

Description

Occlusion culling is a crucial yet challenging task in mobile gaming, where hardware resources are constrained. Despite advancements in state-of-the-art (SOTA) software rasterization-based occlusion culling (SOC), performance bottlenecks remain a significant issue. This paper introduces a novel SOC framework designed for mobile environments, aiming to maintain high occlusion accuracy while significantly reducing computational overhead.

The key contributions include an automated occluder generation algorithm that minimizes false culling in complex scenes, a mobile-optimized SOC technique that restructures occluder clustering and introduces a novel visibility test, and extensive evaluations across multiple datasets and mobile devices. The framework leverages SIMD architecture and memory pre-fetching to enhance performance while maintaining scene integrity through conservative occluder modifications.

Experimental results show a 4× reduction in CPU time and a 10× lower false culling error rate compared to previous methods. However, limitations include a lack of GPU-based occlusion culling comparisons (e.g., HOQ), potential valid rate degradation in tiny triangle-heavy scenes, and possible visibility update delays due to smooth blending in fast-moving environments.

Overall, this paper presents a practical and efficient approach to occlusion culling for resource-constrained scenarios. While some limitations remain, it offers a strong contribution to real-time rendering by reducing false culling without additional computational cost and optimizing performance for mobile platforms.

Clarity of Exposition

The paper is well-written and follows a logical progression. However, certain parts are slightly difficult to understand intuitively. For instance, the modification of QEM by adding a bias term to the plane equation is mathematically straightforward and conceptually reasonable. However, it would be more convincing if the authors provided a comparative visualization showing how the standard QEM results in mesh contraction while the proposed method leads to a more conservative occlusion.

Additionally, it would be beneficial to see further experimental validation of the conditions introduced by the authors, such as the criteria for "thin" triangles in the simplification process and the energy functions used for determining the offset direction. More empirical analysis in these areas would enhance the credibility of the proposed approach.

Apart from these aspects, the figures and comparative tables are well-structured and clearly explained. Overall, the paper is well-written, but improving the explanations in certain areas would make it more intuitive and accessible to others.

Quality of References

The references include key related studies and state-of-the-art (SOTA) research, which is a positive aspect. The inclusion of Jiaxian Wu et al. (2023) and Silvennoinen et al. (2014) allows for a comparison between recent occlusion culling techniques and traditional methods, which is particularly valuable. Additionally, the references to Epic Games' Unreal Engine documentation and related papers demonstrate the practical applicability of the proposed approach in real-world scenarios.

However, some of the Unreal Engine references are sourced from blogs or online websites documents, which slightly reduces their reliability.

Technical Correctness and Reproducibility

The preprocessing steps and mathematical filters described in the code are logically structured and well-documented. Additionally, the algorithm's pseudo-code is clear and well-presented, making it easy to follow. The figures effectively illustrate fixed parameter and comparisons; however, including a number of parameter value cases in the tables during tests compared to other methods will further improve reproducibility.

Furthermore, the paper lacks a direct comparison with GPU-based occlusion culling methods, which could provide a more comprehensive evaluation. Adding such experiments would strengthen the study by offering a broader perspective on its performance relative to GPU-based techniques.

Validation

The paper evaluates two methods: occluder generation and Mobile-SOC. The occluder generation process is tested on a single CPU configuration, comparing it against four other methods, while Mobile-SOC is evaluated across three different mobile device environments, comparing it with three existing methods. The experiments are well-designed, and the comparative results are clearly presented. However, while the paper highlights the accuracy advantage over HOQ (Hardware Occlusion Queries), it acknowledges that performance could be similar or even worse in some cases. Providing additional evidence to support this claim would further strengthen the validation.

Furthermore, the paper would benefit from a more in-depth analysis of edge cases, such as scenarios where occluder generation struggles with highly detailed geometry or where Mobile-

SOC encounters challenges with rapidly changing visibility conditions. Addressing these situations would enhance the robustness of the evaluation.

Ethics

Experiments were conducted using the data set provided by the previous technology. There is no ethical problem with the dataset or experimental process. Furthermore, it did not disclose any information about Epic Games' gaming scenes.

The study primarily focuses on software occlusion culling for real-time rendering, which has clear applications in mobile graphics. Overall, the research adheres to ethical standards, but a brief discussion of potential unintended applications could improve the ethical consideration of the study.

Private Comments

The study briefly mentions limitations related to tiny triangles, but it does not explore them in detail. Additionally, while the Runtime Visibility Testing introduces smooth blending to prevent flickering, it is unclear how the method reacts in rapidly changing scenes. A deeper analysis of these edge cases would strengthen the study.

The mobile-related optimizations in the paper include the use of the XSIMD library and 128-bit NEON instructions for mobile optimization, as well as vertex data caching and pre-fetching to minimize redundant calculations. However, the primary contributions, Occluder&Occludee Preprocessing, Offset Mesh Generation, and Simplification, are more aligned with general algorithmic optimizations rather than mobile-specific optimizations. It would be beneficial to analyze how much each of these two contributions impacts the final results through experimental comparisons.

Additionally, the comparison with Unreal Engine should be updated to a more recent version. The paper currently uses UE 4.20 from 2018, but newer versions, such as Unreal Engine 5.5, have introduced updates for mobile software occlusion culling. Comparing against a more recent version would provide a more accurate assessment of the proposed method's effectiveness.

paper 요약(review 파트 X)

Occlusion culling is a crucial yet challenging task in the mobile gaming industry, where hardware resources and computational budgets are often limited. Despite the advances in state-of-the-art (SOTA) software rasterization-based occlusion culling (SOC), performance bottlenecks remain a significant issue. This paper presents a novel SOC framework specifically designed for mobile environments, maintaining high occlusion accuracy while significantly reducing computational overhead.

The contributions of this paper can be categorized into three main aspects. First, it introduces an automated occluder generation algorithm that preserves occlusion capability while preventing false culling in complex scenes. Second, the paper proposes a mobile-optimized software occlusion culling technique by restructuring the occluder clustering mechanism and introducing a novel visibility test. Third, extensive evaluations are conducted using various datasets and multiple levels of mobile devices, comparing the proposed method against existing techniques.

The algorithmic structure is mathematically sound and well-improved. The occluder generation process carefully determines the offset direction and distance to reflect the overall geometry. Additionally, simplification issues have been addressed with minor algorithmic modifications. The approach adopts a conservative yet adaptive strategy to modify different conditions, effectively reducing false culling while preserving scene integrity. The framework further leverages SIMD architecture and memory pre-fetching techniques to enhance SOC performance in mobile environments.

Experimental results demonstrate that the proposed method achieves a 4× improvement in CPU time, reducing it from 8-10 seconds to 2-3 seconds, while maintaining similar FPS performance to UE4 SOC. However, the key highlight is the significant reduction in false culling errors, achieving an error rate close to 0%, over 10× lower than previous methods.

Nevertheless, certain limitations exist. The study lacks a detailed comparison of GPU-based occlusion culling techniques such as HOQ, which could provide further insights into accuracy trade-offs. Additionally, the framework does not always guarantee a high valid rate, particularly when dealing with objects composed of numerous tiny triangles, where valid rate degradation may occur. Furthermore, while the Runtime Visibility Testing introduces a smooth transition mechanism by halving the visibility score "v" to prevent flickering in small objects, this approach may also be considered a limitation. The gradual decrease in vvv reduces false culling errors but could lead to delayed visibility updates, potentially causing noticeable visual inconsistencies in fast-moving scenes.

Overall, this paper makes a substantial contribution to real-time rendering in mobile environments. Despite some limitations, it effectively overcomes algorithmic challenges by constructing heuristic occluders that reduce false culling without additional computational cost. This novel approach offers a practical and efficient solution for occlusion culling in resource-constrained scenarios.