Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 9, Issue. 6, June 2020, pg.55 – 64

3D OBJECTS OCCLUSION ESTIMATION USING MONTE CARLO SIMULATION

Mustapha Musa[1]; Isiaka Shuaibu[2]; Bello Muhammad Zaidu[3]
School of Computer Science and Technology & Changchun University of Science and Technology, China
[1] 2018300036@mails.cust.edu.cn, [2] isiakashuaibu@yahoo.com, [3] bellozaid@yahoo.com

Abstract: Monte Carlo simulation is applied in many fields of science and remains a useful tool for complex testing experiments under uncertain conditions. 3D object occlusion estimation remains the most challenging task in the field of computer vision, with multiple impacts on autonomous driving, robotics, pedestrian detection and virtual reality. In this paper, we presented experiments with five groups of arbitrary 3D objects in Euclidean Space and estimated the area of the occluded region between 3D objects using the Monte Carlo method. The objects vertices data points were obtained from [CUST] and the simulation was carried out with MATLAB R2019b.

Keywords: occlusion detection, simulation, point-cloud, occlusion handling

1. INTRODUCTION

The Two-Dimensional(2D) shapes (square, circle, rectangle, triangle and polygons) are on a flat plane represented by mutually perpendicular axes of x and y axes. The length, area and volume of these shapes are relatively easy to calculate compared to the three-dimensional (3D) objects. 3D objects are the solid shapes that we interact within the real world; it has height, width and depth represented in(x, y, z)axes[1]. Object Occlusion can be defined mathematically as a discontinuity in a smoothed function of a surface. The goal of occlusion detection is to determine whether a significant discontinuity (within the discrete domain) is present in adjacent regions. Occlude detection has many applications in engineering simulation, robotics and computer vision such as real-time rendering, virtual reality, medical-tissue modelling and 3D printing[2]. Various approaches for detecting an occlusion in three-dimension (3D) objects in the Euclidean space proposed. The detection of actively occluded objects is still challenging. Reddy *et al.* [3] introduced Occluded net a framework for predicting 2D and 3D locations of occluded object key points with the aid of MaskRCNN, which has been trained only on visible key point annotations. A graph encoder network, then explicitly classifies invisible edges and a graph decoder network corrects the occluded key point locations from the first detector. Jun *et al.* [4] have designed a 3D object detection framework on the point cloud using the Backbone Network to fuse features of the lower level with high-level features. The car, cyclist and pedestrian detection works have experimented, and the results are efficient with average accuracy. Georgis et al. [5] proposed a method that combines planar and solid occluders using a unified selection approach for dynamic environments on precalculated visibility data. The algorithm applied to deploy virtual reality systems commercially, and results provided from actual virtual reality shows.

Wang et al. [6] also presented an approach for recognition and localization of occluded apples using the K-means clustering algorithm and convex hull. The experimental results show that the method could reach a much better location rate than the

Hugh transformation method and contour curvature method, so an efficient way of recognizing and locating occluded apples concluded.

This paper presents an estimate of the occlusion of 3D objects using the Monte Carlo method. This technique is efficient and applied to many of the quantitative problems in science [7].

2. GENERAL MONTE CARLO METHOD STAGES

Simulations stages can vary depending on the number of factors involved. As described in [8] at a basic level, all Monte Carlo simulations have four simple steps as follows:

1. **Identify the Transfer Equation**
   The Monte Carlo simulation requires a quantitative model of the business activity, plan or process explored. The mathematical expression process is called the "transfer equation." It may be a known engineering or business formula, or it based on a model of a designed experiment (DOE) or a regression analysis.
2. **Define the Input Parameters**
   For each factor in the transfer equation, determine how its data distributed. Inputs may be in a uniform distribution, normal distribution, or triangular. Determine distribution parameters for each input.
3. **Create Random Data**
   Create a vast, random data set for each input to perform an accurate simulation. Such random data points represent the values seen for each input over a long period.
4. **Simulate and Analyze Process Output**
   Using the simulated data in place, use the transfer equation to calculate the simulated results. The fact that the model runs sufficiently simulated input data should provide a reasonable indicator of the performance of the method over time, despite the expected variability in inputs.

2.1 SAMPLING OF RANDOM VARIABLE

A sample$(X^1, \dots, X^N)$ is a measurable function from Ω_1 to Ω_2^N A realization of this sample is the value taken by the function at $\in \Omega_1$ $(x^1, \dots, x^N) = \left(X^1(\omega), \dots, X^N(\omega)\right)$. In Monte Carlo estimation, we prove the unbiasedness of the estimator by considering the samples $(X^n)_{n=1}^N$ to be the functions. If an expectation of a random variable X defined as

$$\mathbb{E}[X] = \int_{\Omega_1} X(\omega_1) dP(\omega_1)$$

and assuming that X^n are functions and $X^n = X$, we can proceed as follows:

$$\mathbb{E}\left[\frac{1}{N}\sum_{n=1}^{N} f(X^n)\right] = \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}[f(X^n)]$$

$$= \frac{1}{N}\sum_{n=1}^{N} \text{o}E[f(X)]$$

$$= \mathbb{E}[f(X)].$$

2.2 MONTE CARLO INTEGRATION

Monte Carlo is a statistical integration technique based on taking random samples from the domain of an integrand and using them to estimate the value of an integral[9]. The standard Monte Carlo estimator gives the value of an integral $\int_D f(x)dx$ over some domain D as the expected value.

$$x = \frac{1}{N}\sum_{i=1}^{N} \frac{f(Xi)}{p(Xi)},$$

Where X_i are random variables drawn from a sampling distribution $p(x)$ Over D. This distribution must be non-zero for all x where $f(x) > 0$. Monte Carlo methods involve randomness source. For convenience, random numbers delivered as a stream of independent $U(0,1)$ random variables. Integration of a one-dimensional function f(x) from a to b can be defined as:

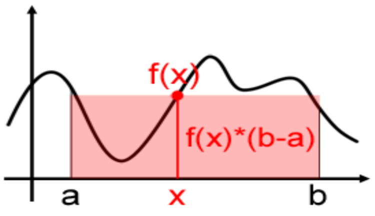


Fig. 1 Area of Integration

The curve evaluated at x, and the result can be multiplied by (b - a). This defines a rectangle which seen as a very crude approximation of the integral. As demonstrated in [10], integration can be done using the reject and accept method. Let I be the area

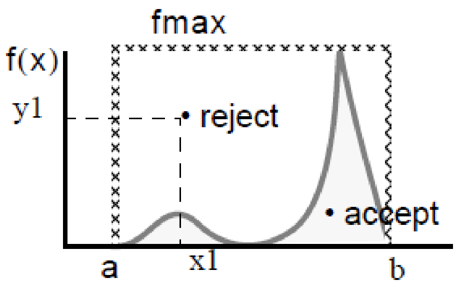


Fig. 2 Integration Using Accept and Reject Method

$I = \int_a^b f(x)\ dx$ – area under the function $f(x), R = (b-a)f_{max}$ – area of a rectangle.

$P = \frac{I}{R}$ is a random point probability that it lies under $f(x)$, thus $I = RP$

Step 1: Choose a random point$(x1, y1)$: $x1 = a + (b-1)\xi$ and $y1 = f_{max}\xi_2$
Step 2: check if $y1 \le f(x1)$- accept the point, $y1 < f(x1)$ – reject the point.
Step 3: Repeat this process N times, Ni- the number of accepted points.
Step 4: determine $P = \frac{Ni}{N}$ and the value of integral $I = R\frac{Ni}{N}$
The probability distribution functions usually have two properties[11]:

1. They must be nonnegative on the whole support interval of the random variable.
2. They must sum or integrate to 1.

3. PROBLEM DEFINITION

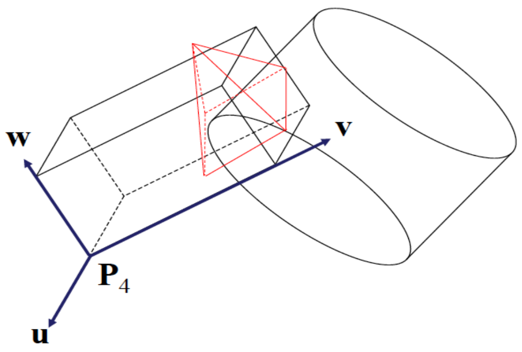


Fig. 3 Occluded 3D shapes comprises of rectangular box, Cylinder and Square pyramid

In a 3D space, there is a rectangular box, a cylinder and a square pyramid with arbitrary sizes, locations and orientations that may or may not intersect with each other. If intersected, the volume of their intersection is expressed as an integration over the occluded 3D objects. Otherwise, their intersection volume will be zero. The rectangular box described by its eight vertices $P1, P2, P3, \cdots, P8$ and the cylinder, is described by the two centre points **C**1, and **C**2 (top and bottom faces) with radius R. A square pyramid is described by five edges $E0, E1, E2, E3, E4$.

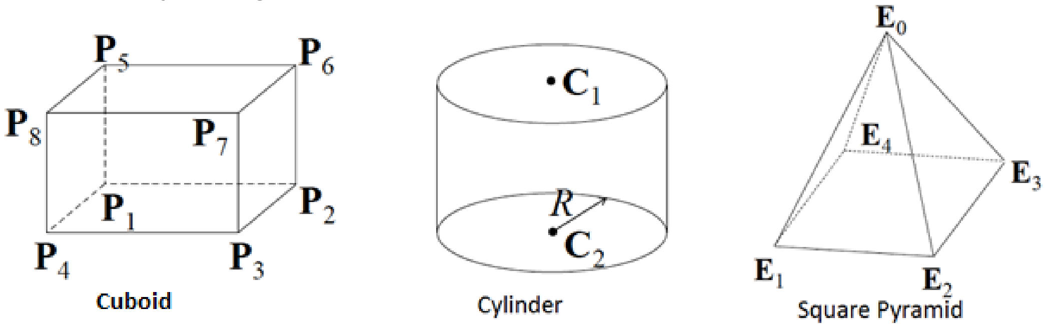


Fig. 4 Collections of 3D shapes with their vertices defined

2.1 RANDOM POINTS SAMPLING

a. Cuboid: A cuboid is a 3D shape with six faces forming a convex polyhedron, the cuboid face can be any quadrilateral, and the cuboid is composed of six rectangles, placed at right angles. It has twelve edges, six sides and eight angles or vertices, the vertices are $P1, P2, P3, \cdots, P8$. l = length, h = height and b = breadth. To get the breadth, length and height of the cuboid, we need to use the Euclidean distance equation to calculate the distance between two points by using the vertical points.

$$d(P_i, P_j) = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2}$$

$$d = \sqrt{x^2 + y^2 + z^2}$$

$$l = d(P_4, P_1)$$

$$h = d(P_5, P_1)$$

$$b = d(P_2, P_1)$$

$$P(x, y, z) = \begin{pmatrix} x_1 \times l \\ \mathrm{x}_2 \times h \\ \mathrm{x}_3 \times b \end{pmatrix}$$

$$P = [x + y + z] + P1$$

$x_1, x_2, x_3 \in [0,1]$ are random points.

The volume of a cuboid $v = l \times b \times h$

b. Cylinder A right circular cylinder is a three-dimensional object where the two ends are circles. The height 'h' is the perpendicular distance between the bases when calculating the volume of an oblique cylinder, it is essential to use the perpendicular height, or 'altitude ' radius 'r' of a cylinder is the base radius. Cylindrical coordinates are defined as follows.

$$P(x, y, z) = \begin{pmatrix} \sqrt{x} \times r\cos(\theta) \\ \sqrt{x} \times r\sin(\theta) \\ \sqrt{x} \times h \end{pmatrix}$$

$r \geq 0, \theta \in [0, 2\pi y]$, $h \in R$. $x, y \in [0,1]$ are random points

The volume of cylinder $v = \pi r^2 h$

Where $h = \frac{v}{\pi r^2} \equiv d(C_1, C_2)$.

The cylinder height is the distance between two bases of the cylinder.

c. **Pyramid** A regular pyramid is a right pyramid whose base is a regular polygon, a pyramid with a square base is called a square pyramid[12]. In general, the surface area of a pyramid is the sum of the areas of all pyramid faces.

$$P(x,y,z)=\begin{pmatrix} -\sqrt[3]{\alpha}+\alpha\times\sqrt[3]{\alpha}\times l \\ -\sqrt[3]{\alpha}+\beta\times\sqrt[3]{\alpha}\times w \\ (1-\alpha)h \end{pmatrix}$$

$\alpha,\beta\in[0,1]$ random points.

$\boldsymbol{l}$ is the length, $\boldsymbol{w}$ is the width and $\boldsymbol{h}$ is the height of the pyramid respectively

Since five points $E0, E1, E2, E3, E4$. Describe the square pyramid, let $E1, E2, E3, E4$ be the surface of the pyramid. Such that $\boldsymbol{l}$ and $\boldsymbol{w}$ obtained by taking the normal of the appropriate vertices.

$$\boldsymbol{l}=\boldsymbol{d}(E1,\ E2)\quad \boldsymbol{w}=\boldsymbol{d}(E3,\ E4)$$

The volume of the pyramid

$$V=\frac{1}{3}b\times h$$

Where $b=l\times w$

3.2 ALGORITHM FOR EVALUATING OBJECT OCCLUSION

We choose a randomly generated point from the cuboid and check if the point falls on the volume of cylinder and pyramid, if the point lies within the volumes, accept the point otherwise rejects the point. The volume of the occluded area is the total number of points accepted, divided by the total area of the cuboid volume. The Accept-Reject method is a classical sampling method that allows one to sample a distribution that is difficult or impossible to simulate by inverse transformation [13].

a. **Check whether Cuboid point intersected with cylinder**

To check whether a point from the cuboid lies within the cylinder volume. Given C1 and C2 as the bases of the cylinder, R radius and P is the test point.

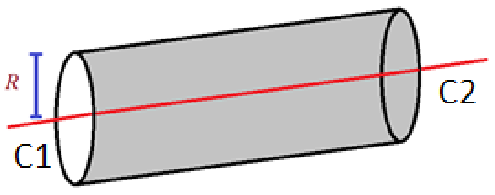


Fig. 5 Demonstrating boundary of Cylinder

First, we will check whether the points lie within the height of the cylinder.

$$(\vec{P}-\vec{C_1})\cdot(\vec{C_2}-\vec{C_1})\geq 0\ \&\&\ (\vec{P}-\vec{C_2})\cdot(\vec{C_2}-\vec{C_1})\leq 0$$

This will determine whether P is between the planes of the cylinder's two circular faces

$$\frac{|(\vec{P}-\vec{C_1})\times(\vec{C_2}-\vec{C_1})|}{(\vec{C_2}-\vec{C_1})}\leq R$$

The above equation confirms whether q lies inside the curved surface of the cylinder

b. **Check whether cuboid point intersected pyramid volume**

The pyramid parameters are E0 to E4; the base of the pyramid is E1 to E4, and the top-level is E0. let Pbase= E1 to E4 and Ph=E0; P as giving test points

$$k=\frac{(Pbase)\cdot(\vec{P}-\vec{Ph})}{\vec{ph}}$$

$$\text{if}\left(\overrightarrow{\text{P(z)}}\geq 0\text{and}\leq\overrightarrow{\text{Ph(z)}}\right)\&\&\left(\overrightarrow{\text{P(x)}}\geq \text{K(x)and}\leq\overrightarrow{\text{K(y)}}\right)\&\&\left(\overrightarrow{\text{P(y)}}\geq \text{K(z)and}\leq\overrightarrow{\text{Ph(x)}}\right)$$

Algorithm for Counting Intersection

```
1. for all the points i on cuboid do
2.      loop pyramid
3.           if intersect () then
4.                loop cylinder
5.                     if intersect () then
6.                          count intersect + i
7.           end
8. end
9. result = (count/N) *Cuboid Volume;
```

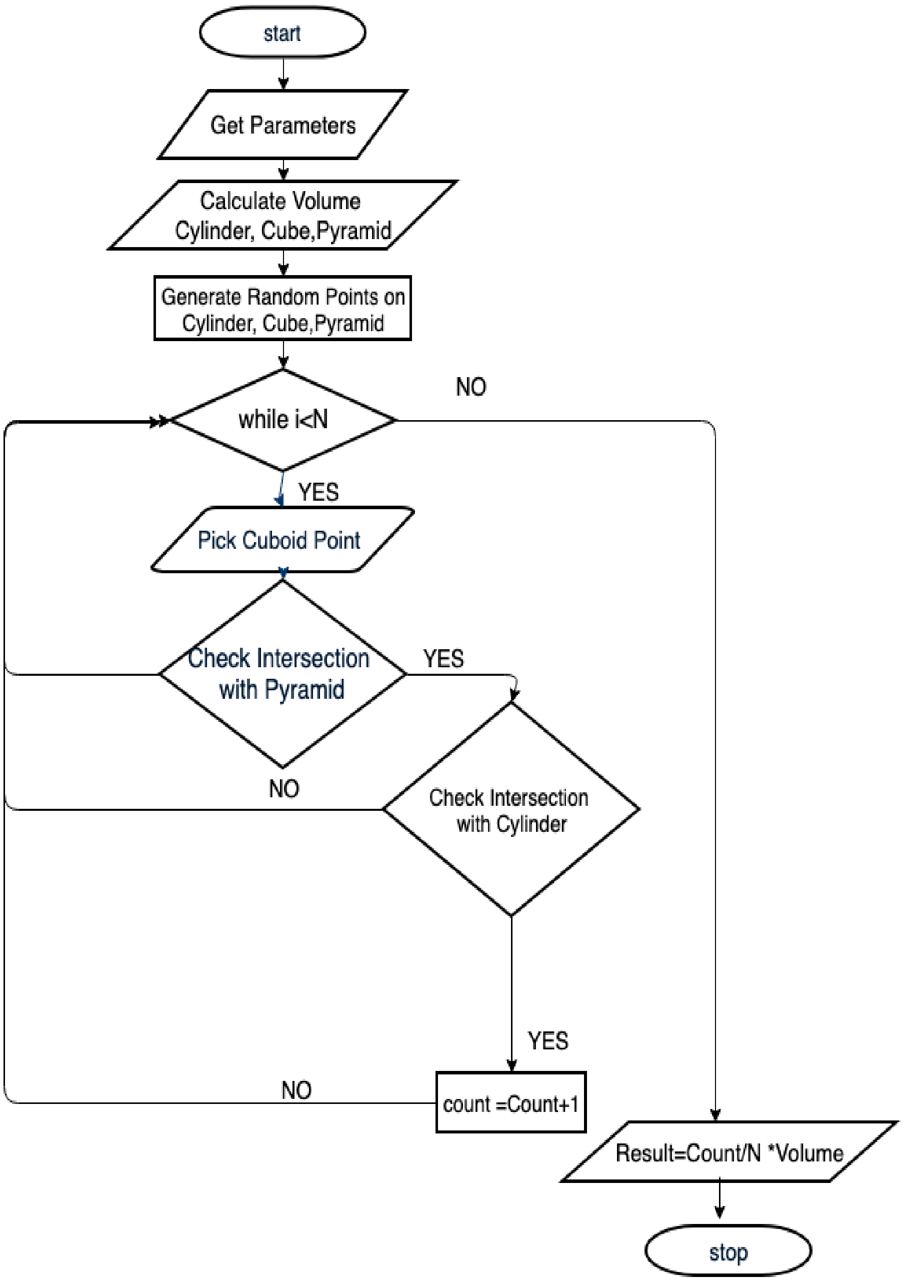

start
Get Parameters
Calculate Volume Cylinder, Cube,Pyramid
Generate Random Points on Cylinder, Cube,Pyramid
while i<N
NO
YES
Pick Cuboid Point
Check Intersection with Pyramid
YES
NO
Check Intersection with Cylinder
YES
NO
count =Count+1
Result=Count/N *Volume
stop

Figure 6 Flow Chart

TABLE 1

EXPERIMENT- Datapoints

The vertices data points of the 3D Shapes namely Cuboid, Pyramid and Cylinder

Objects	vertices	GROUP 1			GROUP 2			GROUP 3			GROUP 4		
		X-axis	**Y-axis**	**Z-axis**	**X-axis**	**Y-axis**	**Z-axis**	**X-axis**	**Y-axis**	**Z-axis**	**X-axis**	**Y-axis**	**Z-axis**
Cuboid	**P1**	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	P2	0.00E+00	1.00E+00	0.00E+00	3.31E-01	3.39E-01	8.81E-01	1.33E-02	8.67E-01	4.98E-01	2.92E-01	-1.84E-01	9.38E-01
	P3	1.00E+00	1.00E+00	0.00E+00	1.17E+00	6.69E-01	4.40E-01	1.01E+00	8.81E-01	4.48E-01	-6.60E-01	-1.57E-01	1.24E+00
	P4	1.00E+00	0.00E+00	0.00E+00	8.35E-01	3.31E-01	-4.40E-01	9.99E-01	1.33E-02	-4.98E-02	-9.53E-01	2.77E-02	3.02E-01
	P5	0.00E+00	0.00E+00	1.00E+00	4.40E-01	-8.81E-01	1.74E-01	4.98E-02	-4.98E-01	8.66E-01	8.18E-02	9.82E-01	1.68E-01
	P6	0.00E+00	1.00E+00	1.00E+00	7.71E-01	-5.42E-01	1.05E+00	6.30E-02	3.70E-01	1.36E+00	3.74E-01	7.98E-01	1.11E+00
	P7	1.00E+00	1.00E+00	1.00E+00	1.61E+00	-2.11E-01	6.14E-01	1.06E+00	3.83E-01	1.31E+00	-5.79E-01	8.26E-01	1.41E+00
	P8	1.00E+00	0.00E+00	1.00E+00	1.28E+00	-5.50E-01	-2.67E-01	1.05E+00	-4.84E-01	8.16E-01	-8.71E-01	1.01E+00	4.70E-01
Cylinder	**C1**	5.00E-01	5.00E-01	1.00E+00	1.02E+00	-5.46E-01	3.94E-01	5.56E-01	-5.72E-02	1.09E+00	5.70E-01	3.16E-01	1.04E+00
	C2	5.00E-01	5.00E-01	0.00E+00	5.83E-01	3.35E-01	2.20E-01	5.06E-01	4.40E-01	2.24E-01	4.21E-01	5.62E-01	7.94E-02
	R	5.00E-01			5.00E-01			1.00E+00			5.00E-01		
Pyramid	**E0**	0	0	1	0	0	1	0	0	1	0	0	1
	E1	1	1	0	1	1	0	1	1	0	1	1	0
	E2	-1	1	0	-1	1	0	-1	1	0	-1	1	0
	E3	-1	-1	0	-1	-1	0	-1	-1	0	-1	-1	0
	E4	1	-1	0	1	-1	0	1	-1	0	1	-1	0

3. SIMULATION RESULT

Using the 3D points vertices data points provided from Table 1, we ran two simulation tests using 200,000 and 300,000 uniform random points using the built-in MATLAB function rand () and fill the volume of 3D objects shown in Fig 7.

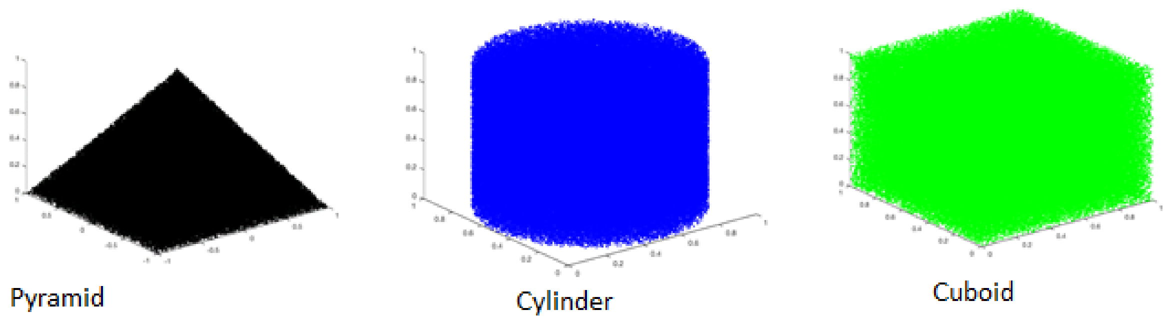
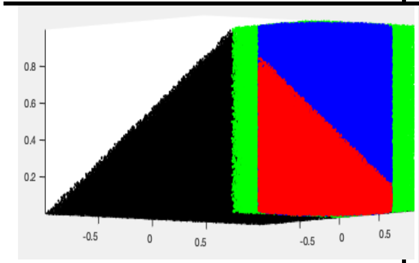


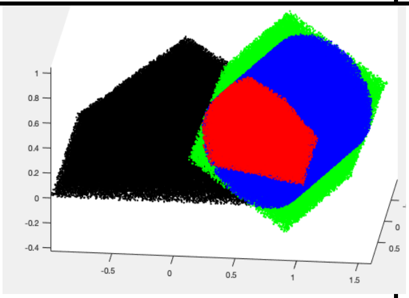
Fig. 6 3D objects filled with random points, we choose different color to distinguished the objects.

The experiment conducted on MacBook Pro Core-i7 2.2GHz CPU, 8 GB RAM and MATLAB software version 2019b, Fig 8 showing the estimated occluded part of the 3D objects in the four(4) groups.

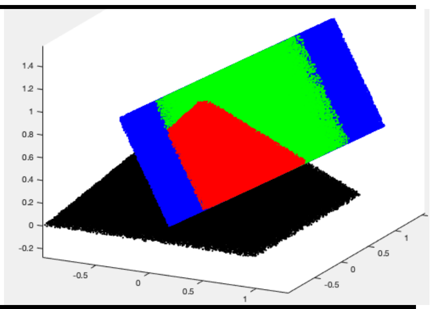
Group 1 = 0.27



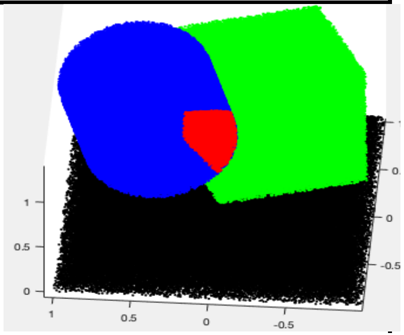
Group 2 = 0.21



Group 3 = 0.26



Group 4 = 0.03



Key	
🟥	Intersection
⬛	Pyramid
🟩	Cuboid
🟦	Cylinder

Fig. 7 occluded result, the red portion indicated the occlusion region between the 3d objects, the result of the occluded portion presented above each group

The source code is available in GitHub repository https://github.com/mustapha-musa/3d-occluder

TABLE 2 SIMULATION RESULTS

			Volumes		Intersected	
Group	N -sample	Cuboid	Cylinder	Pyramid	Points	Volume
1	200,000	1	0.7854	1.3333	54,446	0.2723
	300,000	1	0.7854	1.3333	82,568	0.2753
2	200,000	1	0.7854	1.3333	43,174	0.2159
	300,000	1	0.7854	1.3333	64,343	0.2145
3	200,000	1	0.7854	1.3333	51,574	0.2579
	300,000	1	0.7854	1.3333	77,851	0.2595
4	200,000	1	0.7854	1.3333	7,373	0.0359
	300,000	1	0.7854	1.3333	10,669	0.0356

4.1 RESULT ACCURACY VS COMPUTATION TIME

The Monte-Carlo simulation does not always yield the same result, and the result may slightly vary, as shown in the result table due nature of randomness. However, it is still the most reliable and practical method of evaluating measurement in practice under uncertain condition. The results converge only asymptotically as the number of trial sample increases. The error in a Monte Carlo computation is proportional to $\sigma/\sqrt{n}$. The accuracy depends on the number of values N that we use for the average[14].

We can reduce the error by increasing the number of samples. Note, however, that to reduce the error by half, we must increase the number of samples by a factor of four. Another way to improve the accuracy is to minimize the variance σ 2. A possible measure of the error is the variance σ^2 defined by

$$\sigma^2 = \langle f^2 \rangle - \langle f \rangle^2,$$

Where

$$\langle f^2\rangle = \frac{1}{N}\sum_{i=1}^{N} f(x_i)$$

and

$$\langle f\rangle^2 = \frac{1}{N}\sum_{i=1}^{N} f(x_i)^2$$

However, we should be expecting that the error will decrease with N points, and the sigma quantity will not decrease. Hence, this is not a suitable method for measuring errors. Imagine that we are executing several integrated measurements, each resulting I_n. These values obtained through N random numbers of different sequences. According to the central limit theorem, these values usually distributed around a mean $\langle I\rangle$. Suppose that we have a set of M such measurements I_n. A suitable way of measuring the differences of these measurements is the standard deviation of the means σM:

$$\sigma_M^2 = \langle I^2\rangle - \langle I\rangle^2,$$

Where

$$\langle I\rangle = \frac{1}{M}\sum_{n=1}^{N} I_n$$

and

$$\langle I\rangle^2 = \frac{1}{M}\sum_{n=1}^{N} I_n^2$$

it can be proven that

$$\sigma_M \approx \sigma/\sqrt{N}.$$

This relationship is accurate at the limits of a considerable number of measures. It is important to note that this expression indicates that the error decreases with the square root of the trials so that we need an average of 100 times more points if we want to reduce the error by a factor 10.

4. Conclusion

In this work, we estimated the occluded region in Euclidean space for the given arbitrary 3d objects. To achieve this, we generated and distributed random points to fill the volumes of giving 3d objects, we evaluate each generated point to determine whether it falls within the volume of all given objects. We evaluated the occlusion estimation efficiency using two separate random points sample categories. The result of the experiment varies slightly from 0.01% to 0.015%, due to the nature of the random number.

ACKNOWLEDGEMENT

I would also like to show our gratitude to Professor Chen Chunyi for sharing his pearls of wisdom with us during the course of this research, and also reviewers for their valuable comments 加油啊。

References

[1] D. P. Mukherjee and D. Jana, *Computer Graphics: Algorithms and Implementations*. PHI Learning, 2010.

[2] D. J. Bullock and J. S. Zelek, "Real-time tracking for visual interface applications in cluttered and occluding situations," *Image and Vision Computing*, vol. 22, no. 12, pp. 1083-1091, 2004/10/01/ 2004.

[3] N. D. Reddy, M. Vo, and S. G. Narasimhan, "Occlusion-Net: 2D/3D Occluded Keypoint Localization Using Graph Networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7318-7327.

[4] J. Xu, Y. Ma, S. He, and J. Zhu, "3D-GIoU: 3D Generalized Intersection over Union for Object Detection in Point Cloud," *Sensors (Basel),* vol. 19, no. 19, p. 4093, Sep 22 2019.
[5] G. Papaioannou, A. Gaitatzes, and D. Christopoulos, *Efficient occlusion culling using solid occluders*. 2006.
[6] D. Wang, H. Song, Z. Tie, W. Zhang, and D. He, "Recognition and localization of occluded apples using K-means clustering algorithm and convex hull theory: a comparison," *Multimedia Tools and Applications,* vol. 75, no. 6, pp. 3177-3198, 2015.
[7] D. P. Kroese, T. Brereton, T. Taimur, and Z. I. Botev, "Why the Monte Carlo method is so important today," *Wiley Interdisciplinary Reviews: Computational Statistics,* vol. 6, no. 6, pp. 386-392, 11/01 2014.
[8] P. Sheehy; and E. Martz. (2012, 26). *Doing Monte Carlo Simulation in Minitab Statistical Software*.
[9] M. M. Pharr, "Monte Carlo Solution of scattering Equations for Computer Graphics," PhD Dissertation, Department of Computer Science, Stanford University, 2005.
[10] J. L. Vujic, "Monte Carlo Sampling Methods," ed: Nuclear Engineering Department, University of California, USA, 2008.
[11] K. Lukacsy, "Generating Random Samples from User-defined Distributions," *The Stata Journal: Promoting communications on statistics and Stata,* vol. 11, no. 2, pp. 299-304, 01/01 2018.
[12] B. C. Leech, *Pyramids*. Rosen Publishing Group's PowerKids Press, 2007.
[13] C. Robert and G. Casella, *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.
[14] A. E. Feigin, "Phys 5870: Modern Computational Methods in Solids," Northeastern University 2012.