Vegetation Realistic Visualization in Virtual Reality Scenarios

Laura Ozoliņa
Vidzeme University of Applied Sciences
Valmiera, Latvia
laura.ozolina15@va.lv

Arnis Cīrulis
Vidzeme University of Applied Sciences
Valmiera, Latvia
arnis.cirulis@va.lv

Lauris Taube
Vidzeme University of Applied Sciences
Valmiera, Latvia
lauris.taube@va.lv

***Abstract*-This paper focuses on optimizing realistic 3D vegetation models for standalone virtual reality experiences considering the visualization of the plant growth over time. Most of the existing solutions are focusing on personal computer based virtual reality utilization, which allows for more computational power, yet it restricts the movement of the user. Experiences meant for standalone virtual reality headsets mainly feature low-poly and unrealistic plant representations. Since plants, especially trees, can be rather complex structures it is important to find a balance between realistic depiction and performance. By achieving this it would allow the headset user to have better immersion in the virtual world and would reduce the risks of cybersickness. To test and find out the most appropriate solution we created a 3D model of an apple tree and its growth animation in SpeedTree software and then created a scene of varying numbers of trees using Unreal Engine 5.3 and tested framerate count using Meta Quest 2 and OVR Metrics tool. Tests were split into two big groups – static mesh tests, where we observed the changes in framerate before and after optimizing and skeletal mesh tests, where framerate changes were noted as the tree growth animation was running. The results of the tests showcased that it might be difficult to visualize growth process of the plants by using animated models, since their performance was way below the target rate for the headset, while using static meshes and swapping the models over time to simulate growth yielded a better framerate.**

Keywords— performance optimization, standalone virtual reality, vegetation visualization, virtual reality.

I. Introduction

As of now virtual reality has a wide range of uses, such as entertainment, education, art and so on. Unlike the classic way where the virtual worlds are experienced through computer screens, virtual reality has been able to provide a higher level of immersion by considering the user's body and hands [1]. Virtual reality can also be experienced by people with mobility issues by using flexible and adaptive control schemes [2]. However, it is important to note that there are two main ways how virtual reality can be experienced and there are noticeable differences between them. One of the options is PCVR which stands for personal computer based virtual reality, where virtual headset is tethered to a computer and uses it for rendering and calculations. While this way allows the use of more complex objects and shaders within the environment it also restricts the movement of the user, since the headset needs to be connected to the computer. Standalone VR fixes this concern, but new ones arise – standalone headsets have less computational power.

One of the metrics that can be used to compare headset performance against a computer is FLOPS, which stands for floating-point operations per second [3]. This measure plays a vital role in three-dimensional scene rendering and higher FLOPS would provide more realistic and complex graphics and make the experience smoother and engaging [4]. For example, most sold headset, as of now, Meta Quest 2 [5] uses Adreno 650 GPU which offers approximately 1250 gigaflops [6], meanwhile NVIDIA GeForce RTX 4090 can provide 82600 gigaflops [7], which 66 times more than Quest 2. This means that it might be complicated to render realistic and complex geometry in standalone devices.

One of the most complex geometries to render are trees since they consist of a trunk, multiple branches and countless leaves, which all together increase the polygon count of the object and make rendering of multiple meshes a challenging ordeal [8]. There is previous research available regarding the improvement of tree rendering in virtual reality, but most of these papers focus on tethered headsets. It is also important to note that there is a distinct lack of research done regarding vegetation simulation in a timescale, which would help to showcase the growth of the vegetation after a certain time based on various factors,

Online ISSN 2256-070X
https://doi.org/10.17770/etr2025vol2.8584

such as light intensity, proximity to other vegetation, etc. Such visualizations could prove useful in botanic garden education, allowing future professionals to experiment with gardening and get predictions of how the garden would look in the future. It has been proven before that use of virtual and augmented reality in education can enhance learning and memory of the students and improve their social and collaborative skills [9]. Virtual gardens could help to improve human wellbeing. A short walk through a virtual forest can improve the emotional state of young adults and seniors who may not have a chance to explore the real environment due to sickness or restrictions, such as ones imposed during the Covid-19 pandemic [10]. Urban landscaping also could benefit from such visualizations, since it would allow for a long-term planning and management of tree planting initiatives, which can prove to be rather challenging due to environmental conditions in the cities and the maintenance, which may negatively impact the growth of the tree [11]. Thus, a virtual reality solution which could not only visualize various types of vegetation but also could showcase the growth of it over a time would prove to be useful in many areas.

The purpose of this study is to develop a methodology which would allow to efficiently visualize changes in plants over a time in a virtual reality scenario for a standalone device while considering the performance and visual fidelity of the environment.

We propose a hypothesis that by combining optimizing techniques, such as instancing and level of detail, it is possible to enable realistic vegetation visualization in standalone virtual headsets and keep visual fidelity and optimal performance.

A. Plant generation techniques

Historically there are multiple available algorithms which have been used to procedurally generate various types of vegetation, such as L-system, also known as Lindenmayer system, which uses grammar-based system and iterative rules [12]. A space colonization algorithm, where plant growth starts from the root and reaches towards attraction points, competing for space [13]. Even though space colonization algorithm can provide the end user with a tree, which has a desired shape, L-system algorithm will be able to create multiple trees faster. Despite the differences, both algorithms can be used to generate trees for various cases [14].

As it can be seen there is a decent amount of research done regarding vegetation modelling in three dimensional environments, yet very little can be found about the vegetation asset suitability for virtual reality scenarios. An approach has been proposed, which allows to generate multiple trees with similar branch patterns and uses LOD based optimization method to change the complexity of the models based on distance [15]. Nevertheless, very little is known about the performance of the generated tree models in a standalone virtual reality headset. Another solution was offered by allowing users themselves to create 3D trees in virtual reality by combining procedural modelling and allowing for a user input in virtual reality, such as sculpting and editing the geometry and using silhouette-based modelling [16]. A rather similar idea was presented where using L-systems and sketch-based modelling could allow users to avoid the learning curves which comes with classic tree modelling software and create simple tree models in virtual reality more efficiently [17].

The techniques mentioned before can help their users to experiment and create their own tree models, yet they do not guarantee how well they would perform on mobile devices and in virtual reality. It is important to further delve into techniques, which could help improve the usability of the 3D models in virtual reality.

B. Asset optimization for virtual reality

While previously mentioned studies were carried out in virtual reality environments, some of them still relied on external computational power and only used headsets to access the environment. This makes it hard to predict how well the created vegetation models will perform in a standalone device. There are multiple factors that need to be considered when optimizing 3D assets for virtual reality headsets.

One of them is the polygon count of the 3D model. By keeping the polygon count as low as possible it will help to reduce the count of calculations needed to render the object [18]. As of now according to Meta documentation Quest 2 can handle from 750000 to 1.8 million polygons per frame, but Quest 3 – 1.3 million to 1.8 million [19]. Yet even though a smaller number of polygons could help to keep the experience smooth it also can make the object appear less detailed and thus less realistic. Still, it may not apply to all 3D objects and it has been found that both position of the object in the scene and the original polygon count of the mesh could impact user perception of the mesh, noting that for meshes with more than 100000 vertices it is possible to remove up to 95% of them and it would not significantly affect the perception [20].

Alongside polygon count an important metric is the draw calls, which happen whenever GPU receives meshes and materials for drawing. Actions, like spawning meshes, changing materials can impact performance if not optimized properly. It is crucial to keep draw calls as low as possible to retain an acceptable framerate [21]. One of the ways it can be achieved is by combining meshes into one, although it can be done only before the application is built, so to make the approach more adaptable a GPU instancing can be applied, which allows GPU to render one mesh with multiple transforms at once [22].

Materials and textures also play a big part in VR experiences, and it has been studied before that the users are way more perceptive of the material properties in virtual reality viewing mode than when the materials were observed through a computer screen [23]. Optimizing them also can improve the performance of the experience. Just like meshes, each material that needs to be rendered would cost a draw call, so trying to keep material count as low as possible or making multiple 3D assets share one material can help lower the performance load [24]. Similar principle could be applied to textures and one of the most used techniques to keep texture count low and optimize performance is to use texture atlases, which are images made up from multiple smaller images [25].

By focusing on these factors, it could help to improve the performance of the 3D assets. Beyond optimizing static 3D assets, dynamic elements such as vegetation growth play a vital role in enhancing realism in virtual environments.

C. Vegetation growth over time period

There are many commercial solutions available, like garden simulation games and experiences, which can visualize changes in vegetation, only a small number of them are meant for virtual reality headsets. For example, Potioneer: The VR Gardening Simulator is an early access game, yet it is only available for PCVR, and the plant growth visualization is very simplified – plant mesh gets scaled up until it resembles a fully grown plant [26]. At Meta Connect 2022 Meta unveiled their open-source meditative gardening experience for WebXR called Project Flowerbed. Just like Potioneer, Project Flowerbed also contained simplified plant meshes and similar growth mechanics which were not accurate to real life [27]. A visually more realistic experience was created by Greener Games called Nature Trek VR, where, just like previously mentioned games, user can plant vegetation and immerse themselves into the scenario [28]. While those experiences can prove to be entertaining and even yield positive results regarding mental wellbeing, they rarely mimic real-life visuals, especially when time factor is being considered. It can be rather challenging to depict realistic plant visuals, especially for a standalone device, so in the next chapter we investigate ways how to make scenarios like these more doable.

II. MATERIALS AND METHODS

A. Virtual reality scene creation

To test the possible gains regarding optimization of plant meshes a realistic apple tree model and its growth animation were created using SpeedTree software. There the number of vertices was reduced to an acceptable amount, at the same time making it certain that the model does not lose its visual fidelity.

To test the model before and after optimization a simple virtual reality scene was created using Unreal Engine 5.3 and the scene was tested using Meta Quest 2 headset and the performance metrics were gained using OVR Metrics Tool.

A simple virtual reality scene was created consisting of ground and sky, with dynamic lighting turned off. Multiple tests were run, each of them featuring a different number of trees, which were arranged in a grid (See Fig.1). Even though it's possible that in real time such an arrangement might be unlikely, to stress test the model and keep the object placement consistent we chose to use this approach.

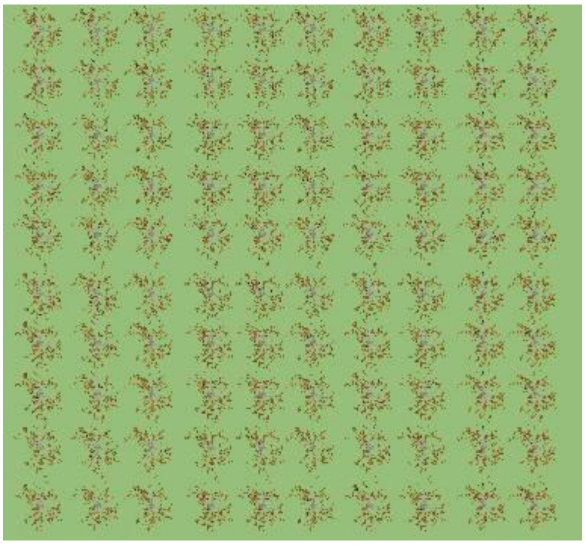

Fig. 1. Top view of the arranged apple tree models.

B. Testing the model performance

The tests were divided into two groups – one of them tested the performance of static tree models, which do not change their geometry at runtime, while the other focused on skeletal tree meshes, which can change their geometry. For the static model a fully grown representation of apple tree was used, which is the most complex one, while in skeletal meshes a growth animation was being played.

For the first test 30 static apple tree meshes were placed in the scene, each of them being rendered separately by the engine. Models were not further optimized in the engine and were kept the same as they were when they got imported into the project. No LOD methods were applied, which meant that the polygon count of the mesh would remain unchanged despite the distance from the camera. We then ran the test two more times with 50 trees and after that with 100 trees.

In the next testing phase instead of using separate meshes an instancing was used, which combined all the meshes into one entity. In Unreal Engine that was done by utilizing Hierarchical Instanced Static Mesh component, which would allow us to later combine this method with LOD usage [29]. Just like before, we run the tests with 30, 50 and 100 trees. For the last test we used both instancing and LOD. Four LODs were generated, where the most complex one consisted of 30643 triangles and the least complex one of 3831 triangles (See Fig. 2).

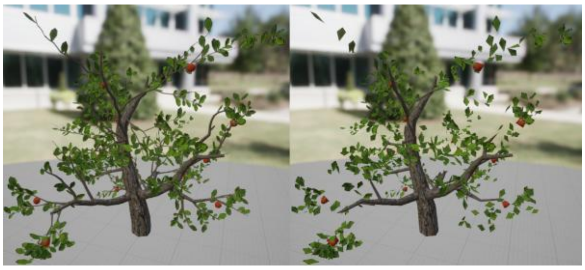

Fig. 2. LOD models of the apple tree. To left – the highest complexity model, to right – the least complex model.

For the tree animations we had two tests – one with two trees and the other one with ten trees. Here we took notice of the framerate when the animation was playing and when it was stopped.

III. RESULTS AND DISCUSSION

Running the scene with 30 static meshes on the headset showed that the average number of frames per second was 50 which is 30.56% less than what is recommended for Meta Quest 2, where interactive applications should achieve 72 frames per second [19]. In the next test with 50 trees, we noticed that the frames dropped by approximately 20% when comparing to the previous test. In the last test where 100 trees were rendered the number of frames remained around 25 and compared to the second test the difference between results had reached 33%. The results show that these unoptimized models cannot reach the target framerate of 72 and by adding more trees the performance would only decrease (See Table 1).

TABLE 1 FRAMERATE FOR STATIC TREES BEFORE AND AFTER USING INSTANCING AND LOD

Number of trees	Framerate before instancing and LOD, fps	Framerate when using instancing only, fps	Framerate when using both instancing and LOD, fps
30	50	72	72
50	40	50	70
100	25	45	55

The next tests using instancing yielded an improved framerate. The test with 30 trees had reached the recommended amount of 72. Then, the number of trees was changed to 50 and frame count decreased by 29%. Furthermore, after we increased the number of trees to 100, the frame count was approximately 45.

In the last phase of testing the static meshes, where both instancing and LOD was applied it was noted that this approach proved to be the most effective, since as seen in Table 1 the framerate is acceptable for up to 50 trees and for 100 trees the difference from the optimal framerate is 26%.

It can be noticed, that by increasing tree count the performance drops (see Fig. 3) even when both optimization techniques have been applied. Still, the performance drop is less rapid than the one which happens without any optimization applied.

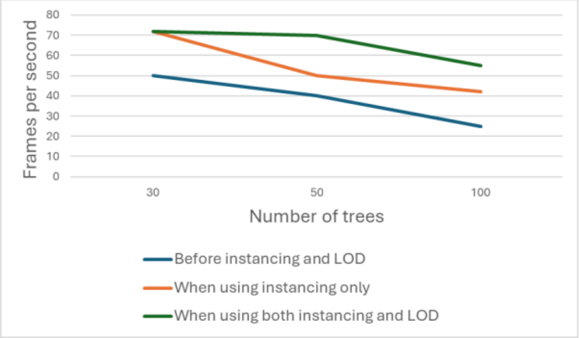

Fig. 3. Performance comparison for varying tree count with different levels of optimization applied.

The animated tree models performed rather poorly, and the framerate neared the target framerate only when the animation on the models was paused. The results are shown in Table 2.

TABLE 2 FRAMERATE FOR ANIMATED TREES WHEN ANIMATION IS RUNNING AND WHEN IT IS PAUSED

Number of trees	Framerate when animation is played, fps	Framerate when animation is paused, fps
2	45	70
10	28	65

IV. CONCLUSIONS

The gained results show that it can be rather challenging to optimize tree models for standalone virtual reality applications, especially if one wishes to visualize tree growth over time. As we saw in Table 2, where we observed the changes in the framerate when the tree was being animated, and its geometry became more complex, the end performance might not be acceptable as it is way below the target framerate for Meta Quest 2 headsets. It is easy to notice that by increasing the number of trees the performance becomes worse, thus it may cause the loss of immersion and overall discomfort for the end user.

We can see in Table 1 that the performance for static meshes when combined with various optimization techniques proves to be better than the animated counterparts. To showcase the growth of the tree we propose the idea of generating multiple static meshes from the animated tree file and swapping them to show the changes in the vegetation. By applying instancing and LOD it is possible to keep the performance at an acceptable level if no more than fifty trees are being rendered. While these results support the hypothesis, further studies are needed to evaluate how the methodology performs with different types of vegetation or on newer generations of VR hardware. Also, further tests with an increased number of trees would be fruitful.

V. ACKNOWLEDGMENTS

The authors would like to express gratitude to Socio-technical systems engineering institute of Vidzeme University of Applied Sciences for the grant funding "Sociotehnisko sistēmu inženierijas institūta grants.", project name "Veģetācijas reālistiska vizualizācija virtuālās realitātes scenārijos." to make this study possible.

We also want to thank for the received information and support to Institute of Horticulture.

REFERENCES

[1] M. Slater, "Immersion and the illusion of presence in virtual reality," British Journal of Psychology, vol. 109, no. 3, pp. 431–433, 2018, doi: https://doi.org/10.1111/bjop.12305.

[2] K. Gerling, P. Dickinson, K. Hicks, L. Mason, A. L. Simeone, and K. Spiel, "Virtual Reality Games for People Using Wheelchairs", in Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 2020, pp.1–11 https://doi.org/10.1145/3313831.3376265.

[3] Indiana University, "Understand measures of supercomputer performance and storage system capacity-IUKB," 2016. [Online] Available: https://servicenow.iu.edu/kb?id=kb_article_view&sysparm_article=KB0023145 [Accessed: November 22, 2024]

[4] Lenovo, "The Importance of FLOPS and its Impact on Your PC's Speed and Efficiency | Lenovo US," 2021. [Online] Available: https://www.lenovo.com/us/en/glossary/flops/ [Accessed: November 22, 2024]

[5] T. Alsop, "VR headset unit sales worldwide 2024," Statista, 2021. [Online] Available: https://www.statista.com/statistics/677096/vr-headsets-worldwide/ [Accessed: November 22, 2024]

[6] D. Heaney, "So Just How Powerful Is Oculus Quest 2, Really?," UploadVR, 2020. [Online] Available: https://www.uploadvr.com/oculus-quest-2-benchmarks/ [Accessed: November 22, 2024]

[7] A. Bradley, "The best graphics card in 2019," PCgamer, 2019. [Online] Available: https://www.pcgamer.com/the-best-graphics-cards/ [Accessed: December 1, 2024]

[8] Q.-L. Zhang and M.-Y. Pang, "A Survey of Modeling and Rendering Trees," Lecture Notes in Computer Science, pp. 757–764, 2008, doi: https://doi.org/10.1007/978-3-540-69736-7_80.

[9] G. Papanastasiou, A. Drigas, C. Skianis, M. Lytras, and E. Papanastasiou, "Virtual and augmented reality effects on K-12, higher and tertiary education students' twenty-first century skills," Virtual Reality, vol. 23, no. 4, Aug. 2019, doi: https://doi.org/10.1007/s10055-018-0363-2.

[10] S. H. M. Chan, L. Qiu, G. Esposito, K. P. Mai, K.-P. Tam, and J. Cui, "Nature in virtual reality improves mood and reduces stress: evidence from young adults and senior citizens," Virtual Reality, vol. 27, Nov. 2021, doi: https://doi.org/10.1007/s10055-021-00604-4.

[11] J. M. Vogt, S. L. Watkins, S. K. Mincey, M. S. Patterson, and B. C. Fischer, "Explaining planted-tree survival and growth in urban neighborhoods: A social–ecological approach to studying recently-planted trees in Indianapolis," Landscape and Urban Planning, vol. 136, pp. 130–143, Apr. 2015, doi: https://doi.org/10.1016/j.landurbplan.2014.11.021.

[12] P. Prusinkiewicz and A. Lindenmayer, The algorithmic beauty of plants. Berlin, Heidelberg: Springer-Verlag, 1990.

[13] A. Runions, B. Lane, and P. Prusinkiewicz, Modeling Trees with a Space Colonization Algorithm. The Eurographics Association, 2025. [Online].Available: https://doi.org/10.2312/NPH/NPH07/063-070 [Accessed: December 1, 2024]

[14] H. Nuic and Z. Mihajlovic, "Algorithms for procedural generation and display of trees," 2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 230–235, May 2019,doi: https://doi.org/10.23919/mipro.2019.8757140.

[15] J. Kim, "Modeling and Optimization of a Tree Based on Virtual Reality for Immersive Virtual Landscape Generation," Symmetry, vol. 8, no. 9, p. 93, Sep. 2016, doi: https://doi.org/10.3390/sym8090093 .

[16] Z. Liu et al., "Interactive Modeling of Trees Using VR Devices," 2019 International Conference on Virtual Reality and Visualization (ICVRV), pp. 69–75, Nov. 2019, doi: https://doi.org/10.1109/icvrv47840.2019.00020.

[17] M. Steiner, "Sketch based L-systems for tree modeling in virtual reality" M.S. thesis. Vienna University of Technology, Vienna, 2023.

[18] R. Leskovsky, E. Kucera, O. Haffner, Jakub Matisak, D. Rosinová, and E. Stark, "A Contribution to Workplace Ergonomics Evaluation Using Multimedia Tools and Virtual Reality," Federated Conference on Computer Science and Information Systems, Sep. 2019, doi: https://doi.org/10.15439/2019f292.

[19] Meta. "Meta Developers," 2024. [Online] Available: https://developers.meta.com/horizon/documentation/unity/unity-perf/ [Accessed: January 19, 2025]

[20] D. V. Nguyen, T. T. Hien, and T. T. Huong, "A Subjective Quality Evaluation of 3D Mesh With Dynamic Level of Detail in Virtual Reality," 2024 IEEE International Conference on Image Processing (ICIP), pp. 1225–1231, Oct. 2024, doi: https://doi.org/10.1109/icip51287.2024.10647962.

[21] Meta. "Meta Developers," 2024. [Online] Available: https://developers.meta.com/horizon/documentation/unity/po-draw-call-analysis/ [Accessed: January 28, 2025]

[22] Y. S. Sadek Hosny, M. A.-M. Salem, and A. Wahby, "Performance Optimization for Standalone Virtual Reality Headsets," 2020 IEEE Graphics and Multimedia (GAME), Nov. 2020, doi: https://doi.org/10.1109/game50158.2020.9315059.

[23] M. Niu and C.-H. Lo, "Effects of Roughness and Texture on Surface Material Perception in Virtuļa Environments: The Psychophysics Approach," International Conference on Computational & Experimental Engineering and Sciences, vol. 24, no. 1, pp. 1–3, 2022, doi: https://doi.org/10.32604/icces.2022.08669.

[24] S. Sukaridhoto, A. L. Haz, E. D. Fajrianti, and R. P. N. Budiarti, "Comparative Study of 3D Assets Optimization of Virtual Reality Application on VR Standalone Device," International Journal on Advanced Science, Engineering and Information Technology, vol. 13, no. 3, pp. 999–1008, Jun. 2023, doi: https://doi.org/10.18517/ijaseit.13.3.18375.

[25] M. Oksanen, M. "3D interior environment optimization for VR" B.S. thesis. Turku University of Applied Sciences, Turku, 2022.

[26] Focus on Fun Games. Potioneer: The VR Gardening Simulator [VR game], 2016.

[27] Meta. Project Flowerbed [WebXR game], 2022.

[28] J. Carline. Nature Treks VR. [VR game]. Greener Games, 2018.

[29] Epic Games. "Instanced Static Mesh Component in Unreal Engine | Unreal Engine 5.5 Documentation | Epic Developer Community," Epic Games Developer, 2025. [Online] Available: https://dev.epicgames.com/documentation/en-us/unreal-engine/instanced-static-mesh-component-in-unreal-engine [Accessed: February 5, 2025]