ARL-SR-0347 ● Dec 2015

US Army Research Laboratory

An Investigation into Conversion from Non-Uniform Rational B-Spline Boundary Representation Geometry to Constructive Solid Geometry

by Clifford W Yapp

NOTICES

Disclaimers

ARL-SR-0347 ● DEC 2015

US Army Research Laboratory

An Investigation into Conversion from Non-Uniform Rational B-Spline Boundary Representation Geometry to Constructive Solid Geometry

by Clifford W Yapp
Survivability/Lethality Analysis Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
December 2015

2. REPORT TYPE
Final

3. DATES COVERED (From - To)
1 October 2014–30 September 2015

4. TITLE AND SUBTITLE
An Investigation into Conversion from Non-Uniform Rational B-Spline Boundary Representation Geometry to Constructive Solid Geometry

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Clifford W Yapp

5d. PROJECT NUMBER
MRI

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
US Army Research Laboratory
ATTN: RDRL-SLB-S
Aberdeen Proving Ground, MD 5068

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-SR-0347

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Modern Ballistic Vulnerability/Lethality (V/L) analysis as conducted by the US Army Research Laboratory (ARL) almost always relies on geometry that has been created by suppliers using Non-Uniform Rational B-Spline (NURBS) geometric primitives. This project explored the possibility of an automatic conversion process to convert these models to a more traditional constructive solid geometry format that would offer speed improvements in analysis runs. Initial development has resulted in successful conversion of simple models and partial simplification of more complex targets, as well as suggesting future development directions to achieve higher conversion success rates.

15. SUBJECT TERMS
CAD, NURBS, CSG, conversion

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Clifford W Yapp
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER (Include area code)**
Unclassified	Unclassified	Unclassified	UU	24	410-278-1382

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

1. Introduction

The US Army Research Laboratory (ARL) and its predecessor the Ballistic Research Laboratory (BRL) have developed many predictive vehicle vulnerability analysis techniques over the last 50 years. All of these techniques rely fundamentally on the ability to represent a vehicle's shape in a computer in a form that can be interrogated by geometry ray intersection. The BRL was involved with early work by the Mathematical Applications Group, Inc. team in the late 1960s,[1] which created and implemented a geometry representation technique called Constructive Solid Geometry (CSG). CSG combines compact mathematical descriptions of simple mathematical volumes (spheres, cylinders, etc.) with Boolean set theory. For example, a subtraction relationship between a planar volume and a cylinder can define a more complex volume not representable by either shape individually (Fig. 1).

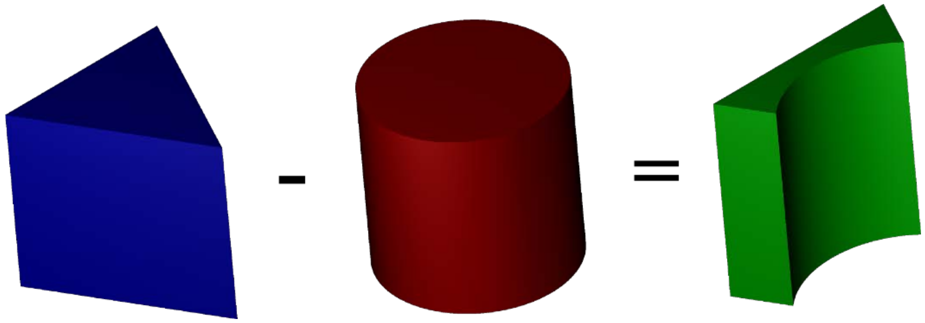

Fig. 1 An example of CSG Boolean shape definition

For several decades CSG was the primary storage format for the US Army's Vulnerability/Lethality (V/L) related vehicle geometry assets. CSG offers advantages in terms of compactness (the earliest systems to use it stored entire vehicle descriptions on punch cards) and solidity guarantees that are advantageous in analysis applications.

Despite these advantages, the use of CSG has declined in V/L applications over the last 10 to 15 years. The commercial Computer Aided Design (CAD) industry, which didn't really exist when CSG was originally invented, has since become a staple of modern manufacturing and it is quite common today for detailed CAD geometry of vehicle assets to be supplied from vehicle vendors. While these existing resources are often useful, the commercial CAD industry does not typically use CSG to describe their geometry. They instead rely on a general boundary surface representation technique known as Non-Uniform Rational B-Spline (NURBS). In the early days, BRL-CAD (i.e., ARL's open-source CAD system) did not have a practical way to perform ray interrogations of NURBS-based geometry, which is quite mathematically complex compared to basic CSG implicit shapes. This in turn meant that taking advantage of pre-existing geometric assets required performing error-prone conversions of these models to triangle-based representations. While still faster than recreating CSG representations of vehicles from scratch, the time to perform conversions and the quality control issues associated with them have long been a target for process improvements. BRL-CAD has recently implemented a capability to directly intersect rays with NURBS geometry which, although providing a definite step forward, is typically slow compared to CSG raytracing.

In general, there has been relatively little modern interest in converting geometry to CSG representations. CSG is not a primary representation format for most of the CAD industry. Generally, shape recognition research has focused on either reverse engineering geometry from point cloud scans or applications such as simplification for Finite Element mesh generation or machine tool path optimization, which means there is no pre-existing solution for the problem of converting NURBS geometric models to BRL-CAD style CSG representations.

This work seeks to investigate the possibility of automatically recognizing when a NURBS-based geometry model is mathematically representable by CSG implicit geometry and deducing that CSG model using the existing NURBS geometry as a guide.

2. The CSG Conversion Pipeline

The entire approach to this problem rests on a single fundamental observation: many NURBS-based geometry targets describe volumes that are in fact bound by simple geometric shapes such as planar volumes, cylinders, and cones. Given that simplicity, it should be possible to characterize those shapes and from them deduce implicit primitive parameters that recreate them. While the idea is simple, the implementation of it is not. Deep introspection of the NURBS data and careful attention to detail are necessary to achieve a successful conversion.

There are several overall guiding principles that dictate the requirements:

1) Accuracy is essential. If we cannot convert a given boundary representation (B-Rep) to a CSG representation that correctly defines the same volume as the original B-Rep, the original B-Rep is preserved.

2) Simpler shapes are preferred over complex shapes. For example, a cylinder should be recognized before the conversation maps that shape to a cone.

3) To keep complexity manageable, relationships between individual shape components that are connected must be planar. This will be explained in more detail below.

To illustrate the various steps of the process, we will use as an example one of the National Institute of Standards and Technology (NIST) Product Manufacturing Information (PMI) example models, shown in Fig. 2.

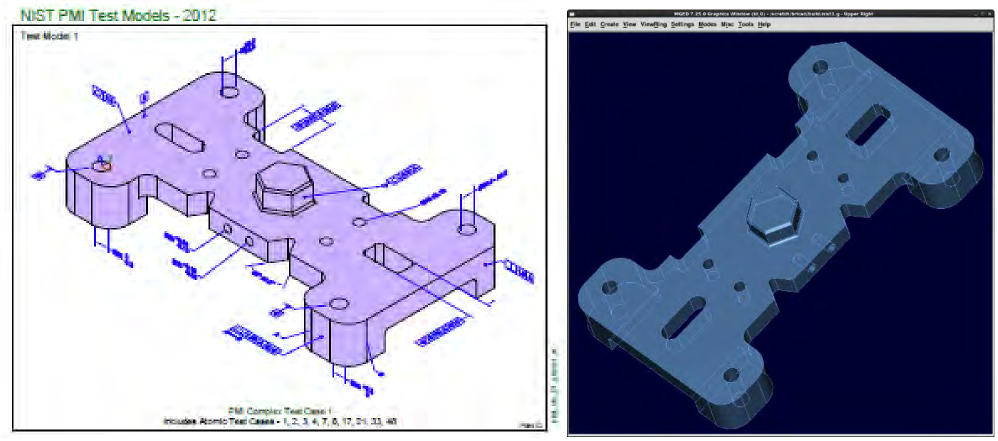

Fig. 2 NIST PMI test case no. 1: a) Left is Original Visualization provided by NIST,[2] b) right is the same model imported and raytraced in BRL-CAD.

2.1 NURBS Topology: Boundary Representations

A volume described by NURBS surfaces is called a boundary representation. Geometry is described by surfaces and curves, but by themselves they are not enough to define a boundary representation. Welding a collection of surfaces and curves into an object requires a great deal of topological connectivity information. This topological information is especially essential to CSG conversion, because it provides relationship information between B-Rep components that allow otherwise disjointed geometry pieces to be acted on systematically.

The major pieces of B-Rep topology used in this process are:

1) Vertex points.

2) Edges, which link a 3-dimensional (3-D) geometric curve with start and end Vertex points.

3) Trims, which associate a 2-dimensional (2-D) curve in a NURBS parametric surface space with a 3-D edge.

4) Trimming loops, which bound areas of a NURBS surface. Trimming loops are defined using ordered arrays of individual 2-D trims.

5) Faces, which combine sets of trimming loops with a geometric NURBS surface. Trims and trimming loops are specific to a single face, while edges exist independently (i.e., a mated edge, the primary type used here, links to 2 trims in 2 different faces).

Trimming loops in faces can serve 2 roles—outer loops, which outline the parts of the NURBS surface that contribute to the volume, and inner loops that identify portions of the surface to be regarded as holes in the face.

2.2 Recognizing Shapes

Given a set of faces from a B-Rep, the single most critical piece of information a CSG conversion needs is: "What implicit shape corresponds to the surface of this face?" Fortunately, the openNURBS library[3] that BRL-CAD uses to support its own NURBS capabilities supplies pre-existing routines for recognizing planes, cylinders, cones, spheres, and tori. Testing of these routines has demonstrated that they are functional and work as expected; and they form the basis on which the CSG conversion capabilities achieved to date have been built.

Surface recognition by itself is not enough—curve characterization is also essential at various stages of the process. The 2 primary tests used thus far, both supplied by the openNURBS libraries, are routines for testing the linearity of a curve and whether a curve is planar. Also useful are a number of distance and vector math operations.

2.3 Subdividing the Problem

While openNURBS provides many extremely useful basic tools, simply recognizing that a face surface is cylindrical is only the beginning of constructing a CSG representation. The basic approach of CSG conversion is similar to that used for most mathematical problems: take a large and complex problem and break it down into a set of simpler problems we know how to solve. This is complicated by the fact that many cylindrical faces (and other nonplanar faces) in a NURBS B-Rep object are not complete implicit shapes but instead use part of that implicit shape to define a more complex shape. An example is the use of cylindrical surface sections to round shape edges between planes, as seen in Fig. 3.

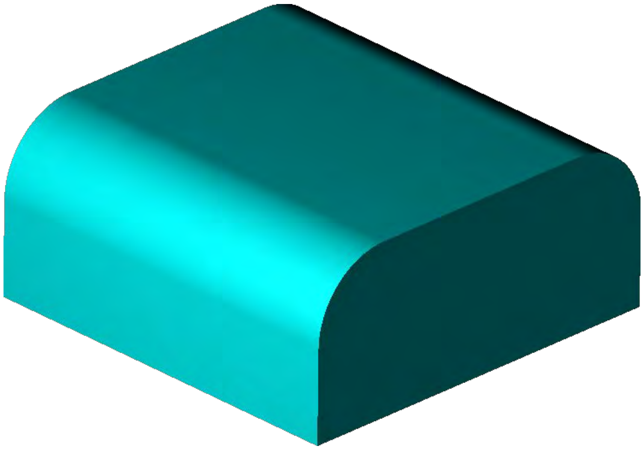

Fig. 3 Rounded edges defined with cylindrical surfaces on a NURBS shape

Given that the plan is to divide and conquer, the question then becomes: "How do we identify and segregate specific portions of the B-Rep for analysis?"

3. Topological Islands: Separable B-Rep Components

The first stage of splitting up a B-Rep into simpler problems involves looking for sets of trimming loops that form independent graphs—that is, no loop in an independent set of loops will share an edge or vertex with any loop outside of the set. These sets are referred to as topological islands, and they identify portions of the B-Rep that can be successfully expressed as independent B-Rep objects. Any island that is defined by a set of loops containing the inner loop of a face is defined as a child island of the island defined using the outer loop of that same face. This relationship establishes a hierarchy of islands and is essential for eventual CSG assembly. Figure 4 illustrates the islands present in NIST no. 1, with each island having its own color.

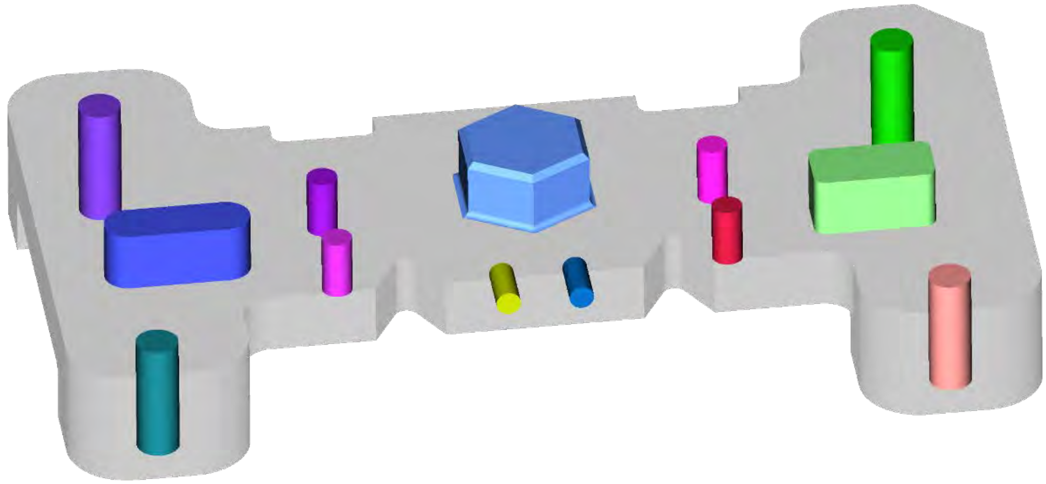

Fig. 4 NIST PMI no. 1 topological islands, identified by color

Once we have identified these islands, shapes that might be modeled by CSG implicit primitives begin to emerge. In this particular case, 8 of the islands form cylinders that can be modeled directly without the need for any additional CSG complexities. Five, however, require further subdivision and this time-trimming loop network will not split them up.

4. Topological Shoals: Breaking Islands into Components

The next stage of simplification is to use the faces in an island to isolate topological shoals—that is, pieces of an island that can define a portion of the volume but not the whole. Before defining a shoal, the surface types of the island faces are checked to verify that all surfaces in the island can theoretically be represented exactly by a CSG implicit shape. If not, the island is considered unconvertible and a B-Rep defining that shape is constructed using the subset of the original B-Rep's data identified as part of that island. If the island is convertible, shoals are identified and converted. To demonstrate the general process of breaking down an island, we will focus on the example shown in Fig. 5.

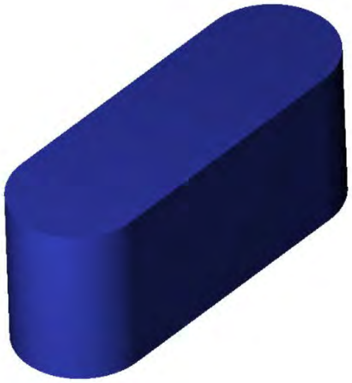

Fig. 5 NIST PMI no. 1 topological islands, identified by color

This island is made up of a combination of cylindrical and planar faces. To define a shoal, we select one of the cylindrical faces (Fig. 6).

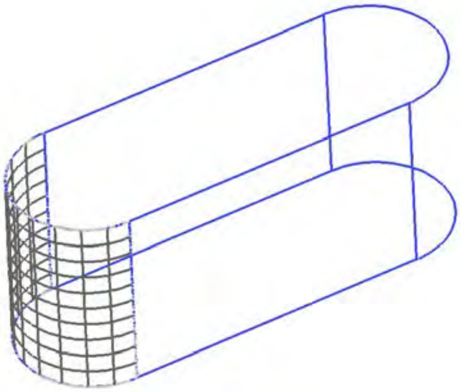

Fig. 6 NIST PMI no. 1 island cylindrical face

That face provides information for defining cylinder parameters, but clearly the face does not represent a full cylinder. We also need to check whether the top and bottom planes of the cylinder do in fact have a normal parallel to the cylinder axis defined by the face. The shoal loop associated with this face establishes the face connectivity, which lets us find and test the 2 planar faces that bound the cylinder (Fig. 7). (In this particular case they happen to be parallel—if they had not been, additional primitives would be needed to refine the shape.)

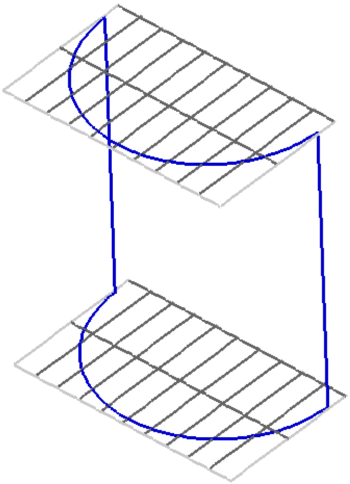

Fig. 7 NIST PMI no. 1 island top and bottom planar bounding faces

While the faces from the top and bottom are part of the cylinder definition, there is still clearly a need for part of the cylinder to be trimmed and no explicit face from the original shape will do so. The information for defining this "pseudo" face comes from the linear edges that mate the shoal to its parent island—they allow another plane to be constructed (Fig. 8).

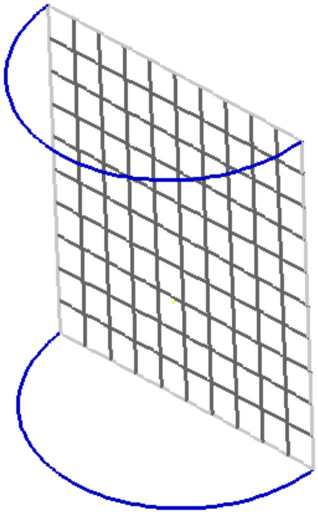

Fig. 8 NIST PMI no. 1 island implicit plane

Knowing this plane, there are multiple possible strategies for defining a primitive to "clip away" the portion of the cylinder not needed for the shoal. Two methods that have been tried are constructing a subtraction using polyhedrons defined by corner points, and constructing an intersection using a polyhedron defined by face planes. Currently, the latter method looks to be more promising but both have proven feasible.

A similar procedure is performed for the cylindrical face on the opposite side of the island—this results in 2 simple CSG hierarchies that express the end volumes of the island (Fig. 9). That leaves the middle, which is handled in a slightly different fashion.

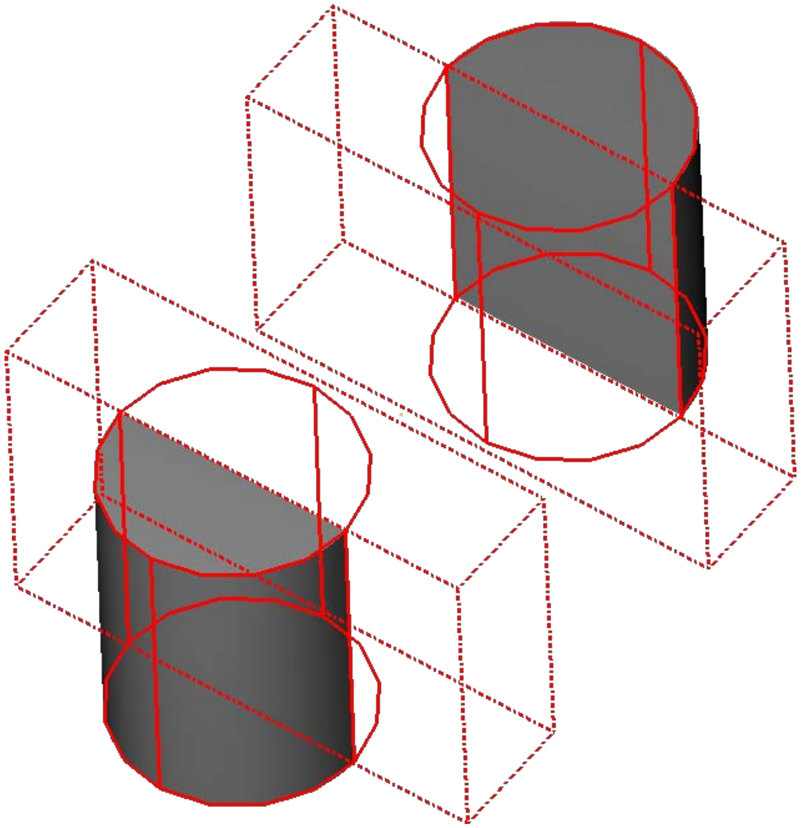

Fig. 9 NIST PMI no. 1 CSG end cap shoals described with cylinders and subtracting arbs

5. Topological Nuclei: Core Island Volumes

Having processed the shoals of an island, it is common to find that there is a planar subset of faces that is not handled by any shoal. This region is termed the topological nucleus of the island, and its final form is a combination of planar faces from the original B-Rep and the implicit faces defined by the shoals.

Examining the example used to create shoals previously, there are 4 faces that are not accounted for by shoal shapes (Fig. 10).

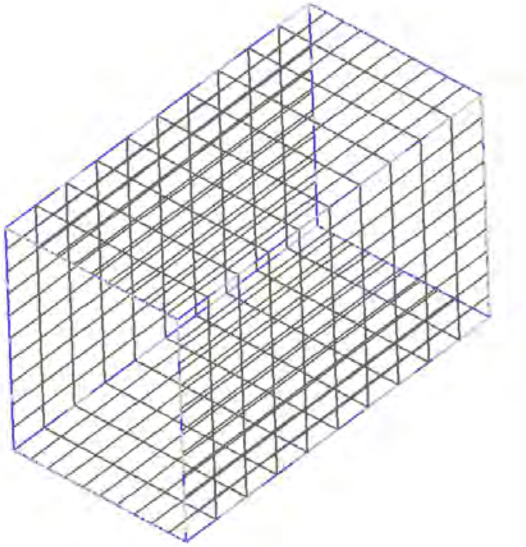

Fig. 10 NIST PMI no. 1 island planar faces

To define a closed volume, we add the 2 implicit planes from the shoals (Fig. 11).

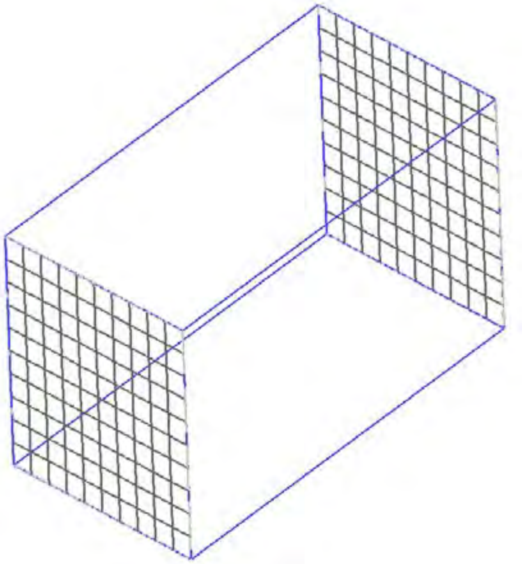

Fig. 11 NIST PMI no. 1 island implicit planes

This results in a closed volume, which may easily be represented using BRL-CAD's planar primitive shapes (Fig. 12).

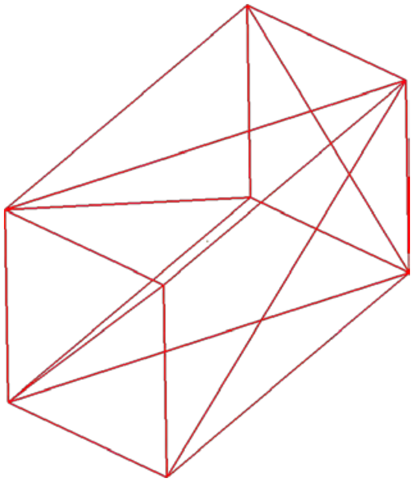

Fig. 12 NIST PMI no. 1 nucleus CSG primitive

Given the shoals and a nucleus, it is now possible to define a CSG hierarchy for the island. In this case, because all shoals are net positive contributors to the volume, the nucleus and shoals are combined together into a single object with union operators. Had a negative shoal been present, all positive shoals would have been combined into a union object and the subtraction would be made from that union, in case a subtracting shoal removes volume from a union shoal. Figure 13 shows the final CSG shoal structure.

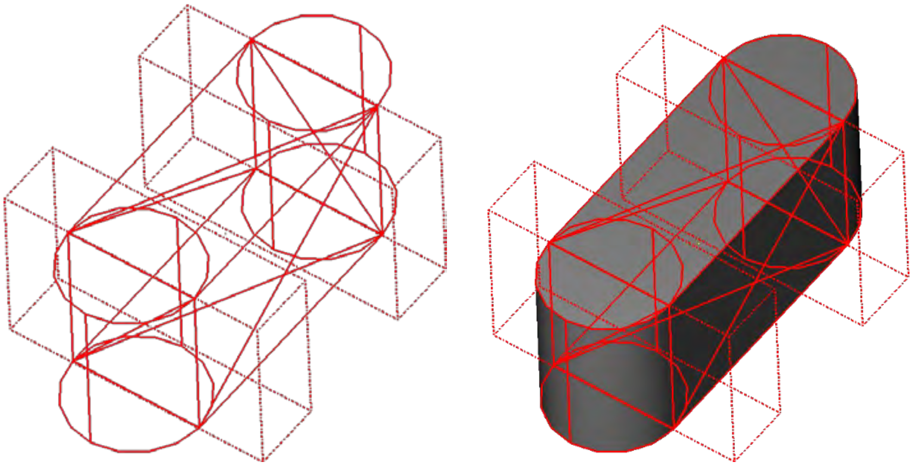

Fig. 13 NIST PMI no. 1 final island CSG structure (wireframe only on left, wireframe and final CSG rendered volume on right).

6. Island Boolean Hierarchy

After processing the islands and shoals, it is time to assemble the collection of individual CSG shapes and hierarchies into an overall Boolean hierarchy that accurately represents the final shape. At the moment, the assembly rules are fairly straightforward, based on the plane shared between a parent island and its child island. The inner/outer loop mappings mentioned at the beginning of this discussion now come into play because they inform the processing routines that islands need to be checked against other islands for subtracting. Given the plane from the shared loop, are the child edge midpoints above that plane or below it? If above, that child is combined with the parent using a union operation and if below, a subtraction operation.

Taking all of the island results from the NIST example, this gives us a CSG shape that accurately represents the original NIST B-Rep (Fig. 14).

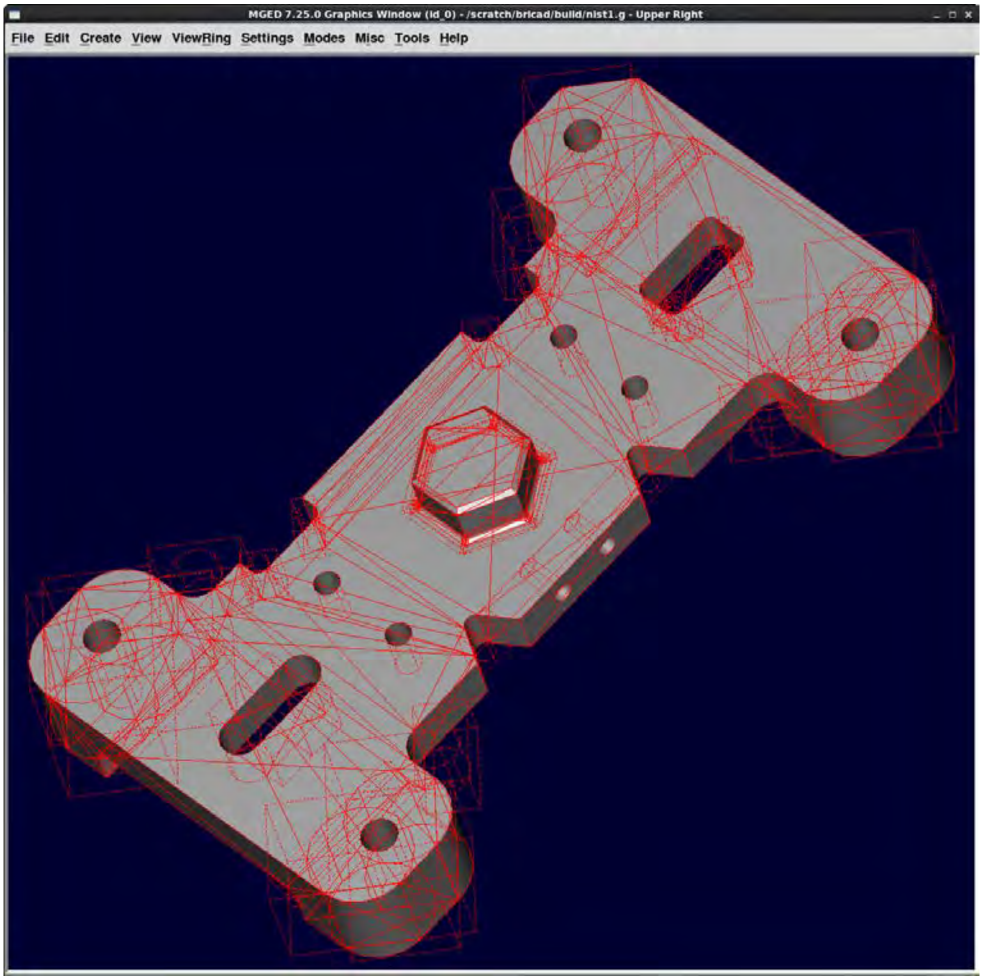

Fig. 14 NIST PMI no. 1 final CSG assembly

This same approach is used in the case where some islands of a B-Rep are not CSG convertible—even a partial conversion can offer some performance enhancements.

There are definite limitations on what types of geometry can be processed assuming planar relationships between islands, but a surprisingly large number of B-Reps have been successfully converted using this method.

7. Raytracing-Based Validation

After processing, the heuristic nature of many of the methods used in this process (i.e., plane-based Boolean determinations) requires a validation step to eliminate incorrectly converted CSG hierarchies. Raytracing is a capability BRL-CAD possesses for both CSG and NURBS B-Rep geometry and it is the performance domain in which ARL's V/L tools need the geometry to work correctly, which makes a raytracing-based test a natural fit.

The basic ray validity test is quite simple; both the B-Rep and the CSG hierarchy are loaded together into a scene and a grid of rays is fired from the x, y, and z planes with grid density based on the bounding box size of the objects. In order to avoid issues with differing grazing ray hit behaviors (a problem even between different types of CSG implicit shapes) a ray segment result is considered for processing only if 4 surrounding rays fired at small offsets from the original ray report similar results. Figure 15 illustrates an issue observed with NIST Test Case no. 2 where cylinders were being created but they did not precisely line up with the parent planar shapes. The small areas of blue are rays reporting solid line segments present in the CSG hierarchy that were not present in the original shape. This would result in a rejection.

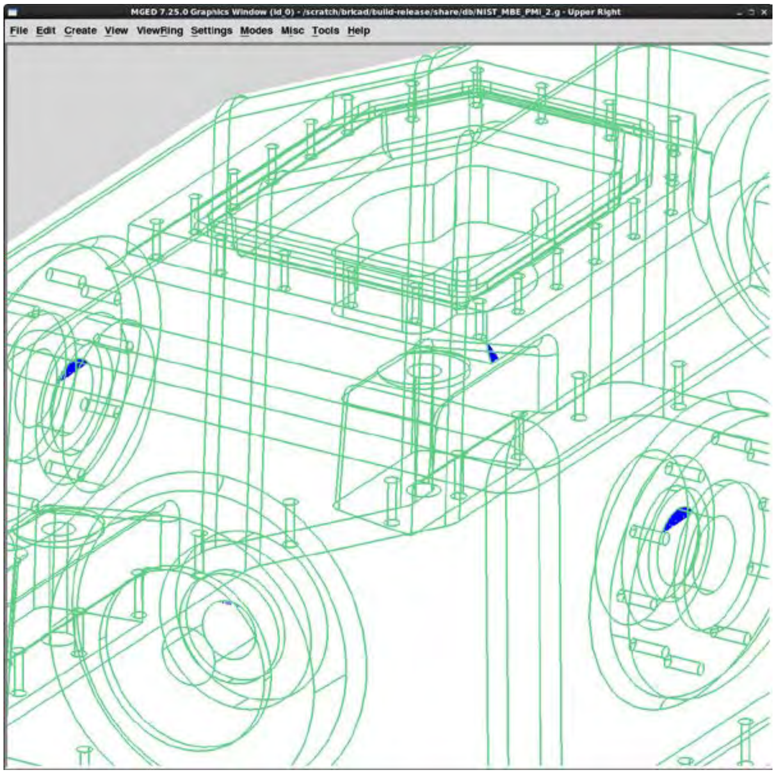

Fig. 15 NIST PMI no. 2 example of ray difference processing

8. Performance Implications

Using the 5 NIST PMI examples as test cases, the comparative performance numbers are reported in the table.

Table **Performance and size characteristics of NIST PMI CSG conversions**

NIST no.	CSG size vs. NURBS (Mb)	CSG performance vs. NURBS	Conversion status
1	0.03×	31×	Full
2	0.48×	2×	Partial
3	0.03×	13×	Full
4	0.51×	3×	Partial
5	0.62×	2×	Partial

The bottom line appears to be that full CSG conversions, when achieved, improve performance by an order of magnitude while also reducing size by an order of magnitude, for equivalent raytracing results—even when the conversion is only partial size and performance gains were observed.

9. Conclusions and Future Work

The potential for improved performance in raytracing using this approach looks quite significant, provided accurate conversion can be achieved on a large enough percentage of a vehicle target geometry. More work needs to be done to support shapes beyond cylinders; currently there exists only some basic support for cones and a few experiments with spheres. Torus shapes appear to be important as a common smoothing surface between cylinders and planes, but they present some additional challenges compared to other primitives and have not yet been seriously studied.

For practical large-scale purposes, the heuristic approaches currently used for tasks such as Boolean evaluation and detection of unsupported geometry configurations will have to be supplemented by slower but more robust fallbacks to achieve practical results. Experiments were done to use the raytracing evaluation of both B-Rep and CSG hierarchy not just for validation, but to guide Boolean assembly, which showed promise but was not working beyond a slow proof-of-concept stage demonstration by the end of fiscal year 2016 (FY16).

Also, for maximum impact on vehicle analysis, the improvements to robustness of methodology will likely need to be combined with research into performing the CSG hierarchy conversion process using approximate fitting (i.e., possibly using Random Sample Consensus methods such as those used to find shapes in point clouds) rather than (or more probably, in addition to) the exact-within-tolerance approach used in the current work. Many NURBS surfaces can be approximated with CSG shapes with only minimal loss of precision, which is likely to be acceptable in many V/L applications. The introduction of nonexact matching also opens up additional problems such as the possibility of introducing new overlaps, which will need careful thought.

10. References

1. Guber W, Nagel R, Goldstein R, Mittelman PS, Kalos MH. A geometric description technique suitable for computer analysis of both the nuclear and conventional vulnerability of armored military vehicles. Army Ballistic Research Laboratory (US); 1967 Aug. Report No.: AD0847576.

2. National Institute of Standards and Technology. PMI Test CAD Model Dataset 1, Combined/Complex Test Cases; Dec 2014 [accessed 2015 Oct 14]. http://www.nist.gov/el/msid/infotest/mbe-pmi-validation.cfm.

3. Robert McNeel and Associates. The openNURBS Initiative; 1993–2013 [accessed 2015 Oct 14]. http://www.rhino3d.com/opennurbs.

List of Symbols, Abbreviations, and Acronyms

2-D	2-dimensional
3-D	3-dimensional
ARL	US Army Research Laboratory
B-Rep	boundary representation
BRL	Ballistic Research Laboratory
CAD	Computer Aided Design
CSG	Constructive Solid Geometry
FY	fiscal year
NIST	National Institute of Standards and Technology
NURBS	Non-Uniform Rational B-Spline
PMI	Product Manufacturing Information
V/L	Vulnerability/Lethality

1 (PDF)	1 DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
2 (PDF)	DIRECTOR US ARMY RESEARCH LAB RDRL CIO LL IMAL HRA MAIL & RECORDS MGMT
1 (PDF)	GOVT PRINTG OFC A MALHOTRA
1 (PDF)	DIR US ARMY EVALUATION CTR HQ TEAE SV PA THOMPSON
3 (PDF)	DIR USARL RDRL SL D BAYLOR P BAKER RDRL SLE R FLORES
18 (PDF)	DIR USARL RDRL SL S COARD RDRL SLB B BOWEN RDRL SLB D R GROTE RDRL SLB E M MAHAFFEY RDRL SLB G D JORDAN J ABELL J BAKER V CERICOLE D KREGEL C HEAPS RDRL SLB S M PERRY G SAUERBORN W BOWMAN R PARKER S MORRISON N REED C YAPP RDRL SLB W S SNEAD