A CAD Assembly Simplification Approach with Ray Casting

Landon Wright[1] , Brendon Forsgren[1] , Trent Owens[1] , Mathew Haskell[1] , John Salmon[1] , Eric Bowman[2]

[1]Brigham Young University,
[2]Ford Motor Company,

Corresponding author: John Salmon, johnsalmon@byu.edu

Abstract. We present novel methods for the removal of interior bodies from complex assemblies using ray casting. These methods locate and preserve bodies that represent the high-fidelity exterior surface of an assembly while removing all interior bodies that do not contribute to the exterior assembly surface. In so doing we create parts that can accurately be used for assembly packaging and other tasks without suffering from the inefficiencies that come from working with the full assembly. We further present an analysis of the process on assemblies of known properties and several use cases with simplification results. Finally, we present directions for future research that could enhance this work.

Keywords: Ray Casting, Assembly Simplification, Computer-Aided Design, External Geometry
DOI: https://doi.org/10.14733/cadaps.2019.864-877

1 INTRODUCTION

CAD assembly models used to design systems in the automotive, aerospace, and other industries often contain thousands of parts. An assembly file of this size slows file load times, makes model manipulation difficult, and reduces computational performance, making common tasks such as checking for part interferences unnecessarily difficult and time-intensive. In the case of supplier provided sub-assemblies, internal bodies in the assembly are beyond the scope of such tasks and could be neglected entirely. A simplified assembly model which preserves the geometry of outer surfaces in high fidelity and does not contain internal bodies would serve as a means by which such tasks could be performed efficiently while maintaining analytical accuracy. An example of where this reduced complexity is beneficial can be seen in Fig. 1. In this example, a designer working to develop a nacelle around the turbofan is unconcerned with the detailed interior bodies that are shown through the translucent body, and these bodies only serve to slow file load time, model manipulation speed, and the computation time of the CAD engine. Thus, it is advantageous to remove them from the assembly model. Previous solutions to this problem found in the literature can identify exterior bodies using tessellations [10]

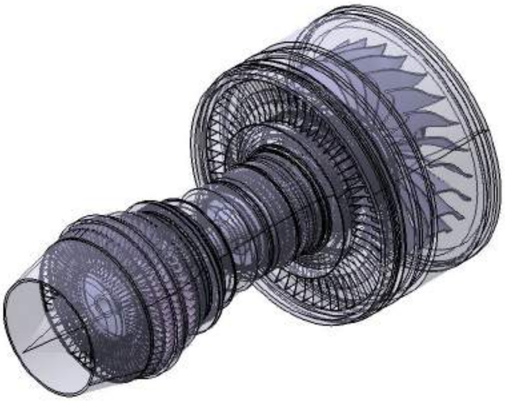


Figure 1: Image of a turbofan engine with translucent casing showing internal bodies that are irrelevant when designing interfaces/checking clearances, etc. with the exterior casing of the assembly.

[7]. While these tessellations certainly reduce time spent loading, manipulating, and processing the model, they result in a loss of accuracy that may be unacceptable for analyses that require very high accuracy.

We present a new CAD assembly simplification method which features ray casting as the enabler to identify internal bodies of the assembly. After identification, external bodies are copied into a new CAD part and properly positioned via the CAD system API (application programming interface). The method is a relatively straightforward and effective process while preserving external geometry exactly rather than through geometrical approximations. The percent reduction in number of parts is heavily dependent on the part being simplified, but test cases have shown reductions of up to 71% in the number of bodies present in the final assembly.

2 PREVIOUS WORK

Research related to simplifying CAD assembly models has been approached in generally two different ways: mesh model simplification and B-rep or feature based solid model simplification [5].

The former category seeks to create a mesh from a set of 3D data and afterwards simplify it. A well known example of mesh reconstruction was presented by [10], and was followed by many variants, improving the algorithm in different ways such as that discussed in [4] and [7] who focused on sharp feature preservation. Kuo and Yau [8] proposed another route to mesh model simplification, applying an approach which combines surface reconstruction with sharp feature recovery. Various mesh simplification algorithms have been put forth, most of which implement techniques like vertex-elimination and edge-collapse [12], [3]. Alternative routes to the conventional three-step mesh processing method include a dual contouring algorithm to extract a simplified surface directly [2] and an octree-based isocontouring algorithm to create hexahedral meshes [11]. However, the meshes still suffer from issues such as from holes in the mesh and some error in the approximated mesh surface.

Methods in the second category generally are characterized by an algorithm to detect invisible or internal features or bodies. For example, Kanai et al. [5] pre-rendered models from multiple view directions and read the rendering results from the frame buffer to determine invisible features. Yu et al. [13] used a similar method, but accessed the CAD system frame buffer thereby avoiding format conversion. Kwon et al. [9] explored ray casting as a means to find internal features. These methods were effective and fast at detecting

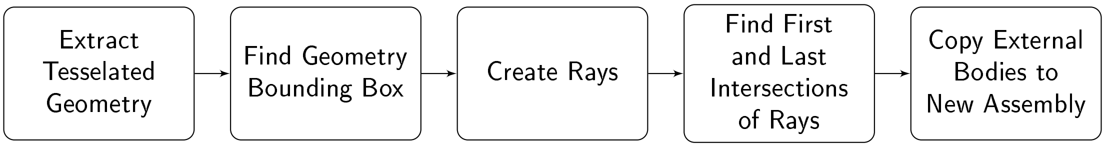


Figure 2: Process flowchart.

invisible features in different assembly configurations, but the simplified end result was not exact; some external features were removed or simplified into something other than the original features. The method described in [9] runs the risk of losing features if the level of detail is set too low. Our algorithm, which falls into this category of feature/body detection, attempts to perform assembly simplification though the identification of all exterior bodies using ray casting. The interior bodies are then identified through exclusion and removed while leaving the details on the exterior of the assembly unaffected. This approach results in a simplified part that contains an exact copy of the original geometry pertaining to the exterior surfaces of the assembly.

Ray casting has been extensively studied in the fields of computational geometry and computer graphics since the first ray casting algorithm in 1968 [1]. In traditional ray casting, rays are "cast" from a common point and traced through the scene until they intersect with objects. The algorithm may then use the intersection information, in the field of computer graphics, for example, to determine how to represent and compute the scene: certain objects may be left out, others may be shaded, still other visual characteristics may be applied such as a disfigured object due to rays passing through a thick transparent material [6].

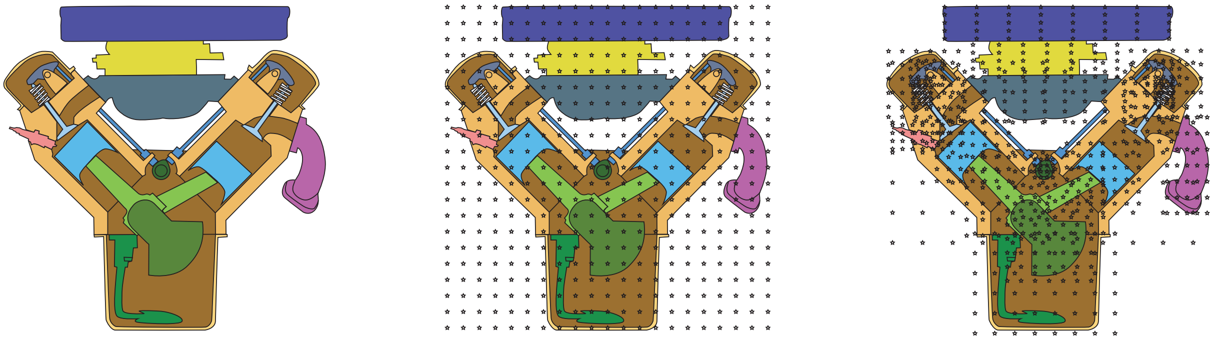
3 METHODOLOGY

3.1 Process Overview

As introduced above, the objective of this research was to create an algorithm that would remove all of the interior parts of a CAD assembly while preserving the exterior surface features of the assembly such that interfaces, alignment, and other geometric relationships can be evaluated in the context of a larger assembly. To achieve this, a ray casting algorithm was developed, implemented, and evaluated. The algorithm extracts the geometry from the CAD system, determines which bodies of the assembly are pertinent to the outermost surface of the assembly, and creates a new assembly that contains only the relevant bodies with the correct position and orientation. An overview of this process is shown in Fig. 2 and discussed in the following paragraphs.

3.2 Geometry Extraction

The process described in this paper can successfully be implemented entirely through a CAD system API, however the long wait times associated with interfacing through the API results in exceptionally long program run times. Initial trials suggested that assemblies of as little as 12 parts would take over 36 hours to complete. As a result, the algorithm is designed to operate and perform computations on tessellated data that is obtained from the relevant CAD package while creating the simplified assembly from the original geometry. Using tesselated data we are able to reduce the time required to perform the ray tracing operations on the same 12 part assembly from over 36 hours to 1 minute 4 seconds. The tessellation data can be obtained in a variety of ways; one of the simplest is through the exportation and parsing of .STL files, but direct extraction through the API or through the use of alternative file formats is also possible. A side benefit of this approach is that the actual simplification process is not dependent on any particular API. As a result the process is easily extensible to other CAD systems as the only portion that requires adjustment is the extraction of the geometry and the creation of the simplified assembly.



(a) Cross section of engine with no rays overlaid

(b) Engine cross section overlaid with evenly spaced rays

(c) Engine cross section overlaid with rays whose spacing is dependent on the bounding boxes of possible intersecting parts

Figure 3: Illustrative engine cross sections used to show the differences between the two ray spacing methods. Note that in Fig. 3c the density of the rays varies depending on the underlying parts, and becomes especially dense around the valve covers where there are numerous small parts.

3.3 Bounding-box Calculation

The tessellation data results in a mesh that corresponds to each body in the original assembly. Each of these meshes is then positioned spatially using the rotation and translation matrices that define its location in the assembly. These matrices are typically available through the CAD system's API. Once the meshes have been obtained by the process software, bounding boxes are computed by finding the minimum and maximum extents of the meshes of interest in the X, Y, and Z directions. The surface of this bounding box is then used as a basis for the creation of a collection of ray emitters that lay on the surface of the bounding box.

3.4 Ray Creation

In order to cast rays the initial step is to define the emitter locations of the rays. Our method differs from the traditional ray tracing method described in [1] in that we generate one emitter for each ray rather than one emitter that acts as the origin for all of the rays. The process of placing the emitters can be carried out through one of two methods. The first places the emitters in an evenly spaced grid on the surface of the bounding box for the entire assembly. The spacing of this grid is a user defined value that corresponds to the orthogonal distance between each emitter in the grid. The second is a more intelligent and adaptive spacing of the emitters. In this second method a bounding box of the same orientation as the assembly bounding box is calculated for each body in the assembly. The emitters are then placed such that a minimum number of rays will traverse the bounding box of each body in the assembly. The difference between these two methods is illustrated in various models presented in Fig. 3. In the image on the far left (Fig. 3a), a cross section of an engine with no rays overlaid is presented. The center image (Fig. 3b) shows the ray pattern produced by spacing the rays evenly throughout the entire assembly bounding box. The rightmost image (Fig. 3c) shows the same engine overlaid with rays spaced using the adaptive spacing method. As can be seen in this last image (Fig. 3c), there are comparatively fewer rays that intersect with no bodies along the direction into the page. We can note that the density of the rays decreases at the top of the model where there is only one large body and increases significantly in the vicinity of the valves and rockers. This increase of ray density serves to ensure that the smaller parts present in this section are detected if they are in fact external bodies while

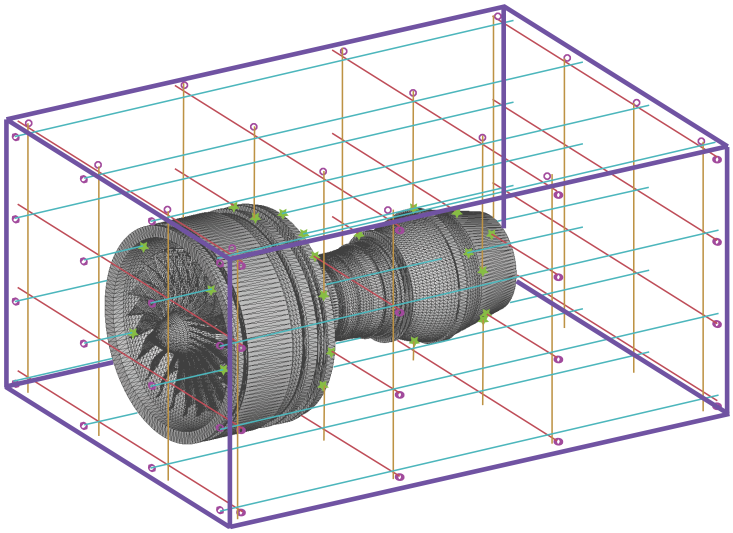


Figure 4: Three-dimensional visualization of the ray tracing process with only 48 rays shown on the a turbo-fan engine. Note the green stars showing the intersections of the rays with the bodies.

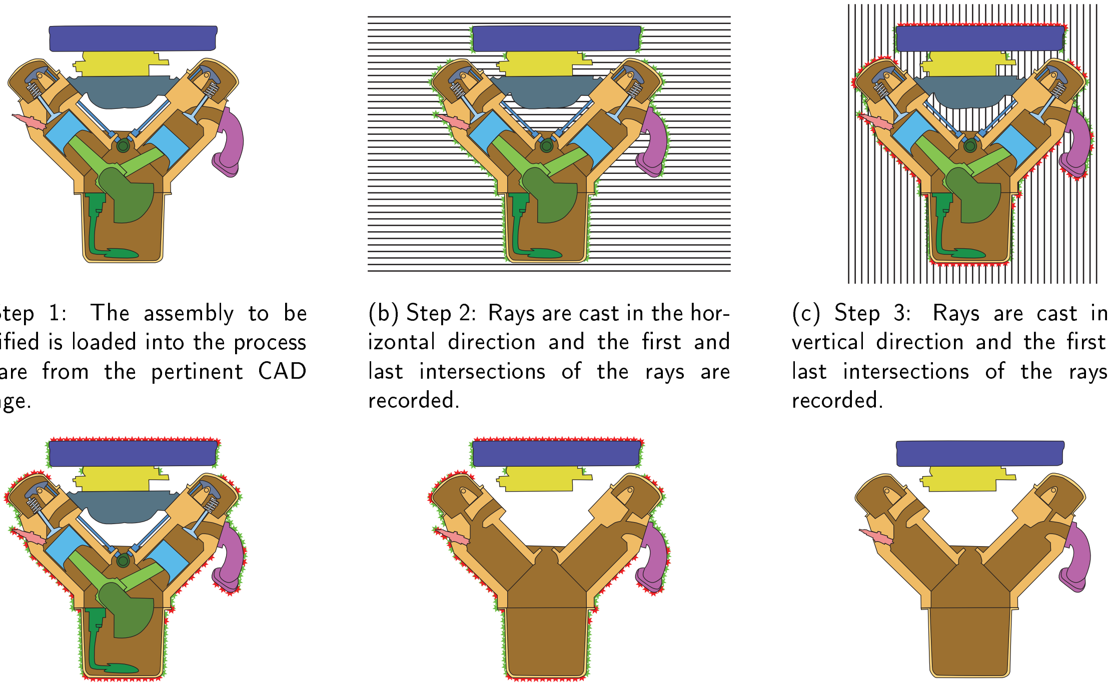
also reducing the number of rays required in areas where there are few large bodies. The method of adaptive spacing theoretically results in fewer rays and a faster computation time to achieve comparable results to the first method as it will selectively place rays where there are more bodies of possible intersection. However, it can also experience exponential growth in the number of rays if there is a large number of small parts in the assembly which may cause this method to be slower as exhibited in Tab. 2. We have noted that assemblies with several hundred fasteners could also be problematic when using the second method as the number of rays may grow to an unsustainable level, and in this case the first method of evenly spaced rays may be required.

3.5 Determine Ray Intersections

Once the emitter locations have been placed, the rays are cast in a direction perpendicular to the face of the bounding box and are followed through the bodies of the assembly until each ray intersects the plane on the opposite side of the bounding box. Using the meshes, the ray will be compared to each triangle in the mesh to determine if the ray intersects a given triangle. Once all the intersections have been found, the algorithm next considers the nearest and the farthest intersections from the emitter, thereby identifying the location where the ray enters and leaves the assembly. These two points indicate which bodies are the outer bodies (on the near and far side) of the assembly for a given emitter. A reference to these two bodies is stored for later use when creating the simplified assembly. By examining the first and last bodies that are intersected by each ray, we reduce the number of rays that are required for the algorithm by half. This process is then repeated for each emitter and corresponding ray.

3.6 Copy External Bodies to New Assembly

Upon completion of ray casting the creation of the new simplified assembly begins. Using the references for the external bodies that were stored in the previous step of the algorithm, the original bodies from the assembly are copied into a new assembly while preserving their location and rotation information. This ensures that the exterior of the created assembly is geometrically identical to the original while containing significantly less



(a) Step 1: The assembly to be simplified is loaded into the process software from the pertinent CAD package.

(b) Step 2: Rays are cast in the horizontal direction and the first and last intersections of the rays are recorded.

(c) Step 3: Rays are cast in the vertical direction and the first and last intersections of the rays are recorded.

(d) Step 4: The ray intersection points are examined to determine the bodies that correspond to the exterior of the assembly.

(e) Step 5: The bodies not comprising the exterior of the assembly are removed

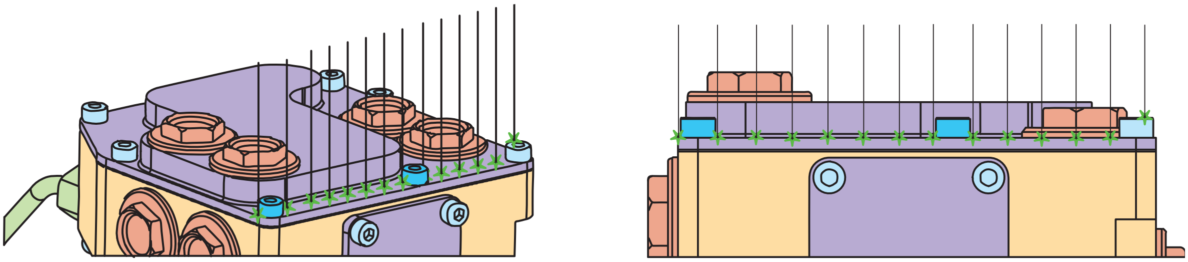
(f) Step 6: A new assembly file is created which is comprised solely of bodies that pertain to the exterior of the assembly.

Figure 5: Two-dimensional example of the process steps that lead to the simplification of a complex assembly.

complexity than is present in the original model. A visual representation of the process can be found in Fig. 5 where the simplification is performed on a 2D cross section of an engine. The case for a 3D assembly follows the same sequence except an additional step is added for a third direction of ray casting orthogonal to the original two directions. This is shown in Fig. 4 where a selection or sample of the rays that would be cast and the associated emitters for one face of the bounding box are shown. This example further shows the placement of emitters using the evenly spaced method described previously.

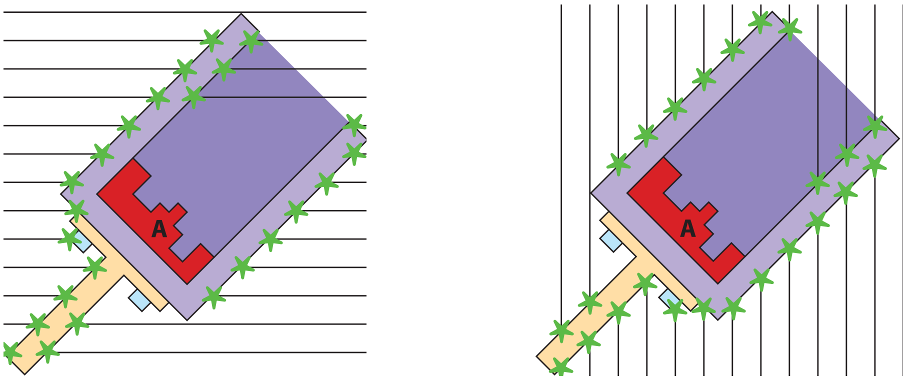
3.7 Limitations

There are several cases where the effectiveness of the discussed algorithm breaks down. One case where the algorithm fails to perform as expected is when there are small pieces on the exterior of the assembly and the user elects to use the even ray spacing method. The algorithm will only find all the exterior bodies if the spacing of the rays is less than the smallest exposed portion of each body to be detected. As shown in Fig. 6 the rays are spaced evenly across the top of the part but at a distance greater than the width of the small socket head screws. As a result the two front left bolts are not identified as exterior bodies by the algorithm. This can be solved by simply decreasing the separation between rays. However, the trade-off for doing so is an increase in the run time of the algorithm. An informed designer or engineer should be able to identify the point where the increased accuracy is not worth the time required to execute the algorithm with a very fine or small spacing distance. One simple method to determine this value is to set the spacing slightly smaller



(a) (b)

Figure 6: Small components may fail to be detected in the ray tracing step if the ray spacing distance is too large. In this example it can be seen that two of the bolts on the top front edge of this engine cylinder were not detected by the rays cast in the vertical direction. This highlights the need for careful selection of ray spacing or the use of adaptive ray spacing if these bodies are crucial to the exterior.



(a) (b)

Figure 7: Rays may fail to identify bodies that could be considered an exterior part. In this example the red object is an interface inside of a cup-like body. Due to the orientation of the assembly the body is not recognized as exterior by any of the rays cast. Depending on the application this body could be considered to be an exterior body. Note that a re-orientation of the part would lead to the correct recognition of the red body labeled "A".

than the smallest body in the model. The intelligent designer may recognize however, that the analysis he or she is performing is not dependent on the smallest bodies in the assembly. Rather it may be dependent on bodies that are 5 times larger than the smallest bodies. He or she is then able to set the ray spacing to 5 times the smallest body. In so doing the runtime of the algorithm is reduced and the efficacy of the analysis is uneffected. An additional option to counteract this limitation is the use of the adaptive ray spacing algorithm discussed in section 3.4. This method ensures that there are several rays cast from each direction that will intersect with each body. As a result, the body, if it is truly an exterior body, will be detected as such.

Another case where the algorithm fails is when a body is hidden from rays in all directions due to orientation or part geometry. Fig. 7 shows an example of this pitfall. The rays, both horizontally and vertically traced, fail to hit the red piece labeled "A" in the center of the assembly either first or last. Unlike the previous pitfall, a finer spacing of the rays will not solve this problem but requires that either the orientation of the part or the rays be modified. In this case if the assembly were to be rotated by 45 degrees in either the clockwise, or counterclockwise, direction the body would be detected as an exterior body. While the solution to this issue in the given example is trivial to solve, such a solution may not be possible when dealing with more complicated assemblies. For the purposes of this work, bodies that are effectively hidden inside of cavities within the part, such as body "A" in Fig. 7 were considered interior bodies. However, if these bodies are considered as exterior for the purposes of an analysis then alternative methods such as multiple part re-orientations or non ray casting methods will need to be evaluated. Lastly this work assumes that the provided assembly is whole and complete. If however, there are missing components within the assembly, many false positives may be generated.

4 RESULTS AND DISCUSSION

4.1 Process complexity and scalability

To evaluate the process in a controlled manner we created a series of test case assemblies that consisted of a series of identical rhombic dodecahedrons tessellated in three dimensions to form cubelike assemblies of NxNxN rhobic dodecahedron, an example of which is shown in Fig. 8. We chose to use tessellated rhombic dodecahedron for these tests because it easily tessellates via translation without rotation to form a large geometric assembly, but it does not result in the larger assembly consisting of planar faces of identical size for each body as would be present in an assembly of cubes. For each of the created assemblies the total number of bodies, the number of exterior bodies, and the number of internal bodies was known. Using this information we were able to perform the simplification of each assembly and determine the computation time required for both the evenly spaced rays and the adaptively spaced rays with respect to these different parameters. The results of these trials are shown in Tab. 1.

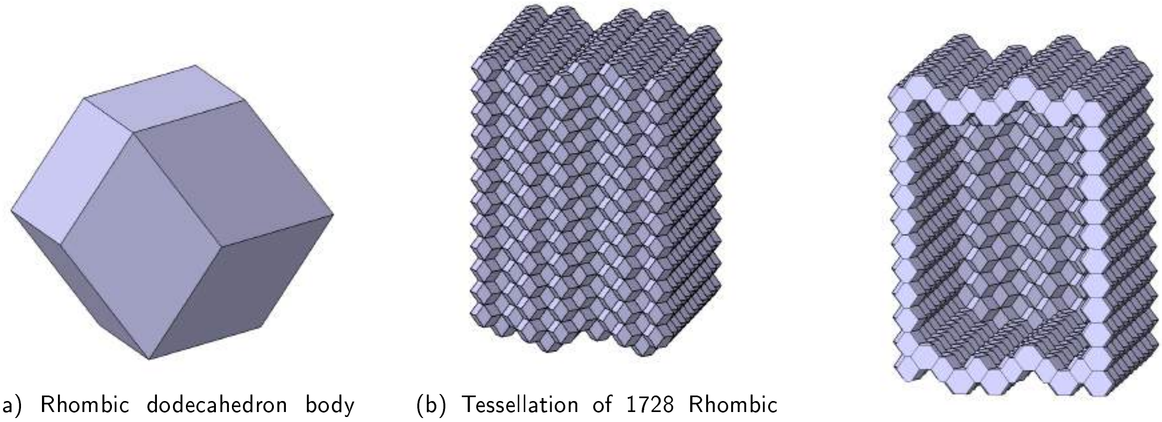
The run time of the process as a whole is dependent on the number of rays that need to be traced and the number of triangles in the tessellation of each body as well as the number of bodies in the assembly. The limiting factor in both methods is the time required to obtain the tessellated data from the CAD software. Our implementation saves each body as a .STL file which with large numbers of bodies (>10,000) can take upwards of one hour to accomplish. As such we expect that the time required to simplify an assembly should scale linearly with the number of bodies in the assembly.

Assembly size	Num. Bodies	Bodies Remaining	Percent Reduction	Time (min)	.STL Time (min)	Ray Tracing Time (min)
3x3x3	27	26	4%	0.17	0.13	0.05
6x6x6	216	152	30%	1.08	0.83	0.23
12x12x12	1728	728	58%	8.40	7.10	1.28
24x24x24	13824	3176	77%	87.42	77.02	10.40

Table 1: Results of controlled rhombic dodecahedron tessellation trials.

4.2 Use cases

The process we have described relies on the simplification of an assembly through the removal of interior bodies that do not contribute to the exterior surface of the assembly. As a result, the efficacy of the algorithm is highly dependent on the nature of the assembly to be simplified. Tab. 2 presents the results of the algorithm after execution on several example assemblies. The table contains the initial number of bodies contained in the assembly, the number of bodies after the simplification algorithm, the percent reduction in bodies for



(a) Rhombic dodecahedron body that formed the basis of the tessellation.

(b) Tessellation of 1728 Rhombic dodecahedron to form 12X12X12 assembly.

(c) Cutaway view of the simplification of the assembly shown in (b).

Figure 8: A rhombic dodecahedron was tessellated to form the test cases for the controlled trials. This tessellation was then simplified using the described process.

each assembly, and the run time of the algorithm for each method of ray placement. This information is also presented in Fig. 9 to allow increased readability and comparisons of the various assemblies tested.

Since the efficacy of the algorithm largely depends on the nature of the assembly being simplified, we present detailed case studies using assemblies 13, 1, and 2. Assembly 13 is a complicated automotive HVAC Unit presented in Fig. 10. There are 353 bodies in this assembly (a relatively large number of bodies in our test set) and both methods of ray casting simplify the assembly by over 50%.

The percent reduction seen using Method 2 is smaller than the percent reduction seen using Method 1 (54.7 % vs. 72.5%). This is due to the inability of Method 1 to detect parts with a size smaller than the 15mm ray spacing used, and, as a results, mistakenly categorizing them as interior bodies. The run time for this test assembly was the longest of the 14 assemblies, despite another assembly (i.e. Assembly 8) with more bodies. This is likely due to the complicated geometry that existed in Assembly 3 that caused the process of extracting the tessellation of the geometry to take longer than other use cases.

Assembly 1 is a model of a turbine that contains 16 bodies. An image of both the original and simplified turbine are shown in Fig. 11. The turbine represents a special case where the assembly contains relatively few bodies but occupies a very large space of about 3.5 meters by 1.7m by 1.7m. As can be seen from Tab. 2 both methods of ray placement resulted in the same percent reduction in parts. However, Method 2 ran 16 times faster than Method 1. This large disparity in time is due to the fact that, in order to maintain consistency each simplification using Method 1 was performed using a 15mm ray spacing. This small ray spacing combined with the large physical size of the assembly resulted in approximately 80,000 rays which led to an unnecessarily large run-time.

Another test case is when the assembly to be simplified is very small. Assembly 2 is a sensor that is used on many computer numerical control (CNC) machines and whose largest dimension is 60mm. With this part Method 1 removed 89% of the original bodies while Method 2 removed 44% of the original bodies. However, as can be seen in Fig. 12, Method 1 misses several exterior bodies due to the extremely small dimensions of these parts relative to the ray spacing size. Correcting this would require a very fine spacing between rays and would negatively affect the run time of the simplification in a similar way that a small spacing relative to part size negatively affected the run-time of assembly 6.

		Method 1: Even Ray Spacing			Method 2: Adaptive Ray Spacing		
Test Case	Total Number of Bodies	Bodies Remaining	Percent Reduction	Time (min)	Bodies Remaining	Percent Reduction	Time (min)
1	16	5	68.75%	10.02	5	68.75%	0.62
2	18	2	88.89%	0.1	10	44.44%	0.12
3	20	7	65.00%	0.12	15	25.00%	0.15
4	24	14	41.67%	0.17	20	16.67%	0.18
5	25	16	36.00%	0.15	24	4.00%	0.18
6	36	25	30.56%	1.12	27	25.00%	1.07
7	44	27	38.64%	0.4	39	11.36%	0.4
8	60	34	43.33%	0.4	39	35.00%	0.43
9	74	20	72.97%	0.37	51	31.08%	0.4
10	80	59	26.25%	1.25	78	2.50%	1.32
11	110	77	30.00%	0.77	101	8.18%	0.88
12	248	27	89.11%	3.72	67	72.98%	4.02
13	353	97	72.52%	9.43	160	54.67%	9.7
14	508	286	43.70%	2.55	334	34.25%	2.85

Table 2: The results of the algorithm with both ray casting methods. Method 1 used 27 rays per body and Method 2 used 15mm spacing between rays.

5 FUTURE WORK

There are a number of steps that could be explored to improve the efficiency and robustness of the algorithm discussed. Currently the process is performed on the computer's CPU which lends itself to being run on any computer. However, most ray tracing algorithms are run on a GPU due to its optimization for this type of calculation. Running the algorithm on a GPU would allow for the run time of the ray tracing portion of the process to be decreased significantly as a result the ray spacing used in the even spacing method could be reduced by a large amount thereby avoiding many of the limitations of the current implementation.

To increase robustness the algorithm could also be modified to change the angle from which the rays are traced. This would allow the algorithm to overcome some of the limitations described in previous sections. Additionally, the algorithm could be modified to identify features or surfaces. There are many bodies that have features that appear on the exterior but also have various features that can be considered interior. Simplifying the assembly to just the exterior surfaces or features would further increase the overall simplification of the assembly.

Further our algorithm assumes that the complete assembly is in the configuration that is reflective of the desired simplified assembly. However, extracting the correct configuration of complicated assemblies with multiple different states may not be a straightforward process. Additional research needs to be carried out regarding the extraction of exterior geometry from more complicated upstream tools. Lastly, as there are several proposed algorithms in the literature for accomplishing similar tasks an examination of the different methods, their respective strengths and weaknesses, and their performance on a library of various different assemblies of ranging size and complexity would serve to illuminate the various intricacies that are present in

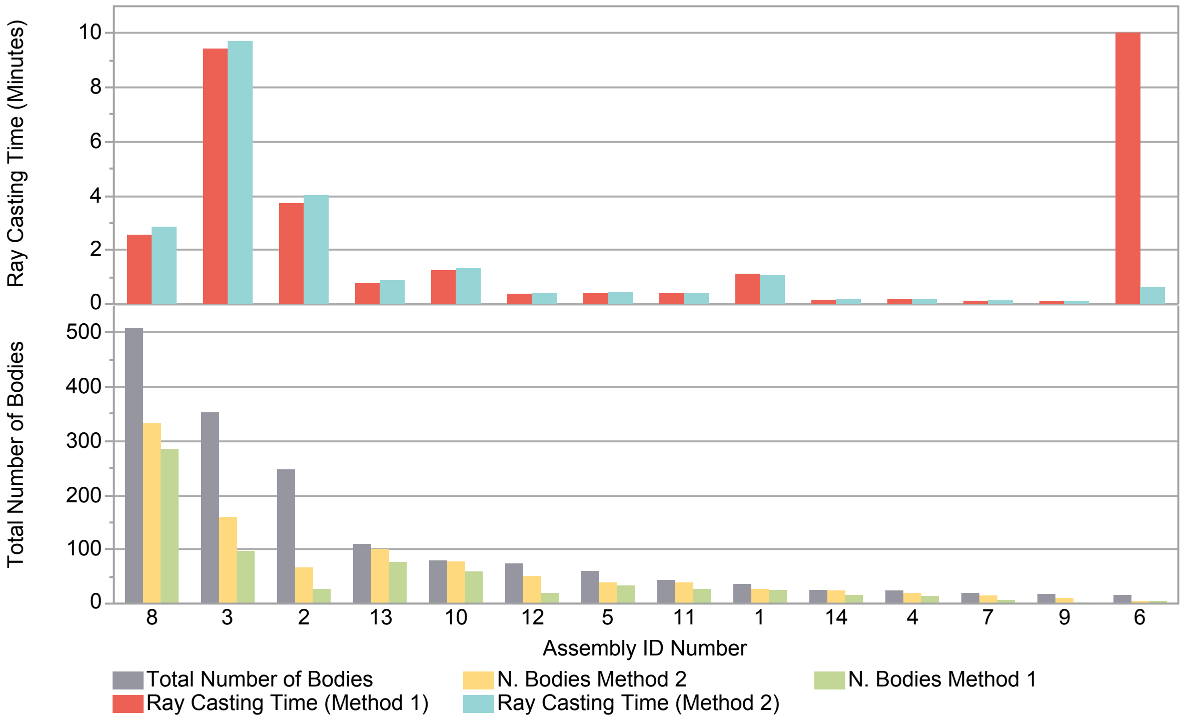
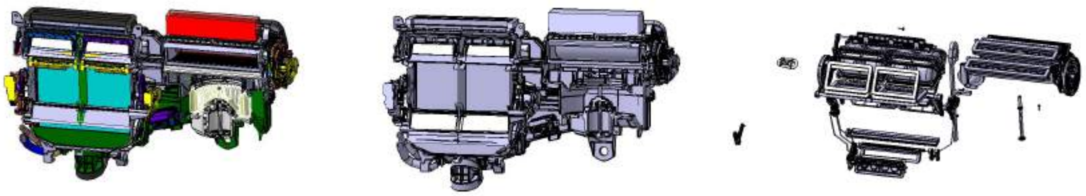


Figure 9: Test assemblies sorted by number of bodies.



(a) Original HVAC cutaway view (352 bodies)

(b) Simplified HVAC cutaway view (160 bodies)

(c) Bodies that were identified as interior bodies to the HVAC unit(192 bodies)

Figure 10: Section views of the original and simplified HVAC unit and the parts that were identified as interior parts.

the different approaches.

6 CONCLUSION

The developed ray casting algorithm for CAD assembly simplification, discussed above, was successful in removing all interior features of the assembly while preserving its exact geometry. This algorithm will help reduce file sizes of CAD assemblies, enhancing load times of the files, improve performance for rotation, general assembly packaging, and other processes.

ACKNOWLEDGEMENTS

This research was funded in part by Ford Motor Company.

ORCID

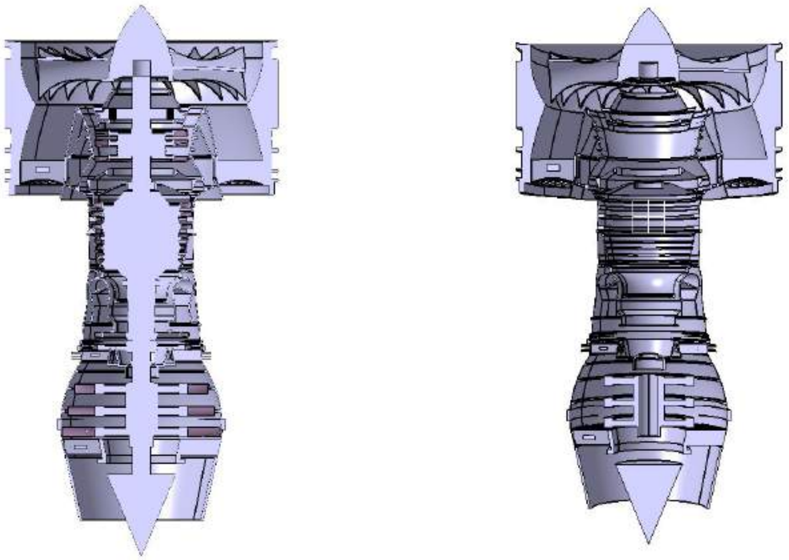
Landon B. Wright, http://orcid.org/0000-0002-4662-6522
John L. Salmon, http://orcid.org/0000-0002-8073-3655
K. Eric Bowman, http://orcid.org/0000-0003-2940-8897

References

[1] Appel, A.: Some techniques for shading machine renderings of solids. In Proceedings of the April 30–May 2, 1968, Spring Joint Computer Conference, AFIPS '68 (Spring), 37–45. ACM, New York, NY, USA, 1968. http://doi.org/10.1145/1468075.1468082.

[2] Barry, M.; Wood, Z.J.: Direct extraction of normal mapped meshes from volume data. International Symposium on Visual Computing Proceedings: Las Vegas, NV, 5358, 816, 2008.

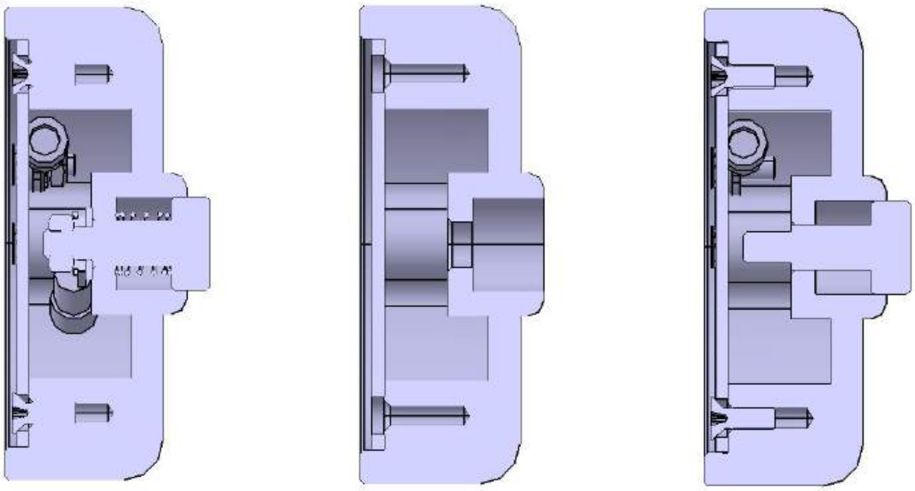
[3] Cohen, J.; Olano, M.; Manocha, D.: Appearance-preserving simplification. In Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '98, 115–122. ACM, New York, NY, USA, 1998. http://doi.org/10.1145/280814.280832.



(a) Section view of the original turbine (16 bodies)

(b) Section view of the simplified turbine (5 bodies)

Figure 11: Images of both the original and simplified turbine.



(a) Section view of the original sensor (18 bodies)

(b) Section view of Method 1 (2 bodies)

(c) Section view of Method 2 (10 bodies)

Figure 12: Simplified views of the CNC Sensor for both Method 1 and Method 2. Note that Method 2 succeeded in recognizing small features such as the two screws on the left side of the assembly.

[4] Ho, C.C.; Wu, F.C.; Chen, B.Y.; Chuang, Y.Y.; Ouhyoung, M.: Cubical marching squares: Adaptive feature preserving surface extraction from volume data. Computer Graphics Forum, 24(3), 537–545, 2005. http://doi.org/10.1111/j.1467-8659.2005.00879.x.

[5] Kanai, S.; Iyoda, D.; Endo, Y.; Sakamoto, H.; Kanatani, N.: Appearance preserving simplification of 3d cad model with large-scale assembly structures. International Journal on Interactive Design and Manufacturing (IJIDeM), 6(3), 139–154, 2012. http://doi.org/10.1007/s12008-012-0145-0.

[6] Kay, D.S.; Greenberg, D.: Transparency for computer synthesized images. SIGGRAPH Computer Graphics, 13(2), 158–164, 1979. http://doi.org/10.1145/965103.807438.

[7] Kobbelt, L.P.; Botsch, M.; Schwanecke, U.; Seidel, H.P.: Feature sensitive surface extraction from volume data. In Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '01, 57–66. ACM, New York, NY, USA, 2001. http://doi.org/10.1145/383259.383265.

[8] Kuo, C.C.; Yau, H.T.: A new combinatorial approach to surface reconstruction with sharp features. IEEE Transactions on Visualization and Computer Graphics, 12(1), 73–82, 2006. http://doi.org/https://doi.org/10.1109/TVCG.2006.2.

[9] Kwon, S.; Kim, B.C.; Mun, D.; Han, S.: Simplification of feature-based 3d cad assembly data of ship and offshore equipment using quantitative evaluation metrics. Computer-Aided Design, 59, 140 – 154, 2015. http://doi.org/https://doi.org/10.1016/j.cad.2014.03.003.

[10] Lorensen, W.E.; Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. SIGGRAPH Computer Graphics, 21(4), 163–169, 1987. http://doi.org/10.1145/37402.37422.

[11] Qian, J.; Zhang, Y.: Sharp Feature Preservation in Octree-Based Hexahedral Mesh Generation for CAD Assembly Models, 243–262. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. http://doi.org/10.1007/978-3-642-15414-0_15.

[12] Wei, J.; Lou, Y.: Feature preserving mesh simplification using feature sensitive metric. Journal of Computer Science and Technology, 25(3), 595–605, 2010. http://doi.org/10.1007/s11390-010-9348-7.

[13] Yu, J.f.; Xiao, H.; Zhang, J.; Cheng, H.; Xin, B.: Cad model simplification for assembly field. The International Journal of Advanced Manufacturing Technology, 68(9), 2335–2347, 2013. http://doi.org/10.1007/s00170-013-4850-z.