GameWorks SDK Shadow Lib v3.0

Agenda

- Features
- Comparison Screenshots
- Integration
- API

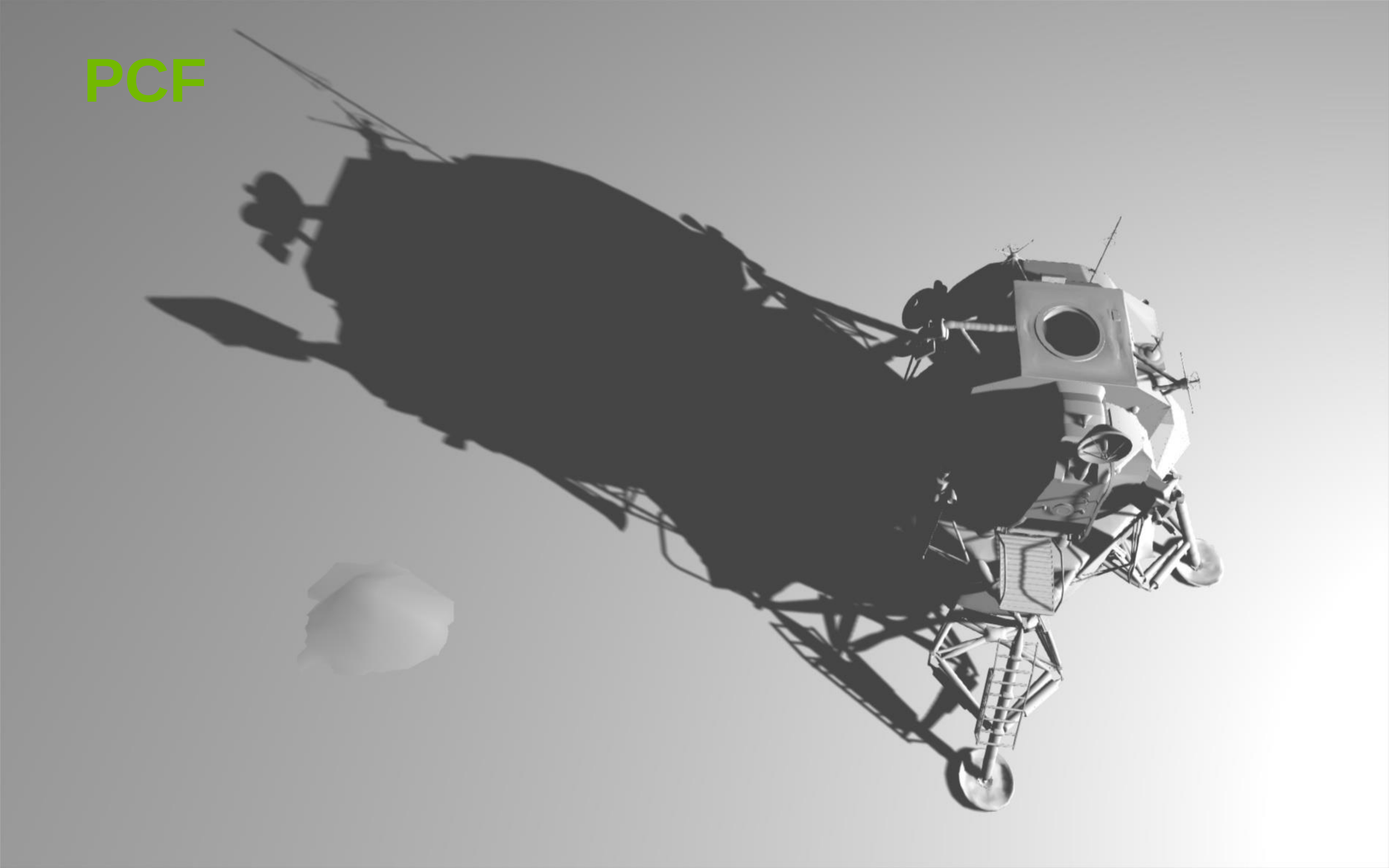
Features - Overview

- Renders various light space maps
  - Depth, Ray Trace, Frustum Trace
  - Insert Begin()/End() hooks in engine
- Cascades
  - SDSM
  - User defined
- Supported light types:
  - Spot
  - Directional (with cascades)
- Renders shadow buffer
  - Application provides depth buffer
  - Supports MSAA
- Array of techniques:
  - Hard, PCF, PCSS, RT, HRTS, FT, HFTS

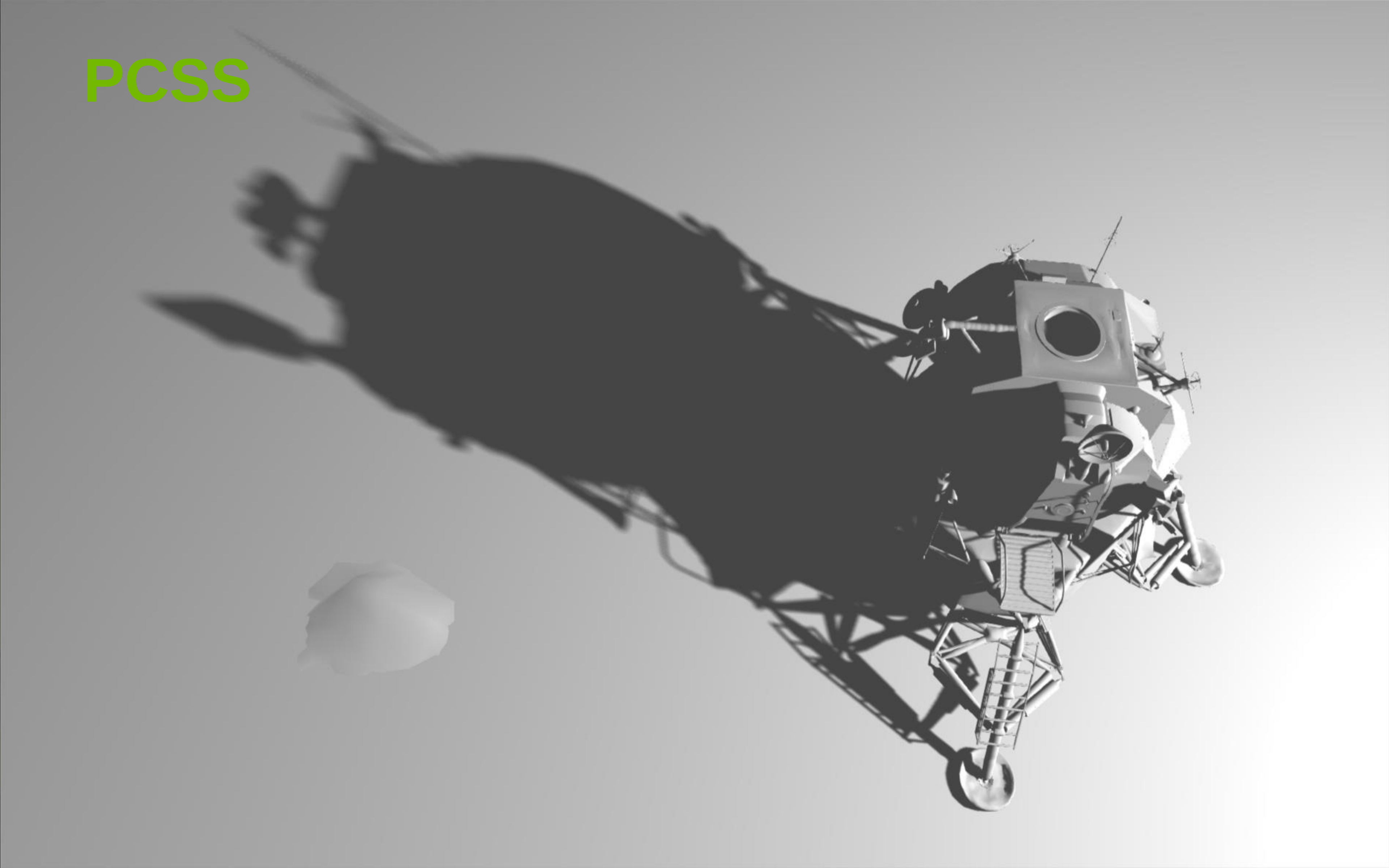
PCSS

- Adapted to support directional light sources
  - Adaptive quality level for cascades
- Aliasing reduction algorithm
  - Penumbra control as blocker depth reaches zero
- Dramatic performance increases with convergence testing algorithm

PCF



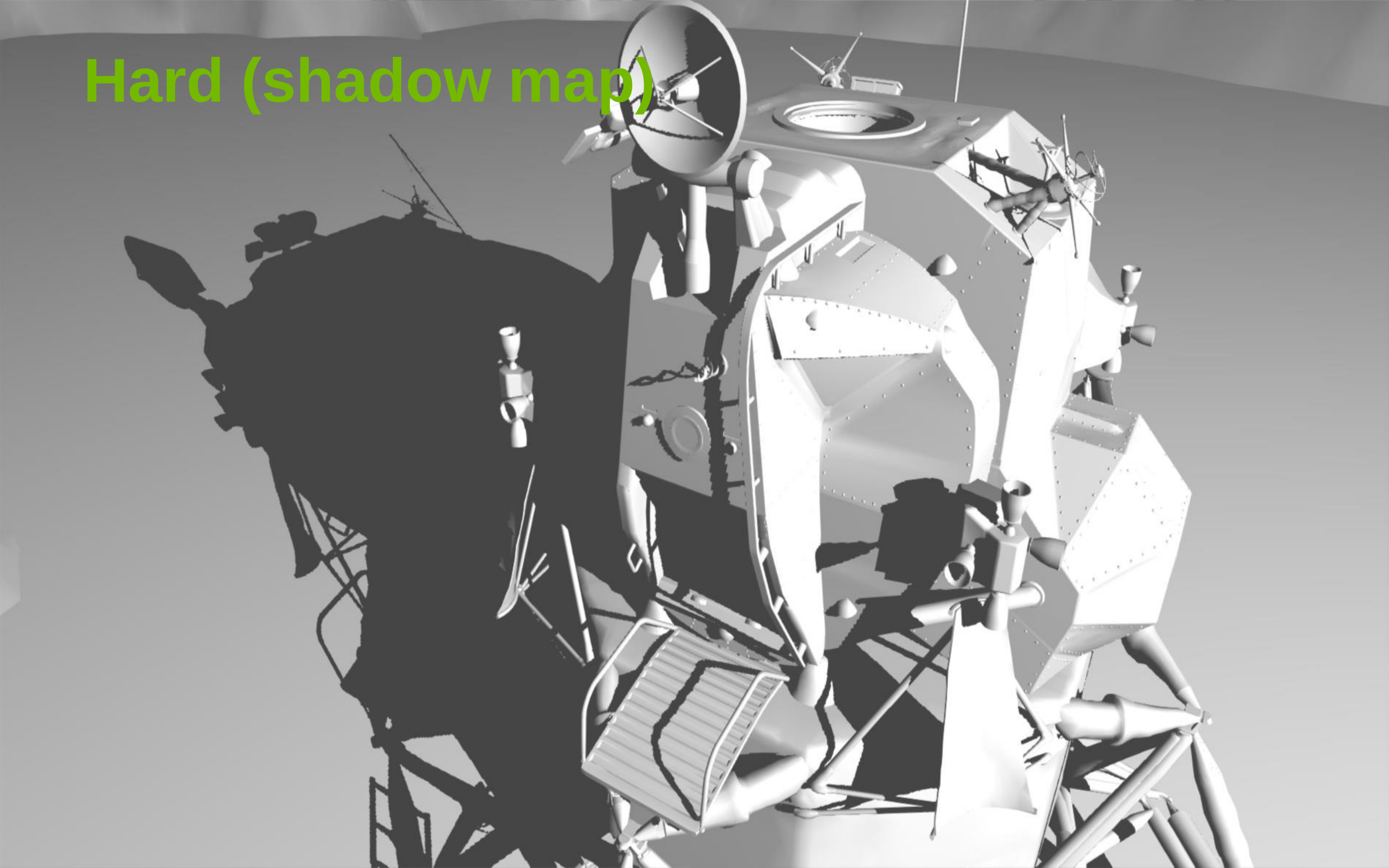
PCSS



Ray Trace & Frustum Trace

- Two new techniques that produce a perfect hard ray traced shadow
- Ray Trace (RT) stores primitives in light space
  - Performs ray-triangle intersection tests
  - More suited to spot light scenes
- Frustum Trace (FT) constructs list of screen pixels mapping to light space texels
  - Performs point in triangle frustum test
  - More suited to cascades
- Solves classic problems
  - Detachment of shadow from object
  - Aliasing

Hard (shadow map)

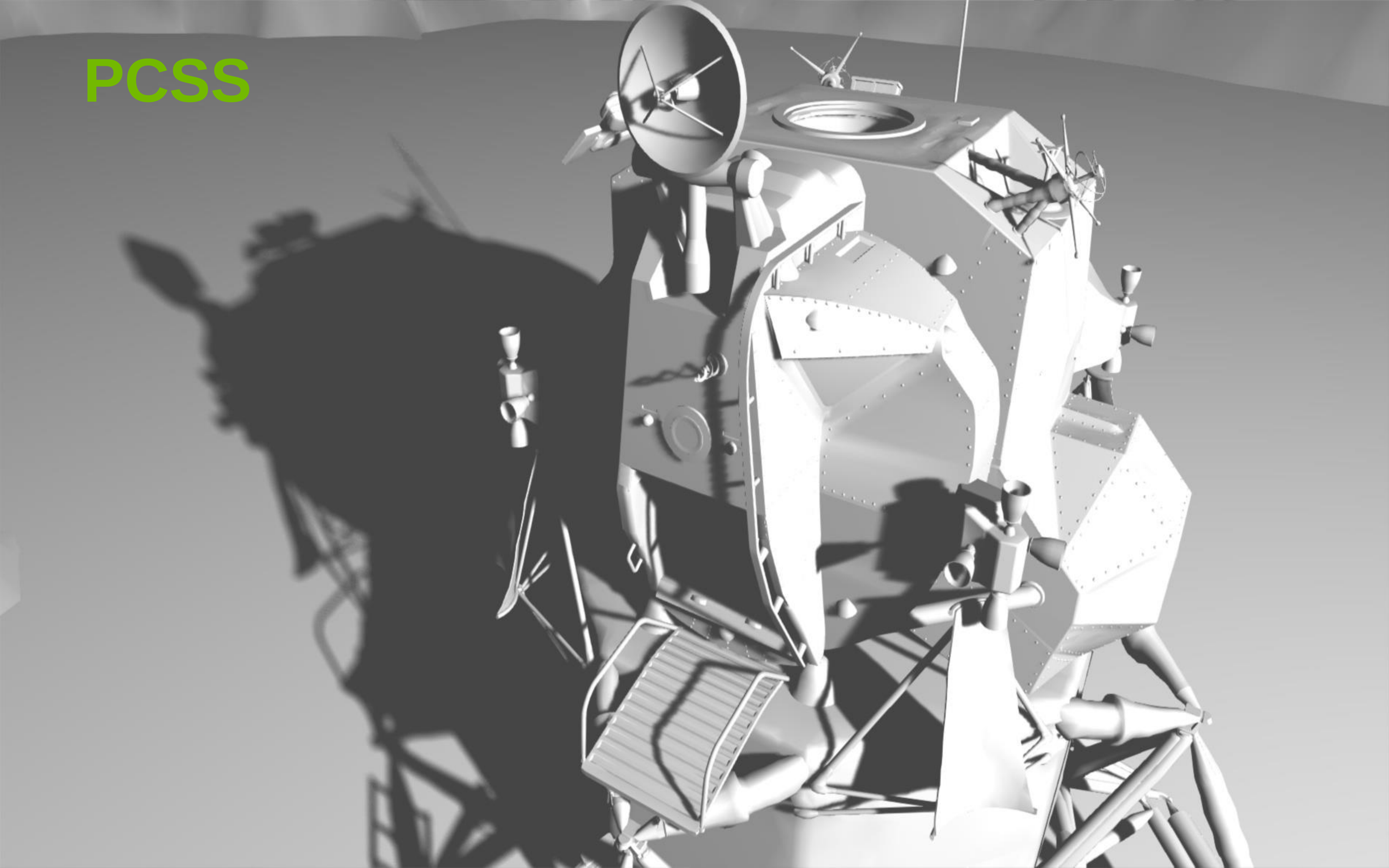


RT / FT

Hybrid Ray/Frustum Traced Shadows

- Two new techniques that smoothly interpolate between RT/FT and PCSS
- Solves classic problems with PCSS
  - Aliasing near point of contact
  - Detachment of shadow from object
  - Interference from overlapping blockers
- Amazing results on fine detail geometry

PCSS



HRTS / HFTS

Integration - Overview

- DirectX 11 capable engine
- Shadow maps:
  - Insert Begin()/End() hooks to wrap light space draw calls
  - Make use of your existing cascade system, by providing frusta exents to the library
- Provide depth/linear depth buffer
  - MSAA or single sample
- Library produces fullscreen shadow buffer
  - Needs combining with color buffer

API – On Create Device

- `GFSDK_ShadowLib_GetVersion()`
  - Check that header file matches DLL version
- `GFSDK_ShadowLib_Create()`
  - Pass in `ID3D11Device` & `ID3D11DeviceContext`
  - Returns a `GFSDK_ShadowLib_Context`
- `GFSDK_ShadowLib_AddBuffer()`
  - Creates a shadow buffer
- `GFSDK_ShadowLib_AddMap()`
  - Optional – if not providing your own shadow maps

API – On Resize Window

- `GFSDK_ShadowLib_RemoveBuffer()`
  - Release the existing shadow buffer
- `GFSDK_ShadowLib_AddBuffer()`
  - Create a new one of the correct size
- `GFSDK_ShadowLib_RemoveMap()`
  - Release the existing shadow map
- `GFSDK_ShadowLib_AddMap()`
  - Create a new one of the correct size

API – On Render (Light Space)

- `GFSDK_ShadowLib_SetMapRenderParams()`
  - Set all of the render params required by the lib
- `GFSDK_ShadowLib_UpdateMapBounds()`
  - Update the frusta based on light/eye position
- `GFSDK_ShadowLib_InitializeMapRendering()`
  - Clears surfaces and other initial work
- `GFSDK_ShadowLib_BeginMapRendering()`
- `GFSDK_ShadowLib_EndMapRendering()`
  - Sets up and restores graphics state for map rendering

API – On Render (Screen Space)

- `GFSDK_ShadowLib_ClearBuffer()`
  - Clears the shadow buffer
- `GFSDK_ShadowLib_RenderBuffer()`
  - Can be called multiple times to composite from different shadow maps
- `GFSDK_ShadowLib_FinalizeBuffer()`
  - Gives back the finalized shadow buffer

API – On Destroy Device

- `GFSDK_ShadowLib_Destroy()`
  - **Fully release all resources and the context**

Questions?

jons@nvidia.com