World Academy of Science, Engineering and Technology
International Journal of Computer and Systems Engineering
Vol:18, No:9, 2024

Development of Wave-Dissipating Block Installation Simulation for Inexperienced Worker Training

Hao Min Chuah, Tatsuya Yamazaki, Ryosui Iwasawa, Tatsumi Suto

Open Science Index, Computer and Systems Engineering Vol:18, No:9, 2024 publications.waset.org/10013823.pdf

***Abstract*—**In recent years, with the advancement of digital technology, the movement to introduce so-called ICT (Information and Communication Technology), such as computer technology and network technology, to civil engineering construction sites and construction sites is accelerating. As part of this movement, attempts are being made in various situations to reproduce actual sites inside computers and use them for designing and construction planning, as well as for training inexperienced engineers. The installation of wave-dissipating blocks on coasts, etc., is a type of work that has been carried out by skilled workers based on their years of experience and is one of the tasks that is difficult for inexperienced workers to carry out on site. Wave-dissipating blocks are structures that are designed to protect coasts, beaches, and so on from erosion by reducing the energy of ocean waves. Wave-dissipating blocks usually weigh more than 1 t and are installed by being suspended by a crane, so it would be time-consuming and costly for inexperienced workers to train on-site. In this paper, therefore, a block installation simulator is developed based on Unity 3D, a game development engine. The simulator computes porosity. Porosity is defined as the ratio of the total volume of the wave breaker blocks inside the structure to the final shape of the ideal structure. Using the evaluation of porosity, the simulator can determine how well the user is able to install the blocks. The voxelization technique is used to calculate the porosity of the structure, simplifying the calculations. Other techniques, such as raycasting and box overlapping, are employed for accurate simulation. In the near future, the simulator will install an automatic block installation algorithm based on combinatorial optimization solutions and compare the user-demonstrated block installation and the appropriate installation solved by the algorithm.

***Keywords*—**3D simulator, porosity, user interface, voxelization, wave-dissipating blocks.

I. Introduction

NOWADAYS, in order to solve the problem of labor shortages at construction sites, it has become an urgent issue to improve the efficiency of operations and construction processes. Therefore, introduction of ICT is progressing at construction sites. Besides, the utilization of three-dimensional (3D) data including the integration of the entire construction process, from surveying to design, construction and maintenance are also being popularized. This approach facilitates the rapid adoption and effective utilization of new technologies, methodologies, and materials, while also aligning with international standardization efforts. Since 2018, there has been a notable expansion in application of ICT in foundational and installation works within port facilities. In Japan, the Ministry of Land, Infrastructure, Transport and Tourism is actively promoting a project called i-Construction with the aim of increasing productivity at construction sites by 20% by 2025 [1].

Wave-dissipating blocks, crucial for dissipating wave energy and protecting coastal areas, are commonly installed in harbors and seawalls. The installation of breakwater blocks relies heavily on the empirical knowledge and experience of skilled workers. Currently, acquiring the necessary expertise entails repetitive trial-and-error installation attempts on-site, leading to significant time and cost constraints.

Mitsui et al. [2] have addressed the challenges by developing a block installation simulation method to facilitate the creation of three-dimensional digital models of breakwater structures. This method focuses on real-time performance and interactive operations but does not specifically address the evaluation of wave-dissipating block installation. In our research, we aim to construct a wave-dissipating block installation simulator. The simulator will replicate the process of installing wave-dissipating blocks to construct breakwater and provide functionality to determine how well the user is able to install the blocks by calculating the porosity of the breakwater.

II. Wave-Dissipating block and Breakwater Model

A. Wave-Dissipating Block

Wave-dissipating blocks come in various types, each tailored for specific applications. Among these, the tetrapod stands out as a commonly used example. For our study, we directed our focus towards the Shake Block shown in Fig. 1, a variant widely employed in Niigata, Japan, and supplied by Honma Concrete Industry Co., Ltd. This block type is available in a range of sizes, spanning from 0.5 tons to 50.0 tons. In our investigation, we specifically utilized the 4-ton variant. Detailed specifications for the Shake Block 4-ton type are provided in Table I.

TABLE I
Shake Block 4-ton Type Specifications

Mass (t)	Weight (kN)	Volume (m^3)	Surface Area (m^2)
3.991	39.138	1.735	9.796

Hao Min Chuah is with Faculty of Engineering, Niigata University, Niigata, Japan (e-mail: t20b345b@mail.cc.niigata-u.ac.jp).

Tatsuya Yamazaki is with Faculty of Engineering, Niigata University, Niigata, Japan (corresponding author, phone: +81- 25-262-6742; fax: +81- 25-262-6742; e-mail: yamazaki.tatsuya@ie.niigata-u.ac.jp).

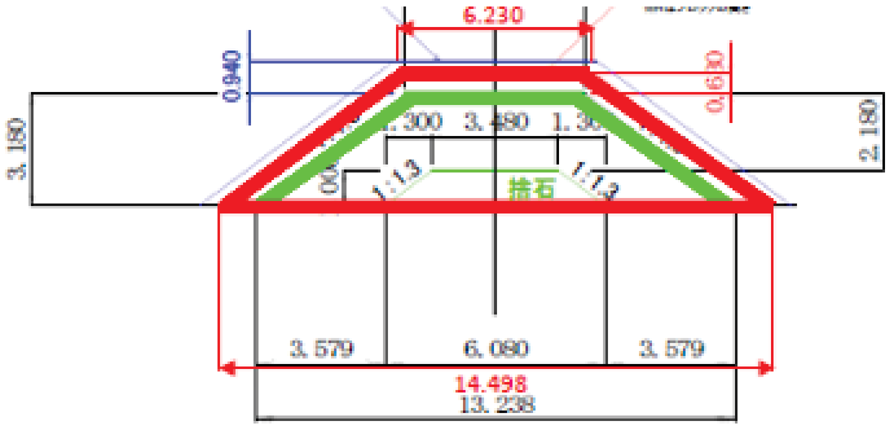
Ryosui Iwasawa and Tatsumi Suto are with Onogumi Co. Ltd., Niigata, Japan.

Fig. 1 Shake Block [3]

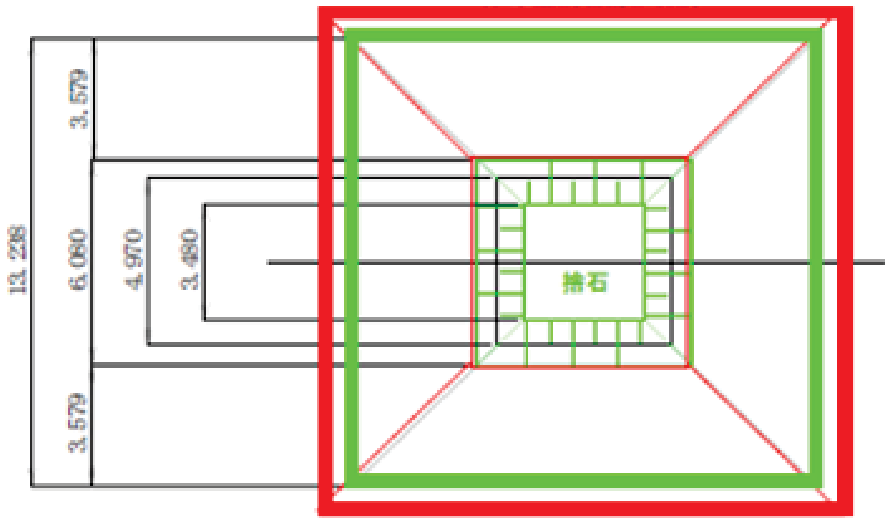
B. Breakwater

For the breakwater model, we utilized a virtual mini model of offshore breakwaters provided by Ono Construction Co., Ltd to facilitate experimentation. Offshore breakwaters are commonly designed in a trapezoidal shape, as depicted in Fig. 2. Design diagrams are typically employed in breakwater design processes. However, accommodating all wave-dissipating blocks within the breakwater can present challenges. To address this issue, a trapezoidal shape, which we address as tolerance area, was devised based on the Irregular Block Finished Shape Management Standards in Table II. The design diagram of the virtual mini model of offshore breakwaters is presented in Fig. 3. In Fig. 3, the bolded inner layer represents the original trapezoidal shape of the breakwater, while the outer layer depicts the newly created trapezoidal shape based on the management standards in Table II.

Fig. 2 Example of offshore breakwater [4]



(a) Side view



(b) Top view

Fig. 3 Design diagram of virtual mini model of offshore breakwater

Open Science Index, Computer and Systems Engineering Vol:18, No:9, 2024 publications.waset.org/10013823.pdf

TABLE II
IRREGULAR BLOCK FINISHED SHAPE MANAGEMENT STANDARDS

Item	Management Standards
Base Height	+1/3H
Top Width	+1/3H
Base Length	+1/3H

H: height of wave-dissipating blocks.

III. SIMULATOR

The steps in building our simulator can be broken down into two parts, designing the 3D model and installing the required features.

A. Designing 3D Model

Based on the design diagrams, the outer layer of the breakwater, the tolerance area and the foundation stone are created using 3D Computer-Aided Design (CAD) software. While there are various types of 3D CAD software available, this research employs AutoDesk Fusion 360. To import these models into the simulator which is based on Unity 3D, the created 3D models are exported as skp files. There is no need to convert the design diagrams of shake block to 3D model as the model is directly provided by Honma Concrete Industry Co., Ltd. The imported breakwater in our Unity simulator is shown in Fig. 4. The outer layer is set to transparent to enable us to observe the stacking of the blocks.

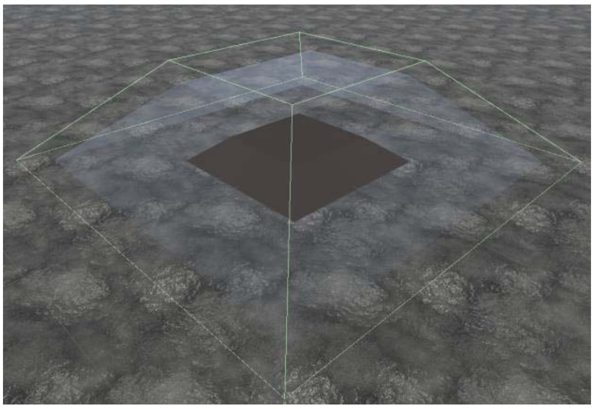
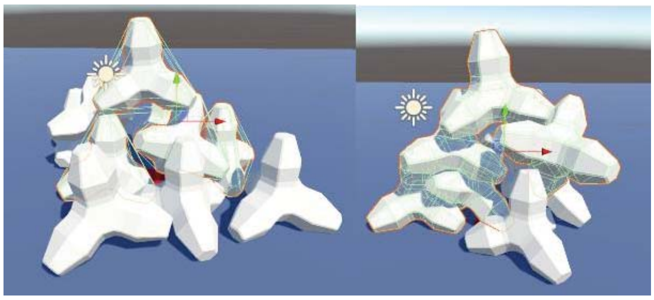


Fig. 4 Imported 3D model of offshore breakwater

B. Unity

The simulator is developed based on Unity, a three-dimensional game development software. We choose Unity as its physics engine, PhysX, which allows for the simulation of realistic physical phenomena. With this physics engine, replicating physical phenomena such as collisions and installation of wave-dissipating blocks, as well as crane movements, becomes simpler. Unity also boasts a powerful graphics engine, enabling the creation of immersive and realistic environments. Furthermore, Unity offers a wide range of assets, plugins, and extensions through its Asset Store. This allows developers to quickly add pre-developed features to the simulator without starting from scratch. One of the notable plugins we utilized is Easy Collider Editor.

Unity utilizes colliders to imbue objects with physical functionality such as collision responses and gravity, akin to real-world objects. Given the complex shape of wave-dissipating blocks, mesh shaped colliders are utilized. However, to conserve computational costs, Unity's default mesh colliders can detect collisions between convex mesh colliders, which means objects such as wave-dissipating blocks which have concave portion will have their collision ignored. To address this issue, Easy Collider Editor uses VHACD (Volumetric Hierarchical Approximate Convex Decomposition) to decompose a block into multiple convex shapes. Examples before and after the introduction of Easy Collider Editor are shown in Fig. 5. As shown in Fig. 5, with the introduction of Easy Collider Editor, collisions between wave-dissipating blocks can be accurately simulated.



(a) Before (b) After

Fig. 5 Examples before and after the implementation of Easy Collider Editor

C. User Interface and Controls

Our simulator User Interface (UI) is shown in Fig. 6 with each function explained in Table III.

In our simulator, wave-dissipating blocks are installed using game controllers or keyboards. The specific control methods are listed in Table IV. Additionally, depending on the mode (block installation mode/block deletion mode), certain functions may not be available or may differ. The game controller used in our research is the Logicool G Gamepad F301r, as shown in Fig. 7 and the terminology used for the operating methods may vary depending on the game controller.

IV. NOTABLE FEATURES

A. Replication of Hanging of Wave-Dissipating Blocks

To achieve a realistic simulation experience, we incorporated a third-party plugin known as Obi Rope to replicate the hanging of wave-dissipating blocks. Obi is an advanced particle-based physics engine renowned for its capability to simulate various deformable material behaviors. This system relies on extended Position-Based Dynamics (XPBD) particles, enhancing their lightweight nature, intricate detail, and unconditional stability. XPBD serves as an extension of the Position-Based Dynamics technique, enabling dynamic capabilities for simulating deformable objects like cloth and soft bodies. Furthermore, at runtime, the length of the rope can be modified, enabling the creation of cuttable ropes. This functionality allows for adjustments to the length of the rope suspending the wave-dissipating blocks and facilitates the detachment of the rope when placing the blocks.

Open Science Index, Computer and Systems Engineering Vol:18, No:9, 2024 publications.waset.org/10013823.pdf

Open Science Index, Computer and Systems Engineering Vol:18, No:9, 2024 publications.waset.org/10013823.pdf

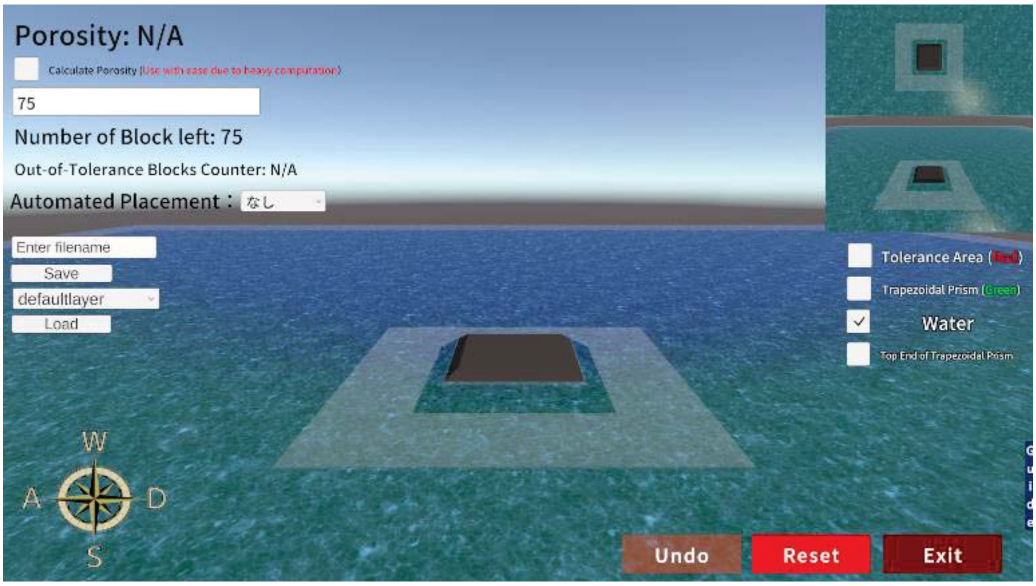


Fig. 6 Starting screen of the simulator

TABLE III
FUNCTION IN UI

Label	Name	Function
1	Porosity Viewer	Enables user to calculate current porosity of the structure
2	Block Requirement Viewer	Displays the total number of blocks required for breakwater based on design calculations
3	Remaining Blocks Counter	Allows user to track the number of blocks remaining for installation
4	Out-of-Tolerance Blocks Counter	Enables user to identify the number of blocks located outside / partially outside of the tolerance area
5	Automated Block Arrangement (Under Development)	Automatically positions the initial layer of blocks according to breakwater design.
6	File Management (Save/Load)	Enables users to save their current block arrangement or load previously saved configurations
7	Directional Indicator	Assists users in confirming the correct horizontal orientation when switching view
8	Other viewpoints	Allows users to check other viewpoints within the simulator
9	Display Settings for Breakwater	Enables users to choose the display of top end of the trapezoidal prism, the breakwater trapezoidal prism, or the tolerance area
10	Undo	Enable user to undo previous placement of block
11	Reset	Reset the entire scene
12	Exit	Close the simulator

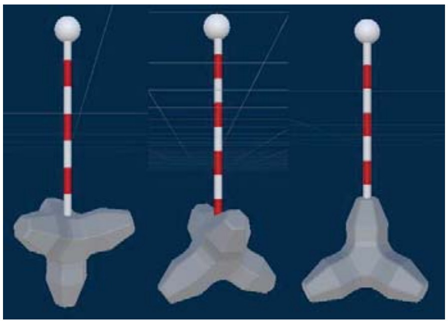
Fig. 7 Logicool G Gamepad F301r

TABLE IV
CONTROL GUIDE FOR SIMULATOR

Control	Function
Keyboard Controls	
1	Instantiate Crown Hanging
2	Instantiate Horizontal Hanging
3	Instantiate level Hanging
Tab	Switching of viewpoints
Space	Detach of rope (Installation of blocks)
WASD/Arrow keys	Movement of crane
QE keys	Vertical movement of rope (Q: Up, W: Down)
ZC keys	Horizontal rotation of block hanging
Shift	Switching between block installation mode / block edit mode
Left right arrow keys (Block edit mode)	Selection of block
Delete (Block edit mode)	Deletion of block
Game Controller	
Left stick	Movement of crane
Right stick	Horizontal rotation of block hanging
Directional pad (Up, Down)	Vertical movement of rope
LT/ RT Button	Switching of viewpoints
Y Button	Switching between block installation mode / block edit mode
B button	Detach of rope (Block installation mode) / Deletion of block (Block edit mode)
LB / RB Button	Selection of hanging of wave-dissipating blocks (Block installation mode) / Selection of block (Block edit mode)
START button	Reset
BACK button	Undo

The hanging of wave-dissipating blocks is categorized into three types: Crown hanging, Horizontal hanging, and Level hanging. Utilizing Obi Rope, these three hanging types are accurately replicated, as depicted in Fig. 8. Compared with real-life examples, as shown in Fig. 9, we can conclude that our replica matches the real-life examples.

Open Science Index, Computer and Systems Engineering Vol:18, No:9, 2024 publications.waset.org/10013823.pdf



(a) Crown Hanging (b) Horizontal Hanging (c) Level Hanging

Fig. 8 Replicas of hanging blocks

B. Toggle Trapezoidal Prism Display

At installation sites, wave-dissipating blocks are installed based on the top end of the trapezoidal prism from the design blueprint to maintain the finished shape of the offshore breakwater. The "finished shape" refers to the completed portion of the construction project, indicating the part where construction work has been finalized. Therefore, displaying the trapezoidal shape of the designed breakwater and the tolerance area enhances user convenience. Users are provided with the flexibility to freely toggle between displaying the top end, tolerance area, and design trapezoid according to their preferences. Fig. 10 illustrates an example of simultaneously displaying the top end, designated trapezoid of offshore-breakwater, and tolerance area.

(a) Crown Hanging (b) Horizontal Hanging (c) Level Hanging

Fig. 9 Real-life examples of hanging blocks

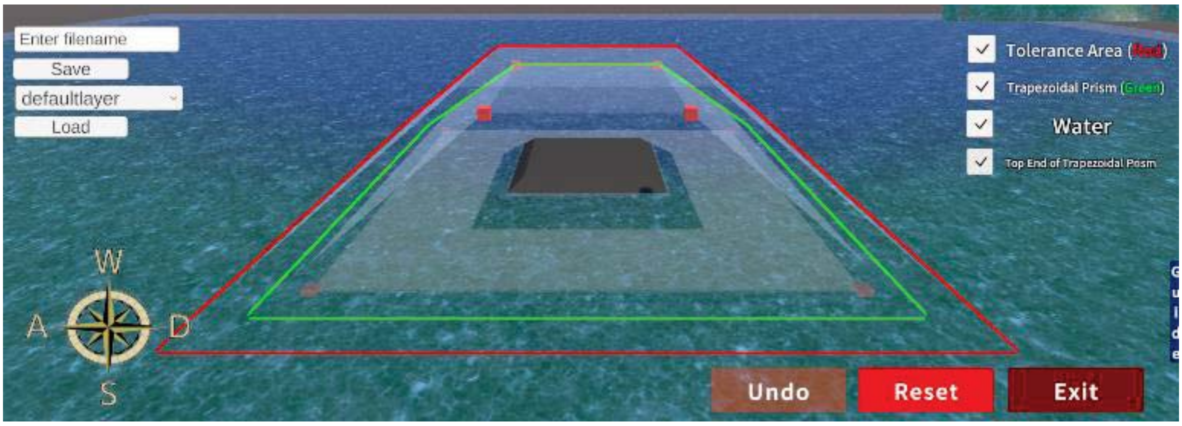


Fig. 10 Simultaneous display of the top end, designated trapezoid of offshore-breakwater, and tolerance area

C. Visual Confirmation of Protruding Blocks

To easily identify wave-dissipating blocks protruding beyond the tolerance area, a visual confirmation feature for protruding blocks has been added. As shown in Fig. 11, protruding blocks are outlined in yellow for clear visibility.

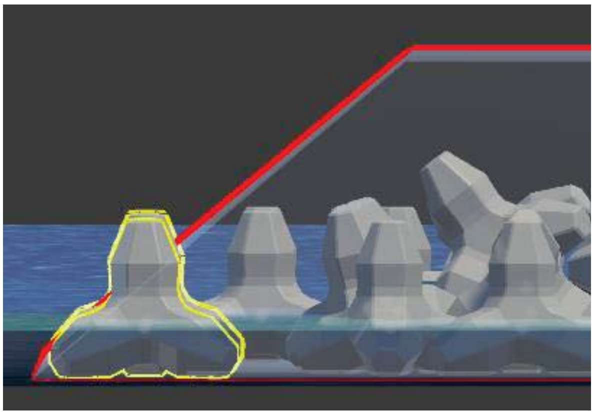


Fig. 11 Example of protruding blocks

To check whether a block is protruding beyond the tolerance area, we implement RayCast function from Unity. The RayCast function allows us to fire a laser beam-like line called a raycast in a specific direction from a point in a Unity's space. It is possible to detect the contact of a laser beam with any object in the space [5]. To determine if a block is entirely contained within the tolerance area, we adopt a similar approach referred to as the "bed of nails" method, as suggested by "hoodlma" on stackoverflow [6]. This method involves casting rays towards each block from various directions. If all rays intersect with the collider of the tolerance area before reaching the block, we can confidently ascertain that the block is fully within the tolerance area. A conceptual illustration of this approach is presented in Fig. 12. In the figure, we are trying to see if the blue point is inside the yellow collider. The green arrows represent successful raycasts (the yellow collider is hit), and the pink one is unsuccessful (the yellow collider is not hit).

D. File Management (Save/Load)

The installation process requires a significant number of blocks. Even in the miniature model of the offshore breakwater used in this study, approximately 75 blocks are necessary, and it is anticipated that installations at actual sites may require more than 75 blocks. Since installing a large number of blocks can take several hours to complete, a feature to save progress midway has been added.

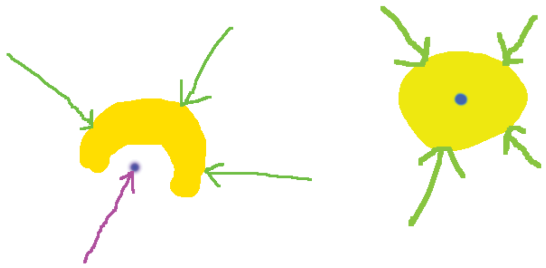


Fig. 12 Conceptual illustration of "bed of nails" method [6]

For the saving process, data are stored in JSON format. Initially, the position and rotation information of the wave-dissipating blocks are recorded and converted into a serializable format, then added to a list. Subsequently, all block data are serialized into a JSON string and saved under the specified file name. When loading a saved file, users can reproduce the installation status up to that point by selecting the desired file name from the dropdown list in the user interface.

V. Evaluation Method

One of the practical examples within this simulator is the focus on the construction method of offshore breakwaters. The 4-ton Shake Block that we focused on this research has an ideal porosity rate of 56% [7]. Therefore, as the porosity rate of the breakwater approaches 56%, the higher the evaluation is considered. In this simulator, users can calculate the porosity from the installed wave-dissipating blocks and evaluate their skills based on the difference from the ideal porosity rate.

A. Current Porosity Calculation Limitations

In geology, hydrogeology, soil science and building science, the porosity (Φ) of a porous medium describes the fraction of void space in an object, where the void may contain, for example in our case, air space. It is defined by (1), where V_B is defined as volume of breakwater while V_b is defined as total volume of wave-dissipating blocks inside the breakwater.

$$\Phi = \frac{V_B - V_b}{V_B} \times 100\% \quad (1)$$

While (1) offers an accurate means of calculating porosity, it presents limitations, particularly in our context. For instance, accurately determining V_b becomes challenging when blocks are partially within the breakwater. The irregular shape of wave-dissipating blocks complicates the calculation of partial volumes, making it less than ideal for simulation purposes.

B. Voxelization

To solve the limitations addressed on (1), we proposed using voxelization to calculate porosity. Voxelization is the process of approximating objects using small cubic elements called voxels, representing them as a collective set of voxels. It finds applications in various fields such as 3D computer graphics, scientific visualization, medical imaging, and game development. The primary advantage of voxelization lies in simplifying computations for irregular objects like trapezoids and wave-dissipating blocks. Unlike polygon-based modeling, where objects are constructed using polygons, voxelization constructs objects using fixed-size cubes, with resolution depending on the size of the voxels. Smaller voxel sizes enable higher resolution and more detailed representations, but come with increased computational complexity, requiring more processing time and resources.

Initially, we initiate the voxelization process by applying it to the trapezoidal prism representing the breakwater. Each object in Unity is enclosed within its bounding box, shown in Fig. 13. We proceed by decomposing the bounding box of the breakwater into a cohesive assembly of voxels, determined by the specified number of voxels allocated along each axis (GridDivisionCount). For instance, if GridDivisionCount is designated as 40, the breakwater's bounding box will be segmented into 40 voxels along each axis, shown in Fig. 14.

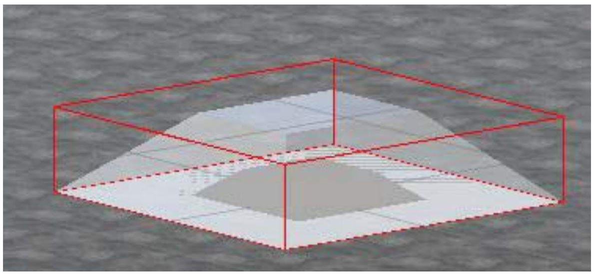


Fig. 13 Bounding box of offshore breakwater

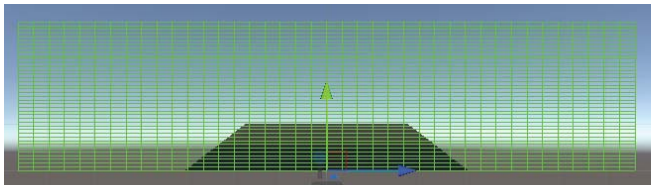


Fig. 14 Decomposition of bounding box for GridDivisionCount = 40

Subsequently, employing a methodology akin to the previously mentioned "bed of nails" technique, we conduct raycasts towards each voxel instead of each block. This enables us to ascertain whether each voxel is entirely contained within the trapezoidal prism. Through this process, redundant voxels are eliminated, leaving behind only those that contribute to forming a shape resembling the trapezoidal prism of the breakwater, shown in Fig. 15.

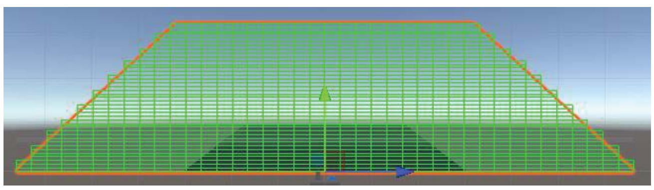


Fig. 15 Using voxels to resemble the trapezoidal prism

Open Science Index, Computer and Systems Engineering Vol:18, No:9, 2024 publications.waset.org/10013823.pdf

Open Science Index, Computer and Systems Engineering Vol:18, No:9, 2024 publications.waset.org/10013823.pdf

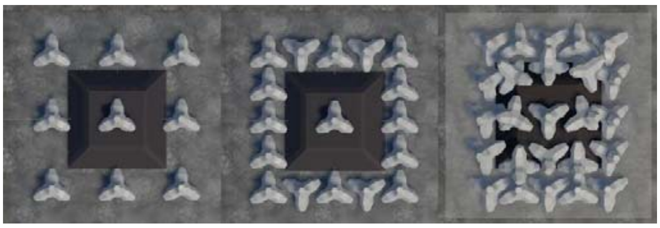
Following the voxelization of the breakwater, the next step involves verifying whether each voxel intersects with wave-dissipating blocks. To accomplish this, we employ the Physics.OverlapBox function available in Unity. Physics.OverlapBox creates an invisible box that defines the tests collisions by outputting any colliders that come into contact with the box [8]. In other words, utilizing Physics.OverlapBox allows us to locate voxels that are either fully or partially enclosed within a block. By applying this process to each voxel within the trapezoidal prism, we can readily ascertain the number of voxels that intersect or contain wave-dissipating blocks.

Once we have determined the total number of voxels representing the trapezoidal prism and the total number of voxels overlapping with wave-dissipating blocks, we can proceed to calculate the porosity using (2) as proposed, where n_B is defined as the total number of voxels representing the trapezoidal prism and n_b is defined as the total number of voxels overlapping with wave-dissipating blocks. Equation (2) does not directly involve volume when calculating porosity. Therefore, we anticipate that it can accurately calculate porosity even in scenarios where blocks are partially within the breakwater.

$$\Phi = \frac{n_B - n_b}{n_B} \times 100\% \quad (2)$$

VI. Experiment

To verify whether the proposed method accurately calculates porosity, we conducted experiments. The experiments consisted of varying the number of wave-dissipating blocks (9, 17, 25) as shown in Fig. 16 and the gridDivisionCount (the number of voxels to divide) to be 50, 100, 150, and 200. We compared the porosity obtained through voxelization with the porosity obtained through the previous method, assuming all blocks had no protrusions. This validation method ensures that we establish the correct porosity as (1) enables us to obtain the correct porosity under the condition where the blocks have no protrusions. This allows us to assess how closely the proposed method approaches the correct porosity. The reason we varied the number of voxels and blocks is to assess their relationship with computation time.



(a) 9 blocks (b) 17 blocks (c) 25 blocks

Fig. 16 Placement of blocks during experiment

The experiment results are presented in Tables V and VI. It is evident from Table V that increasing the number of voxels for division brings the calculated porosity closer to the correct value. However, it leads to heavier computations and longer processing times based on the results from Table VI. Moreover, increasing the number of blocks within the same division of voxels does not significantly alter the computation time. This might be because regardless of how many blocks are within the breakwater, the proposed method still needs to loop through all divided voxels to calculate the porosity.

TABLE V
EXPERIMENT RESULTS (POROSITY (%))

Number of blocks used	Porosity (%)				
	gridDivisionCount				Correct value
	50	100	150	200	
9	89.52	92.40	93.25	93.60	9.68
17	80.5	85.58	87.12	87.82	89.94
25	71.93	79.03	81.10	82.16	85.21

TABLE VI
EXPERIMENT RESULTS (PROCESSING TIME (SECS))

Number of blocks used	Processing time (seconds)			
	gridDivisionCount			
	50	100	150	200
9	0.56	4.76	13.55	31.13
17	0.64	5.11	13.92	32.02
25	0.66	5.22	14.51	32.89

The visualization results of the voxels containing wave-dissipating blocks are shown in Fig. 17. The top column in Fig. 17 shows the visualization results of GridDivisionCount = 50, while the bottom column shows GridDivisionCount = 200. These visualizations demonstrate that increasing the number of voxels for division enhances the fidelity of reproducing wave-dissipating blocks using voxels.

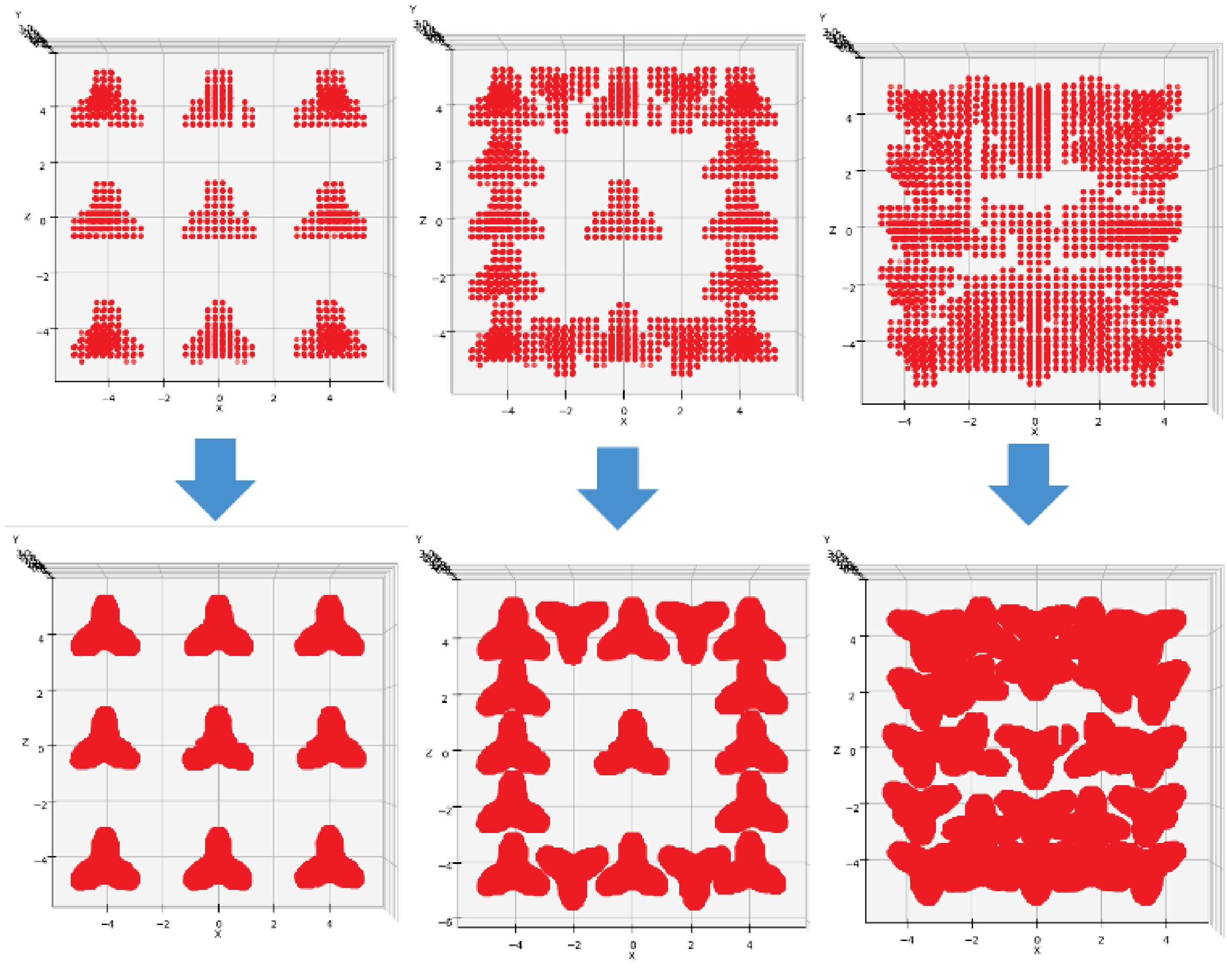
Based on this experiment, it can be concluded that increasing the number of voxels for division improves the accuracy of porosity calculation. In other words, a higher division of voxels yields better results. However, since the goal of this simulator is to enable workers to install wave-dissipating blocks while checking porosity in real time, aiming for computation times within 1 second is crucial. Therefore, it is deemed more important to streamline the porosity calculation through voxelization rather than just increasing the number of voxels for division.

VII. Conclusion

To enhance the proficiency of workers, we built a wave-dissipating block installation simulator and proposed a method for calculating the porosity of breakwaters as an evaluation criterion.

When voxelizing the breakwater, it is believed that the approximation accuracy of voxelization increases with a higher number of voxels, leading to a more accurate determination of porosity. However, this also increases the computational load, potentially resulting in increased latency during operations within the simulator.

A future challenge is to devise methods to optimize the voxelization process to enable workers to perform installation tasks while verifying porosity. It is necessary to determine the optimal value for the grid cell count, considering both accuracy and latency. Additionally, there are still many areas for improvement in the simulator that require enhancements to provide more usability for users.



(a) 9 blocks (b) 17 blocks (c) 25 blocks

Fig. 17 Visualization of blocks using voxelization

ACKNOWLEDGMENT

This research was conducted with the support of the Grant-in-Aid for Joint Course Creation Assistance in Higher Education Institutions in 2023. Additionally, we would like to express our gratitude to Honma Concrete Industry Co., Ltd. for providing the three-dimensional data.

REFERENCES

[1] Ministry of Land, Infrastructure, Transport and Tourism, "*i-Construction no Saranaru Tenkai* (Further development of i-Construction)," i-Construction Promotion Consortium, Document for the 9th Planning Committee Meeting, Item 2, Dec. 2023.

[2] Mitsui Jun, Kubota Shin-ichi, Hashida Masaya,Noboru Satoshi, "Development of a real-time placement simulation method for wave-dissipating blocks," in Journal of Japan Society of Civil Engineers, Ser. B2 (Coastal Engineering) 78 (2), I_685-I_690, 2022

[3] Honma Concrete Industry Co., Ltd, "Shake Block", https://www.honmacon.co.jp/02syouhin_list/01sye-kuburokku/01setumei.html.

[4] Goko-Construction Co., Ltd, "*Kanbara Fuji Kaigan Rigantei Kouji* (Kambra Fuji Coast offshore breakwater construction)," https://www.goko-con.co.jp/2019/08/06/%E5%B9%B3%E6%88%9030%E5%B9%B4%E5%BA%A6-%E8%92%B2%E5%8E%9F%E5%AF%8C%E5%A3%AB%E6%B5%B7%E5%B2%B8%E9%9B%A2%E5%B2%B8%E5%A0%A4%E5%B7%A5%E4%BA%8B/, 2018

[5] 2023 Unity Technologies, "Physics2D.Raycast," https://docs.unity3d.com/ScriptReference/Physics2D.Raycast.html, Unity Documentation, 2023

[6] hoodlma, "How to detect that my object is completely inside a box?," stackoverflow, Feb. 2014

[7] Honma Concrete Industry Co., Ltd, "Shake Block", https://www.honmacon.co.jp/08/pdf/2022/sye-kuburokkuv9_202202.pdf, p. 10

[8] 2023 Unity Technologies, "Physics.OverlapBox," https://docs.unity3d.com/ScriptReference/Physics.OverlapBox.htmll, Unity Documentation, 2023

Open Science Index, Computer and Systems Engineering Vol:18, No:9, 2024 publications.waset.org/10013823.pdf