УДК 004.92

Klyatchenko Ya. , associate professor, PhD;
Samoilenko Denys, student

National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"

RAY-TRACED SHADOW ATLAS FOR MULTIPLE DYNAMIC LIGHT SOURCES INTEGRATED WITH RASTERIZATION RENDERING

Резюме

К.т.н., доцент Клятченко Я.М., студент Самойленко Д.А.
Атлас рей-трейсингових тіней для великої множини динамічних джерел світла та його інтеграція у рендеринг на основі растеризації

У статті розглянуто задачу побудови тіней у реальному часі для великої кількості динамічних джерел світла. Проаналізовано сучасні методи трасування променів та відображення тіней, визначено їхні обмеження. Запропоновано гібридний підхід з використанням атласу тіней та адаптивної фільтрації, що дозволяє одночасно підтримувати сотні джерел світла.

Introduction

Real-time graphics often need to show many lights casting shadows, yet most engines cannot scale well. Shadow maps are fast but consume lots of memory and require extra passes. Full ray tracing gives accuracy but is far too expensive. Screen-space shadows cut cost but miss off-screen geometry. Our approach is to achieve equilibrium: compute visibility in an atlas where each light gets a tile sized by its importance, then reuse those results in screen space with a simple filter that respects the normal surface. This way we avoid per-pixel rays everywhere, save performance, and still get stable shadows even with hundreds of lights.

Problem Statement

The main objective of this research is to reduce the computational resources required for processing shadows from many light sources by developing and implementing a method capable of supporting the maximum possible number of dynamic shadows at the same time. To achieve this goal, it is necessary to review existing methods that allow rendering a large number of shadows from dynamic light sources and identify their key limitations, to propose a new approach for generating such shadows with improved performance and reduced resource consumption, and to experimentally assess the efficacy of and scalability of the developed method in comparison with existing techniques.

Terminology

VSM (Variance Shadow Mapping) - a shadow mapping method that stores depth and depth squared, which allows a shadow probability to be estimated using variance estimation.

ESM (Exponential Shadow Mapping) - a version of standard shadow mapping that transforms the depth values with an exponential function to achieve softer penumbra edges.

BVH (Bounding Volume Hierarchy) - A hierarchical acceleration structure for ray-tracing acceleration of ray-scene intersections.

G-buffer - A set of screen sized textures that store attributes at the pixel level including world position, surface normal, albedo, roughness, metallic, and other material properties.

SSRT (Screen-Space Ray Tracing) - a ray tracing technique that traces ray in screen space depth and normal textures from the G-buffer.

PCF (Percentage Closer Filtering) - a standard shadow filtering technique where several samples from the shadow map are averaged to create a softer shadow boundary. In our approach we generalize PCF with normal aware weights and kernel size adapting to the screen to tile resolution ratio.

Analysis of existing approaches and algorithms

Modern real-time shadow rendering methods can be grouped into several categories, each with specific trade-offs between accuracy, performance, and memory cost. Shadow mapping remains the most common approach, using rasterized depth from the light's point of view. Although efficient, shadow maps suffer from aliasing and perspective distortion. VSM and ESM improve softness and filtering quality but still rely on fixed-resolution depth projections. Ray-traced shadows, both world-space and screen-space, offer physically correct penumbra and precise contact regions, yet their computational cost are significant in large scenes.

Proposed Method

The core idea of the proposed method is to modify SSRT method and instead of tracing rays for every screen pixel or maintaining a full-resolution shadow map per light, we trace one binary shadow ray per atlas texel and then reconstruct smooth shadow factors in the lighting pass. Atlas tile sizes are chosen from a discrete set (64×64, 128×128, 256×256) according to the distance from the camera to the light's influence boundary. This distance-aware sizing reduces the total number of texels and therefore rays while keeping perceptually important shadows detailed.

A small detail which helps to avoid every update search which light is for this tile is the same atlas size which contains for every pixel an 16-bit ID that identifies the owning light. This indirection decouples ownership lookup from layout changes: tiles can be repacked, updated, or partially reuploaded frequently without having to redo expensive searches to discover which light owns which atlas region.

During the ray generation pass we map each atlas texel back to a screen location that corresponds to the geometry sample this texel represents. Before sampling the G-buffer we apply texel snapping - the atlas-derived screen position is rounded to the nearest integer screen sample. Texel snapping is need to pins the ray origin to a discrete grid so that small camera or object motions do not produce high-frequency jitter in which different subpixel samples are traced each frame.

Reconstructing the shadow factor in the lighting pass uses a normal-aware PCF applied inside the tile bounds. For each shading pixel we compute its relative texture coordinates inside the corresponding tile, then sample a small neighborhood of atlas texel. Each tap’s contribution is weighted by a Gaussian of the angular normal difference between the tap’s normal and the shaded fragment normal. Taps whose normals differ significantly are downweighted. This simple bilateral-like weighting preserves geometric edges and reduces undesired blurring across discontinuities while keeping the filter cheap.

Finally, the PCF kernel size is adaptive: it depends on the ratio between the screen sampling density and the tile resolution. When a tile is heavily downsampled, a larger PCF kernel is used to hide staircase artifacts and produce a soft, plausible penumbra.

Results

The proposed method was evaluated on a desktop setup with a mid-range GPU at 1920×1080 resolution. Scenes contained between 200 and 900 dynamic lights with per-light tile sizes ranging from 64 to 256 pixels.

A key observation was the appearance of flickering shadows during camera motion. This artifact stems from the fact that several screen pixels are mapped to the same low-resolution atlas texel; as the camera moves, different mixtures of pixels are sampled, producing temporal instability. Texel snapping noticeably reduces this effect, yet some instability remains at very coarse tiles.

During rendering tests, the standard shadow map algorithm could sustain only about 19 shadow maps at ~15 frames per second. In contrast, the modified atlas-based method was able to render roughly 245 shadow maps at the same frame rate. This advantage highlights the efficiency of combining distance-based tile sizing, atlas reuse, and single-ray-per-texel evaluation.

These experimental results indicate that, while quality refinements are still needed - particularly in smoothing temporal flicker - the proposed method offers a significant performance gain and paves the way for practical use of hundreds of dynamic lights in real-time graphics.

Conclusions

We introduced a distance-aware, texel-snapped, ray-traced shadow atlas with normal-aware, adaptive PCF. The method integrates with rasterization pipelines, scales to hundreds of dynamic lights, and delivers stable shadows at predictable cost. It provides a viable alternative for real-time scenes where classic shadow mapping or brute-force ray tracing are impractical.

However, several improvements remain for future work. First, the problem of jumping shadows under camera motion needs to be fully addressed. Second, the method should incorporate an algorithm for soft shadow rendering near the camera. Third, further research is required to optimize the number of traced rays, balancing atlas resolution and quality to reduce GPU workload without visible quality loss.

References

1. Williams, L. Casting Curved Shadows on Curved Surfaces. – *Proceedings of SIGGRAPH*, 1978. – P. 270–274.
2. Reeves, W.T., Salesin, D.H., Cook, R.L. Rendering Antialiased Shadows with Depth Maps. – *Proceedings of SIGGRAPH*, 1987. – P. 283–291.
3. Fernando, R. Percentage-Closer Soft Shadows. – In: *GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics*. Addison-Wesley, 2007. – P. 303–329.
4. Lauritzen, A. Variance Shadow Mapping. – In: Nguyen H. (ed.) *GPU Gems 3*. Addison-Wesley, 2008. – P. 279–292.
5. Schied, C., Dachsbacher, C., Wimmer, M., Salvi, M. Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path-Traced Global Illumination. – *Proceedings of High Performance Graphics (HPG)*, 2017. – P. 1–12.