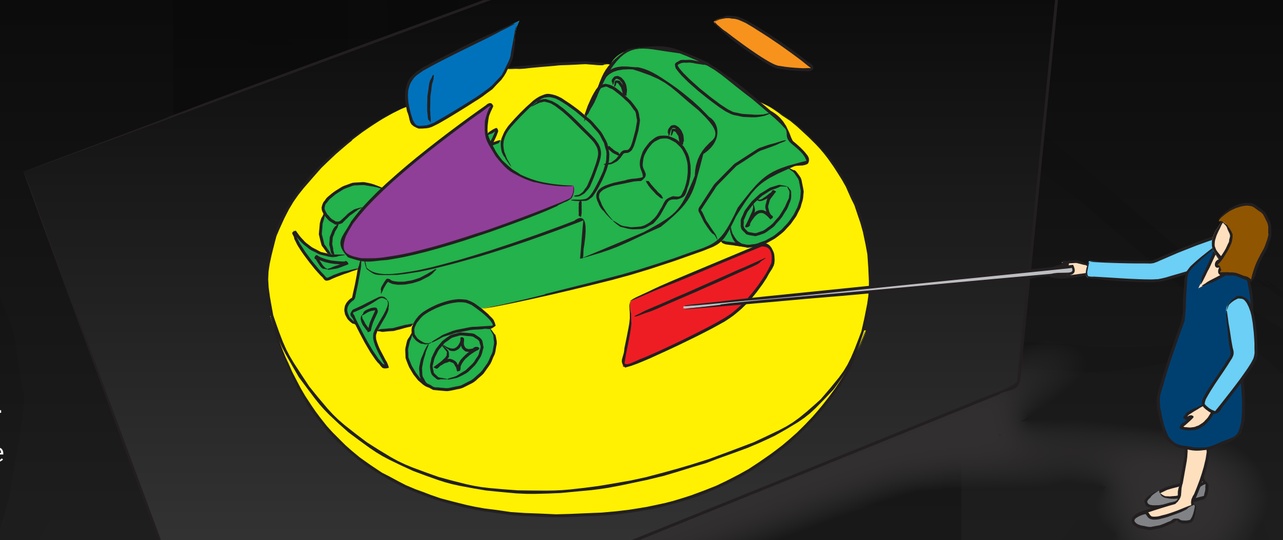
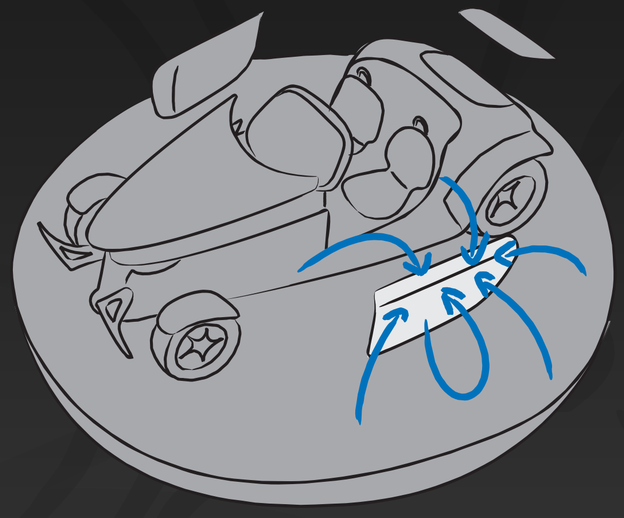
Improving Interaction Performance for Ray Tracing

Daniel Kurz, Christopher Lux, Jan P. Springer, and Bernd Fröhlich — Bauhaus-Universität Weimar, Germany



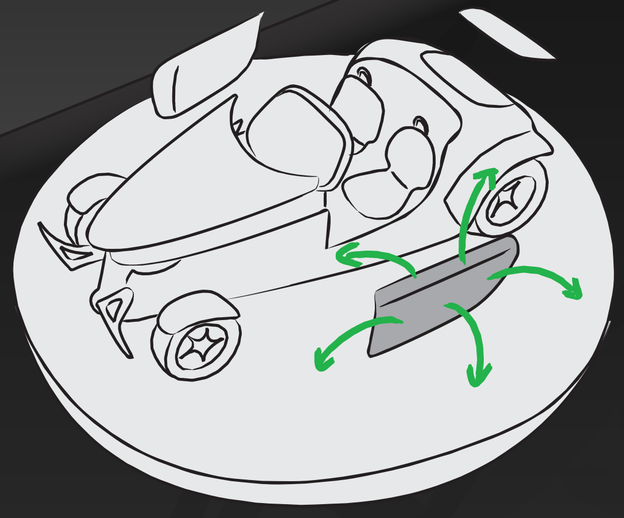
We assume that users *alternate* between navigating a scene and manipulating objects in the scene.

- The scene is divided in objects currently being manipulated and the non-interactive rest.
- During object manipulation the viewpoint is fix.
- First order reflections can then be divided in the following three groups.



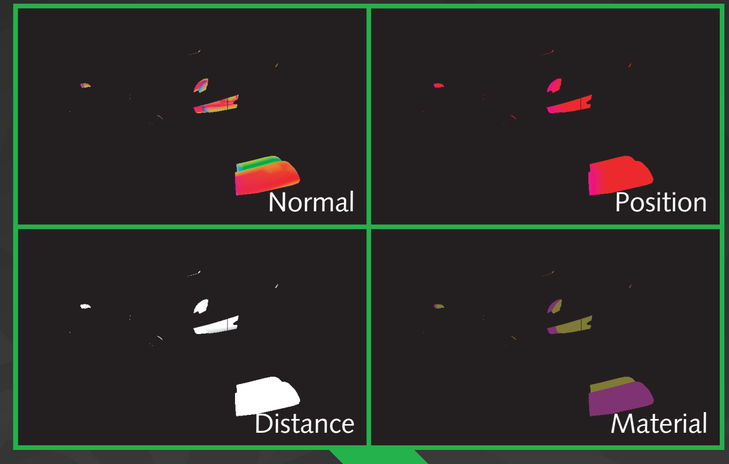
The whole scene reflecting in the manipulated objects

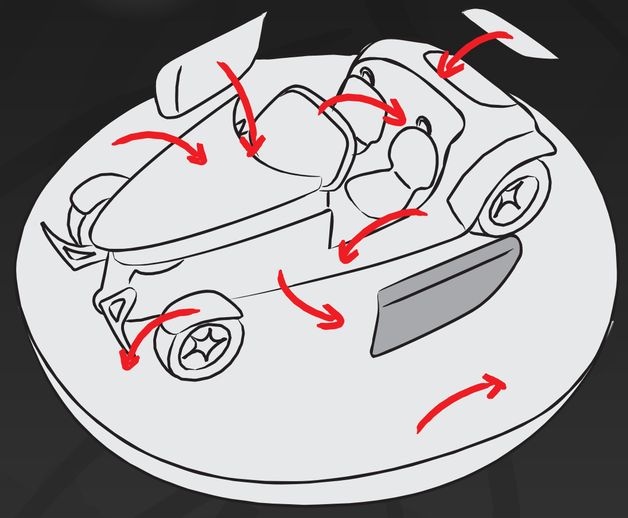
· influences only those pixels showing the active objects
· has to be updated whenever the objects are moved
→ **regular ray tracing** (for the above-named pixels)



The manipulated objects reflecting in the static rest

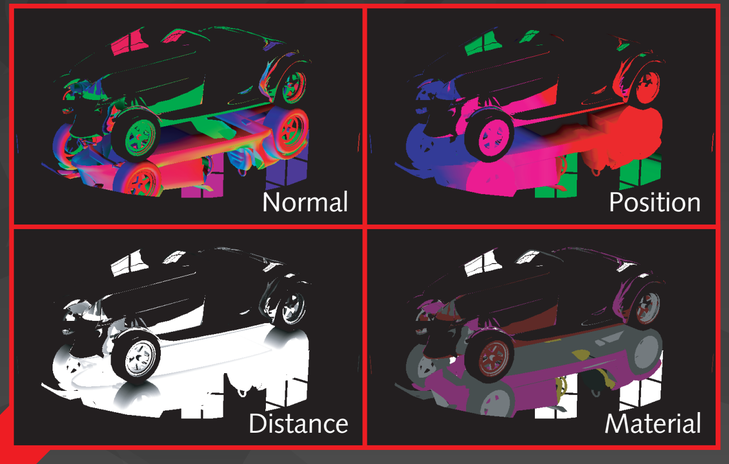
· influences those pixels containing the static rest
· changes when manipulating the objects
→ testing rays against **the manipulated objects only;** intersection properties are stored in G-buffers





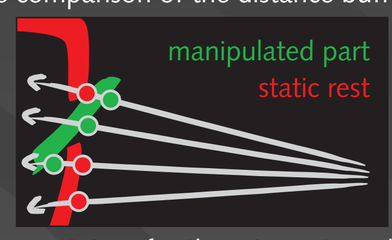
The static rest of the scene reflecting in itself

· influences those pixels containing the static rest
· do **not** change when the objects are moved
→ **pre-computation** (when manipulation starts); properties of the intersection point are stored



Composition of reflections

· pixel-wise comparison of the distance buffers

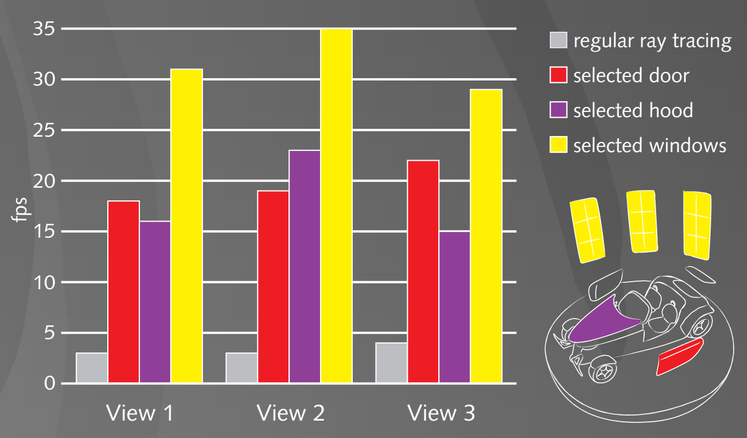


· lighting computation for the intersection closer to the reflecting surface (in case there are two possible)
· parameters from G-buffers (*deferred shading*)
· adding reflections to the local lighting

Final merging

· pixel-wise addition of the two G-buffers

Resulting Performance



Bauhaus-Universität Weimar

email: *{daniel.kurz, christopher.lux, jan.springer, bernd.froehlich}@medien.uni-weimar.de*

web: *http://www.uni-weimar.de/medien/VR*