally efficient algorithm. The visibility algorithm is however completely dependent on the direction of any ray as it travels through the building from box to box. The algorithm therefore depends on the ray-path connecting the n^{th} order and $(n+1)^{th}$ layers of the Method of Images (MoI) tree.

IV. BUILDING DATA STORAGE

The main purpose of this paper is to produce a method that determines a set of convex spaces which are filled or unfilled and arrives at a fast ray-tracing algorithm that is independent of the transmitter location. It is important to state that the building information was input by hand into data files based on sample building structures or were obtained from draftsman plans of buildings. In the future, with wider use of building information modelling (BIM), it should be possible to

automatically discretise buildings into convex spaces that can be fed directly into the ray-tracing and building connectivity algorithms given in this paper. Each convex space was defined as a closed polyhedron consisting of 8 vertices, describing the type of wall/window/door that the convex space represents and attributing relative permittivity, permeability and conductivity values to it as follows:

```
ConvexInteriorWall
0.0 0.0 0.0
31.0 0.0 0.0
31.0 21.0 0.0
0.0 21.0 0.0
0.0 0.0 0.2
31.0 0.0 0.2
31.0 21.0 0.2
0.0 21.0 0.2
DielectricParameters
4.0 1.0 0.04
```

The string at the start is not currently used but serves as a marker for the information that follows.

The algorithm for extracting information automatically from a BIM system is not defined generally but should be achievable by defining, for example, wall materials and thicknesses. Currently, most BIM systems do not include permittivity values for walls which is a major difficulty for the production of the type of algorithm as described in this work. However, given the location of walls in buildings and their thicknesses it is possible to attribute some recommended values for the relative permittivity, permeability and conductivity such as recommended in Chap. 12 of Balanis[2].

The propagation algorithm defined in the following verification tests uses the permittivity values to produce reflection and transmission coefficients as defined by James[6], Balanis[2] for all filled boxes. In addition, the UTD diffraction coefficients given by McNamara[10] are applied to a single diffraction edge.

V. VERIFICATION OF RAY-TRACING AND COMPARISON WITH MEASUREMENTS

The correctness of any ray-path finding method needs to be tested thoroughly. Where possible a verification of the working modules, written in C++, must be compared with previously generated results. Also a comparison with real-world measurements is necessary since this verifies the validity of the assumptions made in the tool and shows that the tool as a whole is working correctly.

A. Correctness of Diffraction and Reflection Ray-Paths

The diffraction method needs to be evaluated for a test case. A simple example that is suitable uses a single room containing four diffraction edges yielding symmetries in the ray-path solution across the lines XX and YY (see Figure 12). The transmitter and receiver are set up in such a way that they always ensure symmetry. If the ray-paths are found to be asymmetric then it is clear that some of the paths are missing. Because there are a number of convex spaces meeting at a

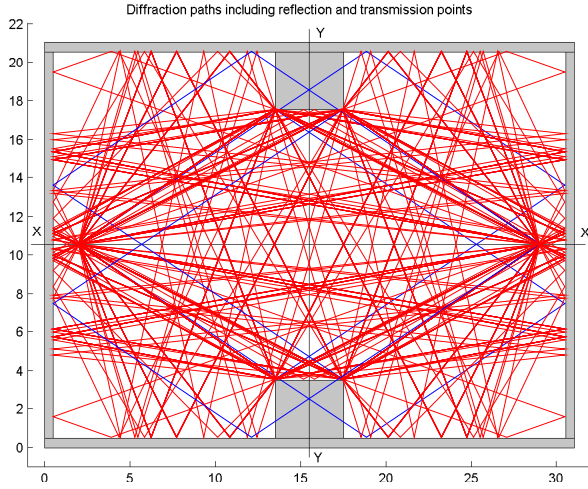


Fig. 12. Diffraction paths and reflection paths inside a single-room with symmetries in XX and YY to ascertain the correctness of the ray-path finding algorithm.

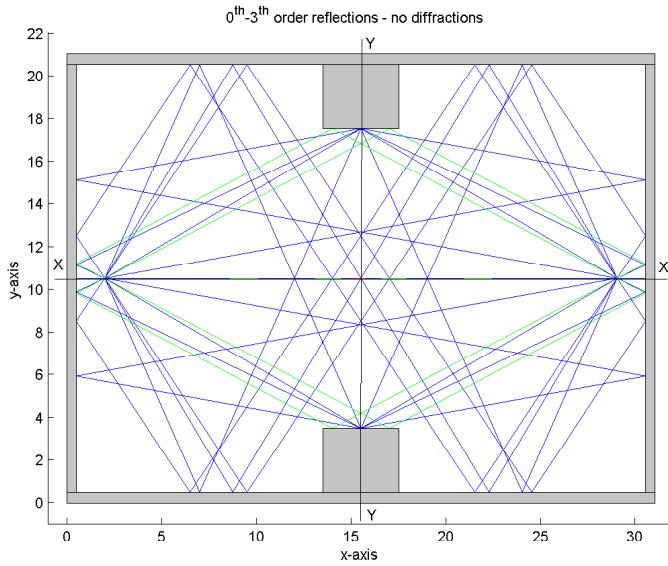


Fig. 13. $0^{th} - 3^{rd}$ order reflections inside a one-room building with symmetries in XX and YY to ascertain the correctness of the ray-path finding algorithm.

single diffraction edge it is not a simple matter to guarantee the correct ray-paths for the calculation. It turned out that when tracing a ray from the diffraction point back to the transmitter it was important to move the starting point by a perturbation along the line to be traced so that the correct box containing the diffraction point was found. Otherwise the algorithm starts off in the wrong box and can detect an additional transmission through a wall - which is incorrect.

Again, setting up the single-room building as for the diffraction case, the valid reflections should form a symmetric pattern through the line segments XX and YY. This is the case as can be seen in Figure 13.

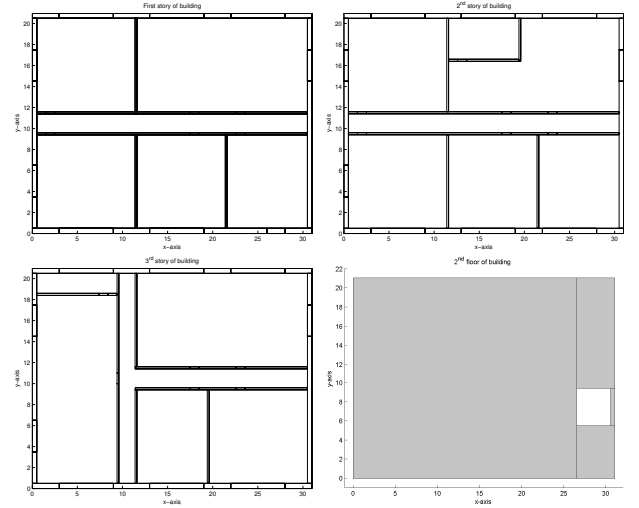


Fig. 14. A plan view of three stories in a sample building and 2^{nd} floor used to test the ray-tracing propagation algorithm.

B. Software Tests

The software was tested as follows: A building was generated comprising of a total of 230 boxes, with 4 floors including the roof space, 3 stories, with internal walls, external walls, doors, windows and 6 or 7 rooms per story. The floor contained apertures to represent stairwells. The building plan can be seen in Figure 14. The operating frequency of the transmitter is $900MHz$. The second story contains an L-shaped room in the top right corner which must be split into two empty convex spaces. The third story contains a corridor (room) which must be split into two or three boxes depending on its design. For the examples that follow the doors of each room are assumed to be closed. All plots in this section were easily generated using Matlab functions once the building was stored as boxes.

The constitutive parameters at $900MHz$ for the main dielectric materials making up the building are presented in Table I.

Constitutive Parameters	ϵ	μ	σ
empty Convex Space	1	1	0
Exterior Walls	4.44	1	0.08
Interior Walls (guess)	4.0	1	0.04
Doors (glass, Balanis[2])	2.32	1	1.45×10^{-3}
Windows (glass)	2.32	1	1.45×10^{-3}

TABLE I
DIELECTRIC MATERIAL PROPERTIES AT $900MHz$ FOR THE SAMPLE BUILDING WITH RELATIVE PERMITTIVITY, PERMEABILITY AND CONDUCTIVITY

C. Test 1: Reflections Inside a Single Story

Presented here is a sample of the type of results that can be obtained from running the ray-tracing algorithm with reflections up to 3^{rd} order and no diffractions. For the purposes of this illustration the number of transmissions was set to be a maximum of three. If a greater number of transmissions were specified then the number of rays would be too numerous to give a presentable plot. The transmitter is placed at location (3, 18, 1.5) and the mobile terminal is placed at location

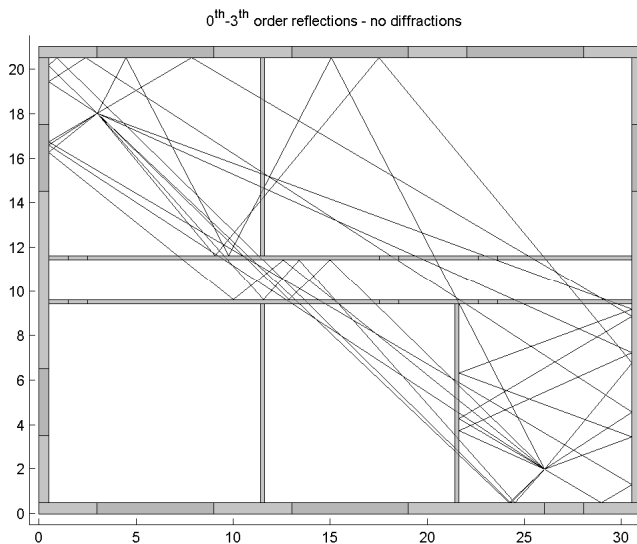


Fig. 15. 0^{th} – 3^{rd} order reflections inside the first story of the sample building.

(26, 2, 1.5). The transmission frequency is at 900MHz. All reflected rays of order zero to three are presented in Figure 15 along with an overlaid plan view of the first story of the building.

This plot does not display the full complexity of the ray paths because it is shown in a two-dimensional cross section. In many cases if there are a sufficient number of transmission allowed, the rays pass into other stories through floors and then re-enter from the opposite side back into the story where the receiver lies.

D. Test 2: Diffractions inside a Building Story

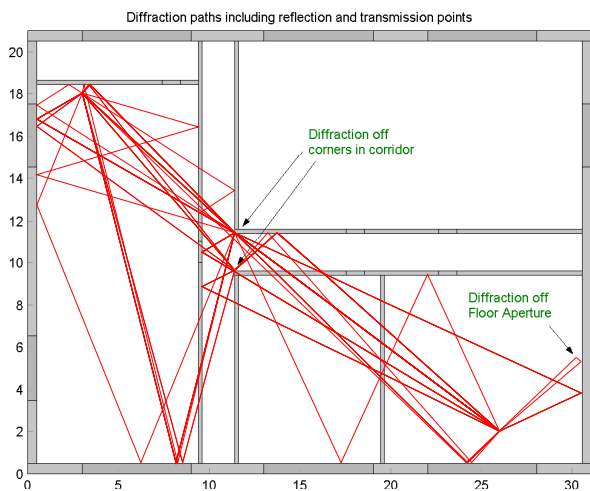


Fig. 16. Diffraction paths mixed with reflections inside the third story of the building.

The transmitter is placed at location (3, 18, 8.5) and the mobile terminal is placed at location (26, 2, 8.5) so that we

expect some diffractions from the edges of the corridor on the 3^{rd} floor of the sample building shown in Figure 14. As can be seen in Figure 16 this is the case. The maximum order of reflections is 3, diffractions 1 and transmissions 3, for this plot. There is also a diffraction from an aperture that exists on the 3^{rd} floor in the room where the receiver is located. The 3^{rd} floor plan is identical to the 2^{nd} floor plan shown in Figure 14.

E. Test 3: Signal Power Grid Plot

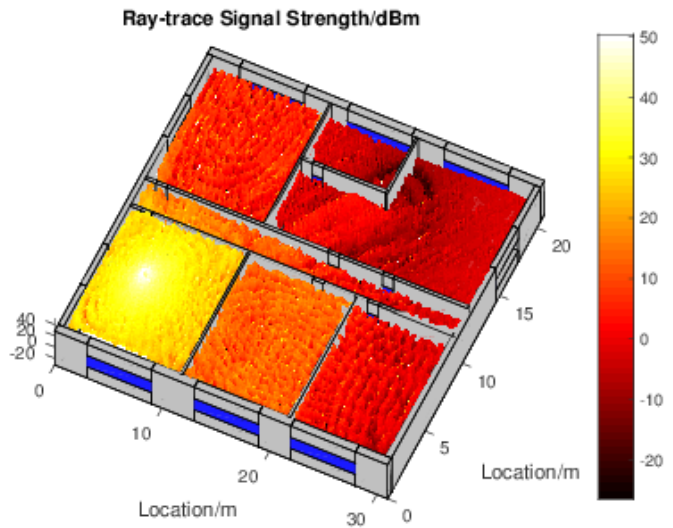


Fig. 17. A signal power grid generated from 9600 points at $\frac{1}{4}$ wavelengths on the second story of the sample building with a frequency of 900MHz and transmitter at (0.75, 0.75, 5.0) calculated in 93 seconds on an i7-3770 processor.

This test assumes a regular grid between (0.75, 0.75, 5.0) and (30.75, 20.75, 5.0) with step size $\Delta x = 0.25$ and $\Delta y = 0.25$ leading to the signal power computation at 9600 points covering the second story of the sample building. A frequency of 900MHz for a z -axis oriented Hertzian half-wave dipole at location (5.0, 5.0, 5.0) leads to the result in Figure 17 where the maximum order of reflections is 3, transmissions is 4 and diffractions, 1. The signal power is plotted in decibel metres. The building height is elongated in the z -axis so that the story and signal power grid can be overlaid. The results show large losses in areas which are heavily shadowed. The code was run on a i7-3770 processor with 8 cores. It executes in 93 seconds showing that it is very useful for use in a point to multipoint optimisation algorithm with multiple transmitters. An individual point to point ray-trace is calculated in approximately 0.01 seconds.

F. Ray-tracing based Propagation Model Verification

A real building located at Trinity College Dublin is the subject of a measurement campaign used to verify the validity of the propagation model used in conjunction with the novel ray-tracing method presented in this paper. The purpose of this comparison is to ensure that the propagation model

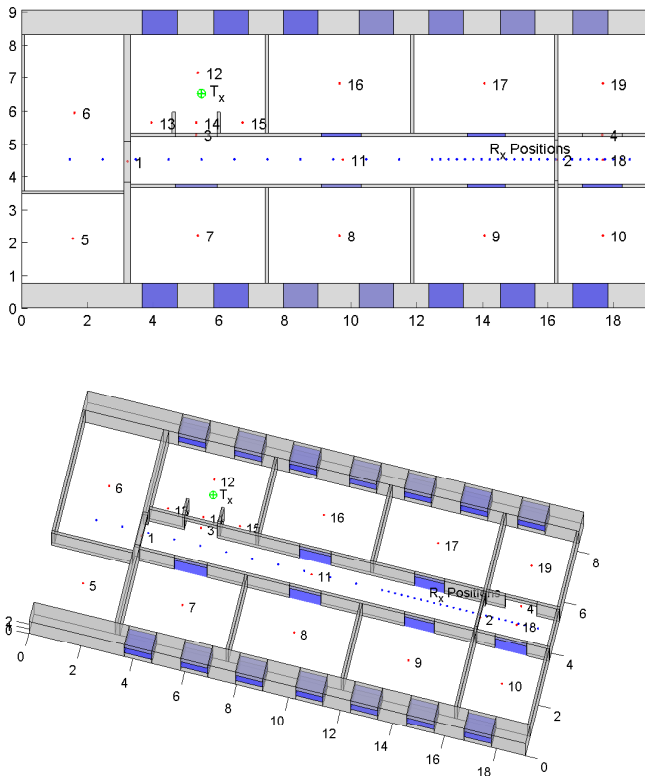


Fig. 18. A three-dimensional view of the top floor of the printing house at Trinity College Dublin including the transmitter point $T_x = (5.468, 6.51, 1.45)$ and multiple receiver points R_x . The transmitter T_x is placed in room 12 with double doors into the room both opened.

was correctly implemented and that the transmitter power and the relative permittivities of the media were given reasonable values. The building layout was input with the dimensions specified by the original draftsman. The building is shown in Figure 18. Two sets of measurement data were recorded and comparisons with measurements now follow for both sets of data.

The measurements were performed as follows: The receiver was placed on a trolley and moved along the centre of the 12.9m long corridor where 36 spot measurements were taken. At each of these locations five measurements were taken in a random walk fashion within a 1λ radius at and about the centrepoint. These results were then averaged to smooth out small scale fading. It was not feasible to use a larger radius since the corridor is narrow (approx 1.56m) and we wished to avoid spurious edge effects near the walls and doors. For each random walk measurement four measurements were taken and averaged to smooth out temporal fades. In total 720 measurements were taken along the corridor. The measurements were taken at shorter intervals as we moved closer to the transmitter which was located in one of the adjoining rooms because in this region we expected the signal to vary more rapidly due to the more pronounced effects of diffraction close to the transmitter NLOS location.

For the first measurement set there were open and shut doors. All open doors are shown in Figure 18.

G. Measurement Set 1

Figure 18 shows 36 receiver point measurement locations on a straight line that vary in position in the x-axis only. A transmitter was placed in one of the rooms of the building at location $T_x = (5.468, 6.51, 1.45)$ inside the empty box number 12. The receiver points were set up along the corridor and out into the landing area.

The room containing the transmitter has its doors open so as to form some diffraction effects down the corridor. The spacing between receiver points was 1 metre at one end of the corridor and 0.25 metres for the other end also shown in Figure 18. The shorter spacing was used to obtain a better measurement of the fluctuations in the field as it propagates through the corridor.

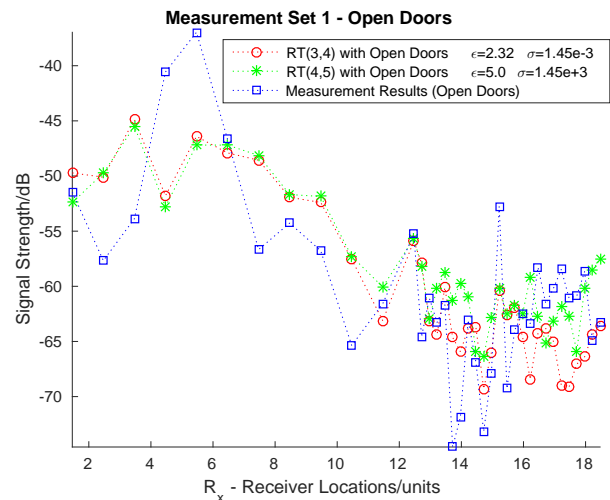


Fig. 19. 36 measurement points compared with ray-tracing results with differing relative permittivity/conductivity values and with double doors opened into room 12.

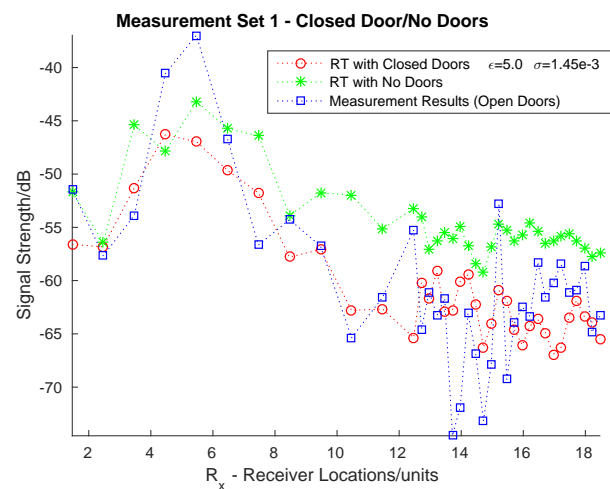


Fig. 20. 36 measurement points compared with ray-tracing propagation model results from a correctly modelled building (including doors) and with the propagation results from an incorrectly modelled building (no doors).

The measurement results along with the ray-tracing prediction at 36 receiver point locations are shown in Figure 19.

Using a relative permittivity of $\epsilon = 2.32$ and conductivity of $\sigma = 1.45 \times 10^{-3}$ the mean signal power deviates from the mean measurement values by 0.15dB whereas the mean signal power deviates by 0.178dB when using a relative permittivity of $\epsilon = 5.0$ and conductivity of $\sigma = 1.45 \times 10^{-3}$. The results show a deviation average of 4.8dB for the realistic permittivity values and 5.4dB deviation average for the unrealistic values but overall compare reasonably well. This might be explained by the fact that the walls are very thick and therefore the choice of conductivity values does not greatly affect the calculation of the signal power. In the next stage of confirmation by measurement we will see that there are other factors more important considerations than the permittivity values.

Again we used the same building for the ray-tracing prediction but this time with the doors closed and in the absence of doors in the office containing the transmitter. This is to show how the electric field can change significantly if the modelling of the building is incomplete/incorrect. These results are shown in Figure 20. It can be seen in the plot that the mean of the predicted signal power is varying much more than the results in Figure 19. In fact, the mean difference between the measurement data and the closed door ray-tracing is 3.3dB and the mean difference between the measurement data and the no doors ray-tracing is 5.9dB. The value for the maximum error in the case where there are no doors results in a much higher maximum error which is in the region of 12 – 13dB. We can conclude that it is extremely important to accurately model the salient features of the building and that the permittivity and conductivity values based on approximate values are sufficient to obtain good ray-tracing predictions of the signal power.

H. Measurement Set 2

For the next measurement set, the transmitter was placed in the corridor at position $T_x = (4.688, 4.516, 1.45)$ and the receiver is placed at the far end of the corridor in front of some fire doors along the line $x = 14.774$ with all doors closed as shown in Figure 21.

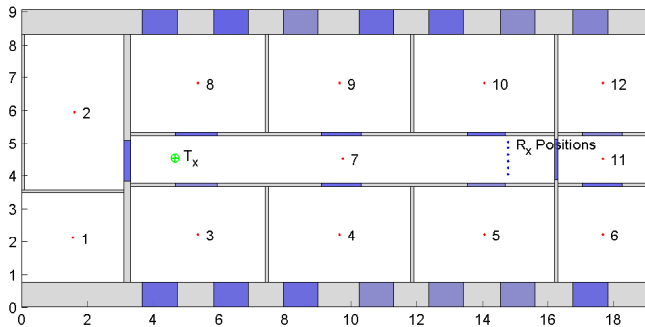


Fig. 21. A plan view of the printing house building at Trinity College Dublin with 6 receiver points at the end of a corridor with closed door used to measure the average signal power from the transmitter at the other end of the corridor.

To obtain an average value of the signal power at each point the signal power of the incoming rays is summed using a

random phase summation technique. This mean value which is a complicated product of Bessel functions, is explained in detail in Takahashi[15] where a computationally efficient numerical solution is given. We will refer to this type of summation of the signal power as the Random Walk Mean.

In this comparison we wish to compare the mean value of the ray-tracing over 6 locations at the end of the corridor and compare them with the mean measurement value at the same location. This is achieved by obtaining a random walk mean for a ray-trace with 1 reflection and 1 diffraction and checking that the result is more accurate with 3 reflections and 1 diffraction. This check also ensures that the transmit power of the antenna and relative permittivity values are approximately correct.

The Random Walk Mean deviates from the measurements by 2.69dB in the case of a ray-tracing computation with order 1 reflections. However, the deviation decreases to 1.4dB in the case of a ray-tracing computation with order 3 reflections. Changing the relative conductivity and permittivity values to account for possible leaded/non-lead glass in the doors did not yield better results. Also, very high order reflection terms did not greatly affect the results either.

VI. DISCUSSION

In this paper the propagation environment has been modeled as a set of convex spaces/boxes in three dimensions which captures the regular structure of a building and leads to a computationally efficient model. The algorithm avoids any loss of ray information by forming a path along which transmissions, reflections and diffractions are easily determined. This is in contrast to some ray-tracing algorithms where finding the reflection and transmission points results in the execution of a sorting algorithm which has to be executed for each path to obtain the correct order of the transmission and reflection points along that path.

The calculation of the Method of Images and diffraction edge images were shown in tests to account for about 9% of the overall computation time when determining the ray-paths for a single ray. This shows that this process is not dominating the overall ray-tracing computation time. In the case of a point-to-multipoint method they are only calculated once so they reduce to accounting for less than 0.09% of the overall computation time when calculating a ray-trace at 100 receiver points.

As in Yun[19] the algorithm is only applied to the existing vertices of edges in the building and no auxiliary ones are introduced. The computational saving achieved by Yun[18][19] of 25% to 38% over conventional visibility techniques is also directly applicable to our own method since the connectivity of the boundaries to other objects is known. However our method, as stated already, can and has been applied to point-to-multipoint problems with changing transmitter location. Our method is extendable to any type of convex space and it has been shown in this paper to be applicable to more complex building structures such as those with multiple storeys and lift shafts. It is important to note, by contrast with our paper, that in other papers the algorithms are essentially reducing the

average number of bounding box checks in the vicinity of an obstacle to locate a reflection or transmission points.

The separation of the diffraction edge list tree from the reflection tree avoids repetition and speeds up the algorithm as a result.

The simulations of the ray-tracing against real measurement results were found to be in reasonable agreement. The Random Walk Mean calculation for signal power at the receiver was found to generate results that agree better with measurements over those obtained by a simple summation of the incident fields.

Although the results are good we find that the only way to improve on the results is to have very accurate data on the locations of the walls, doors and windows in the building and to use precise values for the complex permittivities.

The point to multipoint algorithm is very efficient and makes it directly applicable to the wireless resource optimisation problem which is the subject of another paper by Kenny, O Nuallain[7].

Future work would require the extension of a polyhedron to 4 or 6 sided shapes (4 or 8 vertices). This would be automatically obtained from building information modelling system. This could result in the combination of the tetrahedron method with a generalised convex space method which would include objects such as domes and sickle/torus shaped corridors. As stated already, the overhead in defining the building connectivity is of little consequence so our method should be already directly applicable to computationally intensive optimisation techniques. Although the prototype buildings in this paper were described as a set of interconnected boxes, all formulae applied were implemented using full 3D vectorized classes to ensure that they are applicable to irregularly shaped convex spaces.

There are clearly improvements that can be made to our method to achieve even greater efficiency as discussed in Section II-B. The rooms can be placed in oct-tree buckets to arrive at the transmitter and receiver locations more quickly. The boundaries of a convex space can also be placed in quad-tree buckets that result in less traversals of bounding boxes when searching for reflection and transmission coefficients. This would result in a method similar to the quad-tree, oct-tree and tetrahedron methods where one uses the efficiencies of each algorithm to produce a coarse granularity for large convex spaces and a fine granularity for smaller objects with many edges (20+) such as domes.

VII. CONCLUSION

A transmitter location independent visibility algorithm has been presented based on a spatial decomposition of the building. The algorithm generates the connectivity information between convex spaces. The visibility information obtained by decomposing the building space into a set of convex spaces (in this prototype, a set of boxes) needs only to be calculated once and is then used to calculate all possible ray-paths with a predefined set of reflections and up to one diffraction. This information is used repeatedly in point to multipoint algorithms since it is transmitter location independent. This results

in immediate computational savings. The computation time for the connectivity algorithm was of the order of *2seconds* for 230 convex spaces but this does not contribute to the overall computation time since it is precomputed once only. Even if it took a few seconds to compute the connectivity it would still result in massive computational saving when applied to transmitter optimisation techniques since these methods require point-to-multipoint computations of hundreds or thousands of times to arrive at optimal transmitter locations.

ACKNOWLEDGMENT

We would like to acknowledge the invaluable help and advice of the late Dr. Gerry Butler in this project.

The ADAPT Centre for Digital Content Technology is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund.

REFERENCES

- [1] H. Azodi, U. Siart, and T.F. Eibert. A fast 3-d deterministic ray tracing coverage simulator including creeping rays based on geometry voxelization technique. *Antennas and Propagation, IEEE Transactions on*, 63(1):210–220, Jan 2015.
- [2] Constantine A. Balanis. *Advanced Engineering Electromagnetics*. New York: Wiley, 2012.
- [3] Jiri Bittner, Peter Wonka, and Michael Wimmer. Visibility preprocessing for urban scenes using line space subdivision. In *In Proceedings of Pacific Graphics (PG01)*, pages 276–284. IEEE, 2001.
- [4] Y. S. Feng, L. X. Guo, P. Wang, and Z. Y. Liu. Efficient ray-tracing model for propagation prediction for microcellular wireless communication systems. In *Antennas, Propagation EM Theory (ISAPE), 2012 10th International Symposium on*, pages 432–435, Oct 2012.
- [5] H. Inanoglu and E. Topuz. Hybrid ray-mode modelling of indoor propagation for wireless applications. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 3:1186–, Sept 1995.
- [6] G.L. James and Institution of Electrical Engineers. *Geometrical Theory of Diffraction for Electromagnetic Waves*. Electromagnetics and Radar Series. P. Peregrinus, 1986.
- [7] Eamonn Kenny and Eamonn O Nuallain. An indoor resource optimisation using ray-tracing techniques and signal-to-interference object function to obtain best capacity, 2014. Poster presented at IEEE CEFC, Annecy, France, 25th-28th May,.
- [8] G. Kouyoumjian and P.H. Pathak. A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface. In *Proc. IEEE*, volume 62, Nov 1974.
- [9] K. McGlinn, A. Gerdelan, K. Jones, and D. Lewis. Experiences developing a web-based visualisation and configuration interface for holistic energy management in buildings. In *International Conference in Applied Energy*, Jul 2013.
- [10] D.A. McNamara, C.W.I. Pistorius, and J.A.G. Malherbe. Introduction to the uniform geometrical theory of diffraction. 1990.
- [11] W.M. O'Brien, E.M. Kenny, and P.J. Cullen. An efficient implementation of a three-dimensional microcell propagation tool for indoor and outdoor urban environments. *Vehicular Technology, IEEE Transactions on*, 49(2):622–630, Mar 2000.
- [12] T. Rautiainen, R. Hoppe, and G. Wofle. Measurements and 3d ray tracing propagation predictions of channel characteristics in indoor environments. In *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, pages 1–5, Sept 2007.
- [13] N. Schettino, D.H.D. Carvalho, and F.J.S. Moreira. An efficient algorithm for radiowave coverage prediction in urban microcells. In *IEEE International Telecommunications Symposium (ITS2002)*, Natal, RN, Brazil, September 2002.
- [14] L. Stola, G.L. Urso, and P. Tenani. Indoor propagation: Experimental validation at 1.7 ghz of a utd-based approach. *Wireless Personal Communications*, 3(3):225–241, 1996.

- [15] S. Takahashi, Y. Yamada, and K. Ogura. Ber estimation with random phase summation applied to ray tracing. In *Vehicular Technology Conference, 1998. VTC 98. 48th IEEE*, volume 2, pages 855–859 vol.2, May 1998.
- [16] R.P. Torres, L. Valle, M. Domingo, S. Loredo, and M.C. Diez. Cindoor: an engineering tool for planning and design of wireless systems in enclosed spaces. *Antennas and Propagation Magazine, IEEE*, 41(4):11–22, Aug 1999.
- [17] Kyu-Young Whang, Ju-Won Song, Ji-Woong Chang, Ji-Yun Kim, Wan-Sup Cho, Chong-Mok Park, and Il-Yeol Song. Octree-r: An adaptive octree for efficient ray tracing. *IEEE Transactions on Visualization and Computer Graphics*, 1(4):343–349, December 1995.
- [18] Zhengqing Yun, Zhijun Zhang, and Magdy F. Iskander. A ray-tracing method based on the triangular grid approach and application to propagation prediction in urban environments. *IEEE Trans. Antennas and Propag*, 50(5):742–748, May 2002.
- [19] Z. Zhang, Z. Yun, and M. F. Iskander. 3d tetrahedron ray tracing algorithm. *Electronics Letters*, 37(6):334–335, Mar 2001.



Eamonn M. Kenny was born in Dublin, Ireland, in 1970. He received his B.A in Mathematics from Trinity College Dublin in 1992 and an M.Sc in Numerical Analysis in 1994. From 1995-2003 he implemented and developed many radio wave propagation techniques in macrocellular, microcellular and picocellular environments for the EU FP4 project STORMS and a number of other national projects. His PhD is in Electronic and Electrical Engineering from Trinity College Dublin in 2003 specialising in advanced radio wave propagation techniques for

mobile telecommunications. In 2003, he joined the School of Computer Science and Statistics, Trinity College Dublin as a research fellow and managed the research group of Dr. Brian Coghlan whilst researching and developing technologies for the Grid Computing group at CERN. In a ten year stay with that group he developed the first Playstation 3 worker node connected to the Grid Computing Middleware used by CERN for the Large Hadron Collider. He also generated the first OpenSUSE portability solution to connect the Jülich Grid cluster to CERN. In total he created workernode solutions on 6 new platforms. From 2007-2010 he was the portability coordinator for Europe in the EU FP7 EGEE II and III projects and the metrics lead for quality assurance from 2010 to 2013 in the FP7 EMI project. Since 2013, he is now a Research Fellow in the ADAPT Centre for Digital Content Technology working on machine learning and multimodal technologies. As part of this work he is specialising on optimisation techniques for many problem sets including building resource management.



Eamonn O Nuallain Dr. Eamonn O Nuallain received his B.E(Elect) degree from University College Cork, Ireland and his M.A. (de jure) and Ph.D. degrees from Trinity College Dublin, Ireland. He has worked for a number of blue-chip companies in various roles before joining the School of Computer Science and Statistics at Trinity College Dublin in 2000 where he lectures in Electrotechnology and Computational Mathematics. He has also lectured in Telecommunications. His research activity has centered around Radio Wave Propagation with a

particular emphasis on Computational Electromagnetics, Radio Tomography, Cognitive Radio and Ad-hoc Networks.