Problem

- Shadow mapping has become the first choice to generate shadows in virtual environments for real-time applications like e.g. video games.
- Many methods and ways have been introduced to improve performance, shadow quality and minimize the occurring errors like e.g. shadow map aliasing.
- Problem 1: Because of the large number of methods, choosing the right one is very difficult.
- Problem 2: It can be hard to find an appropriate parameter configuration for specific scenes.

Goals

- Analyze and discuss several commonly used advanced shadow mapping techniques.
- Compare them with each other considering the performance and quality improvements.
- Optimize specific methods and determine the enhancement of their characteristics.
- Investigate the results for various parameter configurations and different scenarios.
- Use several visualization tools to find an appropriate technique for each possible outdoor scene.
- Focus on fully hardware-accelerated techniques for real-time capability (~60 FPS per second).

Related work 1: Basic shadow mapping

1. Render the scene from the light’s point of view and store the depth values into a depth map (shadow map).
2. While rendering the scene from the eye’s point of view, perform a linear transformation that maps each point into light-space and apply a visibility test against the values in the shadow map (shadow test) to determine shadowed and unshadowed points.

Light view

Eye view

Includes a visualization of the shadow map level boundaries.

Related work 2: Warping

1. Perform a perspective transform to the light-space (more samples nearby the eye and less samples distant from the eye) and render the scene from the light’s point of view into the shadow map using the transformed light-space.
2. While rendering the scene from the eye’s point of view, map each point into the transformed light-space and apply the shadow test.

Light view

Eye view

Related work 3: Z-partitioning

1. Split the eye view frustum into several split frustums and render the scene objects of each split frustum from the light’s point of view into a single shadow map.
2. While rendering the scene from the eye’s point of view, determine for each point the corresponding split frustum, map the point into the correct shadow map-space and apply the shadow test using the corresponding shadow map.

Light views

Eye view

Contributions

Features of the new shadow mapping framework:
- It supports basic shadow mapping, warping, z-partitioning and several other advanced shadow mapping techniques.
- It supports reasonable combinations of various methods (e.g. warping + z-partitioning).
- It allows experimenting with the implemented methods and interactively adjusting almost every interesting parameter.
- It provides several visualization tools for comparison and analysis of the applied techniques.

Improvements to existing techniques:
- Optimized calculation of the projection center of the perspective transform for warping to improve robustness.
- Automatically adjusted near plane distance for the eye view frustum to improve the shadow quality of some specific methods.

Results 1: Error visualizations

Sampling rate in the shadow map

Perspective aliasing error along the eye view frustum

Results 2: Common visualizations

Intersection bodies (third person view) and light views

Future work

- Improvements to the presented framework:
  - Complete elimination of shadow flocculation between consecutive frames (e.g. usage of temporal coherence).
  - Visualization of sampling rate in the third person view (e.g. computation independent of current screen-space).

- Improvements to shadow mapping:
  - Complete elimination of shadow map aliasing (e.g. alternative partitioning approach).
  - Simulation of physically correct soft shadows (e.g. appropriate sampling of an area light source).

Contact

Martin Stingl (martin.stingl@gmx.at)
Michael Wimmer (wimmer@cg.tuwien.ac.at)