International Journal of Engineering and Geosciences
https://dergipark.org.tr/en/pub/ijeg
e-ISSN 2548-0960

Multithreaded wedge detection method on triangular 3D CAD objects using mesh traversal method

Özkan Kırık*[1], Caner Özdemir[1]

[1]Mersin University, Engineering Faculty, Department of Electrical-Electronics Engineering, Mersin, Turkey

Keywords
Wedge Detection
Triangular Mesh
CAD
Mesh Traversal

ABSTRACT

In this study, a multithreaded method for triangular mesh three-dimensional computer aided design objects is proposed to detect and extract wedges. Wedge detection is time consuming process for such objects that have large number of facets. To take the advantage of parallel computing opportunities, the algorithm is refactored in this study. Scope of variables, memory management and stack use are optimized for efficient use of computational resources. The proposed method is focused to calculation efficiency and performance on multicore / multithreaded processors and it is evaluated with benchmark, complex and realistic objects.

1. INTRODUCTION

Computer aided design (CAD) objects are widely used in nearly all engineering fields. Triangular mesh modelling is used mostly due to its simplicity and efficiency (Kim et al. 2009). Triangular meshes consist of Cartesian coordinates of points called vertices and collection of triangle corner vertices called facets. Wavefront obj file format is preferred for storing triangular mesh data (Possemiers and Lee 2015). File format of the wavefront obj is shown in Figure.1. For vertex definition, the line should start with a letter “v” and three numbers separated by spaces are x, y, and z Cartesian coordinates respectively should be followed. Afterwards, three-dimensional (3D) facet definition lines are listed with a letter “f” and followed by 3 vertex indexes delimited with a space character. It should be further noticed that the order of vertex indexes is important for calculating surface normal direction. The wavefront obj definition shown in Fig.1, constructs a 2D plate that its center is at origin and has a dimension of 15 x 15. The unit is not stored in wavefront obj file.

Wedge outline of a triangular mesh is a very important feature that used by many calculation techniques such as ray traversal diffraction and physical theory of diffraction methods (Kirik and Ozdemir 2019; Griesser and Balanis 1987; Pyotr 2014). Calculation of wedges of CAD object is a time and processor consuming process for objects that have large number of facets (Kuo et al. 2009; Sun et al. 2002; Wang et al. 2012; Zhang et al. 2003).

```
# Vertex Definitions
v 0.000000 7.500000 -7.500000
v 0.000000 -7.500000 -7.500000
v 0.000000 -7.500000 7.500000
v 0.000000 7.500000 7.500000

# Facet Definitions
f 2 1 4
f 3 4 2
```

Figure 1. Wavefront obj file format

Parallel computation is an important opportunity that provides efficient use of computational resources. Modern central processing units (CPU) have many cores within single package that provides many computational resources even for personal use.

The purpose of this study is to detect wedges of triangular mesh CAD models using parallel computation techniques. Other techniques are using sequential scan principles which is a single threaded operation for detecting wedges. For this paper, 4 benchmark and 2 complex objects' wedge detection is performed using the purposed method.

* **Corresponding Author**

*(ozkan@mersin.edu.tr) ORCID ID 0000-0002-6996-6239
(cozdemir@mersin.edu.tr) ORCID ID 0000-0003-2615-4203

Cite this article

Kirik O & Ozdemir C (2022). Multithreaded wedge detection method on triangular 3D CAD objects using mesh traversal method. International Journal of Engineering and Geosciences, 7(2), 174-178

Research Article / DOI: 10.26833/ijeg.947235 **Received: 03/06/2021; Accepted: 25/06/2021**

2. METHOD

Since the conventional solutions are single threaded, multithreaded mesh traversal is needed and implemented throughout this study.

2.1. Single threaded traversal of triangular mesh

Fully closed 3D CAD models defines a volume. But open-3D CAD models defines a surface instead of volume. Therefore, object wedge detection process was studied within the next two subsections: wedge detection for volumes and wedge detection for surfaces.

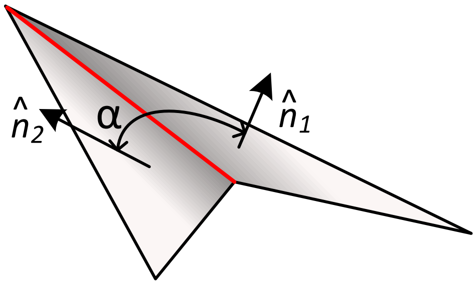

Figure 2. Neighboring triangular facets normal vectors and definition of wedge angle

The α angle shown in Fig. 2 between normal vectors of two neighboring facets which are sharing same edge is called the wedge angle. The α angle defines an important information about sharpness of edge. When the α value is 0 °, the facets are located on the same plane.

2.1.1. Wedge detection approach for volumes

The implementation of deciding whether two triangular facet contains a wedge or nor is given below:

i. Read three vertex indexes of the triangular facet from the CAD file,
ii. Find intersections of these two selected facets' members.
iii. If the intersections result;
   a. contain three elements then the facets are basically the same. The second facet should be removed for simplicity.
   b. contain two elements then the facets are neighbors.
   c. else continue to the next iteration till the end of file.

Found neighborhoods between facets should be checked if the angle between the facet normal vectors is larger than a user-defined threshold value. This threshold value can be changed from application to application. For electromagnetic solvers that uses edges/wedges for diffraction calculations; for instance, the wedge angle usually taken at least 30° since diffraction coefficients do not contribute significant energy to the total scattered field if the wedge angle is smaller than this value (Kirik and Ozdemir 2019; Griesser and Balanis 1987). If the angle is greater than this user-defined value, the intersection is decided to be a wedge so that an edge line should be calculated that is to be used in the selected application.

2.1.2. Wedge detection approach for surfaces

In addition of 3D volumetric wedge detection problem, there is also a need to find the edges for two-dimensional (2D) surfaces. This is because of the fact that some CAD file definitions have outside of surfaces that do not have any neighboring facets such as wing and air-flaps that are constituted of 2D plates. For example, an 2D plate triangular mesh like shown in Fig. 2, the most outsite rectangle edges represent surface wedges which don't have any neighbor facets.

Both of the volume and surface wedge detection approach used together in numerical examples section.

2.2. Multicore / Multithreaded Traversal

Instead of sequentially scanning of collection of facets, the methods described at section 2.1.1 and 2.1.2 are implemented as scalable to multicore and multithreaded computing architectures.

Scalable architecture, depends on worker threads that created at the startup step. The number of worker threads are decided according to the number of available hardware resources. The started worker threads, polls tasks from job queue. Spinlock mechanism is also used to prevent race conditions (Crummey and Scott 1991).

3. NUMERICAL EXAMPLES

The above method was implemented in Microsoft .NET framework with C# language. Both sequential mesh traversal and multithreaded mesh traversal was implemented.

Numerical experiments are accomplished against four benchmark and two complex 3D triangular meshes listed in Table 1. Each facet represents a triangle that consists of 3 vertices. The minimum wedge angle is selected as 30° in all experiments.

Table 1. Numerical experiments for wedge detection

Type	CAD	Vertices	Facet Count	Wedges Found
Benchmark	2D Plate	4	2	4
Benchmark	Dihedral	6	4	7
Benchmark	Cube	24	12	12
Benchmark	Cone-sphere	802	1600	0
Complex	Backhoe Loader	4618	9387	5850
Complex	T72M1 Tank	69042	137384	32659

3.1. Benchmark Object #1: 2D Plate

To examine the purposed method for a surface and validate the detected wedges, a 2D plate object chosen as the first example.

The 2D plate object's dimensions are 15cm x 15cm. Center of the 2D plate shown in Fig. 2 is aligned to be centered at the origin of (0,0,0). As listed in Table.1, this benchmark object consists of 4 vertices and 2 facets.

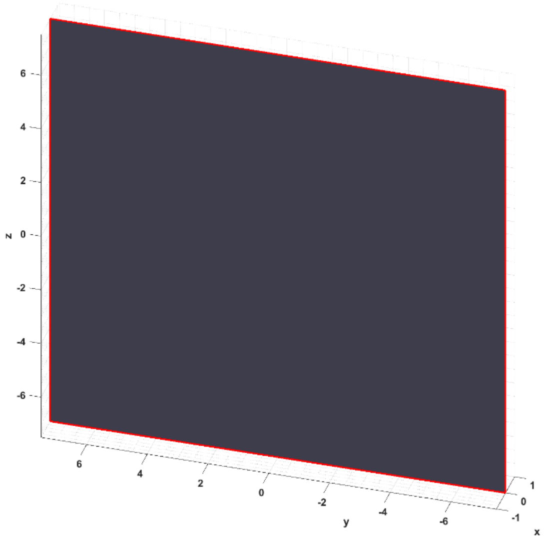

Figure 3. Wedge Detection Result for 2D Plate Object

Result of the method is shown in Fig. 3 with four red lines. Only outside of the rectangle is red colored as expected. The diagonal line that corresponding the neighboring line between the triangular facet is not red colored because the angle between their surface normals is equal to 0° which is not greater than the threshold value of 30°. Execution times are measured as 0.053 ms for the sequential mesh traversal calculation and 0.292 ms for the multithreaded mesh traversal calculation.

3.2. Benchmark Object #2: Dihedral Corner Reflector

As another benchmark object a Dihedral corner reflector model is chosen for the different assessment. This object is built to be a Wavefront obj file with six vertices and four facets in total.

As shown in Fig. 4, the corner edge is highlighted with red color in addition to perimeter of the rectangles as expected. Execution times are measured as 0.075 ms for the sequential mesh traversal and 0.292 ms for the multithreaded mesh traversal as listed in Table 2.

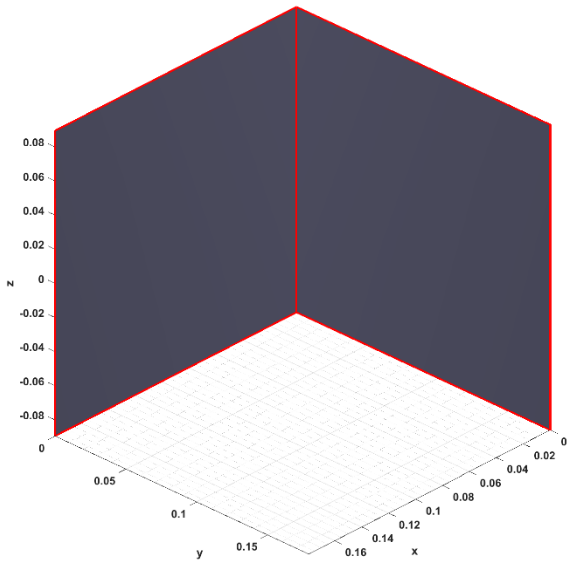

Figure 4. Wedge Detection Result for Dihedral Corner Reflector Object

3.3. Benchmark Object #3: Cube

Another benchmark target is selected to be a cube that has the dimension of 16 cm x 16 cm x 16 cm as shown in Fig. 5. It is aligned to be centered at the origin of the coordinate system. This target is used to validate the volume wedge detection of a closed 3D triangular mesh. This particular triangular mesh contains twenty-four vertices and twelve facets.

After applying the proposed approach given above, the total number of detected wedges is found to be twelve as shown in Table.1 which is the correct result. Execution times of the algorithms are measured as 0.083 ms for the sequential mesh traversal process and 0.295 ms for the multithreaded mesh traversal process as listed in Table 1.

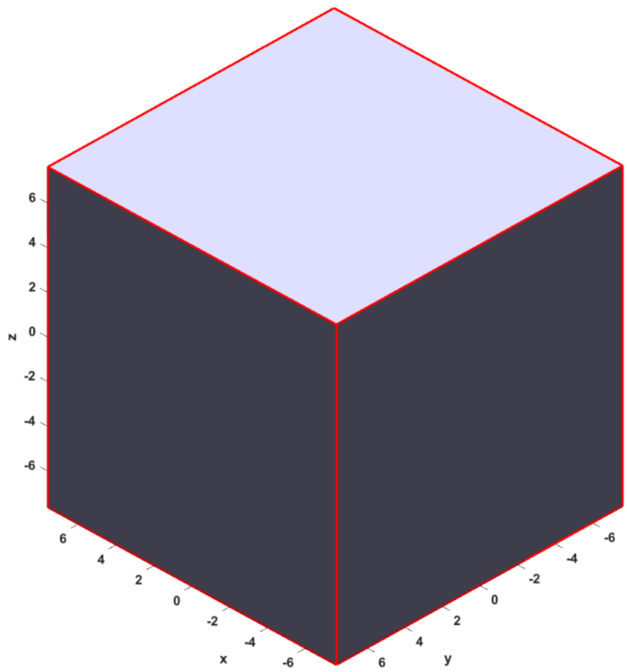

Figure 5. Wedge Detection Result for Cube Object

3.4. Benchmark Object #4: Cone-sphere

Cone-sphere benchmark target has no neighbor facets which exceeds the angle threshold. So this experiment is also important to validate the correctness of the proposed wedge detection method. The cone-sphere structure shown in Fig. 6 with the half-sphere part with a radius of 74 mm and the cone part with the height of 605 mm (Griesser et al. 1989).

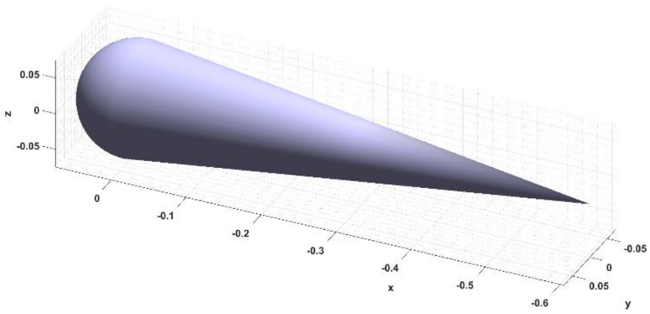

Figure 6. Cone-sphere Object Wedge Detection Result

After applying the proposed method, there is no detection of wedges listed at the end as expected. Therefore, we see no red lines or curves in Fig. 6. Run times for this particular object are measured as 329 ms for the sequential mesh traversal calculation and 115 ms for the multithreaded mesh traversal calculation.

3.5. Complex Object #1: Backhoe Loader

To implement the proposed method on a more realistic complex-shaped object, a Backhoe loader model is chosen whose CAD file can be viewed in Fig. 7. This wavefront obj CAD model has a detailed design that is constituted via 4618 vertices and 9387 facets. The dimensions of the target are 6.33 m (length), 1.99 m (width) and 2.33 m (height) (Demirci et al. 2020).

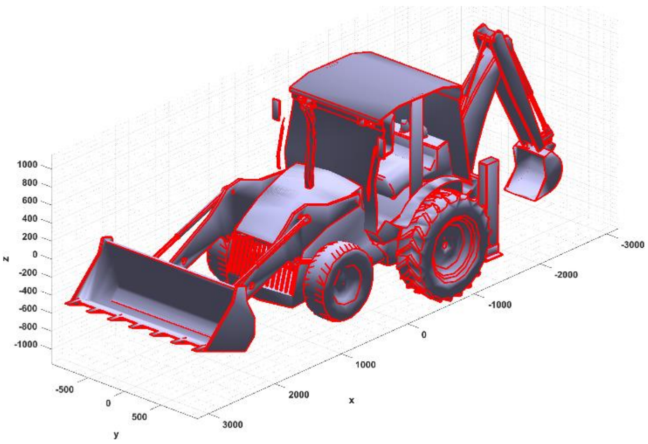

Figure 7. Wedge Detection Result of Backhoe Loader Object

After applying the proposed wedge detection algorithm to this complex object, a total of 5850 different wedges are detected. As can be fairly seen from Fig. 7, all the joint and sharp lines are found and highlighted with good fidelity. Execution times are measured as 4.007 s for the sequential mesh traversal process and 1.33 s for the multithreaded mesh traversal process.

3.6. Complex Object #2: T72M1 Tank

For the last example, a much more complex object that is a tank model is chosen to comprehend the effectiveness of the proposed method. The CAD model of T72M1 tank is depicted in Fig. 8. This triangular mesh model is highly detailed and complex such that a total of 137384 facets are used to form the structure (Demirci et al. 2020). The size of this tank model is 10.4 m in length, 3.74 m in width and 2.76 m in height. It is no doubt that this model is much more complex when compared to previous backhoe loader object.

After applying the proposed wedge detection method, the wedges are extracted perfectly as there are pointed out red lines in Fig. 8. The total execution times are measured as 1439 s for the sequential mesh traversal calculation and 501 s for the multithreaded mesh traversal calculation.

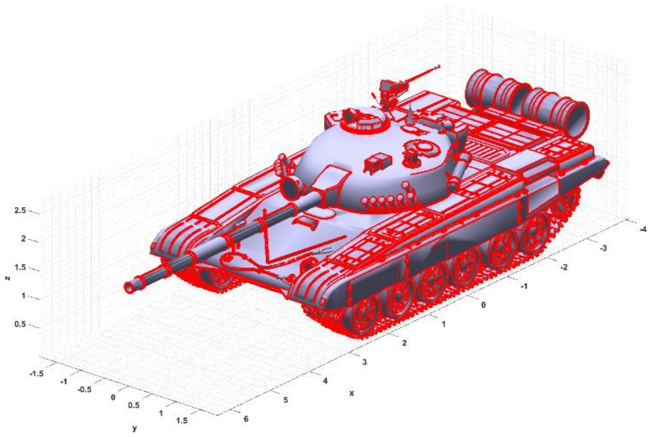

Figure 8. T72M1 Tank Object Wedge Detection Result

Table 2. Execution Times of the Conducted Experiments

Experiment	Facet Count	Seq. Trav. Time	Multi Thr. Trav. Time	Speed-up Ratio
2D Plate	2	53 ns	292 ns	0,18x
Dihedral	4	75 ns	292 ns	0,25x
Cube	12	83 ns	295 ns	0,28x
Cone-sphere	1600	329 ms	115 ms	2,86x
Backhoe Loader	9387	4 s	1.33 s	3,01x
T72M1 Tank	137384	1439 s	501 s	2,87x

4. CONCLUSION

In this study, we have presented an assessment of multithreaded wedge detection method that can be applied universally to any triangularly meshed CAD object from benchmark CAD models to complex-shaped models. The first four experiments, applied to benchmark objects, have been utilized to validate the base functionality of both surface and volume wedge detection approached. Then, much more complex objects were chosen to assess the availability and the effectiveness of the proposed method.

Execution times of all experiments are listed in Table 2. In fact, the mesh traversal process has the order of N^2 complexity. So the execution time increases quadratic manner by total number of facets, N. In the multithreaded mesh traversal method, the main goal is to decrease the execution time of the process by the use of parallel computing opportunity. On the other hand, the main drawback of this method is the initialization cost that independent from the number of facets. The initialization cost of multithreaded applications has the order of N complexity. It is also important to mention that it also depends on number threads to start as expected.

The multithreaded execution times of the 2D plate, the dihedral corner reflector and the cube objects are almost the same although total number of facets are different. This is because the thread initialization cost is the same for all these three experiments. Sequentially mesh traversal is much faster than multithreaded mesh traversal when the facet count of object is relatively very small. As a solution, usage of multithreaded traversal should be decided according to facet count of object. A good speed-up values have been achieved using this method for large and complex-shaped triangular meshed models as obvious from Table 2.

ACKNOWLEDMENT

This work was supported in part by the Mersin University Scientific Research Unit under Grant 2015-TP3-1160.

Author contributions

Barnali Das: Conceptualization, Methodology, Analysis, Mapping, Compilation, Writing-Original draft and its preparation. **Anargha Dhorde:** Visualization, Checking draft and Editing.

Conflicts of interest

The work is original and authors declare no conflicts of interest.

REFERENCES

Crummey J M, Scott M L (1991). Algorithms for scalable synchronization on shared-memory multiprocessors. ACM Transactions on Computer Systems, 9(1), 21–65.

Demirci S, Kirik O, Ozdemir C (2020). Interpretation and Analysis of Target Scattering from Fully-Polarized ISAR Images Using Pauli Decomposition Scheme for Target Recognition. IEEE Access, 8, 155926-155938.

Griesser T, Balanis C A (1987). Backscatter analysis of dihedral corner reflectors using physical optics and the physical theory of diffraction. IEEE Transactions on Antennas and Propagation, 35(10), 1137-1147.

Griesser T, Balanis C A, Liu K (1989). RCS analysis and reduction for lossy dihedral corner reflectors. Proceedings of the IEEE, 77(5), 806-814.

Kim H S, Choi H K, Lee H K (2009). Feature detection of triangular meshes based on tensor voting theory. Computer-Aided Design, 41(1), 47-58.

Kirik O, Ozdemir C (2019). An Accurate and Effective Implementation of Physical Theory of Diffraction to The Shooting and Bouncing Ray Method via Predics Tool. Sigma Journal of Engineering and Natural Sciences, 4, 1153-1166.

Kuo Ct, Cheng Sc, Wu Dc, Chang Cc (2009). A Blind Robust Watermarking Scheme for 3D Triangular Mesh Models Using 3D Edge Vertex Detection. Asian Journal of Health and Information Sciences, 4(1), 36-63.

Possemiers A L, Lee I (2015). Fast OBJ file importing and parsing in CUDA. Computational Visual Media, 1, 229-238.

Pyotr Ya. Ufimtsev (2014) Fundamentals of the Physical Theory of Diffraction, Second Edition. John Wiley & Sons. ISBN: 9781118753668

Sun Y, Page D L, Paik J K, Koschan A, Abidi M A (2002). Triangle mesh-based edge detection and its application to surface segmentation and adaptive surface smoothing. International Conference on Image Processing, 825-828 vol.3, Newyork, USA.

Wang Xc, Cao Jj, Liu Xp. (2012). Feature detection of triangular meshes via neighbor supporting. Journal of Zhejiang University Science C, 13, 440–451.

Zhang X, Zhou M, Geng G (2003). A Method of Detecting the Edge of Triangular Mesh Surface. China Knowledge Network (CNKI) China Academic Journal.