# ULTRASOUND RAY TRACING IN ARBITRARY COMPLEX GEOMETRIES

A. Turnbull
Center for NDE
Iowa State University
Ames, IA 50011

M. Garton
Center for NDE
Iowa State University
Ames, IA 50011

## INTRODUCTION

Ultrasound ray tracing is the process of recursively intersecting vectors with a geometry and applying Snell's Law to the rays at the intersections. The two key elements necessary for solving arbitrary UT ray tracing problems are a topological data structure to represent the geometry and a computational engine to calculate intersections of vectors with the geometry.

## SOLID MODELING TOPOLOGY

To solve arbitrary ultrasound ray tracing problems a topological data structure is required that can represent any geometry in a precise, non-ambiguous way. This data structure must lend itself to the accurate evaluation of any surface point and its derivatives. There are five basic types of solid model topologies [1] primitive instancing, sweeps, boundary-representations (b-reps), spatial-partitioning, and constructive solid geometry.

Primitive instancing is the use of simple parameterized shapes. Each primitive is handled computationally as a new case. Sweep topologies involve rotating or extruding two dimensional curves into three dimensions. The primitive instancing and sweep topologies would be well suited to a particular class of problems, involving pipes or nozzles for instance, but are not flexible enough to model arbitrary, complex geometries.

Spatial-partitioning represents a solid by creating a collection of smaller, non-intersecting entities (bricks for example). Spatial-partitioning topologies rarely contain the information required to calculate precise surface normals required for Snell's law. One type of spatial-partitioning topology, finite elements, would adequately represent a large class of geometry. The shape and corresponding accuracy are limited by the order of the finite element. The additional task of creating a mesh is also a drawback of finite elements.

*Review of Progress in Quantitative Nondestructive Evaluation, Vol. 14*
Edited by D.O. Thompson and D.E. Chimenti, Plenum Press, New York, 1995

Constructive solid geometry (CSG), is the process of applying regularized Boolean set operations to simple primitives. Hybrid CSG includes sweeps, half spaces, and other topologies as primitives. This method allows for intuitive creation of geometry using operations like cut, drill, and glue. The major drawback of the method is that the tree of Boolean operations on primitives is stored in "unevaluated" form. To use the geometry, one must mathematically process the tree, performing the Boolean operations. Each CAD package supports a (different) particular set of primitives, parameters, and operations. An intersection engine for ray tracing could be written to handle hybrid CSG entities from a particular CAD package, but would have to be rewritten for geometry originating elsewhere.

Boundary representation topologies come in two common forms, precise and faceted. Faceted b-reps approximate a geometry with a collection of planar polygons. This representation has a direct trade off between storage size and accuracy. Faceted b-reps are not well suited for UT ray tracing unless one has access to another, more precise definition of the geometry, which can be converted to facets, while controlling the accuracy of the polygon approximations.

The other type of boundary representation, precise b-rep, includes precise mathematical definitions of an object's surfaces, boundaries, and the relationships between them. Precise b-reps are used alone or with CSG by most CAD vendors [2]. The b-rep is an evaluated model with a hierarchical structure. An object has pointers to shells, pointing to faces, pointing to boundary loops, pointing to loop segments, pointing to vertexes. It can be written out in a standard format such as IGES and transferred between CAD packages with little or no ambiguity [3]. The precise b-rep structure utilizing parametric surfaces (particularly non-uniform-rational-b-spline surfaces) is capable of accurately representing arbitrary geometries from many different sources. For these reasons, the precise b-rep was chosen as the primary topological entity for the UT-Sim software package.

## SURFACE RAY INTERSECTION

With the method of surface definition determined, the next step is to decide upon the proper computational engine for finding the intersections of rays with the surface.

As was noted by Bouville [4], it is absolutely necessary to use some form of data structure to reduce the number of surfaces tested for intersections. The ones most commonly used are hierarchical bounding volumes and spatial subdivision [4]. The use of hierarchical bounding volumes consists of creating some form of bounding volume around objects and surfaces whose intersections are less computationally expensive to determine than the actual surface. With this method the majority of surfaces can quickly be eliminated from consideration. The second method, spatial subdivision, divides the three-dimensional space within which the surfaces exist into a number of smaller regions. With this method the ray traverses from one region to another, only checking the surfaces that exists within that region. Through experience it has generally been the case that there is a small number of fairly large surfaces, which lends itself to the use of bounding volumes, a far simpler structure. The bounding volume chosen was a parallelepiped constructed with the six sides formed parallel to the three principal planes. The method of testing is similar to that used by Glassner [5]. The actual hierarchy is a set of bounding boxes, both around each object and around each individual surface, to quickly eliminate as many surfaces as possible from the next step, surface intersection.

Once the surfaces whose bounding boxes were hit have been determined, it is next necessary to determine the form of the calculation of actual intersection between a ray and the given surface. The two primary methods for surface intersection are subdivision and some form of numerical method. The subdivision technique either divides the surface into a predetermined number of planar facets when the surface is read in, or recursively subdivides the surface into smaller and smaller patches until the necessary precision has been achieved. The first method tends to be somewhat inaccurate as it tends to lose a significant amount of the precision available from the surface definition, the second method is extremely time consuming. It is for these reasons that the numerical method has been chosen, which preserves all of the precision of the surface definition, while also being relatively computationally efficient [4].

The first step in all common numerical methods is to convert the surface into a usable form, usually bivariate parametric polynomial equations for the three dependent variable x, y, and z. All common parametric surfaces can be directly converted into this form. The equations are of the form:

$$x = X(u,v); \quad y = Y(u,v); \quad z = Z(u,v). \tag{1}$$

These equations are then used with the definition of the ray to form a solvable system of equations. There are three methods for combining the surface definition with the ray definition two plane method, point-direction method, and the parametric to analytic method.

The most common method of combining the equations for the surface with the equation for the ray is by defining the ray as the intersection between two planes, intersecting each plane with the surface, and determining the intersection of the two surface curves [4]. The equations for the planes are expressed as:

$$\begin{aligned} A_1x + B_1y + C_1z + D_1 &= 0, \\ A_2x + B_2y + C_2z + D_2 &= 0. \end{aligned} \tag{2}$$

Combining Equation (1) with Equation (2) gives us:

$$\begin{aligned} A_1X(u,v) + B_1Y(u,v) + C_1Z(u,v) + D_1 &= 0, \\ A_2X(u,v) + B_2Y(u,v) + C_2Z(u,v) + D_2 &= 0. \end{aligned} \tag{3}$$

A second method of forming the system of equations is by using the point-direction definition of the ray and forming three equations, one for each of x, y, and z. [6] The definition of the ray is given by:

$$p = p_o + td. \tag{4}$$

Combining Equation (1) with Equation (4) gives:

$$\begin{aligned} p_{ox} + td_x - X(u,v) &= 0, \\ p_{oy} + td_y - Y(u,v) &= 0, \\ p_{oz} + td_z - Z(u,v) &= 0. \end{aligned} \tag{5}$$

As this definition involves three equations and three unknowns, it is used far less than the first method.

The final method is to construct an analytic definition of the surface from the parametric definition. The analytic definition, similar to that for common objects such as spheres and cylinders, is simply a function of x, y, and z set equal to zero:

$$f(x,y,z) = 0. \qquad (6)$$

The components of the point-direction form of the line, Equation (4), can then be inserted into Equation (6) to from a single equation which can be solved for t:

$$f\left(p_{ox} + td_x, p_{oy} + td_y, p_{oz} + td_z\right) = 0. \qquad (7)$$

Numerous difficulties arise with this method. The first is the amount of storage necessary to store the definition of the analytic surface, whose polynomial degree is twice the product of the two degrees of the original surface definition. The other primary concern is the accuracy of the parametric to analytic conversion. The conversion process leads to extremely large floating point errors. This method has been implemented by Manocha [7].

Of the three methods the first method is the most appropriate for use by UT-Sim, considering computational time, and storage availability.

With the system of equations defined, the final step is to determine a method to solve it. Finding the roots of systems of polynomial equations is an extensive topic with significant research devoted to it, as it is useful in every major branch of science and engineering. The methods of solution can be generalized into either Newtonian or elimination methods. Considering the number of texts devoted to the subject, this paper will not go into any depth in the specifics of the actual solution. Nearly any text on numerical methods, such as [8], contains an adequate description of the necessary methods.

The Newtonian methods are methods based on utilizing the gradient of the surface to improve an initial guess towards the correct solution. The gradient of the surface is determined either analytically or by some form of approximation. This is the most common method, and the one used by UT-Sim. The main drawbacks include its tendency to not find solutions when given an initial guess too far from the actual solution, and its inability to find all possible solutions, considering that there is usually more than one solution to a system of equations.

The elimination methods are methods that reduce the system of equations to a single equation in a single variable. From the solution of the single equation the solution of the system is then recovered. There are also a number of methods included in [8] for the solution of single equations. The elimination methods include resultants, grobner bases, and characteristic sets. The method of resultants has been used successfully by Mooshabad [9], and was originally implemented by Kajiya [10].

UT-Sim uses a bounding volume method to eliminate as many surfaces as possible and then intersects two planes defining a ray with the remaining surfaces to create a system of equations which is solved using Newton's method to determine the intersections with the surfaces.

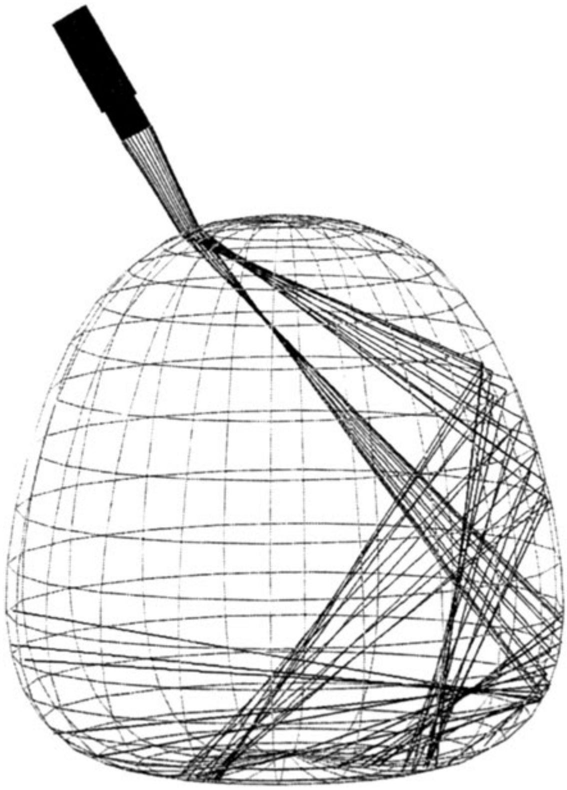

Figure 1. Inspection of an "apple" shape with a spherically focused probe, two levels of mode conversion shown

## REPRESENTATION OF DATA

After the intersections are calculated and recursively mapped through Snell's law, the software has filled an array with intersection points. These are of little use to the user without some data visualization.

UT-Sim draws a wire frame or transparent solid model geometry on the screen with the transducer and rays in three dimensional space (See Figure 1). Compressive waves (rays) are drawn in pink, shear waves in yellow. The user can control by time gate or level of reflection which rays appear on the screen.

## COMBINING RAY TRACING WITH OTHER ULTRASOUND MODELING METHODS

Ray tracing can provide valuable information about the illumination of ultrasound when the specular approximation is valid. It does not directly provide information on diffraction, attenuation, frequency dependencies, and near-field effects. Other modeling techniques are needed to compliment the ray tracing and provide a better picture of the sound field.

The Gauss Hermite beam model of Thompson and Gray [11] has been incorporated into UT-Sim. The Gauss Hermite model can calculate the effects of a smoothly curved interface on a beam. It takes as input the transducer properties, the material properties, and the geometric properties of the interface. To use the beam model, a user selects the beam model mode from a menu and positions the virtual probe relative to the solid model (numerically of with the mouse). The rest is automatic. Internally, the software is using the location of the probe to fire a central ray from the probe and calculate its intersections with the geometry. At the closest point of intersection, derivatives are calculated in the directions of the two parameters of the surface. The derivatives are used to calculate the normal and principle curvatures. These geometric parameters are then combined with the transducer and material parameters as input parameters for the Gauss Hermite beam model. The location(s) of interest where the beam is to be evaluated must also be passed to the routine. If the scattering intensities are to be viewed on a planar cross-section of the geometry, a mesh must be formed and the beam evaluated at each vertex. Forming the mesh involves the intersection of more rays to feel out the edges of the geometry.

## CONCLUSION

Methods for ray tracing arbitrary complex geometries have been presented. The topology chosen to represent the geometry is accurate, non ambiguous, and readily available from most CAD packages. These methods are being tested in a software package called UT-Sim.

## ACKNOWLEDGEMENT

This work was sponsored by NIST under cooperative agreement #70NANB9H0916 and was performed at the Center for NDE, Iowa State University

## REFERENCES

1. Foley, van Dam, Feiner, and Hughes,*Computer Graphics: Principles and Practice*, *Second Edition*, (Addison Wesley Publishing Company Inc.,1992) p. 558.
2. D. Mattei, "New Version of IGES Supports B-Rep Solids", *Mechanical Engineering*, January 1993 p. 51.
3. *The Initial Graphics Exchange Specification (IGES) Version 5.1*, (International TechneGroup Incorporated and the IGES/PDES Organization, 1991) p. 575.
4. C. Bouville and K. Boutouch, "Developments in Ray-Tracing", *Advances in Computer Graphics IV*, eds. W.T. Hewitt, M. Grave and M. Roch, (Springer Verlag, 1986) p 194.
5. A. S. Glassner, *Graphics Gems,* (Academic Press, Boston, 1990).
6. M. E. Mortenson, *Geometric Modeling*, (John Wiley & Sons, New York, 1985)
7. D. Manocha, and J. F. Canny, "A New Approach for Surface Intersection" in *International Journal of Computational Geometry & Applications,* Vol. 1, No. 4, 1994, 491-516, (World Scientific Publishing Company).
8. Press, Teukolsky, Vetterling, and Flannery, *Numerical Recipes in C, Second Edition,* (Cambridge University Press, New York, 1992).
9. I. Yalda-Mooshabad, T.A. Gray, and M. Attaar, "Ultrasonic 3-D Ray Tracing" in *Review of Progress in QNDE*, Vol. 11B, eds. D.O. Thompson and D.E. Chimenti (Plenum, New York, 1991), p 2179.
10. J. T. Kajiya, "Ray Tracing Parametric Patches" in *Computer Graphics,* Volume 16, Number 3, July 1982, 245-254.
11. R. B. Thompson, T.A. Gray, J.H. Rose, V.G. Kogan, and E.F. Lopes, "The radiation of elliptical and bicylindrically focused piston transducers", *Journal of the Acoustical Society of America,* Volume 82, number 5, November 1987 (Acoustical Society of America, 1987) pp. 1818-1828.