This thesis presents optimized visibility algorithms in the context of a cross-platform rendering engine for PC, PlayStation 3, Xbox 360 and Wii. The main challenges due to varying platform properties are the following:

**Unified API:** Each platform has vastly different CPU and GPU specs, but a rendering engine must provide a unified API while still taking advantage of platform-specific features.

**Efficient algorithms:** State-of-the-art visibility algorithms must be tailored to console hardware in order to be efficient.

### Contributions

**Rendering engine:** The thesis describes the design and implementation of a multi-platform rendering engine, detailing different strategies and approaches for unifying vastly different architectures into the same API.

**Novel occlusion query mechanism:** A low-level mechanism for enabling occlusion query functionality on platforms without hardware-support was invented.

**Improved algorithms:** State-of-the-art algorithms were improved using new branch-free data-structures as well as platform-specific features.

**Optimized CHC++ algorithm:** A tool for evaluating optimal parameters of the CHC++ algorithm was developed, resulting in a 5%-10% performance improvement over the original CHC++ algorithm.

**Improved visibility algorithms**

*Left:* Camera viewpoint. *Right:* Visibility visualization. Visible objects must be rendered, while occluded objects are discarded. The used visibility algorithm is able to cull about 80% of all objects in the scene in this case.

By inserting certain instructions into the command-buffer, the CPU and GPU can be synchronized via interrupts in a way that allows the occlusion query functionality to be implemented on platforms which do not natively support it.