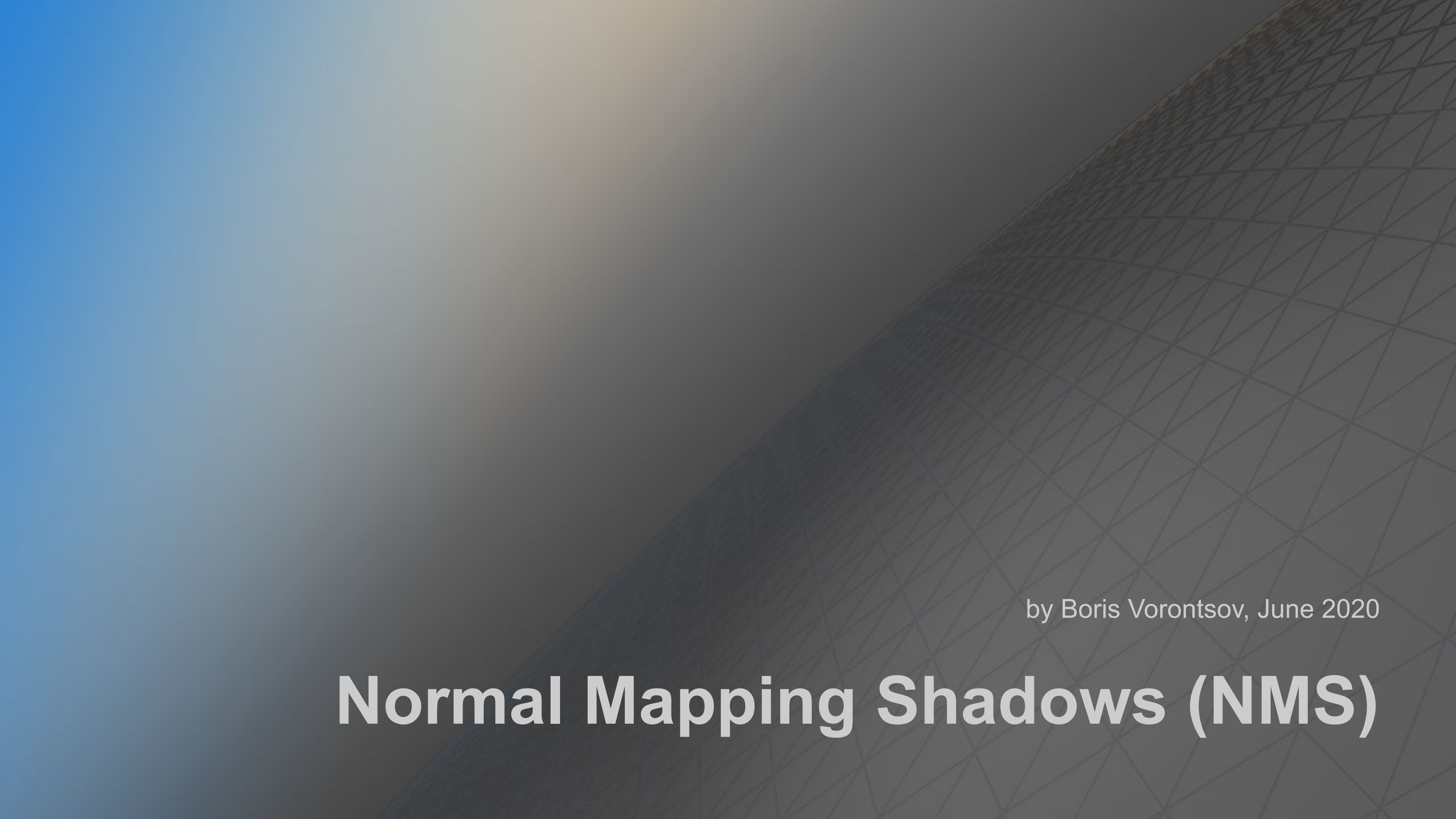


by Boris Vorontsov, June 2020

Normal Mapping Shadows (NMS)

Presenting a Normal Mapping Shadows (NMS) technique to render microsurface self shadowing by using normal maps. No height map is required.

Works well with both forward (directly in the material shader) and deferred (by using GBuffer normal) rendering.

Height map shadows together with parallax effect significantly improve perceived surface depth

+ Great visuals

- Height maps require extra video memory

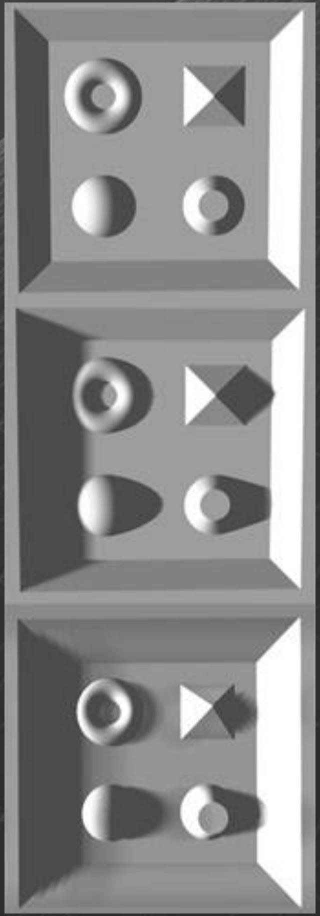
- Often height maps are unavailable or bothersome to author, very few released games have them

But games do have normal maps, why not use them?

- Bump Mapping

- Normal Mapping Shadows (NMS)

- Parallax Occlusion Mapping (POM)



Normal Mapping Shadows

+ Good increase in detail

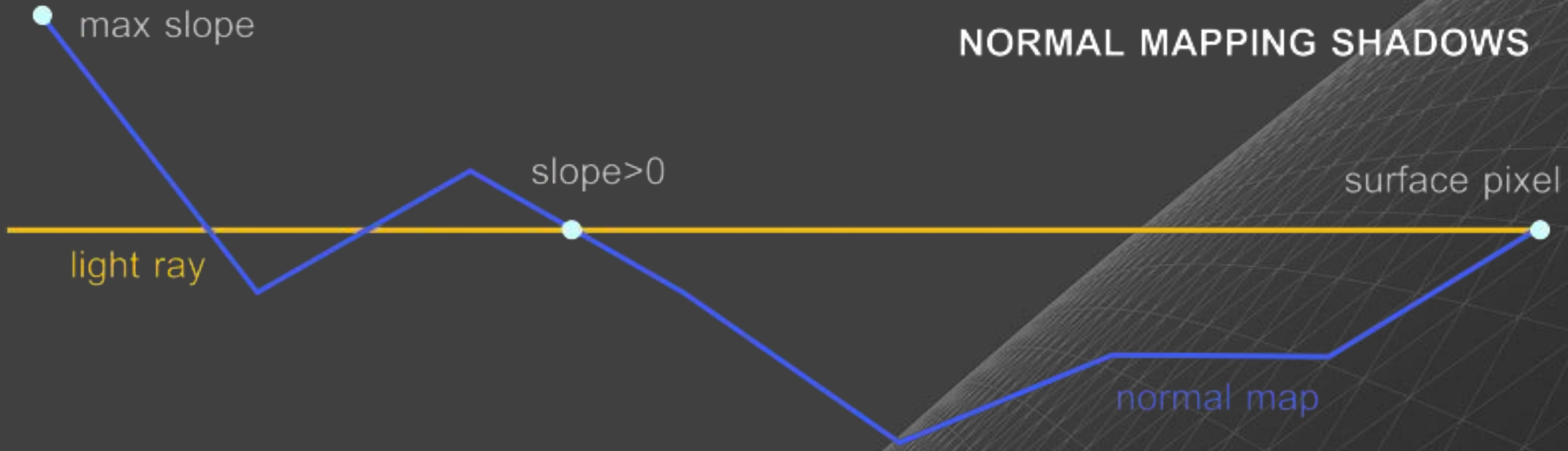
+ Works fine with 2 channels of normal map and reconstructed Z component, saving memory for something more useful than height data (f.e. reflection amount, glossiness)

- Produce wrong result if slopes of normal map are invalid (but the lighting there is wrong anyway)

- No parallax :(

How it works

NORMAL MAPPING SHADOWS



The light ray is an axis for computing slope of normal map. Trace from hit point to light direction and compute sum of dot products between normal map and light direction.

If slope is bigger than 0, pixel is shadowed. If slope is also bigger than previous maximal value, increase hardness of shadow.

- Normal Mapping Shadows effect is similar to parallax technique, but instead of comparing heights of texel and traced vector at each step, compute dot product of lighting and adjust slope as sum of previous dot products
- Optimizations similar to parallax can be applied as well, variable sampling rate and shadow length depending from angle and distance to surface, etc
- Softer shadows possible by various tweaks or interpolation

Forward renderer

+ No issues with occluded objects

+ Wider possibilities for customization, including rendering of translucent objects

- Requires texture "sides" for non tiled maps or transparency to limit tracing distance or visual errors when going outside of UV range of the mesh (same problem as parallax effect)

- Shadows are not cast from other textures and meshes

Deferred renderer

+ No need to change meshes and textures (artists are happy)

+ Better performance by texel and fill rate both for computing NMS (especially when using compute shader)

+ Smooth transition of shadows for all objects, including decals

- Normal map from GBuffer is occluded by objects, in practice barely noticeable as the length of such shadows is short

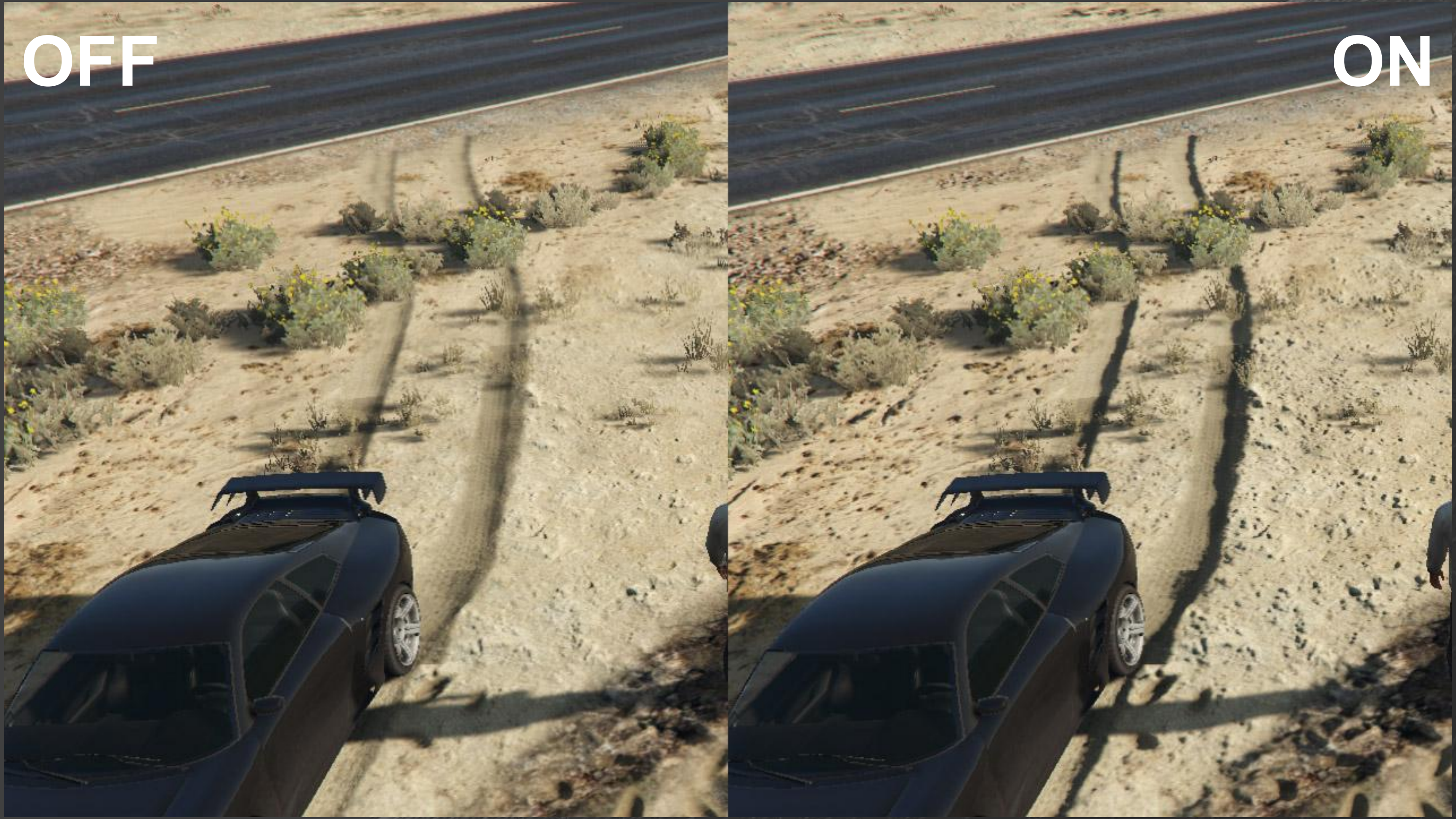
Solutions to issues with deferred render

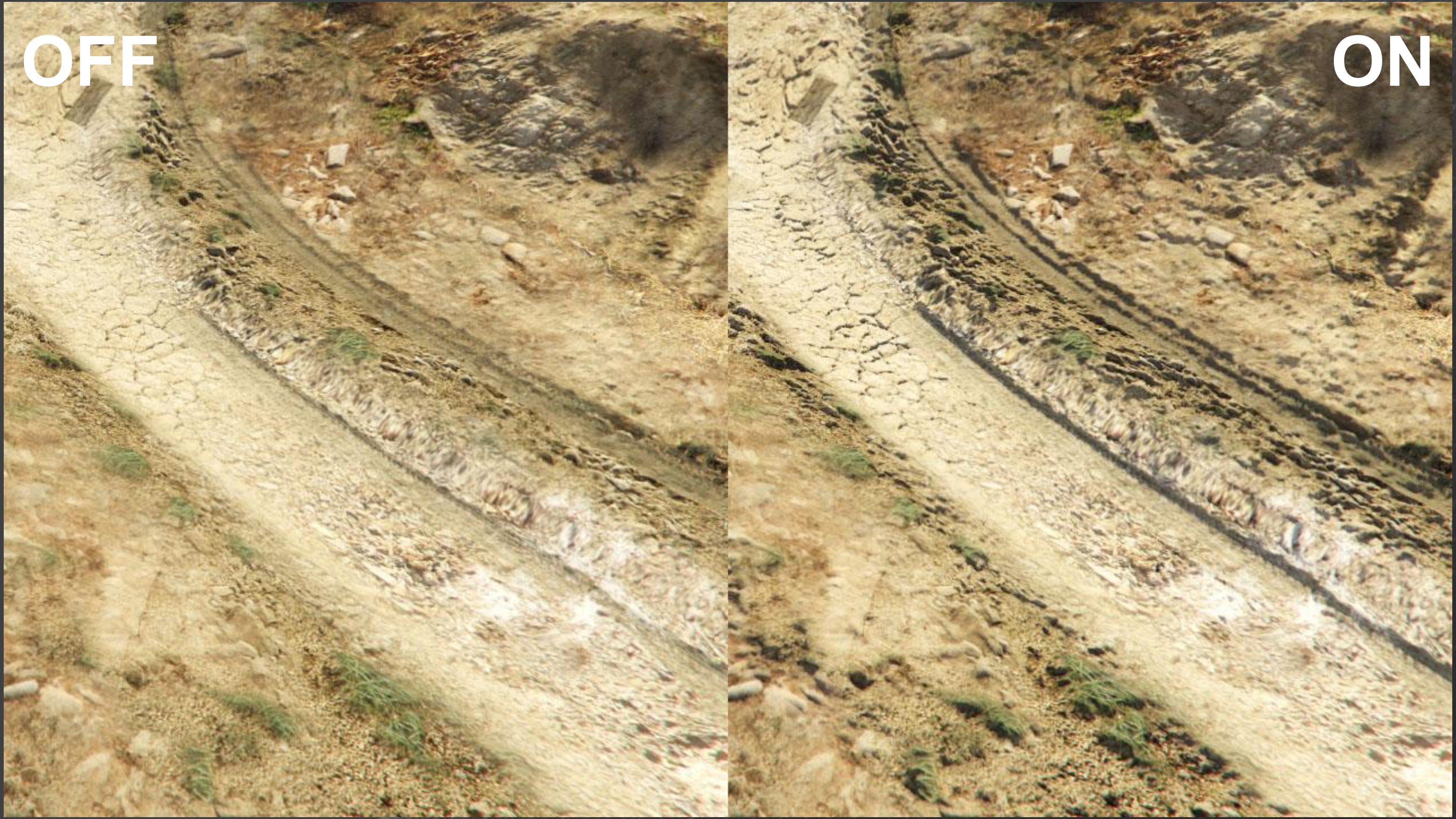
- Stencil mask for characters, vehicles to not apply Normal Mapping Shadows to them

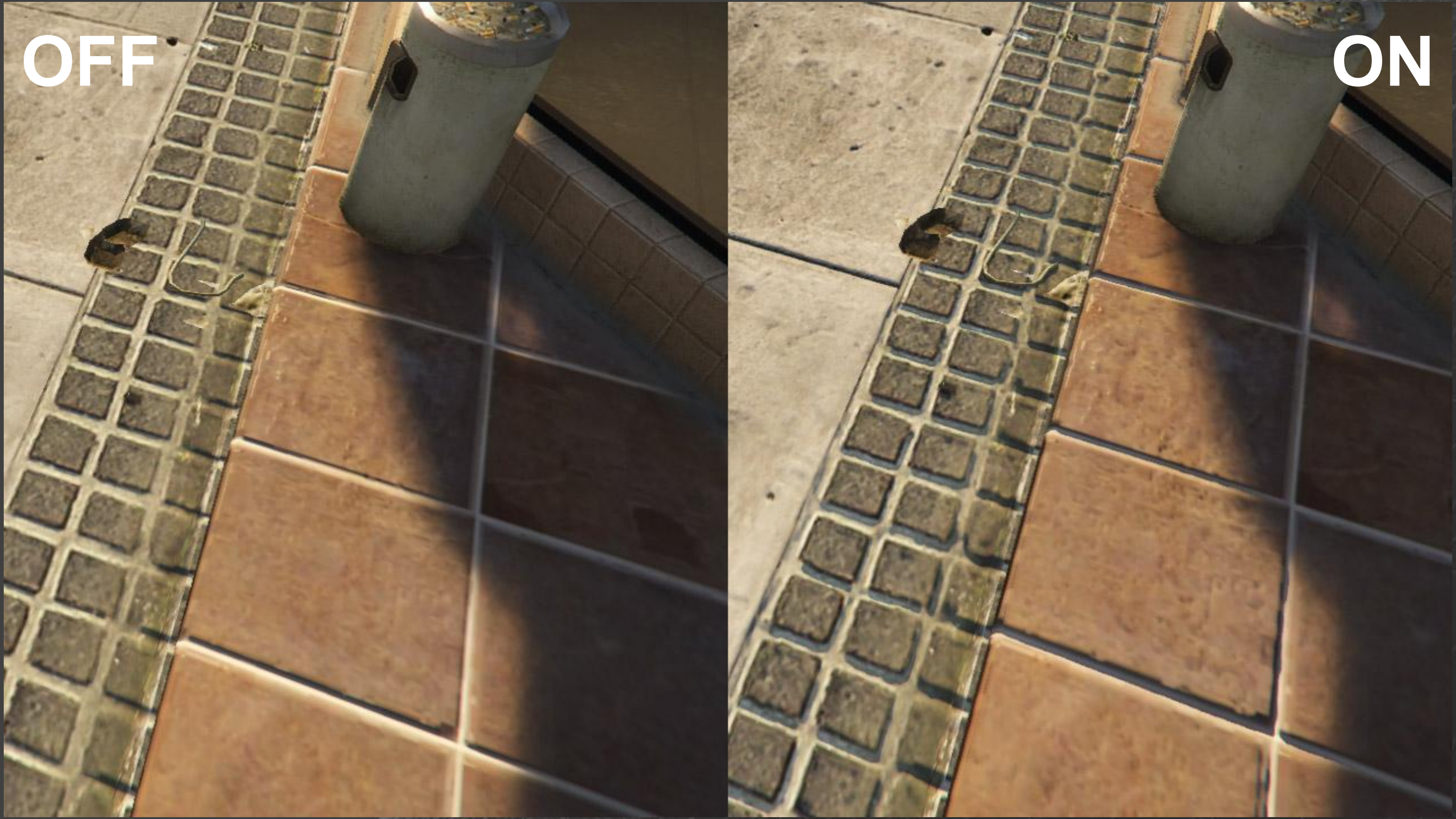
- Compute effect before characters, vehicles and other similar objects rendered

- Depth comparison in the NMS by distance factors to skip occluded objects

Screenshots are taken with deferred version


OFF
ON


OFF
ON

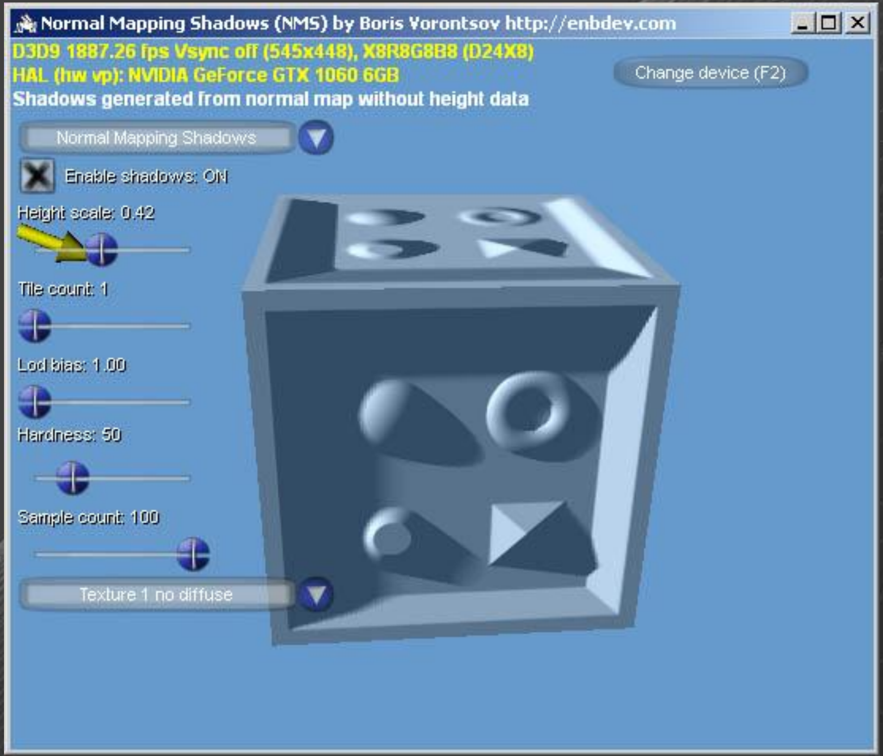

OFF
ON

Demo application

Example and other related materials are available at http://enbdev.com

References

[GDC06] N. Tatarchuk. Practical Parallax Occlusion Mapping For Highly Detailed Surface Rendering, 2006 https://developer.amd.com/wordpress/media/2012/10/Tatarchuk-POM.pdf



Normal Mapping Shadows (NMS) by Boris Vorontsov http://enbdev.com

D3D9 1887.26 fps Vsync off (545x448), X8R8G8B8 (D24X8)
HAL (hw vp): NVIDIA GeForce GTX 1060 6GB
Shadows generated from normal map without height data

☒ Enable shadows: ON

Height scale: 0.42

Tile count: 1

Lod bias: 1.00

Hardness: 50

Sample count: 100

Questions?