forests

Article

Exploring a New Physical Scenario of Virtual Water Molecules in the Application of Measuring Virtual Trees Using Computational Virtual Measurement

Zhichao Wang [1,2,†], Xiaoning Zhang [3,†], Xiaoyuan Zhang [4], Xinli Pan [5], Tiantian Ma [1,*], Zhongke Feng [6,*] and Christiane Schmullius [2]

[1] Precision Forestry Key Laboratory of Beijing, Beijing Forestry University, Beijing 100083, China; zhichao@bjfu.edu.cn

[2] Department for Earth Observation, Friedrich Schiller University Jena, Loebdergraben 32, 07743 Jena, Germany; c.schmullius@uni-jena.de

[3] Faculty of Electrical Engineering and Information Technology, Ruhr University Bochum, Universitaetsstr. 150, D-44801 Bochum, Germany; xiaoning.zhang@edu.ruhr-uni-bochum.de

[4] State Key Laboratory of Chemical Resource Engineering, Beijing Key Laboratory of Advanced Functional Polymer Composites, Beijing University of Chemical Technology, Beijing 100029, China; zhang.xiaoyuan@buct.edu.cn

[5] Instiute of Eco-Environmental Research, Guangxi Academy of Sciences, Nanning 530007, China; pxl@gxas.cn

[6] Surveying and 3S Engineering Research Center, Beijing Forestry University, Beijing 100083, China

* Correspondence: matiantian2019@bjfu.edu.cn (T.M.); zhongkefeng@bjfu.edu.cn (Z.F.)

† Those authors contribute equally to this work.

Abstract: Our previous studies discussed the potential of measuring virtual trees using computational virtual measurement (CVM). CVM is a general methodology that employs observational techniques in lieu of mathematical processing. The advantage of CVM lies in its ability to circumvent mathematical assumptions of tree shapes at the algorithmic level. However, due to the current computational limitations of desktop computers, the previously developed CVM application, namely, virtual water displacement (VWD), could only act as a primary theoretical testimonial using an idealized point cloud of a tree. The key problem was that simulating a massive number of virtual water molecules (VMMs) consumed most of the computational resources. As a consequence, an unexpected empirical formula for volume calibration had to be applied to the output measurement results. Aiming to create a more realistic simulation of what occurs when water displacement is used to measure tree volume in the real world, in this study, we developed a new physical scenario for VWMs. This new scenario, namely, a flood area mechanism (FAM), employed footprints of VWMs instead of quantifying VWM counts. Under a FAM, the number of VMMs was reduced to a few from several thousands, making the empirical mathematical process (of the previously developed physical scenario of VWMs) unnecessary. For the same ideal point clouds as those used in our previous studies, the average volume overestimations were found to be 6.29% and 2.26% for three regular objects and two artificial stems, respectively. Consequently, we contend that FAM represents a closer approximation to actual water displacement methods for measuring tree volume in nature. Therefore, we anticipate that the VWD method will eventually utilize the complete tree point cloud with future advancements in computing power. It is necessary to develop methods such as VWD and more CVM applications for future applications starting now.

Keywords: computational virtual measurement; virtual water displacement; PhysX; unity; future methods; tree volume

Citation: Wang, Z.; Zhang, X.; Zhang, X.; Pan, X.; Ma, T.; Feng, Z.; Schmullius, C. Exploring a New Physical Scenario of Virtual Water Molecules in the Application of Measuring Virtual Trees Using Computational Virtual Measurement. *Forests* **2024**, *15*, 880. https://doi.org/10.3390/f15050880

Academic Editor: Jan Bocianowski

Received: 18 April 2024
Revised: 15 May 2024
Accepted: 17 May 2024
Published: 18 May 2024

1. Introduction

1.1. Trend of Virtualization of Forest Studies

Virtualization in forest studies is on the rise, with advancements in 3D scanning technologies allowing for faster mapping of forests [1]. The use of digital 3D technologies in forestry has seen significant developments, with a focus on 3D reconstruction tools and visualization methods [2]. Initiatives like the "Virtual Forest" project aim to create intelligent planning and decision support tools for forest growth and wood harvest through the integration of aerial survey technology, satellite data, virtual reality, and robotics know-how [3,4]. Various methods are employed in creating virtual forests, including the integration of remote sensing data, three-dimensional modeling, and forest inventory data [5]. One approach involves developing digital twins of forests through computational virtual recreations, utilizing virtual measurement processes like the virtual diameter tape for accurate tree modeling [6]. Additionally, forest scene synthesis technology utilizing OpenGL graphics standards and virtual reality can be used to create virtual forest landscapes with detailed features and functionalities [7]. Evaluation of algorithms for generating virtual forests has shown that different methods, such as plant competition models and random point distribution techniques, produce forestry with varying levels of believability and playability, emphasizing the importance of considering the end goal when selecting an algorithm [8]. Virtual reality technology is also utilized for three-dimensional reconstruction of forest landscapes, aiding in effective forest resource management through reliable data support by providing full view of forests instead of a group of single photos [9].

However, from our point of view, most of the current studies on the virtualization of forests are limited to the field of tree phenotype reconstruction, which could hardly represent live processes in virtual forests. In our previous works, we defined two stages of virtual forests: (1) static virtual forests and (2) dynamic virtual forests [6]. According to our own experience, if we intend to incorporate physical processes into virtual forests/trees, two major drawbacks must be addressed. The first drawback is the lack of a software package for forest science. Therefore, forest scientists have to use packages from unfamiliar fields of study. Then, how to integrate a third-party software package [10–12] with virtual forests/trees would be the next big challenge. The second drawback is the current computational power of desktop computers, which limits the simulation of physical processes in forests [13,14].

Nowadays, the physics engines used in computer games have revealed the potential to be utilized in adding physical dynamics to virtual forests/trees. These engines simulate various physical systems, like rigid-body dynamics, soft-body dynamics, and fluid dynamics [15]. Engines like PhysX, Open Dynamics Engine (ODE), Bullet, Vortex, and MuJoCo may also feature specific tools for handling the unique challenges of certain simulation scenarios, such as the accurate simulation of multiple contact points or the integration with robot control systems for enhanced precision in industrial applications such as the simulation as robot movement [16]. Meanwhile, another key feature of game development platforms is the capability to create large virtual space [17]. These virtual spaces are constructed using methods that define the environment for user immersion, including features like defining the field of view, generating images synchronized with head-mounted display movements, and incorporating guiding regions to direct user sight lines [18].

1.2. Virtual Water Displacement Method

The virtual water displacement (VWD) method is our previously developed method used in the simulation of physical processes to measure point clouds, particularly for assessing the volume of objects like tree stems within a virtual reality [13]. This method uses a virtual measuring instrument (VMI) that comprises various computer algorithms, including physical simulation functions from libraries like PhysX. The process essentially mimics the traditional water displacement method but in a simulated, virtual environment.

In VWD, instead of physically measuring water displacement in a container, the method virtually calculates the mass of water displaced by the object using the number of virtual water molecules (VWMs) displaced. These VWMs are not represented on a molecular scale due to computational limits. Instead, they are balls that simulate two key physical properties of water molecules: possessing a certain volume and the ability to move. The VWD method involves setting up a 3D space in software such as Unity, where the virtual displacement occurs. A virtual object, such as a tree's point cloud, is placed within a virtually represented container, and the displacement of VWMs around this object is calculated to estimate the volume of the object. The dynamic and static properties of VWMs are crucial, as each VWM acts as a volume placeholder, ensuring that it occupies a unique space within the virtual environment, which helps in accurately calculate the displaced volume. This method is particularly useful for applications where physical water displacement is impractical or where digital models need precise volume assessments without physical prototypes.

Figure 1a shows the concept of VWD and Figure 1b the implementation of VWD using the original physical scenario. The major problem with using VWMs in the VWD method is their simplified representation compared to actual water molecules. In the virtual environment, each VWM is significantly enlarged and simplified to facilitate computations, which does not accurately reflect the complex interactions and behaviors of real water molecules on a molecular scale. Due to the inability to precisely simulate the dynamic and static characteristics of water at the molecular level, such simplifications may lead to inaccuracies in volume measurement. Additionally, the process relies on a predefined space within the virtual environment, and the arrangement and distribution of these virtual molecules can impact the precision of the volume calculations. The volume of each VWM does not just include its own volume but also considers the gap volume shared with neighboring VWMs. The calculation of these volumes requires assumptions about the distribution and packing density of the VWMs, which can introduce further sources of error. Therefore, developing a new physical scenario for VWMs is becoming necessary.

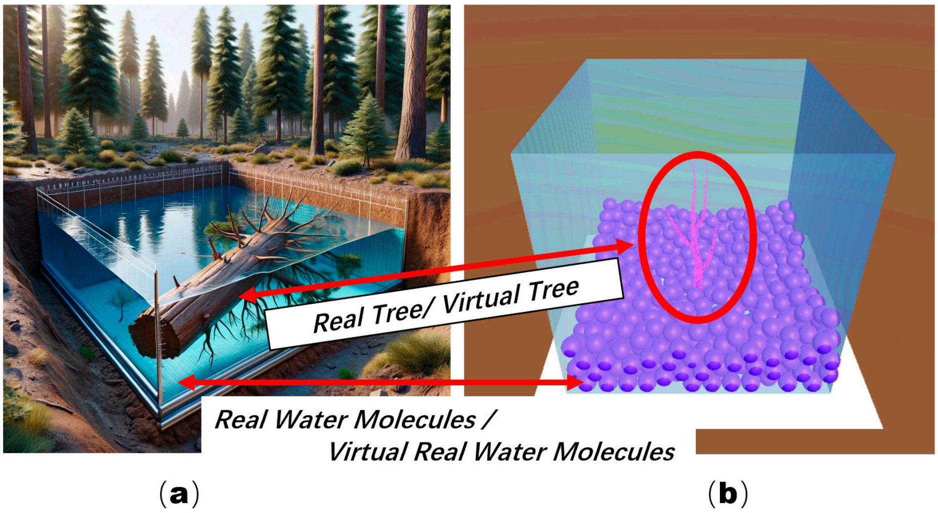

Figure 1. Comparation of water displacement (**a**) and virtual water displacement (**b**).

2. Materials and Methods

2.1. Data

For maintaining the development consistency between two VWD versions, we adopted the same dataset as in our previous study. Due to the limitation on current computer performance and the PhysX SDK 3.3 [19], for example, each point consumes a resource from GameObjects, with an overall limitation of 65,536 GameObjects. The number of point clouds representing real trees far exceeds this limit. Meanwhile, commonly used graphics cards, such as the GTX 1650, cannot support more than approximately 10,000 GameObjects in the same scene during the development of this newly designed physical scenario. Therefore, two kinds of artificial models were used in this development as representatives for the real tree point cloud. We used 3D modeling tools [20,21] to create the artificial trees and calculated their mesh volumes as true volumes [22]. After that, we applied the Point Cloud

Library (PCL) and Visualization Toolkit to re-sample 3D models into point clouds [23,24]. The detailed description of the procedures can be found in our previous works [13].

Regular-shaped objects were selected as the first type of artificial models to test the VWM interactions with basic geometrical shapes. The models created were a sphere, cube, and cylinder. Each model has coordinates, a scale parameter, and a surface mesh in virtual space. Noteworthy is the difference of 2.15% and 1.64% between the mathematical and mesh volumes for the sphere and the cylinder, respectively. Figure 2 presents the models and point clouds for these regular-shaped objects.

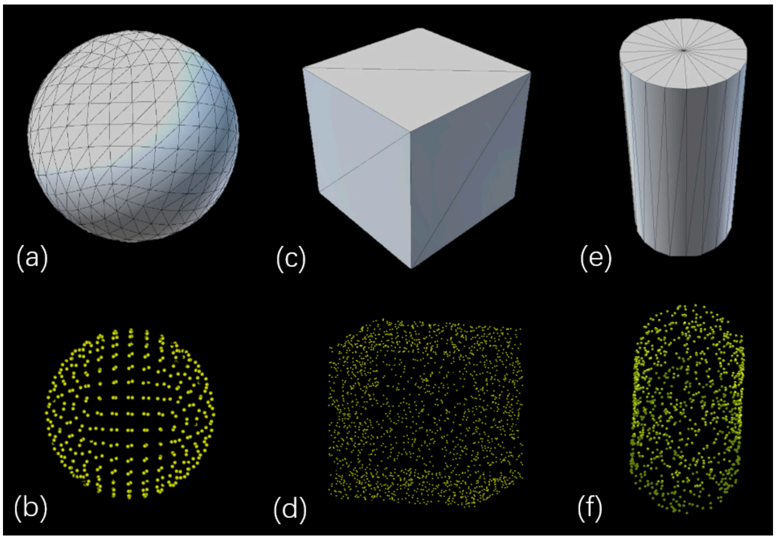

Figure 2. Regular-shaped objects. (**a**) Sphere mesh model; (**b**) sphere point cloud; (**c**) sphere mesh model; (**d**) cube point cloud; (**e**) cube mesh model; (**f**) cylinder point cloud [13].

Artificial stems were explored as the second type of artificial models. We used the Unity (5.6.7) Tree Editor to manually create two distinct artificial stems. The "stem" model consisted solely of the main stem structure, devoid of any branches. As depicted in Figure 3, the "stem with branches" model originated from the "stem" model and was augmented with three branches. Subsequently, these were converted into point clouds. Owing to computational performance constraints, we decimated the number of points to 1000 for the "stem" and 2000 for the "stem with branches" model.

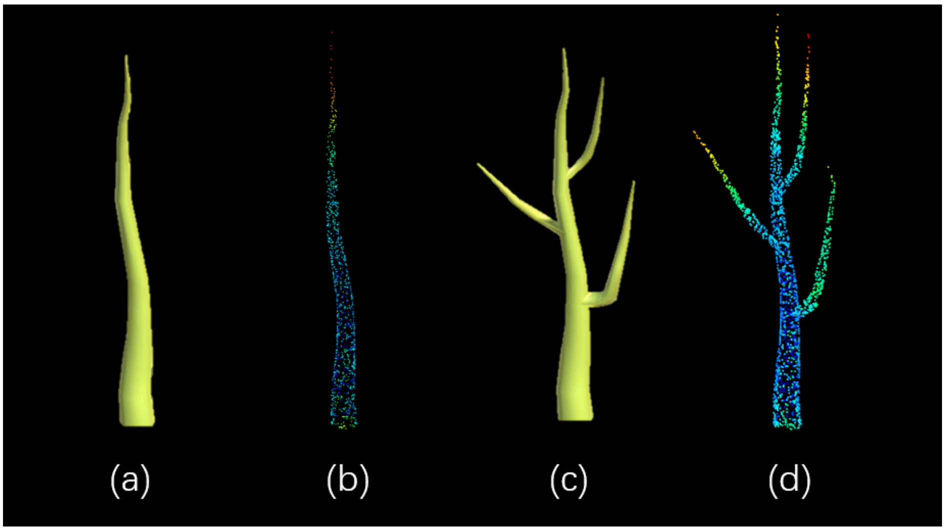

Figure 3. Artificial stems. (**a**) "Stem" mesh model; (**b**) "stem" point cloud; (**c**) "stem with branches" mesh model; (**d**) "stem with branches" point cloud [13].

2.2. The Original VWD Method and the Scale Effect

In order to compare with the new physical scenario of VWMs, this section describes two major drawbacks of the physical scenario of VWMs we previously developed. The first is the need for additional empirical calibration, and the second is the scale effect. However, a water displacement method for measuring tree volume should not require these two processes if applied in reality. For detailed technical information, it is suggested to refer to Ref. [13].

2.2.1. The VWD Method with Empirical Calibration

VWD is an exploratory method proposed in our previous work. It simulated the natural water displacement (WD) procedure in virtual space. In reality, the displaced volume is collected using a graduated cylinder or by weighing the mass of water. In virtual space, the quantity of displaced VWMs was counted one by one to calculate the volume or the mass of water using the following Equation (1), where V is the stem volume; k is the coefficient of volume calibration, N is the predicted number of VWMs for a vast vessel, n is the actually filled number of VWMs in a vessel with a tree point cloud inside, and v_s is the sphere volume of a VWM.

$$V = k(N - n)v_s \tag{1}$$

By applying the VWD procedure in virtual space, the tree volume was a directly measured variable, similar to the measurements for DBH and H in reality, which should be an observation process. Theoretically, there was only one variable (n) in this equation, which limited the source of uncertainty. We applied the methods in the sphere packing problem to determine the theoretical values of N and k [25,26]. However, it was proven that there was an obvious difference between the theoretical and the experimental results in virtual space. Therefore, the empirical parameters, such as an empirical calibration for k in the following equation, had to be introduced for the calibration. Where k is the coefficient of volume calibration from VWM sphere volume to VWM volume (as a volume placeholder), V_{vessel} is the volume of the vessel, n' is the actual filling number of VWMs, and v_s is the sphere volume of a VWM.

$$k = \frac{V_{vessel}}{n' \nu_s} \tag{2}$$

In this equation, n' is the independent variable. However, it was proven that the relative error for the determination of n' was less than 1% in our previous work. Therefore, with the involvement of n', the VWD was not completely a direct measuring method.

2.2.2. The Scale Effect

The scale effect is a phenomenon observed in our prior research. Due to computer performance constraints, the number of virtual world models (VWMs) was restricted to a very small size (a few thousand at most). The size ratio between the VWM and the object affected the accuracy of the virtual world development (VWD) process. Under typical conditions during LiDAR scanning, the average distance between two neighboring points on the objects could be less than 1 cm. It was anticipated that the diameter of the VWM cloud would comprise less than 10% of the stem disk diameter. This size ratio was described by the following equation, where ρ refers to the size ratio between the VWM and (a part of) the target object, $d(VWM)$ refers to the diameter of the VWM, and $d(O)$ refers to the diameter of the circumscribed circle for (part of) the detection object.

$$\rho = \frac{d(VWM)}{d(O)} \tag{3}$$

In our previous works, the ρ varied from 41% to 29% with decreasing $d(VWM)$. Correspondingly, the difference between VWD volumes and true volumes shrank much faster. We applied several regression methods to predict the trend of the scale effect. The result showed that the scale effect was eliminated when ρ changed from 5% to 17%.

However, due to the computer performance of our study and the theoretical limitation from PhysX SDK, it was not possible to have a further investigation of the scale effect. As a compromise, we introduced an additional empirical coefficient to solve this problem. Consequently, the corresponding uncertainty was involved as well.

2.3. Flood Area Mechanism Using a Single Virtual Water Molecule

2.3.1. The Goal of the VWD Revision

The original design of the VWD was a simulation of WD in the virtual space. The VWD regarded the point cloud in the virtual space as representative of the living tree in reality. Therefore, the VWD volume should be derived from direct measurements. Obviously, no empirical calibration was applied in natural WD. In our previous study, we introduced some regression methods as a compromise for the limitation of computer performance. In this study, we would like to modify the VWD method to be independent with empirical regression. Meanwhile, it was believed to prevent the scale effect and to generate the output of the true volume.

2.3.2. Flood Area Mechanism Using Single VWM

Each VWM holds critical importance in two aspects: static and dynamic processes. In static processes, a VWM serves as a volumetric placeholder. In dynamic processes, it emulates the motion of water molecules (or clusters) to achieve uniform distribution in virtual space. To refine the computer performance, we introduced a new flood area mechanism (FAM) to simulate VWM interactions, foregoing the use of multiple VWMs. The flowchart of the FAM is shown in Figure 4.

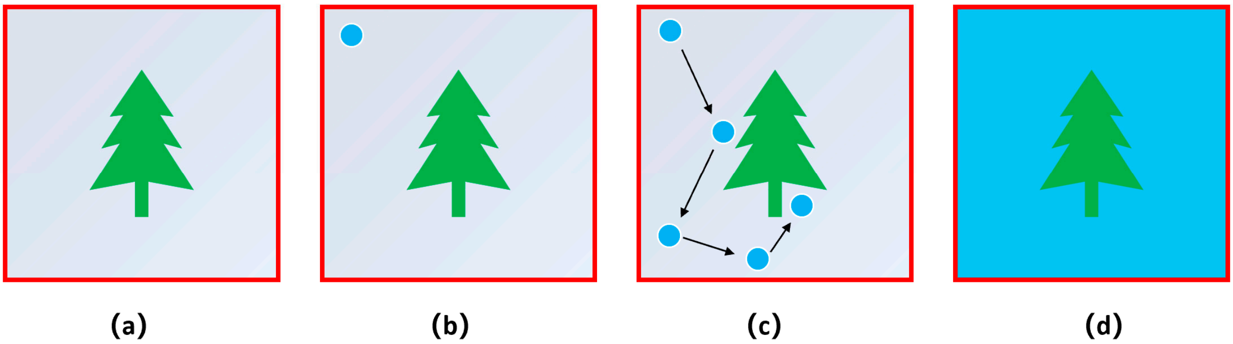

Figure 4. The key steps of the flood area mechanism (FAM). (**a**) a 3D Euclidean virtual space was created; (**b**) a single VWM was created; (**c**) a single VWM moves freely in the virtual water sink; (**d**) final results.

(i) A 3D Euclidean virtual space was created. This virtual space, which resembles a water sink, as shown in Figure 1a, is used to contain the virtual tree and VWMs. Subsequently, the space within the virtual water sink was divided into a grid of voxels. Each voxel was initially marked with a negative indicator, such as 0 or false, to denote unoccupied water space (In Figure 4a).

(ii) In contrast to what is depicted in Figure 1b, a single VWM was created in this study. The diameter of the VWM exceeded the average distance between adjacent points. Upon its creation, the VWM occupied an area within the virtual water sink. Importantly, this occupied area was not considered part of the virtual tree. Meanwhile, the rest area in this virtual water sink could belong to the virtual tree (in Figure 4b).

(iii) This step was to let the single VWM move freely in the virtual water sink. Where the VWM could visit does not belong to the virtual tree. To enable the VWM to move, a gravity field was initiated with a random three-dimensional orientation [13].

(iv) Attracted by this gravity field, the VWM moved, marking its trail in the voxels. The status of voxels traversed by the VWM changed to positive values, such as 1 or true, to signify water occupancy (in Figure 4c).

(v) Upon colliding with the point cloud or vessel boundary, the gravitational direction was randomized anew. The VWM was dragged to a new route of movement.

(vi) Each collision event of the VWM triggered step (v) again.

(vii) There are several termination conditions, such as the preset application runtime, the number of footprints, and the number of collision events. The thresholds for these termination conditions can be auto-detected or user-defined (in Figure 4d).

By performing these steps, the FAM mechanism could be fulfilled using only one VWM, which significantly diminished the consumption of computer performance. Meanwhile, the uncertainty from the interaction of VWMs no longer existed. Thus, Equation (1) (of the previously developed physical scenario of VWMs) can be transformed into a simpler form in the following Equation (4), where V refers to the volume of the tree, V_{vessel} refers to the volume of the vast vessel, and V_{flood} refers to the volume where the VWM has been visited.

$$V = V_{vessel} - V_{flood} \tag{4}$$

From the user′s perspective, the sole responsibility is to deploy the VWM and interpret the outcomes. This procedure aligns closely with traditional methods of measuring diameter at breast height (DBH) and height (H).

2.3.3. Improvement for Time Efficiency

The updated FAM mechanism is both parameter-free and free from empirical regression, effectively circumventing the limitations of computer performance. However, as a trade-off, the execution time of the method significantly increases. This is due to the VWM position being calculated incrementally, frame by frame, utilizing PhysX. Within computer graphics, performance is assessed by frames per second (FPS). Assuming the VWD simulation operates at an FPS of 100, this indicates the VWM is capable of traversing 100 voxels each second. For instance, in a cubic vessel with a side length of 100 units encompassing 1,000,000 voxels, a comprehensive scan of this voxel array would require approximately 10,000 s (~2.7 h). Moreover, since the VWM movement is propelled by a stochastic gravitational field, this could further prolong the duration necessary to achieve a complete scan.

We devised two strategies to enhance time efficiency. The first strategy involved deploying multiple virtual world models (VWMs) simultaneously. By replicating n number of VWMs from the original, the time efficiency could potentially improve up to n times, under the assumption that each VWM operates independently without considering whether voxels are visited by the same or different VWMs. However, utilizing a greater number of VWMs could increase computational demand and accordingly reduce the frames per second (FPS). Henceforth, it would be possible to fine-tune the quantity of VWMs released by applying the Equation (5), where M refers to the count of visited voxels in a second, n refers to the number of VWMs in virtual space, and FPS refers to frames per second.

$$M = n \cdot FPS \tag{5}$$

The second strategy was to use the buffer for the visited voxel. As shown in Figure 5, if we granted a buffer with 1 unit, we could mark 27 voxels as the flood area. Under this configuration, one movement of the VWM could generate 27 footprints instead of just one footprint without buffering. From the same area, VWMs with buffer can leave many more footprints on the 3D space. Consequently, this could significantly reduce the total time required to visit the entire flooded area.

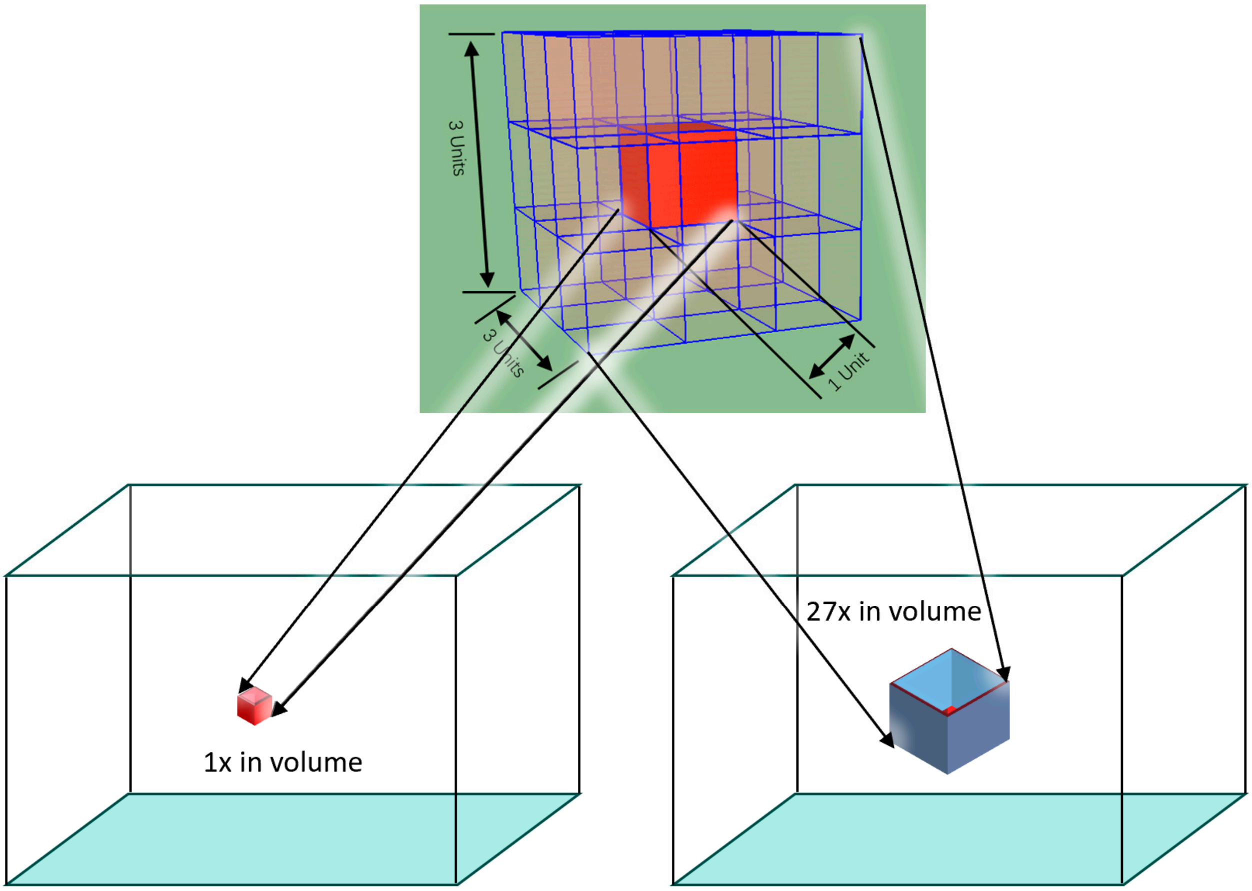

Figure 5. The voxel buffering for the improvement of time efficiency.

2.3.4. Developing Environment

The original developing environment was drawn from our previous paper [13]. A new desktop computer was used in this work. It had an Intel i5-11400f CPU (Intel, Silicon Valley, CA, USA) and a NVIDA GTX1650 (NVIDIA, Santa Clara, CA, USA) graphic card.

3. Results

3.1. Test for the FAM with Regularly Shaped Objects

The flood area mechanism using one VWM is shown in Figure 6. The running time is the only key parameter that affects the accuracy. Compared to mathematical modeling processes, this approach is unrelated to human experience, and thereby avoids human error. No human decisions are required to select the models used in this CVM process. The extent of human involvement is simply to initiate the CVM process and await the measurement output. Once the VWM is released, there is an absolutely true volume for V_{flood} in a specific virtual space. We mark it using $V_{absolute}$. With the flying of time, the difference between V_{flood} and $V_{absolute}$ can diminish to an ultra-small scale.

As shown in Figure 6, the initial application of the VWD process on a cube-shaped object resulted in 1,438,814 footprints. Of these, 561,025 represented unique visited voxels. The VWMs moved in response to a dynamically shifting gravity field. The efficiency of the process was notably impacted by the VWMs revisiting the same voxels. Figure 7a illustrates that directly applying Equation (4) to the original footprints did not yield accurate results. Consequently, buffering each voxel became an essential step. Figure 7b displays the outcome after buffering by one unit (refer to Figure 7), where speckle patterns on the wall area indicate voxels missed by the VWMs. Compared to the interior, the walls of the vessel were less frequently visited. These isolated speckles could be eliminated using a simple outlier filter, as depicted in Figure 7c. Excessive buffering could encroach upon voxels that are part of the object, resulting in an underestimation of volume, as seen in Figure 7d.

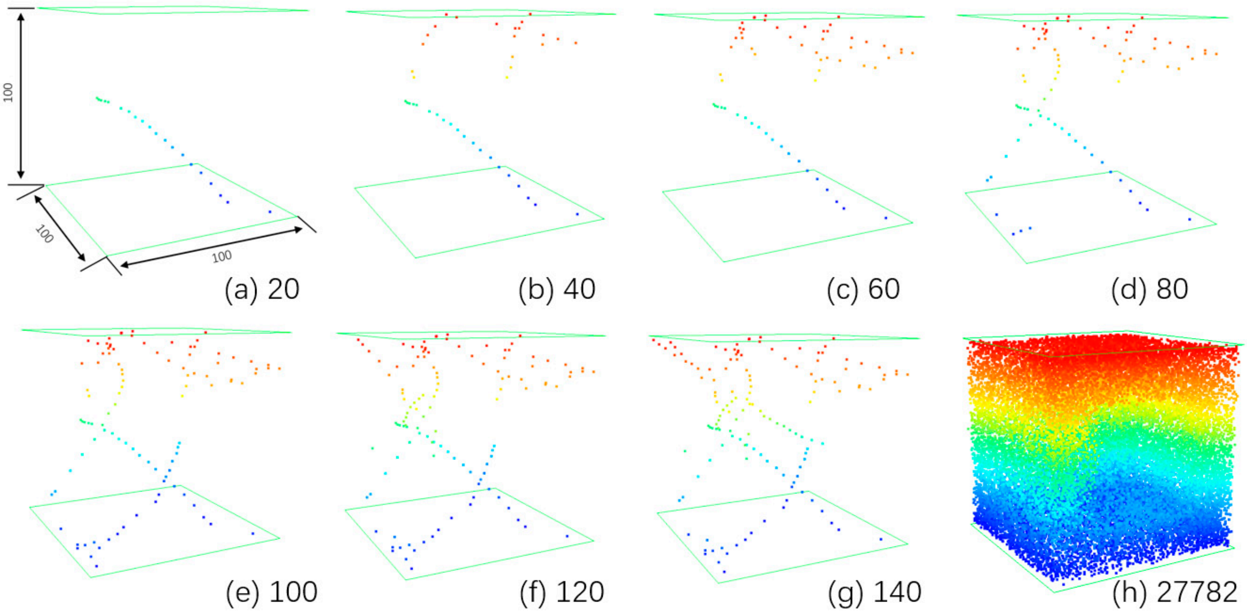

Figure 6. The footprint recording of a VWM. Letters (**a**–**h**) are arranged in ascending order by the application running time. The number is the total generated frames.

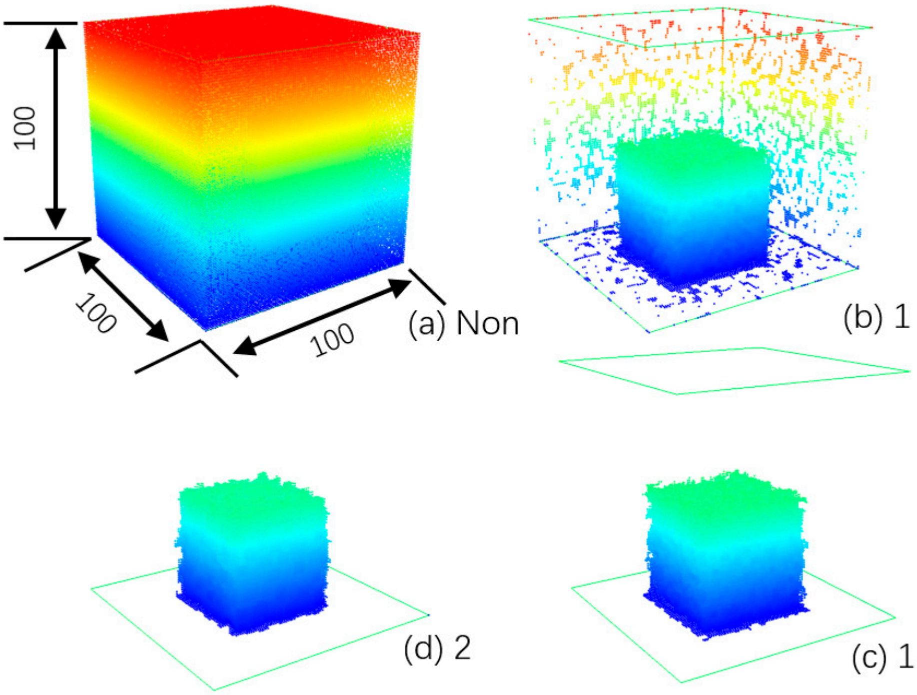

Figure 7. The voxel buffering for the dataset of 1,438,814 footprints. (**a**) No buffering; (**b**) buffered with 1 unit; (**c**) additional outliner cleaning on (**b**); (**d**) buffered with 2 units.

In our preceding research, we noted that the reaction of VWMs differed among three regularly shaped objects. As the diameter of the VWM decreased, the accuracy of the VWD's volume (V) estimation became more sensitive, due to variable responses from the interaction of numerous VWMs. In the current study, we employed either a single VWM or a small number of VWMs to detect the point cloud. The comparative data of the actual volumes and those calculated through VWD evidenced that the area flooding algorithm displayed low sensitivity to the variation in objects (see Figure 8 and Table 1).

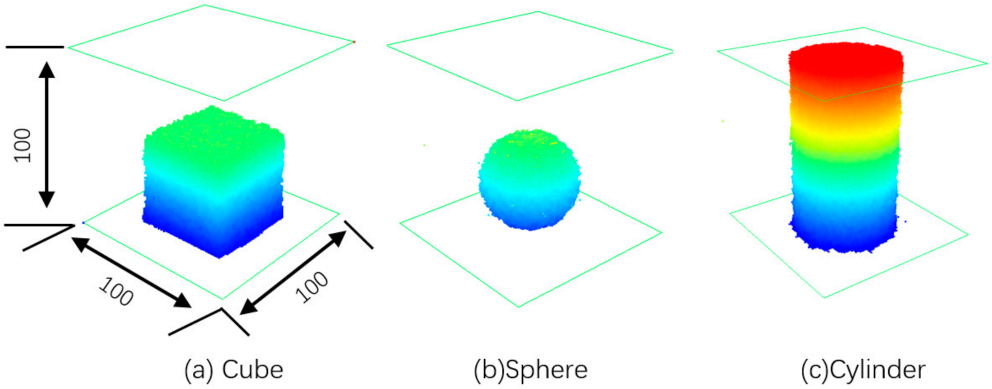

Figure 8. The visualization of the voxels for three regularly shape objects.

Table 1. The results for the VWD process on three regularly shaped objects.

Clouds	True Volume	VWD Volume	Relative Difference	Footprints	Buffering Distance
Cube	125,000	132,845	+6.28%	246,540	1
Sphere	64,043	66,946	+4.53%	436,519	1
Cylinder	193,129	208,673	+8.05%	323,855	1
			Mean: +6.29%		

In Table 1, "clouds" refers to the name list of point clouds; "true volume" refers to the mesh volume of the original 3D models for the clouds; "VWD volume" refers to the point cloud volume measured by the VWD process; "relative difference" refers to the ratio between the volume difference/the true volume; "footprints refers" to the number of VWMs that visited voxels, including the re-visited voxels, in a VWD process; and "buffering distance" refers to the setting of voxel buffer.

3.2. Artificial Stem Results

The applicability of the area flooding algorithm to regular-shaped objects was established in Section 3.1. Advancing from that point, we broadened the application of this algorithm to the point cloud of artificial stems in this analysis. The VWD method furnished data of ground truth caliber, serving as a benchmark for data derived from remote sensing. Consequently, the data visualization (Equation (4) and Figure 9) emerged as a beneficial secondary outcome. This provided a clear illustration of the VWD approach.

Detailed outcomes of the VWD process for both the "stem" and the "stem with branches" point clouds can be examined in Figure 10 and Table 2. The complete filling of the vessel equated to 2,812,500 voxels. Recorded footprints from both instances of the VWD process exceeded this voxel count. Nevertheless, the buffering interval was dependent on the VWD diameter. The buffering unit must be set to less than the theoretical 1 unit because the virtual space represents each position with a tiny sphere rather than a geometric point. This accounts for the likely overestimation of VWD volume. Beyond enhancing time efficiency, buffering further ensures comprehensive coverage by VWMs within all accessible regions. In the case of the "stem" point cloud, a buffering distance of three units was utilized. Every actual footprint affected 343 voxels, leading to a total of 2,428,355,965 visited voxels, approximately 863-fold more than the vessel's voxel capacity. Applying a two-unit buffering to the "stem with branches" point cloud resulted in each physical footprint affecting 125 voxels, tallying to a grand total of 1,148,874,125 visited voxels, or roughly 408 times the full voxel count. Relative to the outcomes pertaining to regular shapes (presented in Section 3.1), overestimations for the "stem" and "stem with branches" point clouds were a mere 2.86% and 1.84%, respectively.

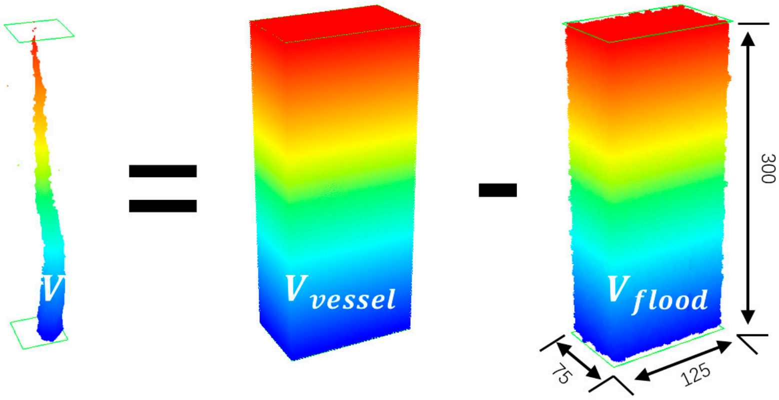

Figure 9. The visual demonstration of Equation (4) using the result of VWD processing on the "stem" data.

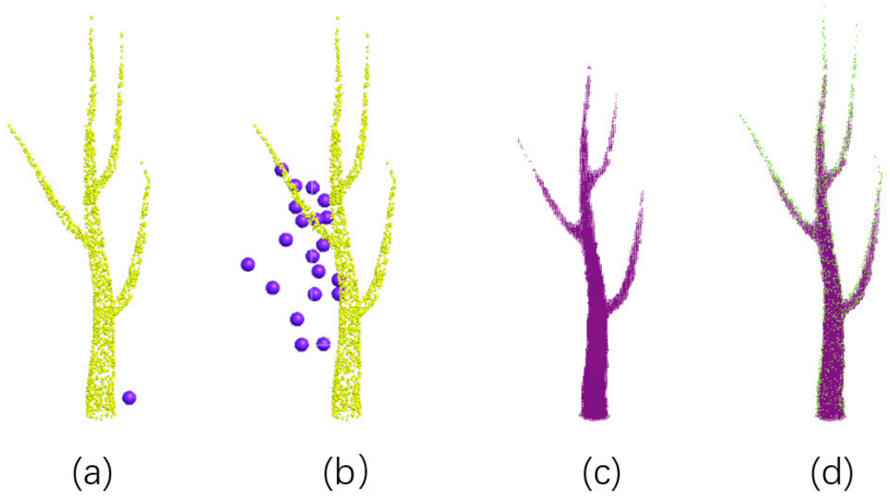

Figure 10. The VWD process for the "stem with branches" point cloud. (**a**) VWD with a single VWM; (**b**) VWD with multiple VWMs; (**c**) VWD result in a voxel; (**d**) overlaying of the VWD result and the original point cloud.

Table 2. The Result for VWD process on two artificial stems.

Clouds	True Volume	VWD Volume	Relative Difference	Footprints	Buffering Distance
Stem	23,709	24,344	+2.68%	7,079,755	3
Branches	27,946	28,459	+1.84%	9,190,993	2
			Mean: +2.26%		

In Table 2, "clouds" refers to the name list of point cloud names; "true volume" refers to the mesh volume of the original 3D models for the clouds; "VWD volume" refers to the point cloud volume measured by the VWD process; "relative difference" refers to the ratio between the volume difference and the true volume; "footprints" refers to the number of VWMs that visited voxels, including the re-visited voxels, in the VWD process; and "buffering distance" refers to the setting of voxel buffer.

4. Discussion

4.1. Tree Volume Determined by Nature Physical Law

Upon inputting a point cloud dataset, a central objective of corresponding computational techniques was to derive specific parameters from each small cluster of points, leveraging their statistical characteristics and geometric traits. The entirety of the tree point cloud might be sectioned into myriad clusters, known as tiny processing units (TPUs), a term coined in our earlier research. At the TPU level, the algorithm consolidates the spatial data from each point to a limited set of parameters. Such a practice could potentially compromise the integrity of the original data, should every point be mistakenly treated as an authoritative representation of its respective segment on the tree. The methodologies in question are vulnerable to three error sources: (i) the segmentation of the complete point cloud into TPUs; (ii) the parameter extraction within each TPU; and (iii) the prediction of the tree's structure.

Considering that LiDAR point cloud data offers a static snapshot of the tree, we envision the full point cloud both as a unified object and a live entity within the virtual space. Concurrently, individual points are perceived as tangible entities within this space. Such is the essential tenet underpinning the exclusion of solid entities by the VWMs. Subsequently, it becomes possible to employ VWMs as a virtual measurement tool to assess the virtual tree, thus circumventing the need for mathematical models aside from physical simulation. This simulation engaged the use of a third-party SDK as a "black box," a component whose internal dynamics we can disregard. Consequently, the VWD method emerges as an approach devoid of parameter dependence, characterized by straightforward mathematical formulation.

4.2. Time as the Only Parameter

Repeated measurements are critical to scientific investigations [27]. Even basic DBH measurements conducted with a diameter tape incorporate gross errors, which necessitate adjustments via further measurements. Moreover, the traditional methodology explained in Equations (1)–(3) generates additional regression errors [28,29]. Our initial VWD study endeavored to eliminate these errors at their origin. In this investigation, however, we utilized one VWM or a few to scan the point cloud, circumventing errors born from VWM interactions.

In an optimal point cloud scenario, no virtual water molecules (VWMs) would infiltrate the tree's structure. As delineated in Section 3.1, deploying designated VWMs within a particular virtual space yields a definitive volume, known as $V_{absolute}$. The incremental operating duration enables the VWD's instantaneous volume, denoted as V_{flood}, to approximate $V_{absolute}$ with increasing precision. This embodies a principal benefit of the computerized simulation. Consequently, we can infer that the VWD approach obviates the necessity for repeated measurements. Thus, the critical factor in VWD execution becomes the duration of operation.

To elucidate the significance of runtime, we engaged a cubic model for footprint record analysis. Findings pertaining to unique voxels identified by the VWMs are depicted in Figure 11 (left). Notably, there was a marked reduction in the absolute value increments subsequent to a pronounced increase during the initial phase. Figure 11 (right) displays the unique voxel increment differentials, computed at intervals corresponding to 0.1% of the overall generated frame count. The diminishing nature of these increments over time is evident, signifying a decreased likelihood of VWMs encountering unvisited voxels as time advances. Ultimately, VWMs are expected to comprehensively examine every reachable region, irrespective of buffering.

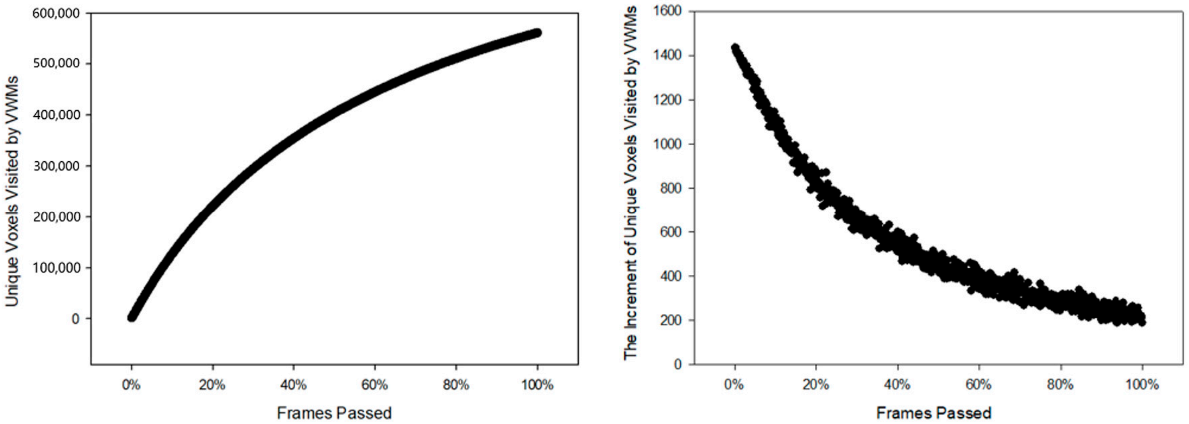

Figure 11. The unique voxels detected by VWMs with the increase in running time of the VWD process for the cube object.

5. Conclusions

In this study, we modified the mechanism of the VWD using one or a limited number of virtual water molecules (VWMs). By applying a gravity field with a random direction and interval in virtual space, the VWMs can visit every accessible area with buffering. This flood area mechanism was testified using three regular objects. The average overestimations of the regular objects were 6.29%, while in the two artificial stems, "stem" and "stem with branches" point clouds, the average overestimations were 2.26%. Thanks to the facilitation of computational simulation, the VWD was established to be free of repeated measurements. With accumulated running time, V_{flood} could approach $V_{absolute}$ as closely as possible. In the next step, we will adopt the VWD method with the current desktop computer system (with PhysX) to obtain a real tree point cloud.

Similar to preceding studies associated with this research, the outcomes of this study cannot be directly applied to processing high-density point clouds scanned by current LiDAR technology. This limitation comes from the collective constraints of current computer hardware technology, general-purpose physical simulation software, and the physical principles underlying LiDAR scanning. As a component of the CVM series research, we remain optimistic that the three aforementioned issues will be individually resolved in the foreseeable future. It is this optimistic anticipation that has motivated us to preemptively undertake a series of studies observing virtual trees in virtual spaces using virtual physical instruments. This anticipates a technological direction that we are confident will be realized in the future.

Author Contributions: Conceptualization, Z.W.; Methodology, Z.W. and T.M.; Software, Z.W., X.Z. (Xiaoning Zhang) and T.M.; Validation, X.Z. (Xiaoyuan Zhang); Formal analysis, Z.W. and X.P.; Resources, X.Z. (Xiaoyuan Zhang), T.M. and Z.F.; Data curation, X.Z. (Xiaoning Zhang); Writing—original draft, Z.W. and T.M.; Writing—review & editing, C.S.; Visualization, Z.W.; Supervision, C.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the 5·5 Engineering Research & Innovation Team Project of Beijing Forestry University (BLRC2023A03) and the Natural Science Foundation of Beijing (8232038, 8234065), National Natural Science Foundation of China (42330507), the Key Research and Development Projects of Ningxia Hui Autonomous Region (2023BEG02050), and China Scholarship Council (201306510005).

Data Availability Statement: Data is contained within the article.

Acknowledgments: We would like to express our gratitude to Yufei Wang for his substantial support towards this work and future CVM studies since 7 March 2024.

Conflicts of Interest: We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work.

References

1. Murtiyoso, A.; Holm, S.; Riihimäki, H.; Krucher, A.; Griess, H.; Griess, V.C.; Schweier, J. Virtual forests: A review on emerging questions in the use and application of 3D data in forestry. *Int. J. For. Eng.* **2024**, *35*, 29–42. [CrossRef]
2. Fol, C.; Murtiyoso, A.; Griess, V. Feasibility study of using virtual reality for interactive and immersive semantic segmentation of single tree stems. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2022**, *48*, 95–99. [CrossRef]
3. Henniger, H.; Huth, A.; Frank, K.; Bohn, F.J. Creating virtual forests around the globe and analysing their state space. *Ecol. Model.* **2023**, *483*, 110404. [CrossRef]
4. Rossmann, J.; Schluse, M.; Schlette, C. The virtual forest: Robotics and simulation technology as the basis for new approaches to the biological and the technical production in the forest. In Proceedings of the 13th World Multi-Conference on Systems, Cybernetics and Informatics, WMSCI, Orlando, FL, USA, 10–13 July 2009; pp. 43–48.
5. Vagizov, M.; Istomin, E.; Miheev, V.; Potapov, A. Visual digital forest model based on a remote sensing data and forest inventory data. *Remote Sens.* **2021**, *13*, 4092. [CrossRef]
6. Wang, Z.; Lu, X.; An, F.; Zhou, L.; Wang, X.; Wang, Z.; Zhang, H.; Yun, T. Integrating Real Tree Skeleton Reconstruction Based on Partial Computational Virtual Measurement (CVM) with Actual Forest Scenario Rendering: A Solid Step Forward for the Realization of the Digital Twins of Trees and Forests. *Remote Sens.* **2022**, *14*, 6041. [CrossRef]
7. Wang, J.; Li, Y.; Liu, H.; Hao, F. Design and implementation for virtual forest scene synthesis-editing system. In Proceedings of the 3rd International Conference on Internet of Things and Smart City (IoTSC 2023), Virtual, 24–26 March 2023; pp. 477–481.
8. Williams, B.; Ritsos, P.D.; Headleand, C. Virtual forestry generation: Evaluating models for tree placement in games. *Computers* **2020**, *9*, 20. [CrossRef]
9. Li, Z.; Zhang, Y.-D. 3D reconstruction method of forest landscape based on virtual reality. *Multimed. Tools Appl.* **2020**, *79*, 16369–16383. [CrossRef]
10. Kortsenshteyn, N.; Gerasimov, G.Y.; Petrov, L.; Shmel'kov, Y.B. A Software Package for Simulating Physicochemical Processes and Properties of Working Fluids. *Therm. Eng.* **2020**, *67*, 591–603. [CrossRef]
11. Mitchell, M.; Udugama, I.; Currie, J.; Yu, W. Software integration for online dynamic simulation applications. In Proceedings of the 2017 6th International Symposium on Advanced Control of Industrial Processes (AdCONIP), Taipei, Taiwan, 28–31 May 2017; pp. 360–364.
12. Slobodyanyk, O.V.; Fedorenko, V.P.; Bolilyi, V.O.; Dmytruk, V.I.; Kushnarov, V.V. Research of the efficiency of using software products for simulation of physical processes. *Inf. Technol. Learn. Tools* **2021**, *82*, 199–214.
13. Wang, Z.; Shen, Y.-J.; Zhang, X.; Zhao, Y.; Schmullius, C. Processing Point Clouds Using Simulated Physical Processes as Replacements of Conventional Mathematically Based Procedures: A Theoretical Virtual Measurement for Stem Volume. *Remote Sens.* **2021**, *13*, 4627. [CrossRef]
14. Li, Z.; Kuang, F.; Zhang, Y. Real-time physically plausible simulation of forest. *Graph. Vis. Comput.* **2021**, *4*, 200025. [CrossRef]
15. Vohera, C.; Chheda, H.; Chouhan, D.; Desai, A.; Jain, V. Game engine architecture and comparative study of different game engines. In Proceedings of the 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 6–8 July 2021; pp. 1–6.
16. Yoon, J.; Son, B.; Lee, D. Comparative study of physics engines for robot simulation with mechanical interaction. *Appl. Sci.* **2023**, *13*, 680. [CrossRef]
17. Gonzalez-Franco, M.; Cohn, B.; Ofek, E.; Burin, D.; Maselli, A. The self-avatar follower effect in virtual reality. In Proceedings of the 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Atlanta, GA, USA, 22–26 March 2020; pp. 18–25.
18. Gabajová, G.; Krajčovič, M.; Matys, M.; Furmannová, B.; Burganová, N. Designing virtual workplace using unity 3D game engine. *Acta Tecnol.* **2021**, *7*, 35–39. [CrossRef]
19. NVIDIA Corporation. PhysX SDK. Available online: https://developer.nvidia.com/physx-sdk (accessed on 10 April 2024).
20. Chopra, A. *Introduction to Google Sketchup*; John Wiley & Sons: Hoboken, NJ, USA, 2012.
21. Unity Technologies. Unity User Manual (2018.2). Available online: https://docs.unity3d.com/Manual/UnityManual.html (accessed on 1 February 2018).
22. Zhang, C.; Chen, T. Efficient feature extraction for 2D/3D objects in mesh representation. In Proceedings of the 2001 International Conference on Image Processing (Cat. No. 01CH37205), Thessaloniki, Greece, 7–10 October 2001; pp. 935–938.
23. Rusu, R.B.; Cousins, S. Point cloud library (pcl). In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 1–4.
24. Schroeder, W.J.; Martin, K.M.; Lorensen, W.E. The design and implementation of an object-oriented toolkit for 3D graphics and visualization. In Proceedings of the Seventh Annual IEEE Visualization'96, San Francisco, CA, USA, 27 October–1 November 1996; pp. 93–100.
25. Hifi, M.; M'hallah, R. A literature review on circle and sphere packing problems: Models and methodologies. *Adv. Oper. Res.* **2009**, *2009*, 150624. [CrossRef]
26. Huang, W.; Yu, L. Serial symmetrical relocation algorithm for the equal sphere packing problem. *arXiv* **2012**, arXiv:1202.4149.
27. Hedayat, A.; Afsarinejad, K. Repeated measurements designs, I. In *A Survey of Statistical Design and Linear Models*; Springer: New York, NY, USA, 1975; pp. 229–242.

28. Gertner, G.Z. The sensitivity of measurement error in stand volume estimation. *Can. J. For. Res.* **1990**, *20*, 800–804. [CrossRef]
29. Pearson, T.R.; Brown, S.L.; Birdsey, R.A. *Measurement Guidelines for the Sequestration of Forest Carbon*; General Technical Report NRS-18; US Department of Agriculture, Forest Service, Northern Research Station: Newtown Square, PA, USA, 2007; Volume 18, 42p.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.