THE JOURNAL OF NAVIGATION (2007), **60**, 95–105. © The Royal Institute of Navigation
doi:10.1017/S0373463307004079 Printed in the United Kingdom

Code Multipath Modelling in the Urban Environment Using Large Virtual Reality City Models: Determining the Local Environment

J. Bradbury, M. Ziebart, P.A. Cross

(*University College London*)
(Email: bradbury@ge.ucl.ac.uk)

P. Boulton, A. Read

(*Spirent Communications*)

Positioning in the urban environment using GNSS is hampered by poor satellite availability due to signal obstruction created by both man-made and natural features of the urban environment. In addition, range measurement to satellites for positioning and for navigation is severely degraded by the multipath effect. The arrival and continuous enhancement of computerised geometric city models makes it possible to tackle these problems through modelling. In this paper description is given of a method for determining the local multipath environment, defined by the surfaces within a city model that will cause disruptive signal reflections to be presented to a receiver. An example simulation is performed, and graphical and numerical results are produced.

KEY WORDS

1. GNSS. 2. Multipath. 3. Urban positioning.

1. INTRODUCTION. Cities are challenging environments for positioning using GNSS. The architectural elements of the urban environment pose a problem to a GNSS receiver by acting as reflectors for satellite signals, resulting in pseudo-range measurement errors due to multipath reflection. Reflected signals can appear in addition to direct signals, or without the presence of the direct signal. For numerous applications there is a serious need to be able to predict the signals that will be present at a receiver located in an urban environment at a defined epoch.

Present day (2006) computerised city models include objects such as buildings, trees, street furniture and road/terrain surfaces. Research is being undertaken at University College London (UCL) in which such city models are being integrated with GNSS observables in order to predict both satellite availability and the level of signal degradation due to the multipath effect on code measurements. This research is

https://doi.org/10.1017/S0373463307004079 Published online by Cambridge University Press

Table 1. Fenchurch St area model specification (exported to VRML).

Fenchurch Street area, London (Model dataset provided by Zmapping[4])	
Approximate area covered (sq km)	0·25
Triangular mesh objects	4953
Triangular mesh surfaces	78780
Total number of index values	236340
Total number of coordinate values	77168
Accompanying normal values	77168
VRML file size (geometry only)	6·45MB

producing software tools for the processing of high resolution geometric city model data, and in particular, processing city model data for use by low resource devices.

A fundamental problem is the rapid and efficient assessment of urban environment data. The ratio of the number of surfaces that cause multipath effects compared to the total number of surfaces in an entire city model is extremely low, and a model may contain many hundreds of thousands or even millions of surfaces. For the work presented in this paper, a three-stage method of rapidly and efficiently determining the candidate reflection surfaces is presented. Stage one organises model data and uses knowledge of receiver specification to take an entire model and determine the surfaces that are within an appropriate range of the receiver such that the resulting multipath signal will disrupt the code correlation process. Stage two determines which of these surfaces are suitably orientated based upon a specular reflection model, and considering a single reflection only. Finally, rigorous testing of the remaining surfaces is employed to cull the list of reflectors that cannot be seen directly by the receiver or a satellite. The final result is a list of surfaces that can be considered as valid reflectors. Also determined is the point of intersection on each candidate surface, about which a specular signal reflection occurs. A single multipath simulation, based upon an arbitrarily chosen position and epoch, is performed on a single city model to illustrate this work.

2. VRML CITY MODEL DATASETS.

Computerised city models contain coordinate geometry information to describe the locations, dimensions and relationships of physical objects, and are available in many file formats. The Virtual Reality Markup Language (VRML) was chosen as the input file format for this work. Most CAD formats used to hold model data can easily be exported into the VRML format. Throughout this paper, reference is made to objects, where an object is defined as a collection of surfaces, and usually represents a single structure or a component of a structure.

The physical size of the urban environment described by a model, and the resolution of data held within the final model file, determine how large a dataset will be in terms of file size and number of coordinate points. For the work described, and the results given, a model of the Fenchurch Street area of London in the United Kingdom has been used. The model originates in an AutoCAD .dwg format, and has been exported into VRML 2.0 using the Rhinoceros NURBS modelling for Windows tool [2]. Specifics of the model used are given in Table 1. All simulations are performed on a high specification PC running Windows XP.

https://doi.org/10.1017/S0373463307004079 Published online by Cambridge University Press

Although a model comprised of triangular mesh surfaces is used, the VRML format is capable of describing objects by using polygons of higher order. The software produced in this research, and the methods described in this paper will work with any polygon order, provided all objects are comprised of planar facets. There is no provision for VRML primitive shapes or linesets in this work. It should be noted, that while a dataset of the size described by Table 1 is used for the purposes of this work, it is possible that a dataset many times larger may be processed using the method described in this paper.

This work assumes that appropriate means of calculating satellite position to the required accuracy have been employed. Simulations performed for this paper use satellite positions determined using the broadcast ephemeris. In addition, satellite positions and model geometry must be expressed in the same Cartesian reference frame.

3. STAGE 1 – DATA ORGANISATION AND USE OF CORRELATOR SPACING.

It is envisaged that the algorithms developed in this study could be used very effectively on mobile devices such as personal digital assistants and mobile phones. For this reason, the method of defining the local multipath environment given seeks to minimise both memory usage, and processing overhead. Also required is the rapid assimilation and processing of city model data.

A proprietary data format to hold geometry-only data has been defined at UCL. This format holds ASCII data extracted from the original VRML input file by way of a VRML reader (also written at UCL). Index values (describing how to join coordinate points to create a polygon), coordinate points, and normal vectors (if present in original VRML file) are grouped by VRML object and written to a '.geo' file. Whilst object data is extracted from the original VRML file, a check is made for the extents of each object. Object extent information is also stored in the '.geo' file.

Normal values per coordinate are generally provided in a VRML file. However it is possible to be presented with index values and corresponding coordinate points only. If this is the case, then a surface normal is calculated by taking three vertices to form two in-surface vectors and performing a cross-product operation to determine the surface normal. It is the case that when using this method, or when relying on given surface normals, there is no guarantee that the vector direction points away from the inside of an object. It is imperative that surface normals point to the outside of an object in order that the orientation of the surface relative to the satellite and receiver coordinates is correctly established. An algorithm has been written to test the direction of the surface normal relative to other coordinate points in the same object. The result of this algorithm is a flag stating whether this normal is to be considered in the given direction, the reverse to the given direction, or a null result. The null result indicates that the test is inconclusive, and this case is dealt with later. A flag accompanies each given (or derived) normal, when written to the .geo file.

At this stage, only a pointer to the file location of each urban object is stored in Random Access Memory (RAM). More specifically, the pointer is referenced to the file locations of the information regarding the extents of each object. It is now possible to rapidly determine the objects in the urban environment that are within an appropriate range such that a multipath reflection would cause distortion of the code

https://doi.org/10.1017/S0373463307004079 Published online by Cambridge University Press

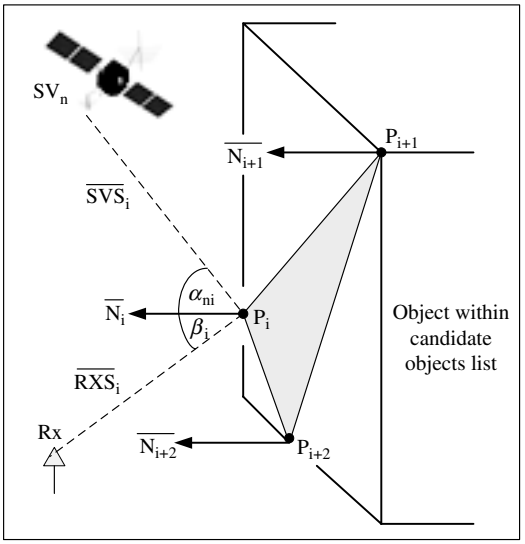


Figure 1. Rejection by orientation.

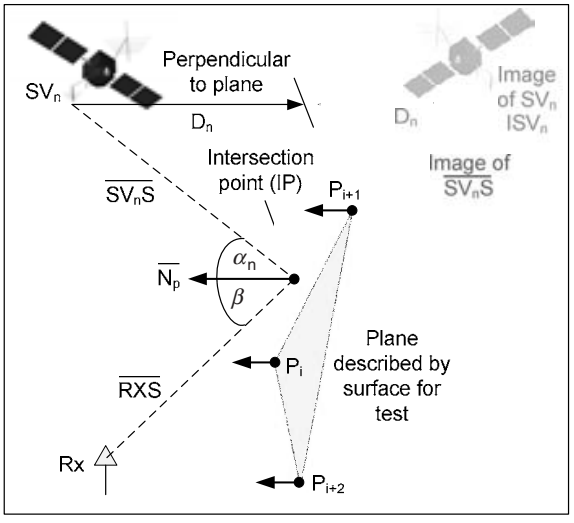


Figure 2. Intersection calculation.

correlation process in a receiver. For correlation peak distortion to occur, a reflected signal must exhibit a path length according to the following, (See [1], p258):

$$P \leqslant 1 + \frac{\tau}{2} \tag{1}$$

where:

P = Difference between direct and reflected signal path lengths (metres)
τ = Correlator Spacing, generally 1 for Standard, 0·1 for Narrow (chips)

https://doi.org/10.1017/S0373463307004079 Published online by Cambridge University Press

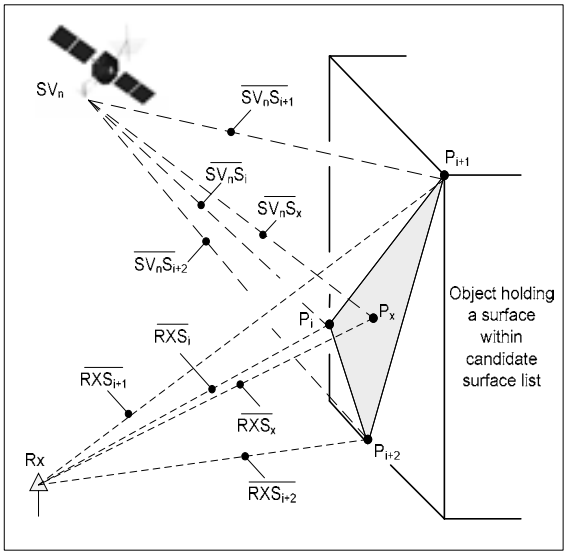


Figure 3. Surface line of sight testing.



Figure 4. Original VRML Fenchurch St model with simulated satellite availability.

The object extents held in the geometry-only file are inspected by jumping to the file locations, and each building number is subsequently stored in memory or disregarded depending on the distance between the receiver and the furthest object point. The original VRML geometry data has now been organised such that rapid assessment of the geometry can be made. Further to this, the specification of the receiver hardware that will have a bearing on the code correlation process (correlation spacing) has also been accounted for. A local environment now exists, defined in terms of the list of objects that could contribute to code multipath. Each of these objects is likely to

https://doi.org/10.1017/S0373463307004079 Published online by Cambridge University Press

consist of many thousands of surfaces, each of which is now a member of the current candidate surface list. The following stages describe the reduction of this list.

4. STAGE 2 – SURFACE ORIENTATION. The orientation of every surface within the candidate surface list, relative to the receiver and to every satellite above the horizon, must be tested to determine whether or not it can act as a reflector. At this point, if an antenna that rejects signals below a specified elevation mask is to be considered, surfaces with all coordinate points existing below the mask angle can be discarded. Figure 1 shows the general scenario for testing a surface within a candidate object, and introduces the notation used.

For each satellite (n) above the horizon at the simulation epoch, the following are determined for each surface point (i):

Rx - Receiver position; SV_n - Satellites above horizon; $\overline{SV_nS_i}$ - Satellite (n) to surface point (i); $\overline{RXS_i}$ - Receiver to surface point (i); $\overline{N_i}$ - Surface normal at surface point (i); β_i - Angle between $\overline{RXS_i}$ and $\overline{N_i}$; α_{ni} Angle between $\overline{SV_nS_i}$ and $\overline{N_i}$.

$\overline{N_i} \cdot \overline{RXS_i}$. yields β_i. A value greater than or equal to 90° means that the receiver is behind, or positioned on the plane described by the surface, and therefore the surface is immediately rejected. The following is then performed for each valid satellite. Angles α_{ni} and β_i are collected for each point describing the surface. These are then assessed to yield α_{max}, α_{min}, β_{max} and β_{min}. Specular reflection cannot occur if $\alpha_{max} < \beta_{min}$ or if $\alpha_{min} > \beta_{max}$. For each surface a list of satellites that could result in a multipath reflection is assembled.

Many surfaces can be rejected on the basis of the above, as demonstrated in the results section. There now exists an updated candidate list, expressed in terms of surfaces instead of objects. This list has been efficiently determined, as no rigorous (and therefore more numerically intensive) geometrical testing has yet been performed.

Three of the surface coordinates are used to determine an equation for the plane which is described by the surface. Figure 2 illustrates the process whereby an intersection point (IP) on that plane is sought. The IP is described as the point at which $\alpha_{ni} = \beta_i$ on the plane described by the surface under test. This is done by first calculating the Cartesian position of the mirror image of SV_n about the plane, denoted by ISV_n. A vector between ISV_n and Rx is now established. It is then possible to determine the point at which this vector intersects the plane (IP), by using parametric equations and knowledge of the perpendicular distance between SV_n and the plane.

Finally, a point in polygon test, such as that described in [3], is employed to reveal whether IP exists inside (valid surface) or outside (invalid surface) of the area bounded by the vertices of the surface under test. In order that a point in polygon test can be performed, it is necessary to transform all polygon points and IP into a 2D space. Once this has been completed for all valid satellites, the next candidate surface is selected. Processing of individual surfaces as described by Figure 2 is expensive in terms of resources and time. It is avoided as far as possible by the preceding steps. Once again the candidate surface list must be updated.

5. STAGE 3 – SATELLITE AND RECEIVER LINE OF SIGHT TO SURFACES. Subsequent to the processes described in stages 1 and 2 being

https://doi.org/10.1017/S0373463307004079 Published online by Cambridge University Press

performed, there exists a list of appropriately orientated surfaces within a radius of the receiver, such that a resulting multipath reflection must be given consideration. It is, however, not known whether a reflection from one of these surfaces could be seen by the receiver, or indeed, whether the surface itself is in direct line of sight of the satellite signal that would cause such a reflection. Stage 3 deals with this, the most computationally intense part of determining the local multipath environment.

Every point of every polygon held within the candidate surface list must be tested for line of sight to the receiver, and then to each satellite. In addition, to mitigate the risk of failure due to the fact that it is possible that the line of sight to all points is obstructed, but that the receiver can still see a portion of the surface, a mid point (P_x) is also determined and tested. This situation is unlikely to occur, but must be accounted for. More artificial points could be established and tested if it is found that failures occur. With reference to Figure 3, vectors $\overline{SV_nS_i}$, $\overline{SV_nS_x}$, $\overline{RXS_i}$, $\overline{RXS_x}$, are to be tested for intersection with any other surface in the entire city model.

A set of algorithms developed at UCL during an earlier stage of this project, were used to determine satellite availability in the urban environment. These algorithms are reused at this stage for determining the local multipath environment. A collection of vectors, (e.g those shown in Figure 3) are taken and assessed against all of the objects in the whole model at a coarse level. This is done by again using the extents of the object coordinates, and rapidly concludes if intersection between any of the vectors and objects is likely to occur. Detailed testing of individual surfaces within the identified objects for intersection with each vector is then feasible. As with all work described for defining the local multipath environment, the determination of line of sight availability is verifiable by way of graphical and numerical output. Throughout the above line of sight testing, references of surfaces that are obscured, and therefore not able to provide a multipath reflection, are recorded. The candidate surface list is reduced for the final time leaving the list of surfaces from which a disruptive multipath signal could be observed by the receiver.

To summarise, the steps described above act to significantly reduce the number of elements of the city model that need to be analysed for the purposes of computing the degree of multipath contamination of the direct signal.

6. GRAPHICAL DEMONSTRATION.

For the demonstration in this paper, a position within the model described by Table 1 has been arbitrarily chosen. An epoch has also been selected, in order that satellite positions can be generated, using the broadcast ephemeris. An example of the city model chosen for this demonstration is provided in Figure 4. The receiver position is indicated at the top left, where the obscured (red) satellite vectors and green (line of sight) vectors meet. These vectors have been generated following assessment of satellite availability, using UCL availability simulation software written as part of this research. Line of sight satellite to receiver availability vectors are of no relevance to the subject of this paper, and are included on this and the following figures as a means of visually identifying the receiver position only.

According to Equation 1, for C/A code measurements, using a receiver with a 1.0 chip correlator spacing, reflections with an excess path length of up to 439·57 m above the direct satellite to receiver vector distance, will degrade the correlator's operation. For P-code measurements this is reduced by a factor of ten. As a starting point all

https://doi.org/10.1017/S0373463307004079 Published online by Cambridge University Press

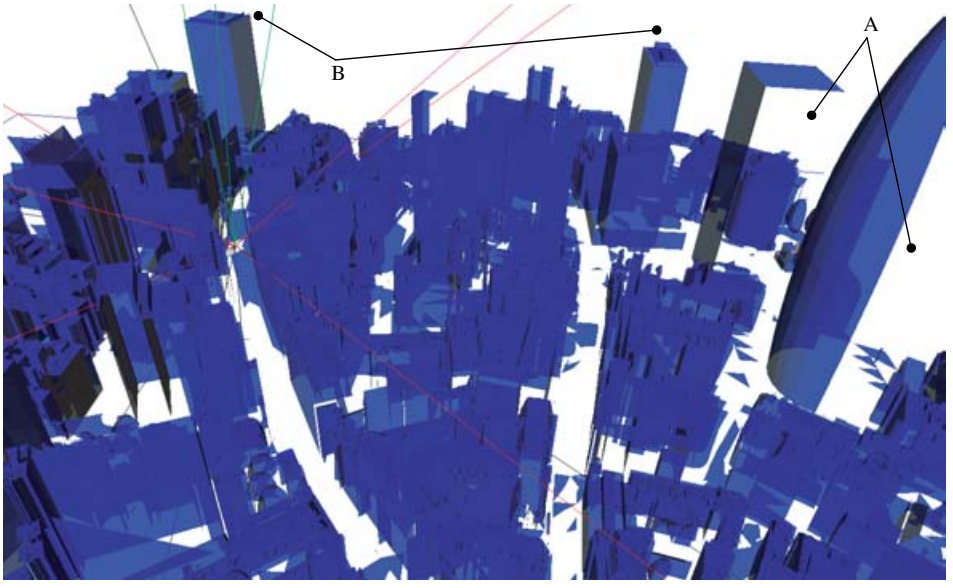


Figure 5. Local multipath environment (Stage 1).

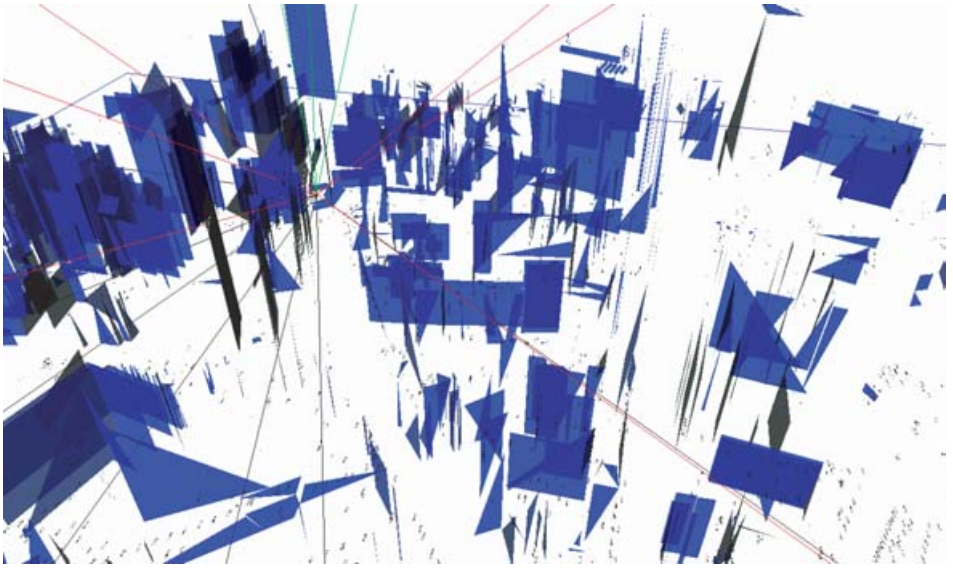


Figure 6. Stage 2 initial candidate surface list.

objects exterior to a circle of 439·57 m are rejected, although as the model used in this example covers a relatively small area, and we are modelling C/A code multipath, all objects are included.

Stage 2 begins with the rejection of surfaces orientated such that the receiver must lay on or behind the plane containing the surface under test. As illustrated by

https://doi.org/10.1017/S0373463307004079 Published online by Cambridge University Press

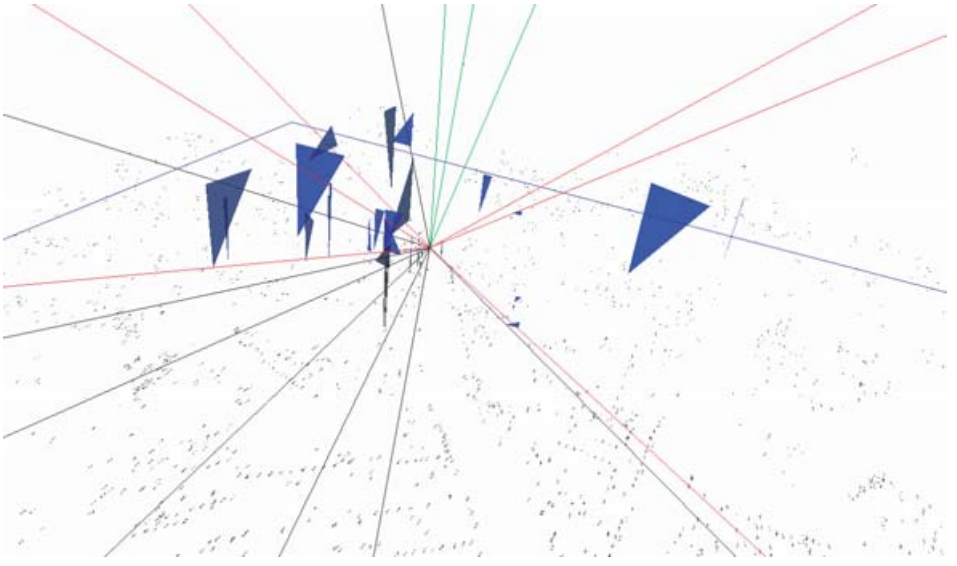


Figure 7. Stage 2 final candidate surface list.

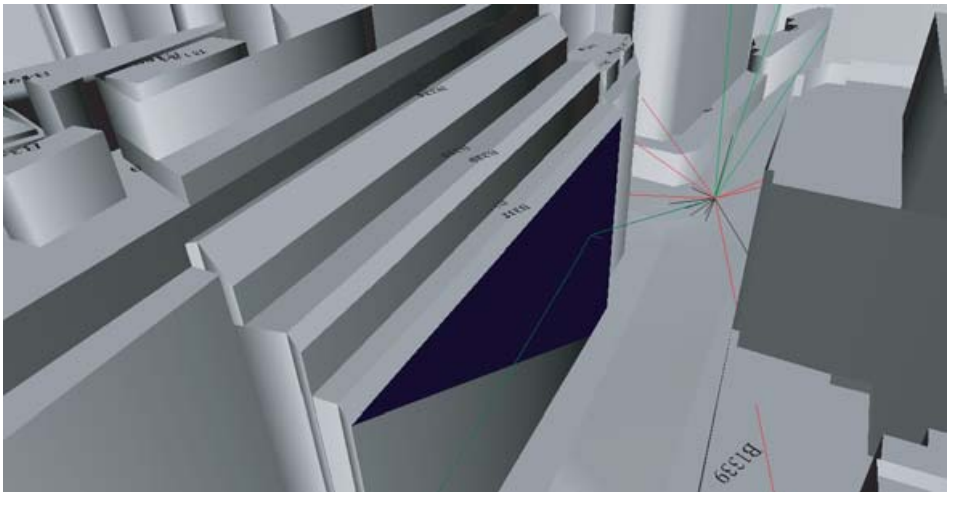


Figure 8. Local multipath environment.

Figure 5, marker A demonstrates the removal of such polygons, leaving surfaces that face the receiver only.

It is the case that surfaces highlighted by B in Figure 5 should ideally be removed at this stage. The appearance of these surfaces here is due to a null result from an earlier algorithm to determine the correct direction of the surface normal. It is not always possible to arrive at a conclusion as to whether a surface normal points into an object or to the outside world (as required) from the original VRML input file. Where there is any uncertainty, it is important to include the surface at this stage. Later processing

https://doi.org/10.1017/S0373463307004079 Published online by Cambridge University Press

Table 2. Example simulation results for Standard and Narrow code correlators.
(* Contains greater than true number of surfaces due to normal direction uncertainty.)

Process	Correlator Type	
	Standard	Narrow
Total model objects	4953	4953
Total model surfaces	78780	78780
C/A code correlator window for multipath (Stage 1)	439·5 m	307·65 m
Objects within correlator window radius (Stage 1)	4953	3837
Surfaces within correlator window radius (Stage 1)	78780	62334
Surfaces where receiver is in front of surface (Stage 2)	34417*	26540*
Surfaces able to cause reflection, assessed with satellite geometry (Stage 2)	3797	3327
Specular reflection with intersection point inside polygon (Stage 2)	30	30
Candidate surfaces visible by receiver and one or more satellites (Stage 3)	1	1

will reject these surfaces, but an increased processing burden, due to their presence, is incurred.

The ability of a surface to cause a specular reflection, with respect to the satellite positions, is assessed. Maximum and minimum values for the incident and reflection angles for each surface point to each satellite are assessed as described in stage 2. Rejected surfaces are removed, and the results are illustrated in Figure 6. The small visible dots in Figures 6 and 7 are object reference numbers included in the graphical display.

Stage 2 concludes with the determination of the plane described by each surface in the current candidate list, and the production of intersection points (IP) as described earlier. A point in polygon test delivers the final candidate list, represented in Figure 7.

Finally, line of sight testing for visibility of the surface by the receiver, and visibility of the surface by one or more satellites, yields the local environment result. As shown in Figure 8, for this demonstration, only one surface will provide a multipath reflection.

7. RESULTS AND DISCUSSION.

Table 2 presents the results of a simulation performed as described for the purpose of illustration throughout this paper. For the scenario and model data chosen during this illustration, no reduction in the list of candidate surfaces has been realised when employing a Standard code correlator (1.0 chip spacing). Simulation to determine the local multipath environment when using a Narrow code correlator (0.1 chip spacing), reveals an immediate reduction of the surfaces to be tested. Consideration of fewer objects leads to noticeably better simulation time accompanied by lower RAM usage.

Before processing city model data at a surface level, it is imperative that data held in RAM is organised effectively. It is also of critical importance that the outward side of each surface is defined, or at least it is acknowledged that an ambiguity exists in order that surfaces are not wrongly discarded from the candidate surface list. Any ambiguity is resolved at a later stage when the line of sight tests are performed.

https://doi.org/10.1017/S0373463307004079 Published online by Cambridge University Press

However, line of sight testing is costly in terms of time and processing resources. For this reason, the preceding stages exist, and their purpose is to reduce the candidate list by as much as possible before line of sight testing begins.

8. CONCLUSION. A method of defining the local multipath environment, for code multipath, has been described. An example has been used to demonstrate the concepts developed during this work, and both graphical and numerical results of this example have been provided to illustrate typical operation. A second set of results has been provided originating from the same simulation but using a different receiver specification. The objective of a means of rapid assessment of the local environment has been achieved. Consequences of incorrectly dismissing potential reflectors are clear, and for this reason the possible failure mode whereby the direction of a surface normal is incorrectly assessed, has been explained and handled. A feature of the described work is that providing acknowledgment of any ambiguity in the direction of a surface normal is made at the outset, the surface will not be mistakenly rejected. Software developed at UCL for the modelling of code multipath is currently in use including the described processes. The software provides a means of visual and numerical verification of the results obtained per simulation.

9. FUTURE WORK. Inclusion of a more sophisticated model to account for the diffuse component of reflections is likely to be added. It is envisaged that the effect of such an enhancement on the processing resources and simulation time will create a trade-off that will be of more importance when considering some applications than others. Currently little in the way of assessment of the resources used and time taken per simulation has been made. This is due, in part, to the fact that the software containing the described processes is under development and has yet to be optimised. Account must be made for the contribution to code multipath error from reflections that occur subsequent to the first reflection. The method for modelling multiple reflections will be defined following analysis of the previously mentioned simulation times and resource usage for single reflection simulations.

REFERENCES

1. Kaplan, E. D. (1996). *Understanding GPS Principles and Applications*.
2. Robert McNeel & Associates. Rhinoceras NURBS modeling for Windows. (2.0). 28-12-2001. Ref Type: Computer Program.
3. Taylor, G. (1994). Point In Polygon Test. *Survey Review*.
4. ZMapping Ltd. Fenchurch St area, London. (2005). Ref Type: Data File.

https://doi.org/10.1017/S0373463307004079 Published online by Cambridge University Press