The lighting technology of Detroit: Become Human

Guillaume Caurant

Thibault Lambert

Contents

- The road to PBR at Quantic Dream
  - From Heavy Rain to Detroit: Become Human
  - Photometric Units
  - Material Calibration
- Direct Lighting
  - Analytical lights
  - Shadows
  - Volumetric lighting
- Indirect Lighting

PBR at Quantic Dream

- Heavy Rain (PS3)
  - Forward Shading
  - Gamma space
  - Blinn-Phong specular BRDF

PBR at Quantic Dream

- Beyond: Two Souls (PS3)
  - Deferred shading
  - Gamma space materials
  - Linear space lighting
  - Micro-facet BRDF approximation
    - Lit Shader: Normalized Blinn-Phong with constant visibility term
    - Skin Shader: Beckmann distribution with custom visibility term

PBR at Quantic Dream

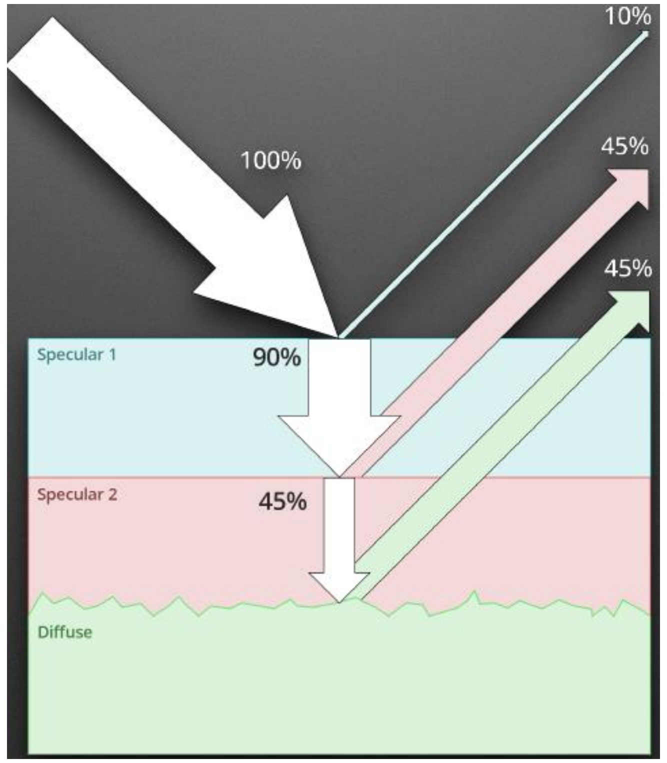
- Detroit: Become Human (PS4)
  - Clustered Forward Shading
  - Full linear space
  - Micro-facet Specular BRDF (GGX)
  - Still use Lambertian diffuse (k/π)
  - Partial Energy conservation
  - Photometric units for lights and emissive materials

PBR at Quantic Dream

- Material Evaluation
  - Shading Tree
  - Lit, unlit, lambert, glass, skin, hair, eye, tooth, car paint, cloth
  - Powerful but hard to control coherency
  - Some materials can have multiple specular lobes

PBR at Quantic Dream

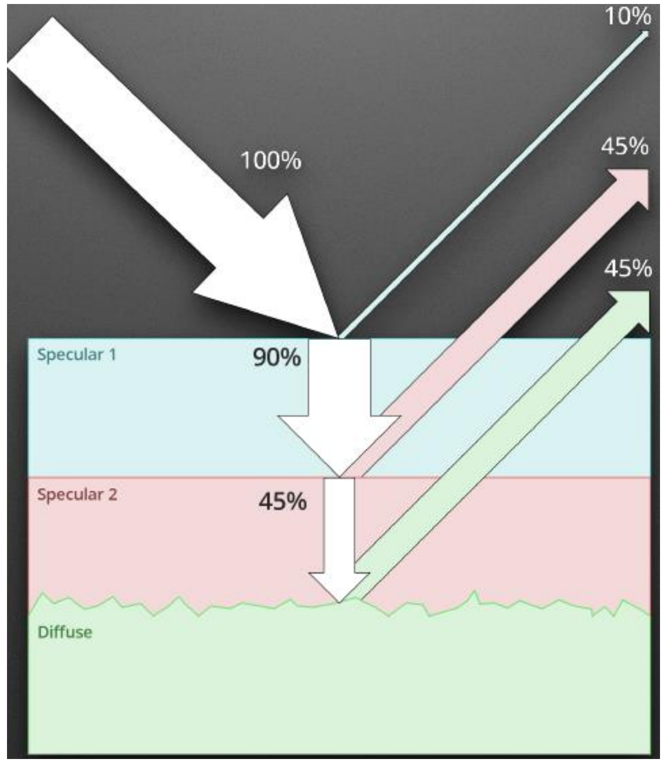
- BRDF layer stack
  - We split our BRDF in multiple layers and stack them
  - Up to 2 Specular lobes + 1 Diffuse + 1 subsurface/backscatter
  - Lit shader uses 1 specular GGX layer
    - Optional layer on metallic surface for rain



[LAN14]

PBR at Quantic Dream

- Energy conservation between BRDF layers
  - Each “layer” computes reflected and transmitted energy
  - Remaining energy is reused to compute next layer
  - Need to pre-compute energy transfer over BDRF
    - Multi-Scattering, Fresnel interfaces
  - Unfortunately we just take into account F0



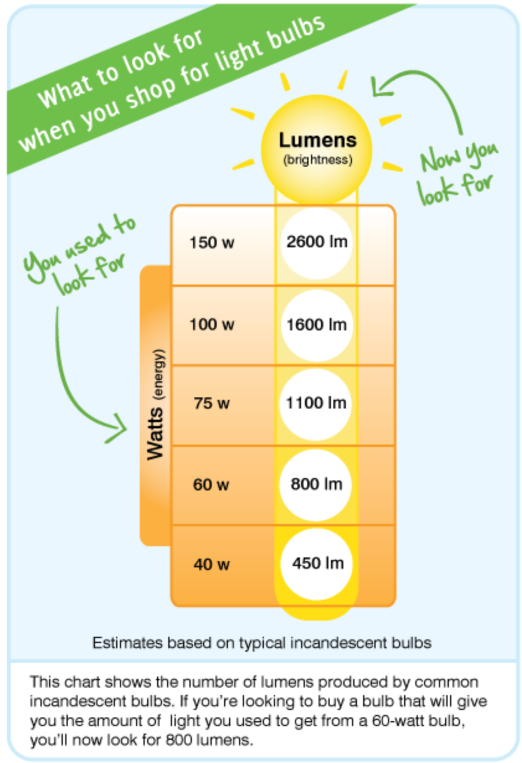
[LAN14]

PBR at Quantic Dream

- Issues faced during production cycle:
  - Poor lighting coherency between scenes
  - Material coherency is also a problem
  - Hard to reuse props in different scenes
  - => We decided to move to Photometric units

Photometric Units

- Lighting coherency was our main goal
  - Easier to compare with real life references
  - Artists can use real life input values
  - Prevents them from baking lighting information into albedo
  - Allows better scene contrast / range
  - [LAG14] "*Moving Frostbite to Physically based rendering*"



Watts (energy)	Lumens (brightness)
150 w	2600 lm
100 w	1600 lm
75 w	1100 lm
60 w	800 lm
40 w	450 lm

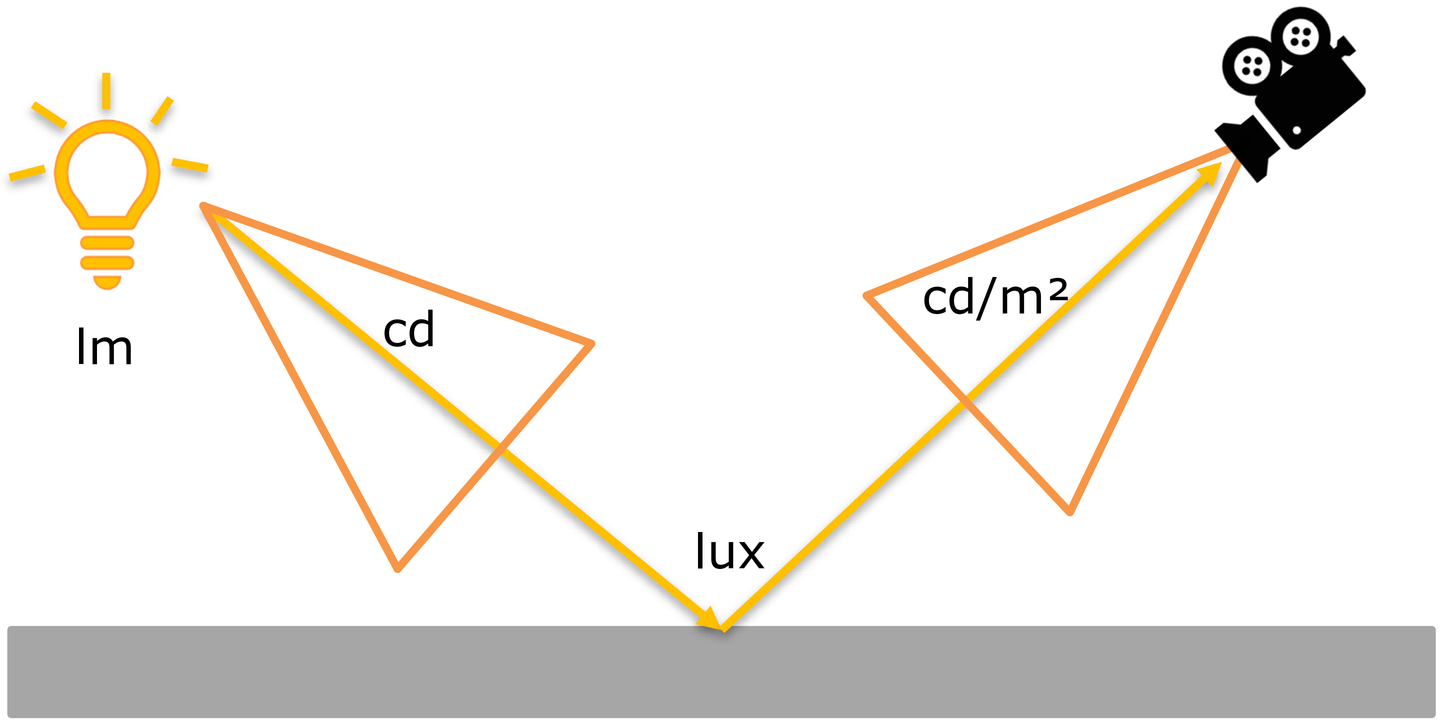
Estimates based on typical incandescent bulbs

This chart shows the number of lumens produced by common incandescent bulbs. If you're looking to buy a bulb that will give you the amount of light you used to get from a 60-watt bulb, you'll now look for 800 lumens.

©Unknown

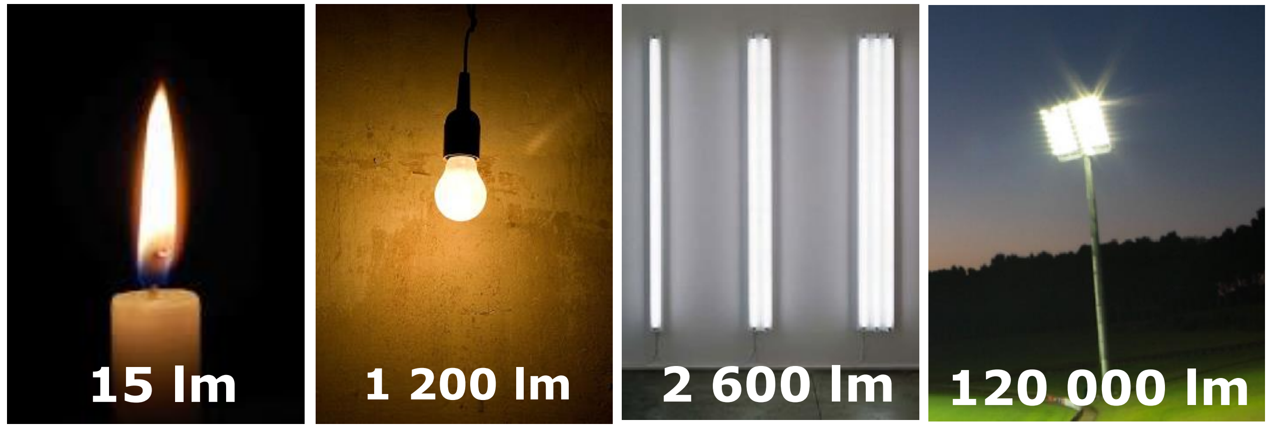
Photometric Units

- Luminous power (lm)
  - Total light amount emitted
- Luminous Intensity (cd)
  - lm per solid angle direction
- Illuminance (lux)
  - Light amount falling on surface
- Luminance (cd/m²)
  - Per unit area of cd in specific direction



Photometric Units

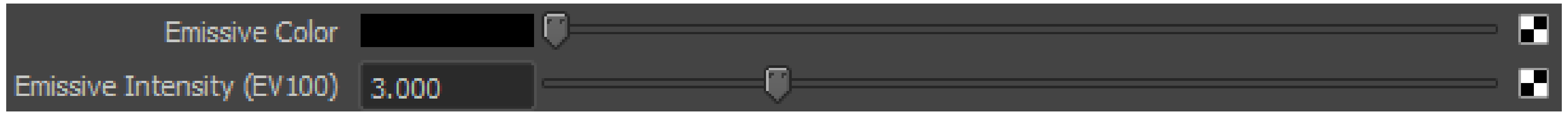
- Directional light in Lux
- Other lights in lumen
- Mandatory quadratic attenuation
- Illuminance is *really* high near punctual light



[LAG14]

Photometric Units

- Emissive Surface
  - Emissive Intensity parameter (+ color) on all materials
  - Expressed in Exposure Value (EV)
    - cd/m² is a linear scale but not perceived as such by the human eye
    - EV is perceptually linear
    - +1 EV double the perceived light intensity



Emissive Color

Emissive Intensity (EV100) 3.000

Photometric Units - Exposure

- Scene Exposure
  - Need to correctly expose our scene in our level editor
    - Auto exposure is not recommended
    - Use measured exposure for typical lighting condition
  - Our levels are split into “Scene Zone” (SZ)
  - Director of Photography provides exposure for each “SZ”
  - Fixed values, no transitions
  - Apply exposure when camera enters a “SZ”

Photometric Units - Exposure

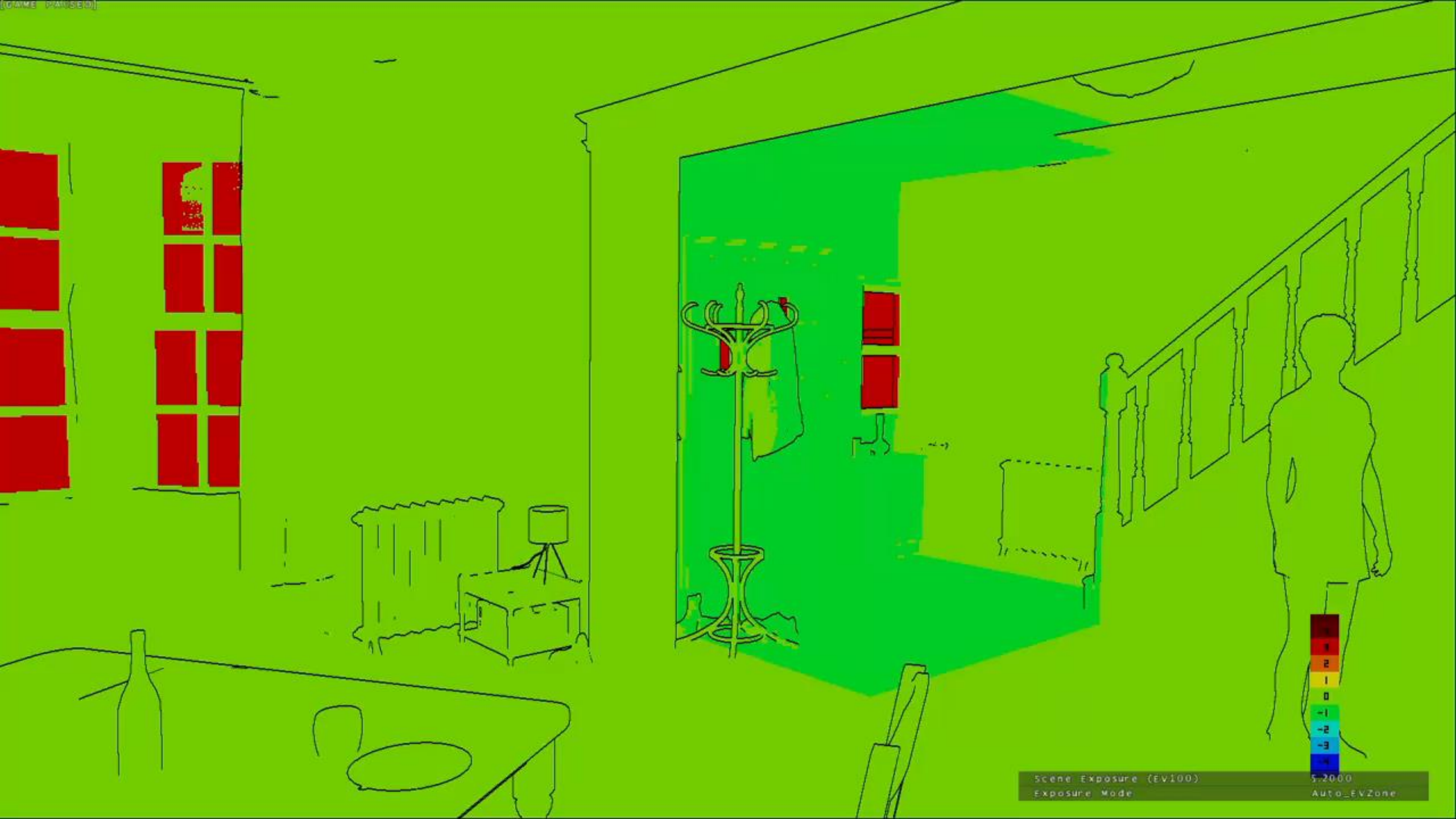
- Scene Exposure
  - Exposures are expressed in EV100
    - Represents a combination of a camera's shutter speed and f-number
    - EV100 is the exposure value for ISO100 sensor sensibility
  - Gives us a framework to ensure coherent lighting range
  - Scene exposure is great to **pre-expose** accumulation buffer
  - Need more control for in-game exposure

Photometric Units - Exposure

- Camera Exposure
  - **Exposure compensation** over the scene exposure
  - Gives control to dynamically change exposure
  - Artists can choose between 4 camera exposure types
  - Auto exposures = gameplay phases (mainly)
  - Manual exposure = cut-scenes (mainly)

Photometric Units - Exposure

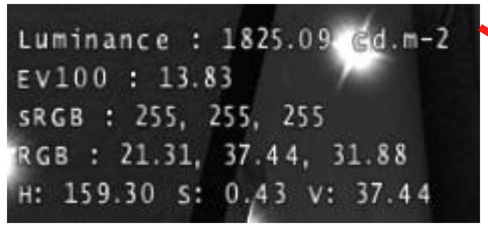
- Camera Exposure type:
  - Manual
    - Exposure value in EV100, can be controlled by animated curves
  - Camera
    - Computed from physical camera settings (f-stop, ISO, shutter time)
  - Auto-Average
    - Computed from Log average luminance of the scene
  - Auto-EVZone
    - Computed from exposure values provided in our “scene zone” + EV “decals” manually placed in the scene



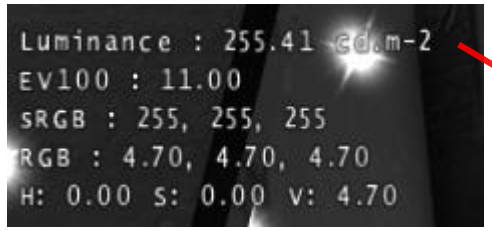
Scene Exposure (EV100) 5.2000
Exposure Mode Auto_EVZone

Photometric Units (DEBUG)

- Virtual Spot Meter
  - Gives pixel absolute luminance in cd/m² and EV100
  - RGB and sRGB values
  - Really useful to
    - Tweak emissive surface
    - Debug high value in specular reflection



```
Luminance : 1825.09 cd.m-2
EV100 : 13.83
sRGB : 255, 255, 255
RGB : 21.31, 37.44, 31.88
H: 159.30 S: 0.43 V: 37.44
```

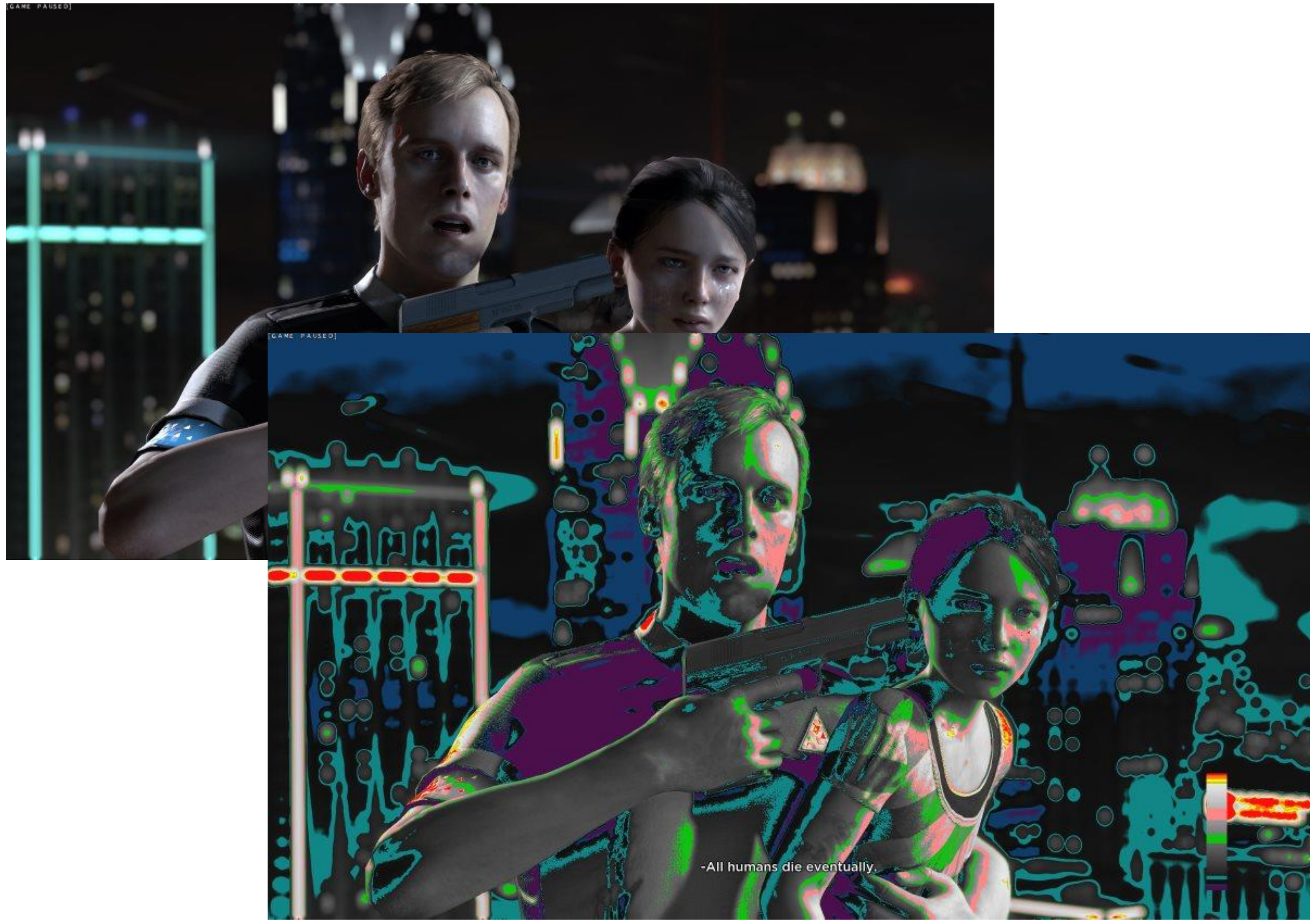


```
Luminance : 255.41 cd.m-2
EV100 : 11.00
sRGB : 255, 255, 255
RGB : 4.70, 4.70, 4.70
H: 0.00 S: 0.00 V: 4.70
```

disson
HOTEL

Photometric Units (DEBUG)

- False Color debug menu
  - Useful to check if a scene is well exposed
  - Green → Middle gray (18%)
  - Pink → Skin tone
  - Purple → Crushed blacks
  - Red → Burnt whites



Material Calibration

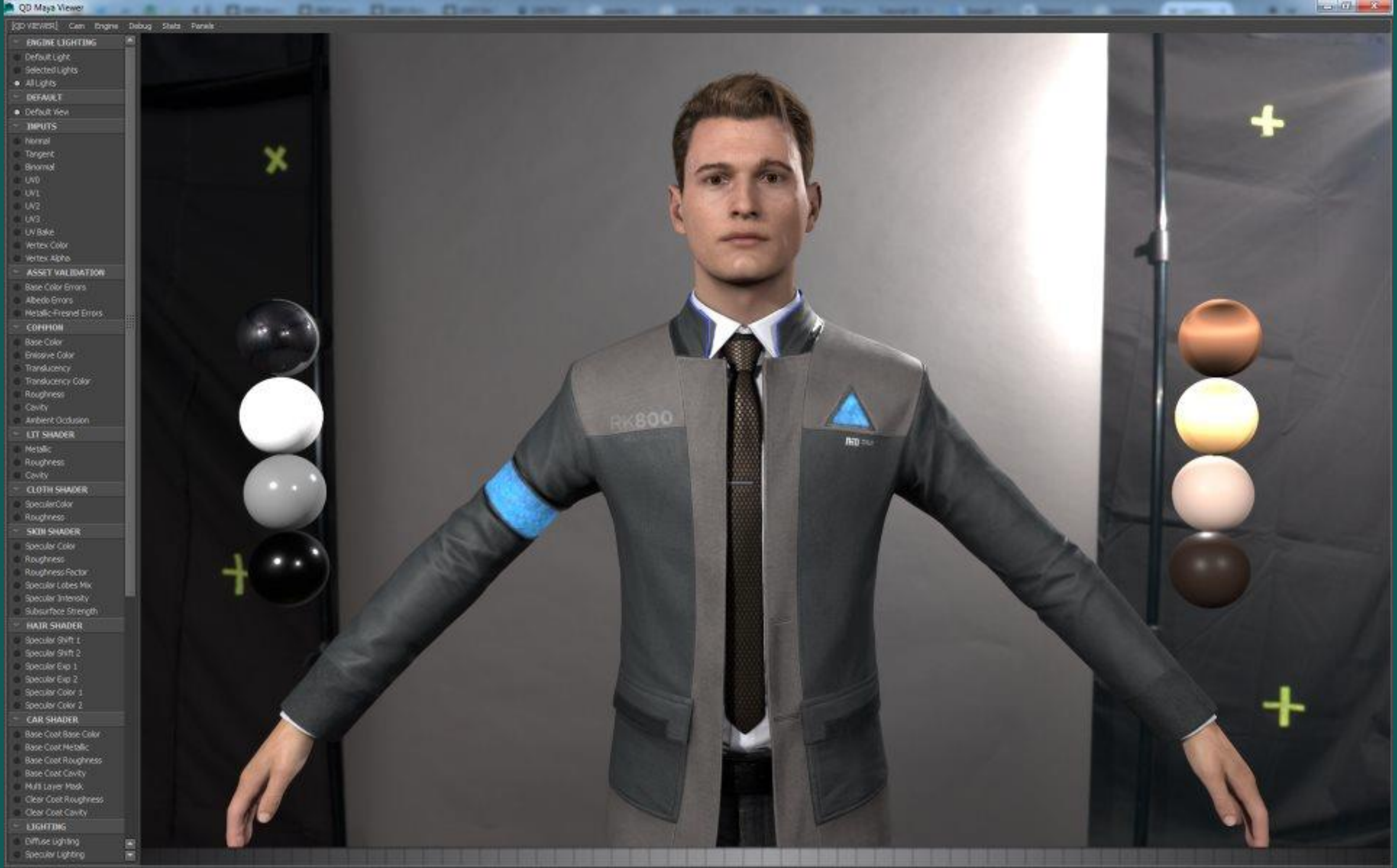
- Now we have a good framework for the lighting
  - real life references
  - coherent values
- Materials need the same treatment
- Impossible to scan all our materials

Material Calibration

- Capture some objects and materials samples
- Setup a room with controlled environment
- Built a black room with 3 incandescent bulbs
- Easy to reproduce in our engine
- Captured materials help us validate our lighting environment

Material Calibration

- Around this we built an “Icing Tool”
  - Provides some calibrated lighting environment
    - Contains our black room
    - And other full range IBL captured on various lighting environment [LAG16]
  - Material properties visualization
  - Comparison with object/materials references
- All props can be validated with this tool

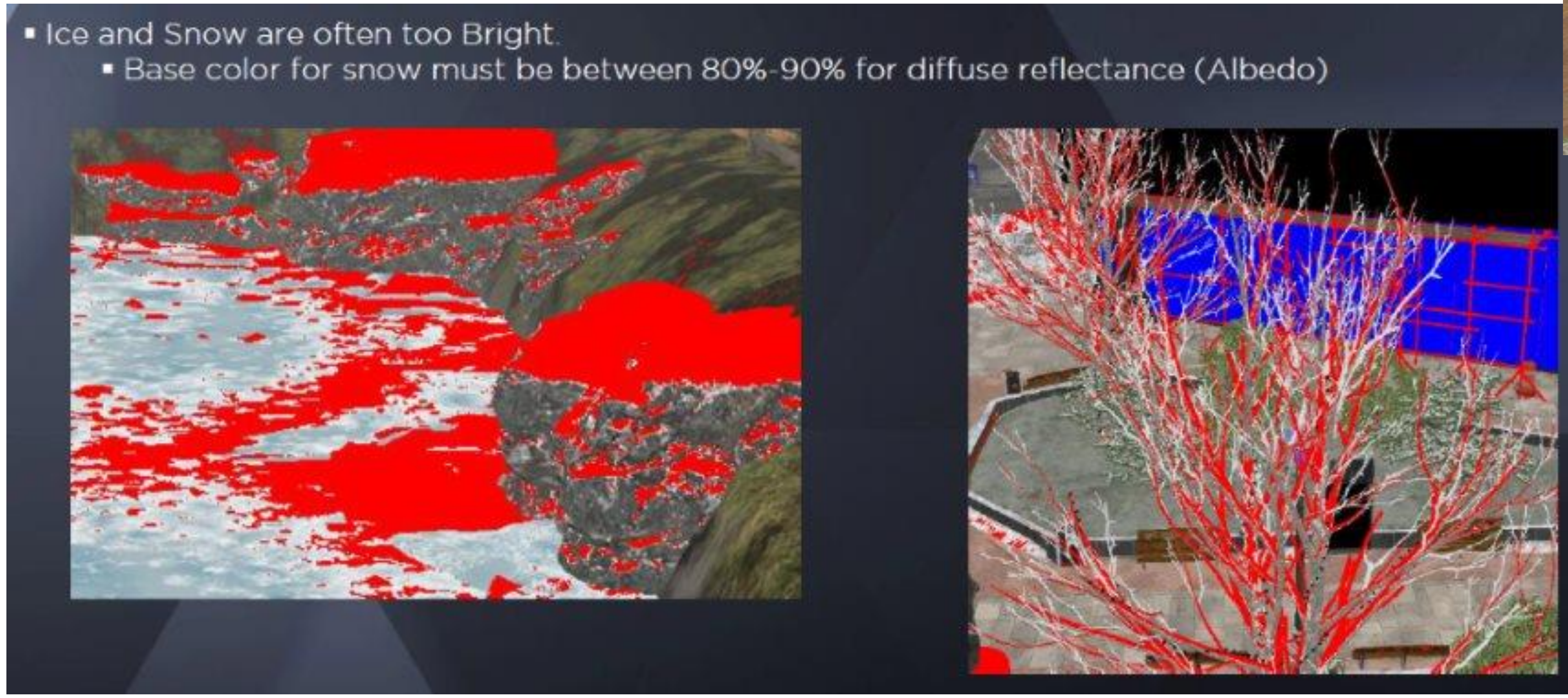

QD Maya Viewer
[QD VIEWER]
Cam
Engine
Debug
Stats
Panels
ENGINE LIGHTING
Default Light
Selected Lights
All Lights
DEFAULT
Default View
INPUTS
Normal
Tangent
Binormal
UV0
UV1
UV2
UV3
UV Bake
Vertex Color
Vertex Alpha
ASSET VALIDATION
Base Color Errors
Albedo Errors
Metallic-Fresnel Errors
COMMON
Base Color
Emissive Color
Translucency
Translucency Color
Roughness
Cavity
Ambient Occlusion
LIT SHADER
Metallic
Roughness
Cavity
CLOTH SHADER
SpecularColor
Roughness
SKIN SHADER
Specular Color
Roughness
Roughness Factor
Specular Lobes Mix
Specular Intensity
Subsurface Strength
HAIR SHADER
Specular Shift 1
Specular Shift 2
Specular Exp 1
Specular Exp 2
Specular Color 1
Specular Color 2
CAR SHADER
Base Coat Base Color
Base Coat Metallic
Base Coat Roughness
Base Coat Cavity
Multi Layer Mask
Clear Coat Roughness
Clear Coat Cavity
LIGHTING
Diffuse Lighting
Specular Lighting
RK800

Material Calibration (DEBUG)

- Highlights values with out of range material properties
  - Red: Wrong Base Color
    - Dielectric materials must be inside [30-240] sRGB
    - Metallic materials must be inside [186-255] sRGB
  - Blue: Wrong Glass Shader reflectance
    - Fresnel reflectance must be inside [52-114] sRGB
  - Yellow: Wrong Metallic parameter
    - Metallic value should be near 0 or 1, in-between values are often an error

Calibration

- Material properties error



- Ice and Snow are often too Bright
  - Base color for snow must be between 80%-90% for diffuse reflectance (Albedo)

Contents

- The road to PBR at Quantic Dream
  - From Heavy Rain to Detroit: Become Human
  - Photometric Units
  - Material Calibration
- Direct Lighting
  - Analytical lights
  - Shadows
  - Volumetric lighting
- Indirect Lighting

Direct Lighting

- All our light sources are punctual
  - Directional
  - Point Light
  - Spot Light
  - Projector Light
    - “Directional” light constrained in a box with attenuation

Direct Lighting

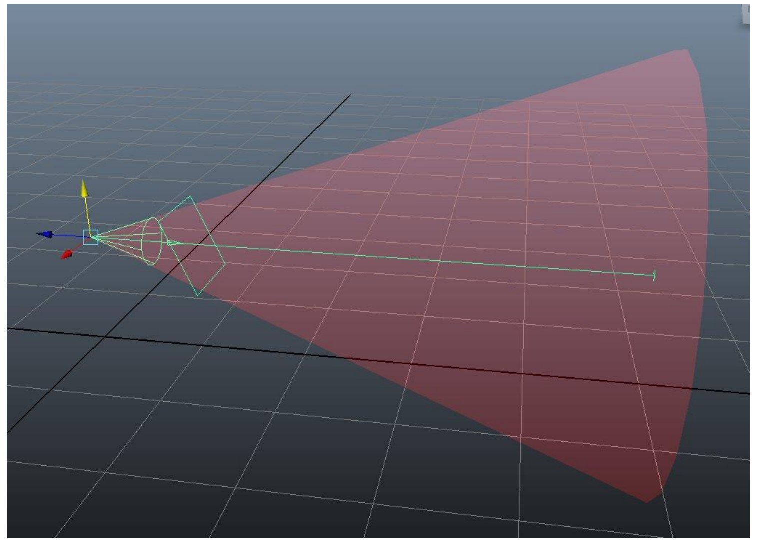
- Attenuation is quadratic by default
  - Artists can tweak the value from 0 to 2
    - Useful to fake bigger light by decreasing the attenuation
  - Combined with attenuation radius for performance reasons
    - Unfortunately it breaks energy conservation on the total energy emitted
    - Based on [KARIS13][LAG14]
- Punctual Light sources can produce really high intensity peak in the specular reflection

Direct Lighting

- Area Lights to the rescue (or not)
  - Implement area lights to fix this issue
  - Forward engine means all lights need to be area lights
  - Too late in the production cycle and the cost was not negligible
- Finally we’ve slightly biased the material roughness to prevent the issue

Direct Lighting

- Custom Near Clip plane
  - Cheap and useful to help with light positioning without custom light geometry
  - Light can be behind a wall or inside an object without affecting it
- Light visibility flag
  - All “SZs”, local “SZ”, visible “SZs”
  - Useful to only lit a local Scene Zone



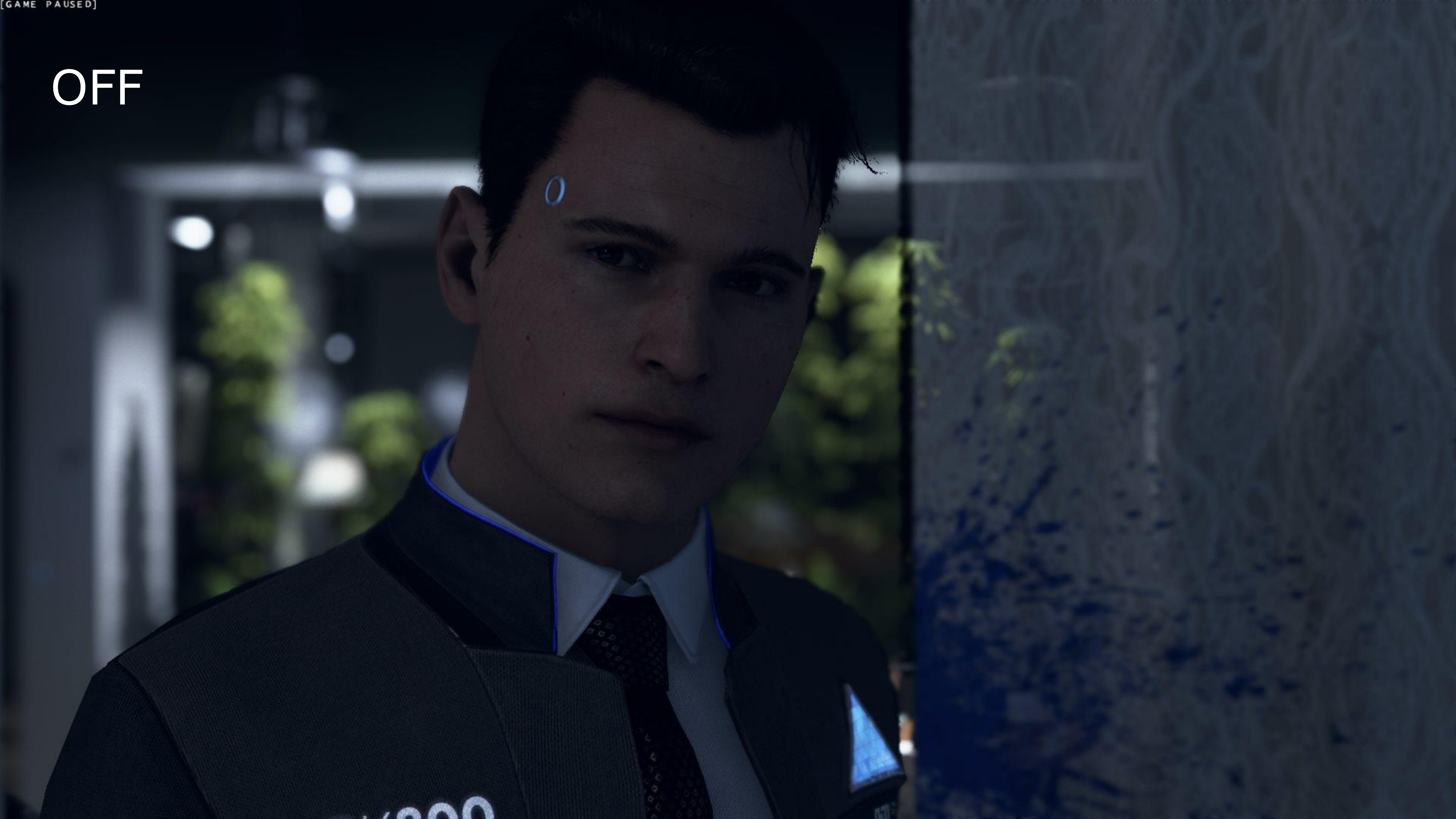
Close-up Lighting

- Story driven games with many close-up camera shots
- Want to lit each camera shot like on a movie set
- Scene lighting edited per shot
  - Light setup and object selection done on our movie editor
  - Can have camera, close-up lighting and exposure track

Close-up Lighting

- Set of lights used to replace regular lighting
- Can flag light in the scene as additional close-up light
- Use close-up shadow (explained later)
- Can affect indirect contribution (GI & IBL color multiplier)
- Have its own light cluster fitted to the bounding volume of the close-up selection
  - 11x11x4

ON



OFF

Shadow

- Shadow map
  - PCF with 8 samples + temporal super sampling
    - Jittered using blue noise
    - 3px default blur radius, can go up to 15px
  - Automatic Shadow Bias computed from geometry normal
    - Customized [HOL11]
  - Tried PCSS, not practical due to heavy register pressure
    - Only used on our tooth shader

Shadow

- Shadow Atlas
  - Shadows are stored on 8192^2 atlas with 16bit precision
    - Split in 256x256 bloc
  - Artists can choose their resolution between 3 different sizes
    - 256,512,1024
  - Shadow are resized depending on the camera distance
    - Resolution can be halved at max
    - Decreasing resolution in 4 step to prevent pixel crawling
    - And repacked in the atlas when reaching multiple of 256px

Shadow

- Shadow Atlas
  - Updated only if something moves in the light frustum
  - Point light shadow faces can be individually excluded
  - Tweakable Shadow Near clip plane
    - Help with numerical precision and light positioning
    - Can be decorrelated from light near clip plane (without rotation)

Shadow

- Directional Cascaded Shadow map
  - PCF with temporal super sampling (8 samples)
  - Smooth transition between splits using jittering and TAA
  - Up to 4 splits of 1440px each and 16bit precision
    - Majority of our scenes use 2 or 3 splits
  - Automatic split distribution

Static Shadow

- Switch to static shadow depending on the camera distance
- Only 1 sample with bilinear comparison
- Static Shadow Atlas
  - Atlas size : 2048^2
  - 64x64 per shadow
  - Up to 1024 shadows
- Directional Static Shadow
  - One big texture with all the static geometry of the level
  - Size : 8192^2

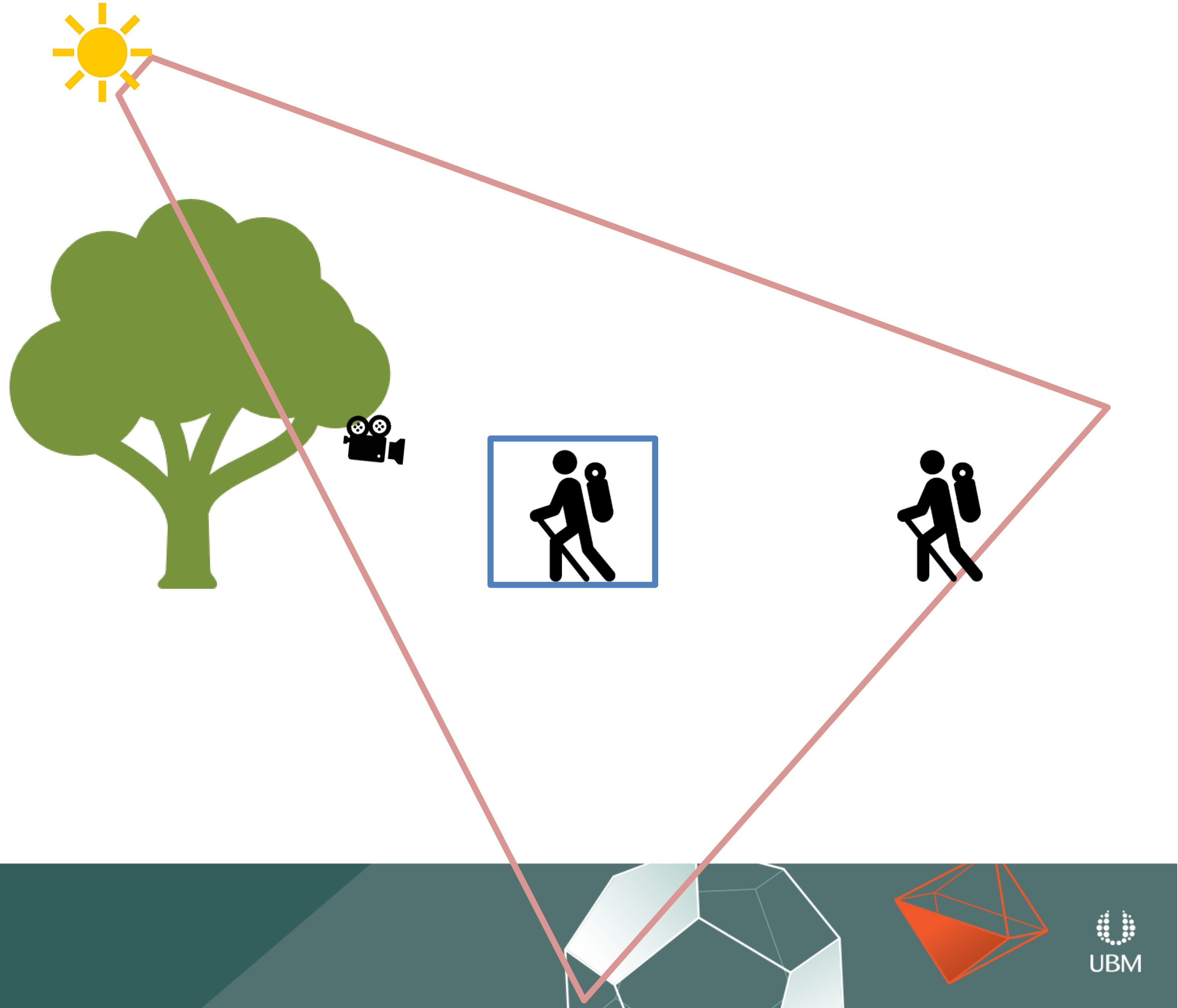
Close-up Shadow

- Add precision on contact and self shadow
- Up to 2 additional shadows at 1536^2px each
- Artists select relevant objects in the scene
  - ex: characters
- Only these objects receive close-up shadows

Close-up Shadow

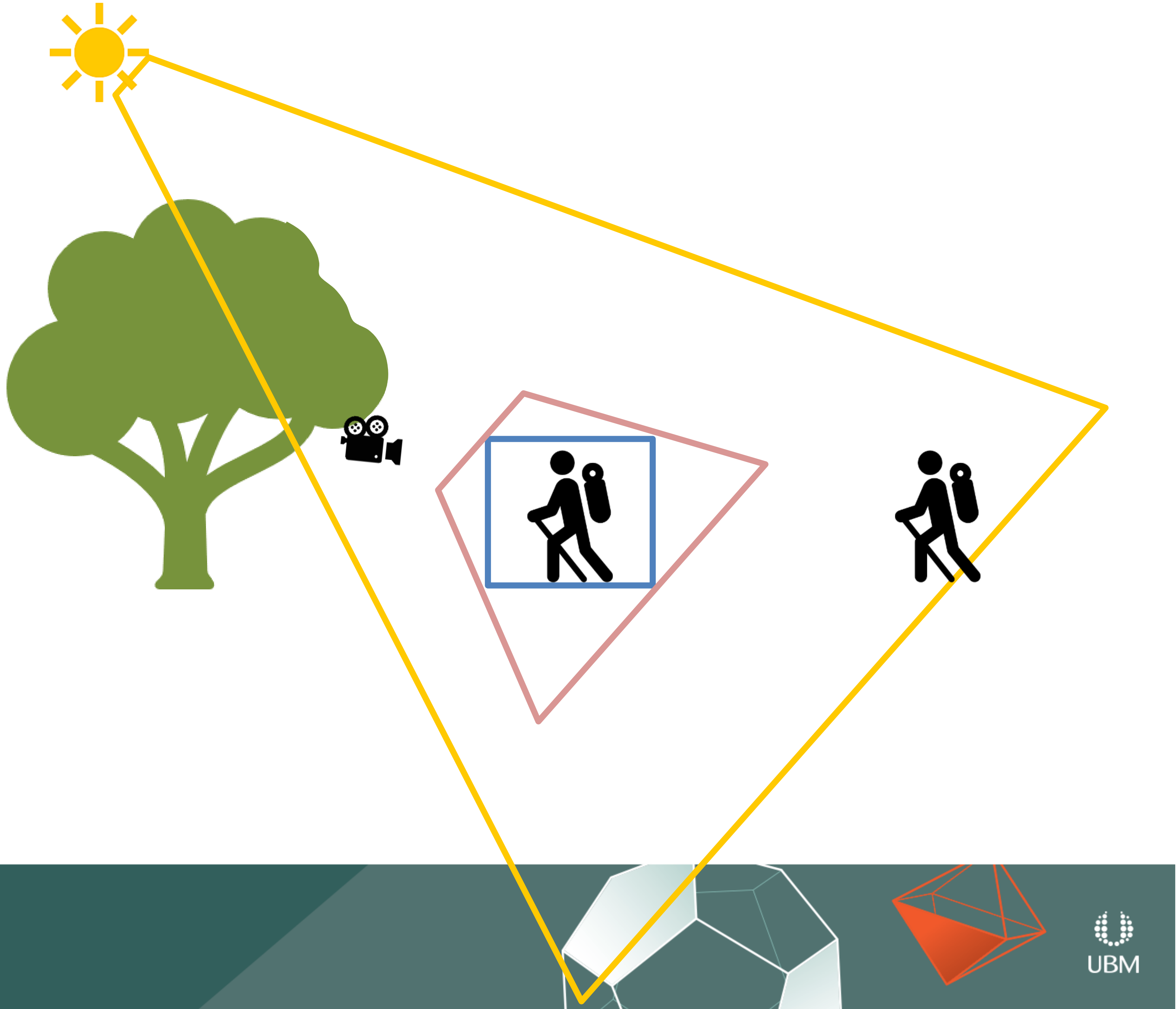
- Object selection:
  - All visible flagged objects inside 10m radius
  - Skinned object bounding volume computed from skinned point cloud

- Near/Far plane fitted to the bounding volume of close-up receivers selection



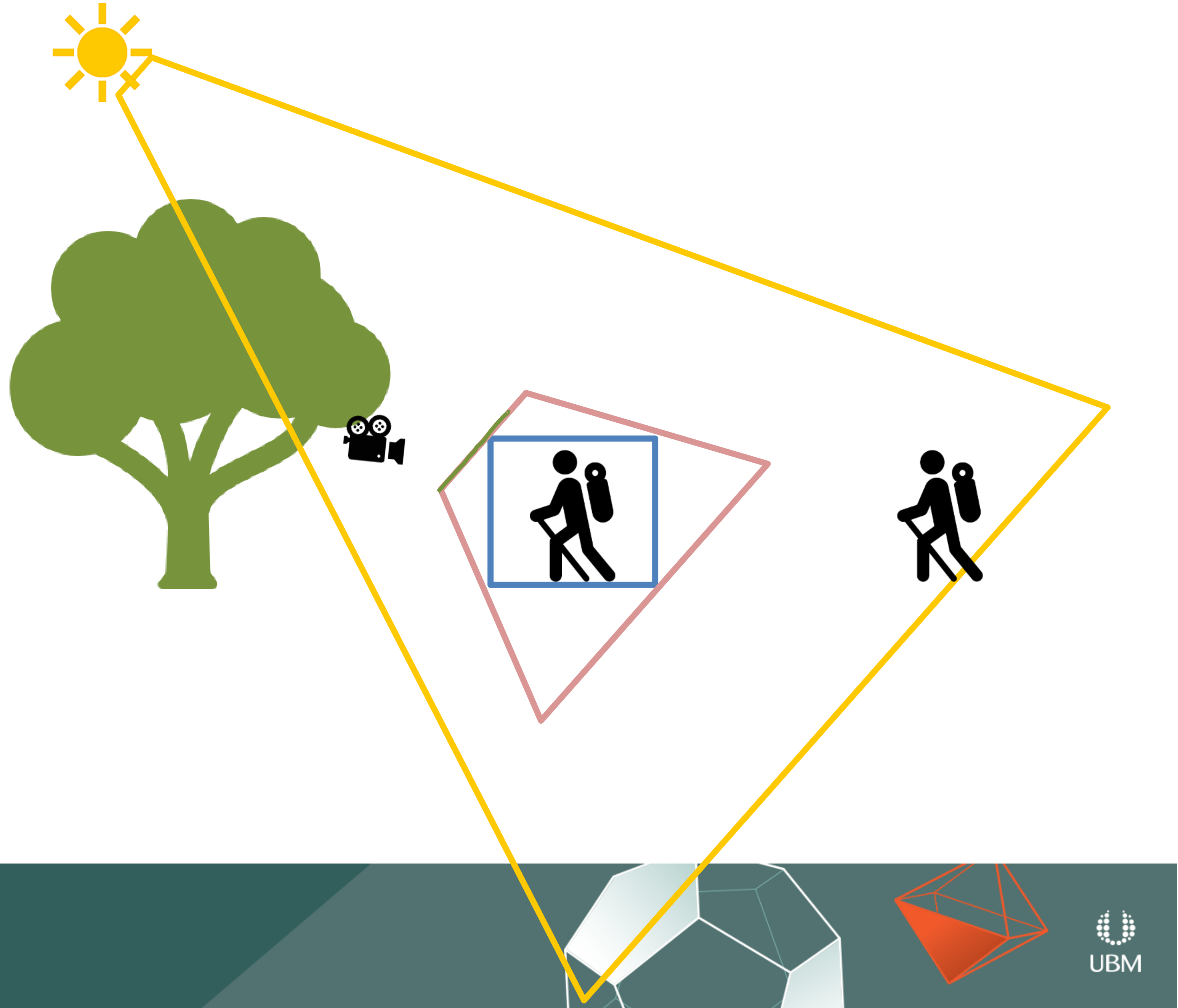
Close-up Shadow

- Object selection:
  - All visible flagged objects inside 10m radius
  - Skinned object bounding volume computed from skinned point cloud
- Near/Far plane fitted to the bounding volume of close-up receivers selection
- Objects outside of the Frustum are projected on the near shadow plane

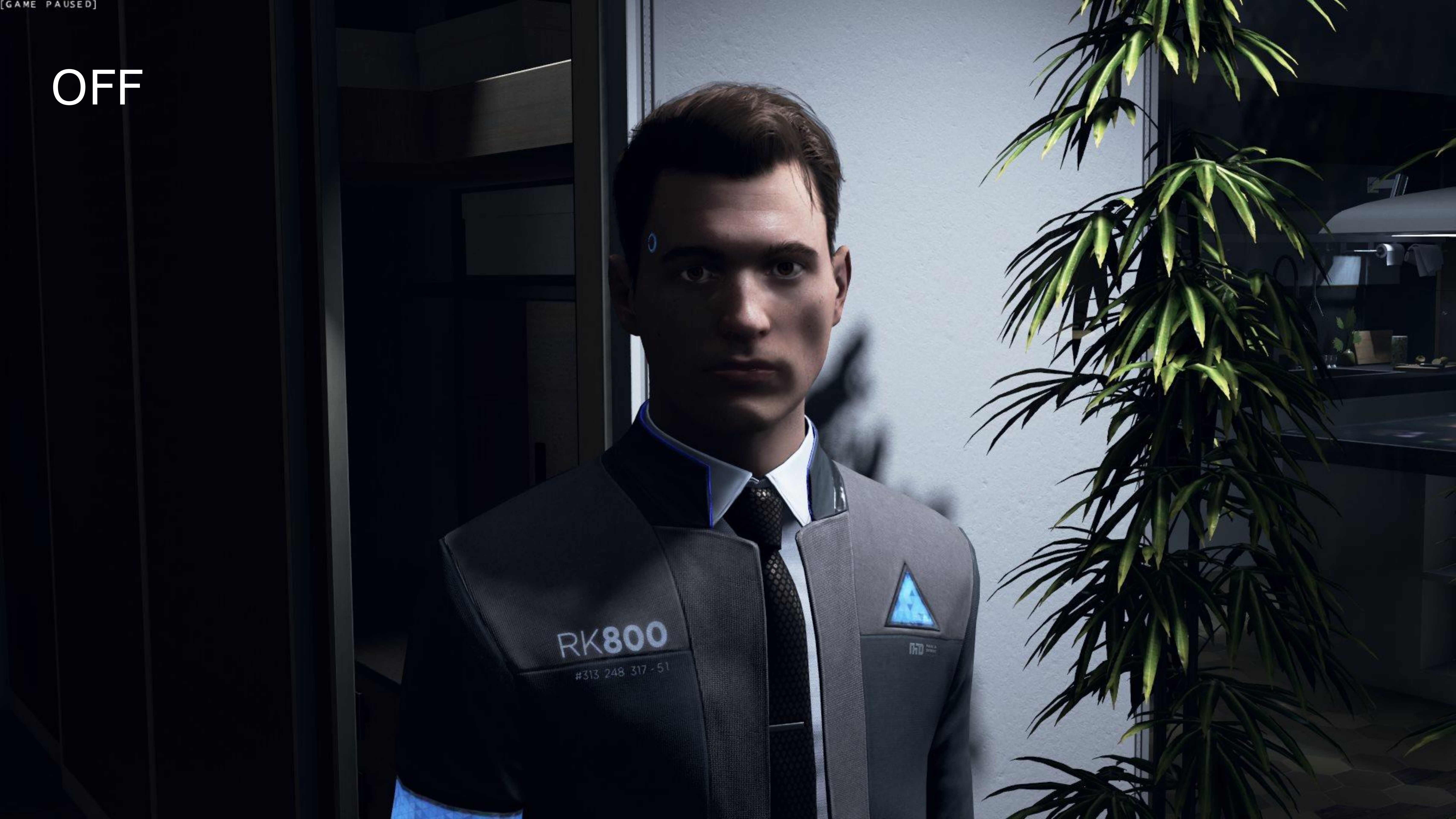


Close-up Shadow

- Object selection:
  - All visible flagged objects inside 10m radius
  - Skinned object bounding volume computed from skinned point cloud
- Near/Far plane fitted to the bounding volume of close-up receivers selection
- Objects outside of the Frustum are projected on the near shadow plane



ON

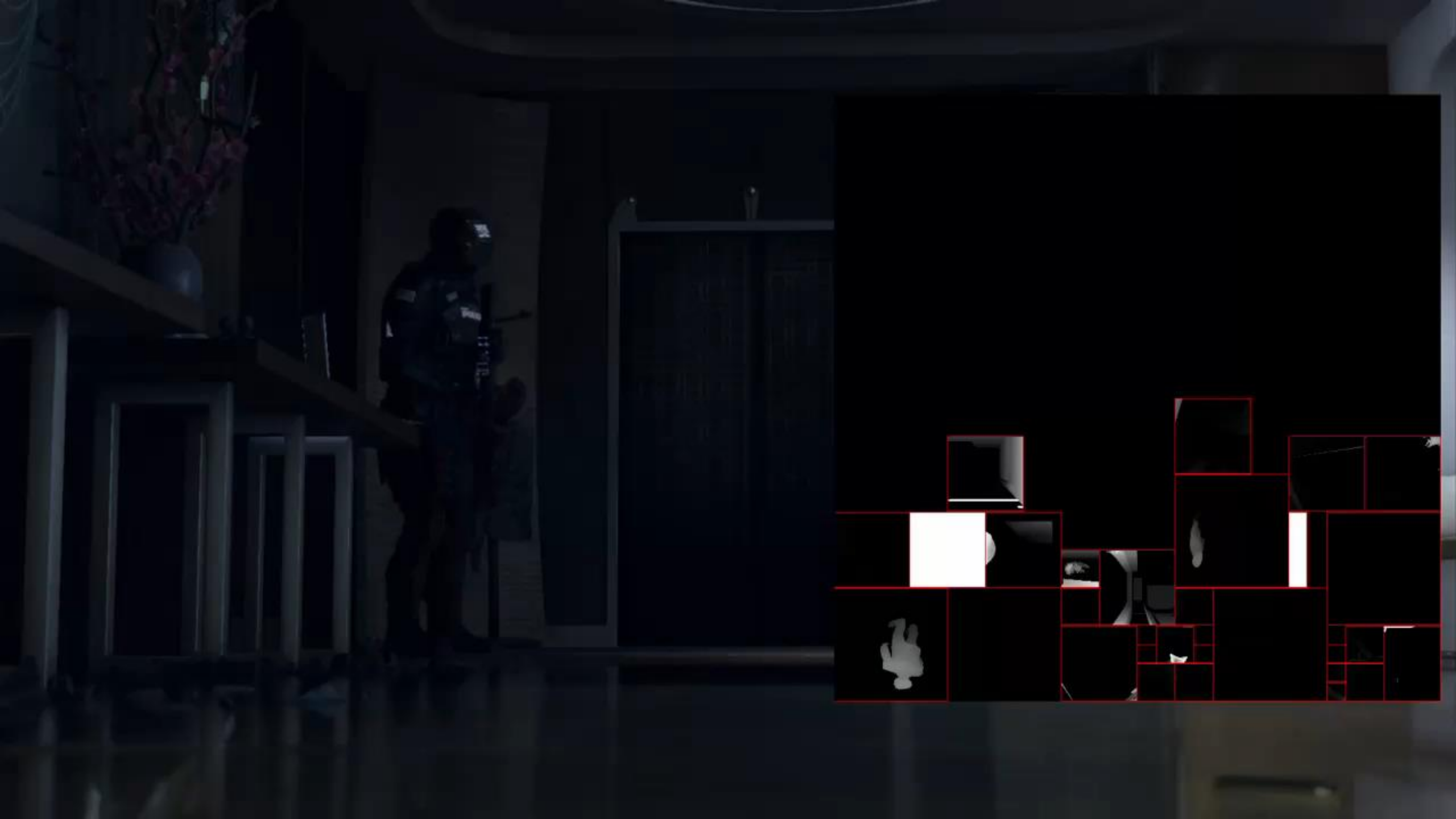


OFF

Shadow Memory Budget

- Shadow Atlas
  - 8192^2, 16bits
  - 10 to 1024 visible shadow
  - Point lights can be huge (6x1024)
- Static Shadow Atlas
  - 2048^2, 16bits
  - 1024 static shadows per level
- Directional Shadow
  - 1440^2 x 3, 16bits
- Directional Static Shadow
  - 8192^2, 16bits

- Static
  - 8MB + 128MB
- Dynamic
  - 128MB + 12MB
- Total Memory
  - 276MB



Shadow Performances

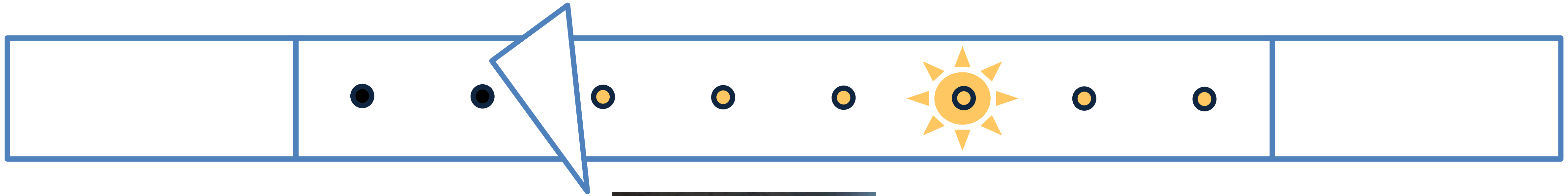
- 15-20 shadow updates on average per frame
  - No update quantity limit, can get wild in some cases
- 1.5ms to 3.5ms on previous video
- Close-up Shadow cost
  - Scene dependent
    - Fast if few re-projected objects on the near plane (< 1ms)
    - Slow if the light is in a tree (huge alpha test coverage)

Volumetric Lighting

- Unified volumetric lighting [WRO14][HIL15]
  - Fitted on light cluster depth
  - Use checkerboard rendering
    - PS4 Base : 192x108x64
    - PS4 Pro : 240x135x64
  - TAA with blue noise jittering
  - Lit by direct light and diffuse probe grid
  - Fog contribution on GI baking
    - Fake Multi-Scattering

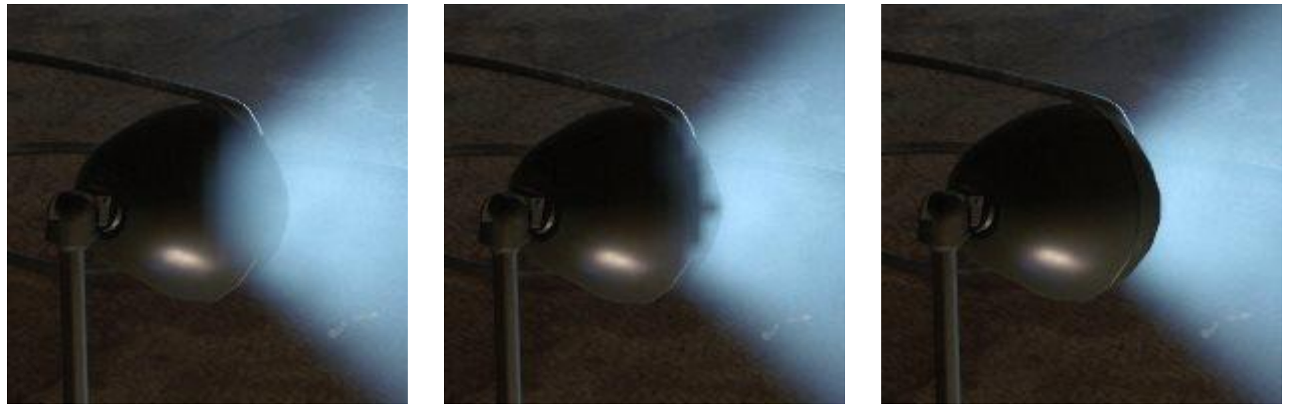
Volumetric Lighting

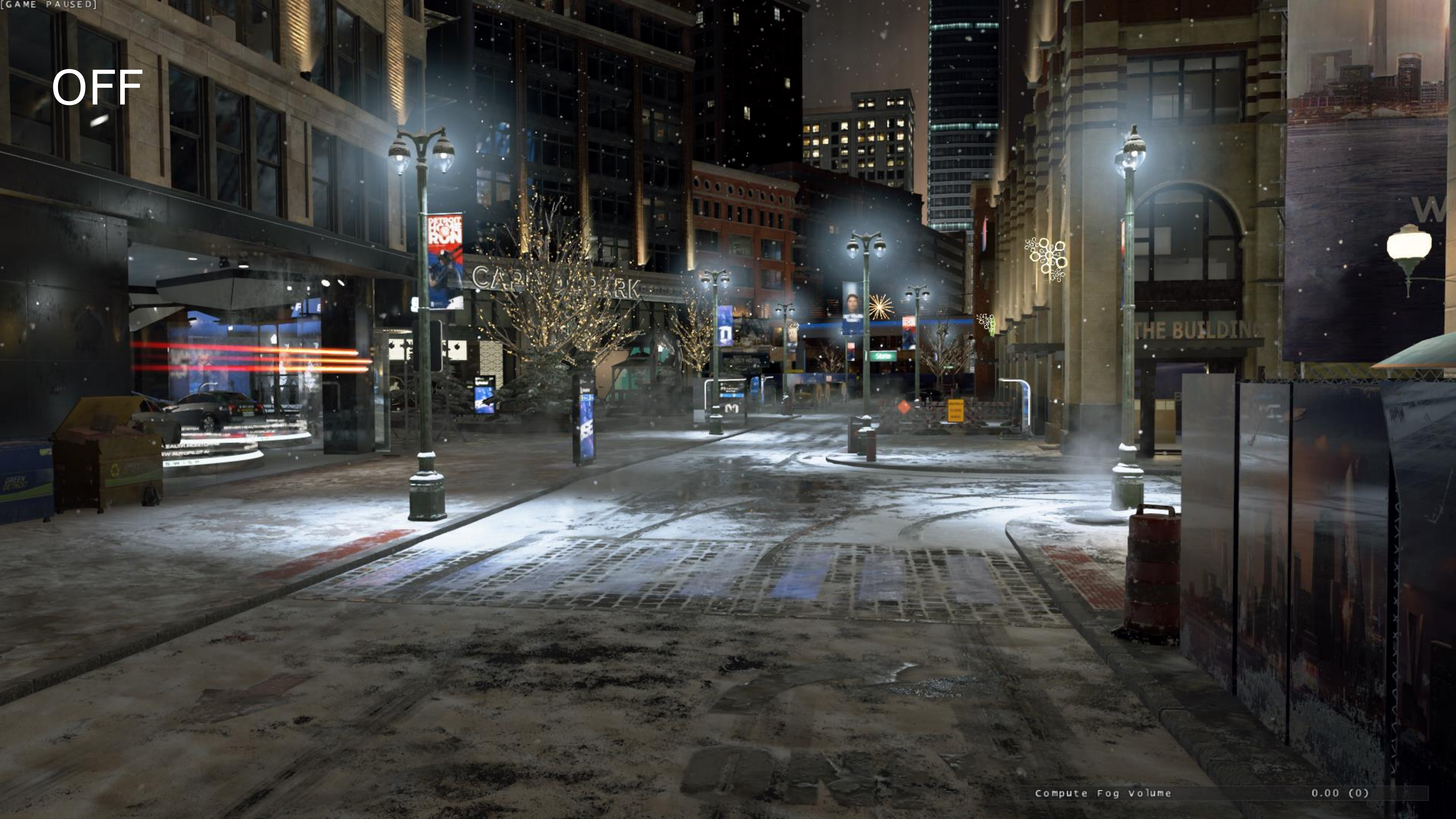
- Volumetric light can leak through surfaces



Volumetric Lighting

- Fix leaking
  - Min/Max depth stored per tile
  - Use max depth to clamp voxel thickness at light evaluation
  - Apply Z bias on volumetric texture sampling
    - (TileDepthVariance > threshold)



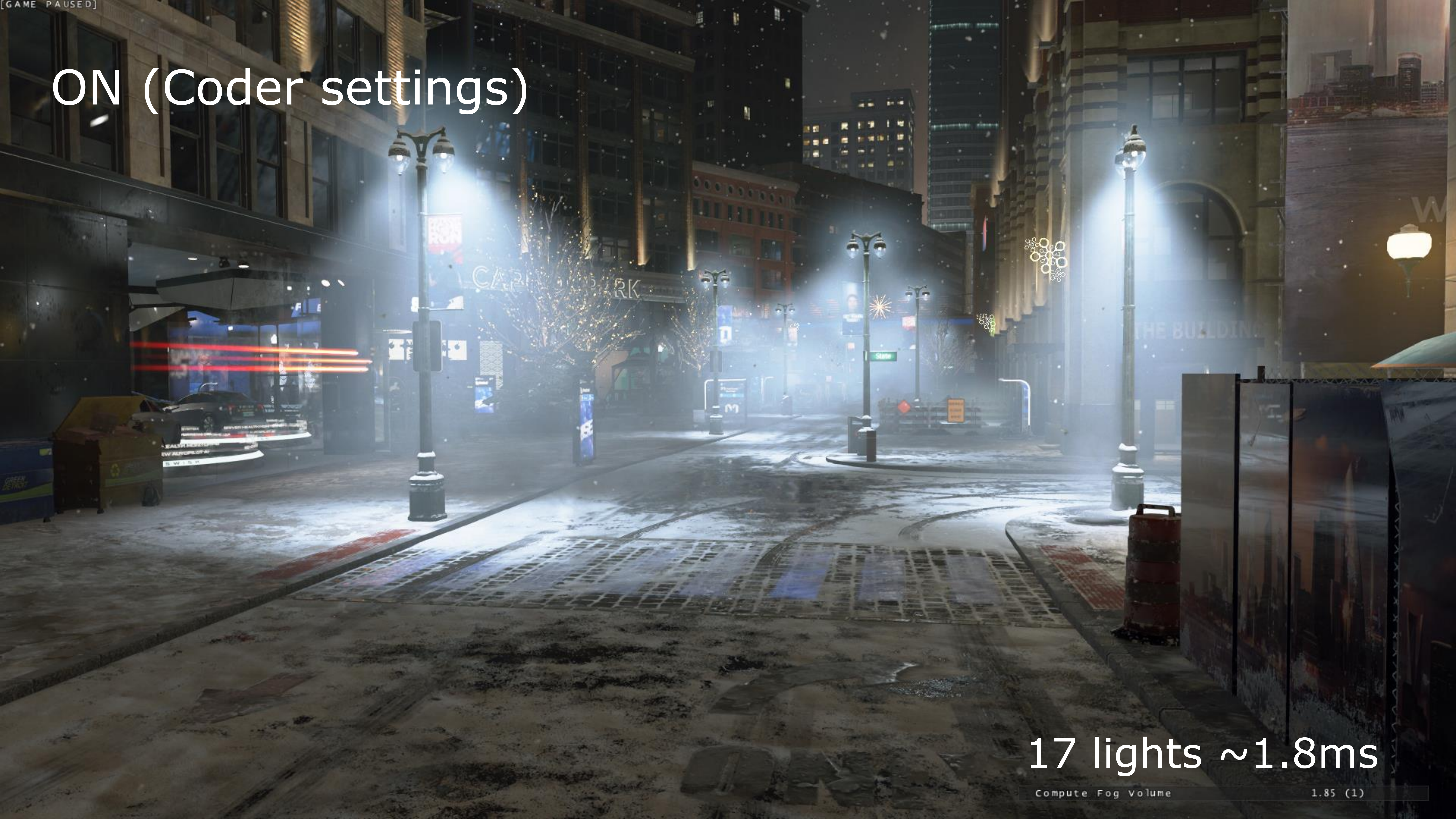


OFF

```
[GAME PAUSED]
```

```
Compute Fog Volume                    0.00 (0)
```

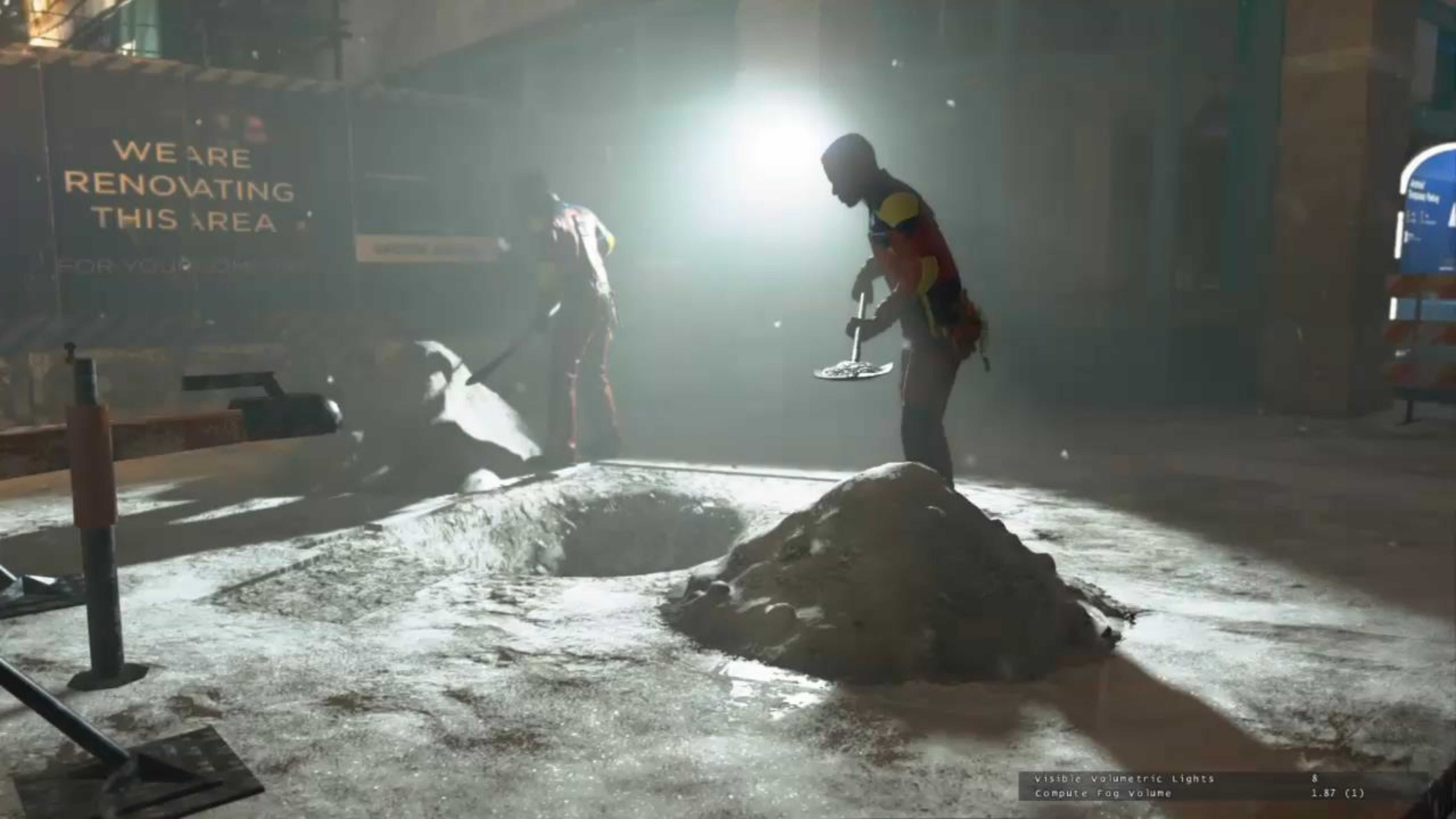
ON (Coder settings)



[GAME PAUSED]

```
Compute Fog Volume                    1.85 (1)
```

17 lights ~1.8ms



```
Visible Volumetric Lights         8
Compute Fog Volume                1.87 (1)
```

Contents

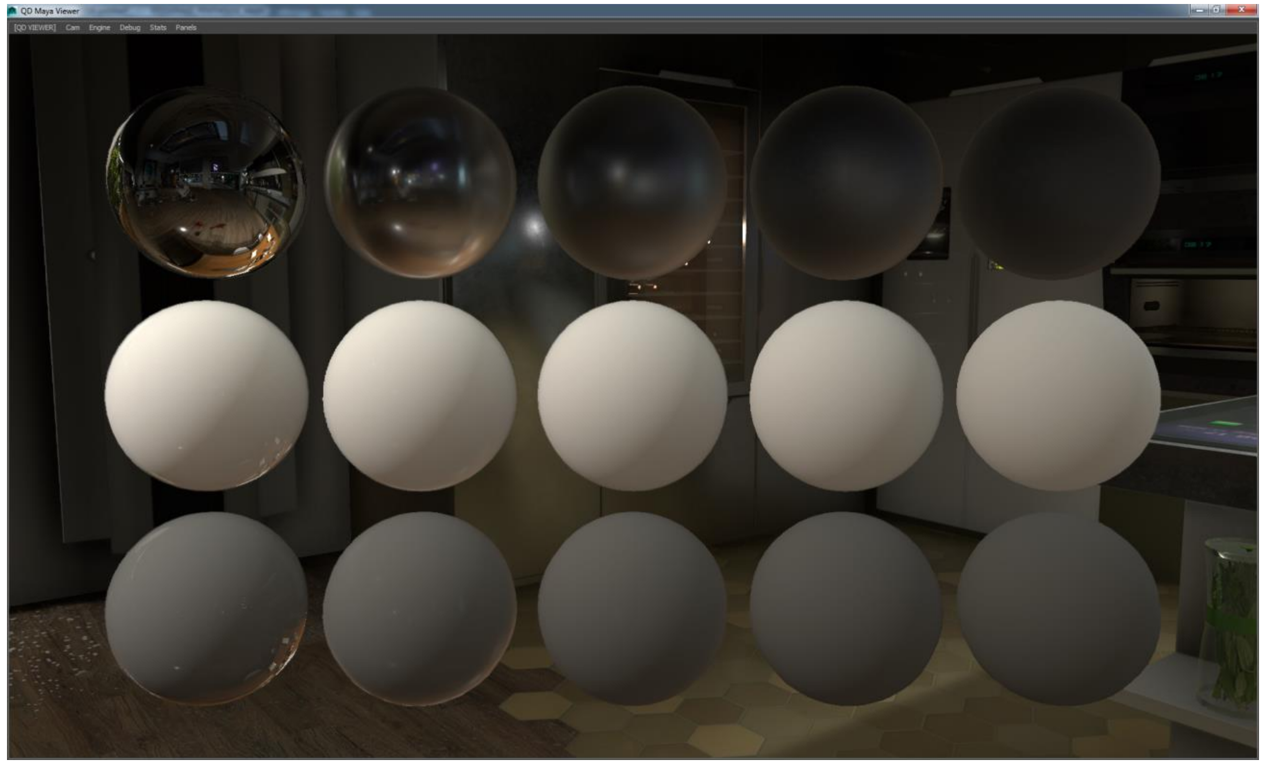
- The road to PBR at Quantic Dream
  - From Heavy Rain to Detroit: Become Human
  - Photometric Units
  - Material Calibration
- Direct Lighting
  - Analytical lights
  - Shadows
  - Volumetric lighting
- Indirect Lighting

Indirect Lighting

- Beyond: Two Souls (PS3)
  - Use HL2 ambient cube
  - Vertex baking for static geometry
  - Light probe for dynamic geometry
  - Fake indirect specular lighting
- Need unified solution for static & dynamic

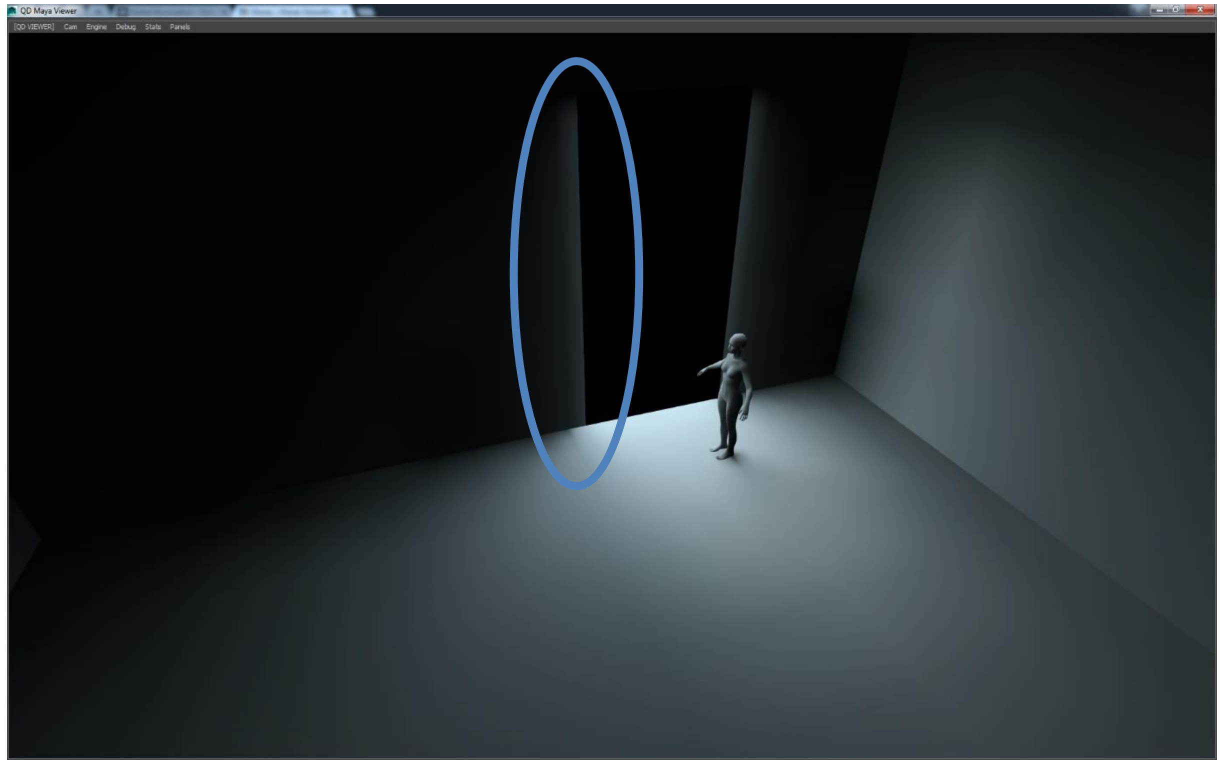
Probe based Lighting

- Probe based solution
  - Ideal for specular Image Based Lighting (IBL)
  - Capture scene cubemaps
  - Bake GGX NDF [WAL07][KARIS14]
    - Filtered importance sampling to prevent fireflies.
  - Artist controlled [LAG12] :
    - Influence boxes
    - Parallax boxes



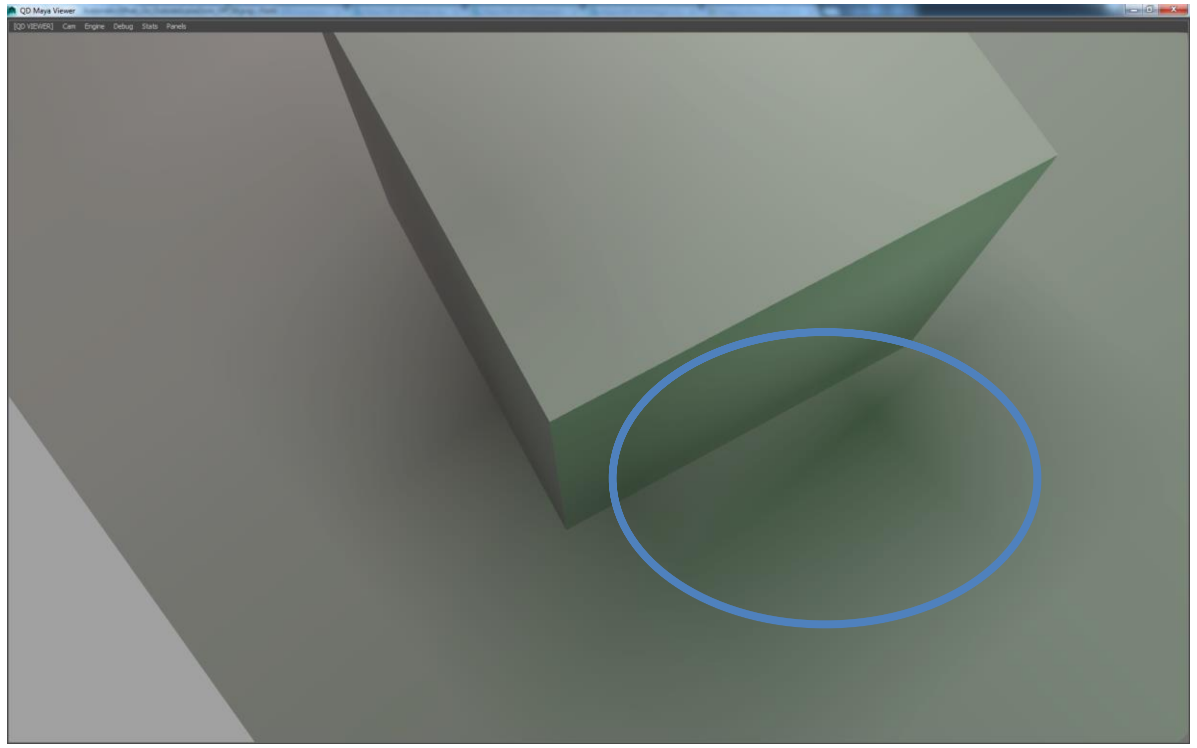
Probe Grid

- Diffuse Probe Grid flaws :
  - Light leaking
    - On-going research area
    - Interesting solutions : [MCG17], [IWA17]
  - Interpolation irregularities
    - Not often addressed but considered problematic for us



Probe Grid

- Rejecting probes based on occlusion often leads to interpolation artefacts
- Ended up never discarding a probe



Irradiance Sparse Octree

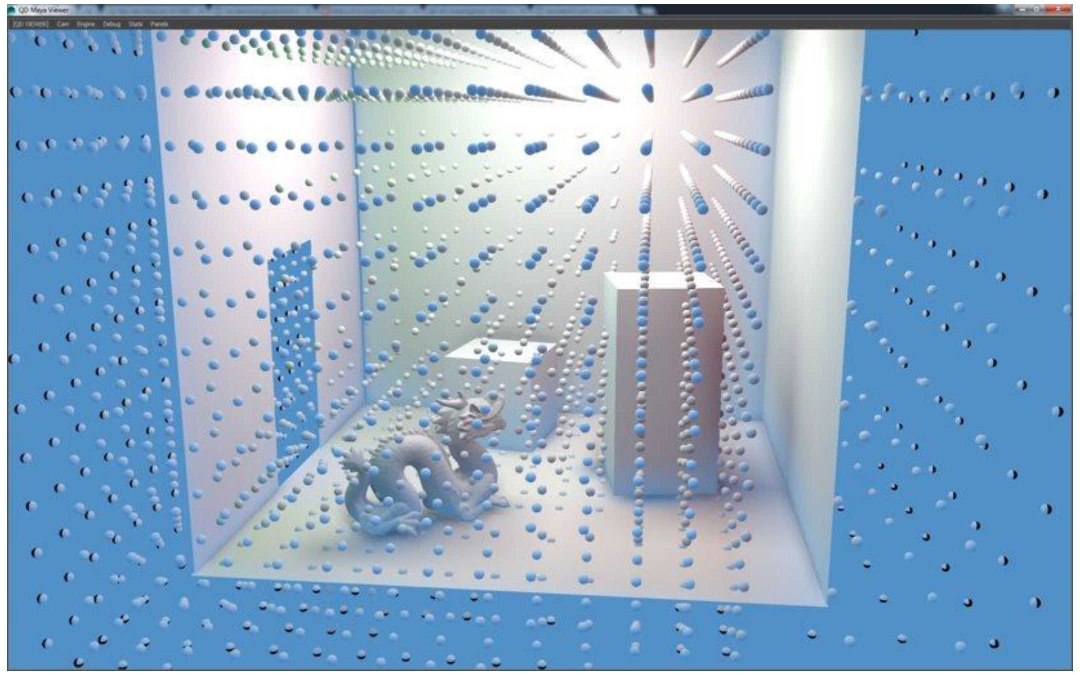
- Irradiance volumes [TAT07]
- Adaptive Sparse Octree
- Automatically built with artist cues
  - Density Zones
  - Automatic scene Voxelization for finer resolution around objects/walls

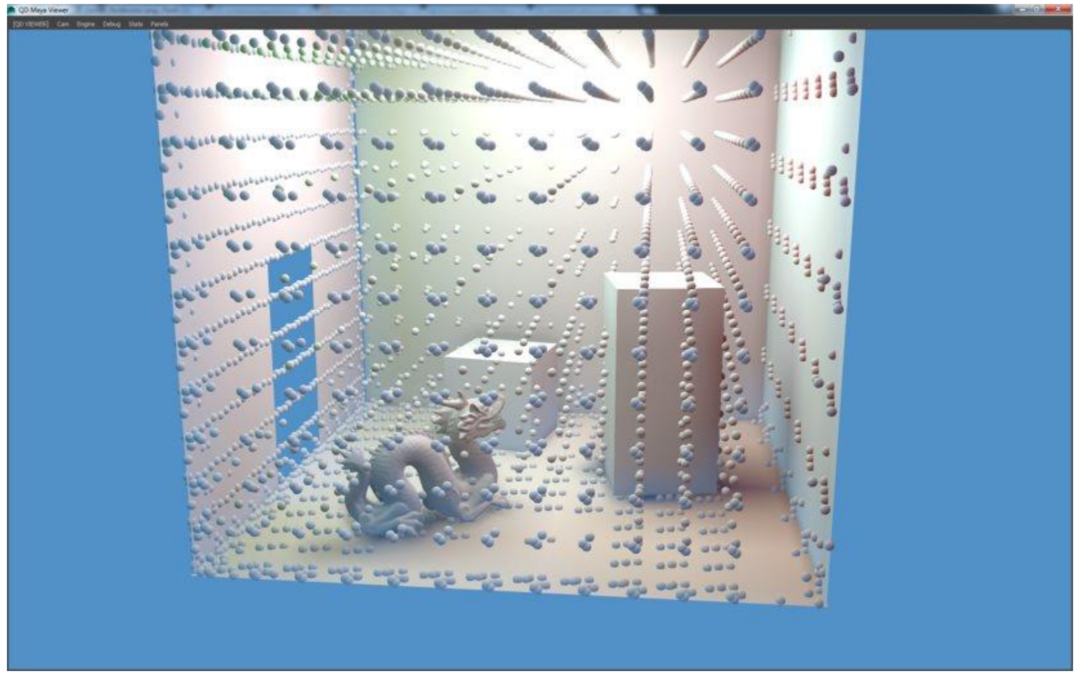
Irradiance Sparse Octree

- One octree cell has 8 probes, one per corner
- Space point always surrounded by 8 probes
- Never discard any probe
- Virtually offset probes instead

Probes attractor

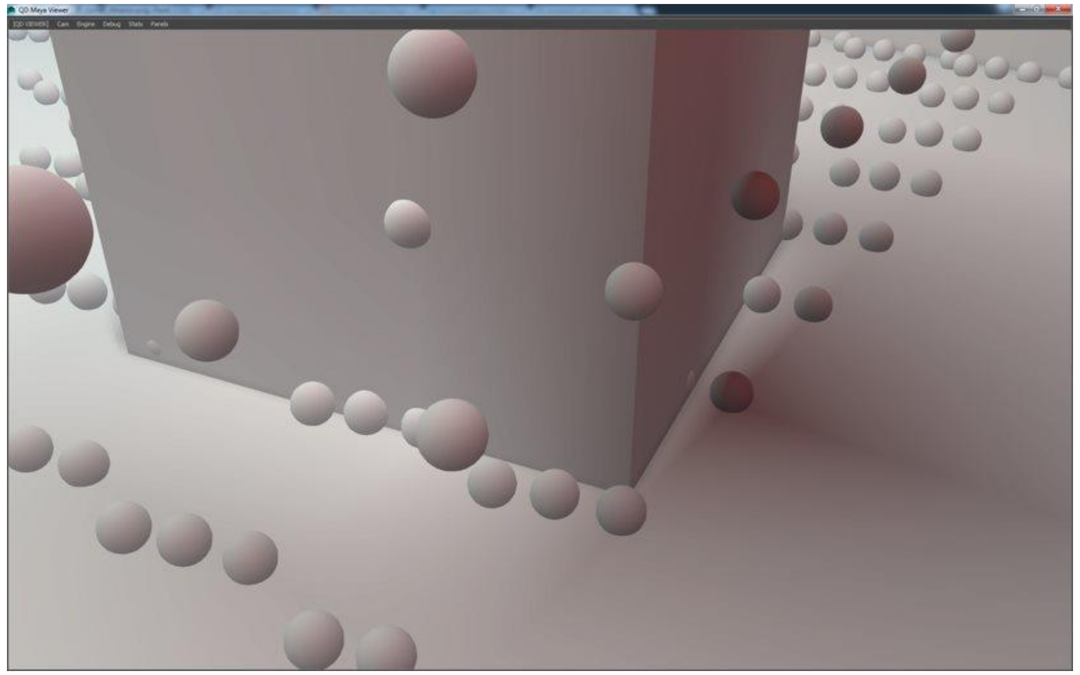
- Artist setup closed mesh that attracts all probes
  - Usually close to *Volume Zone* already used for visibility

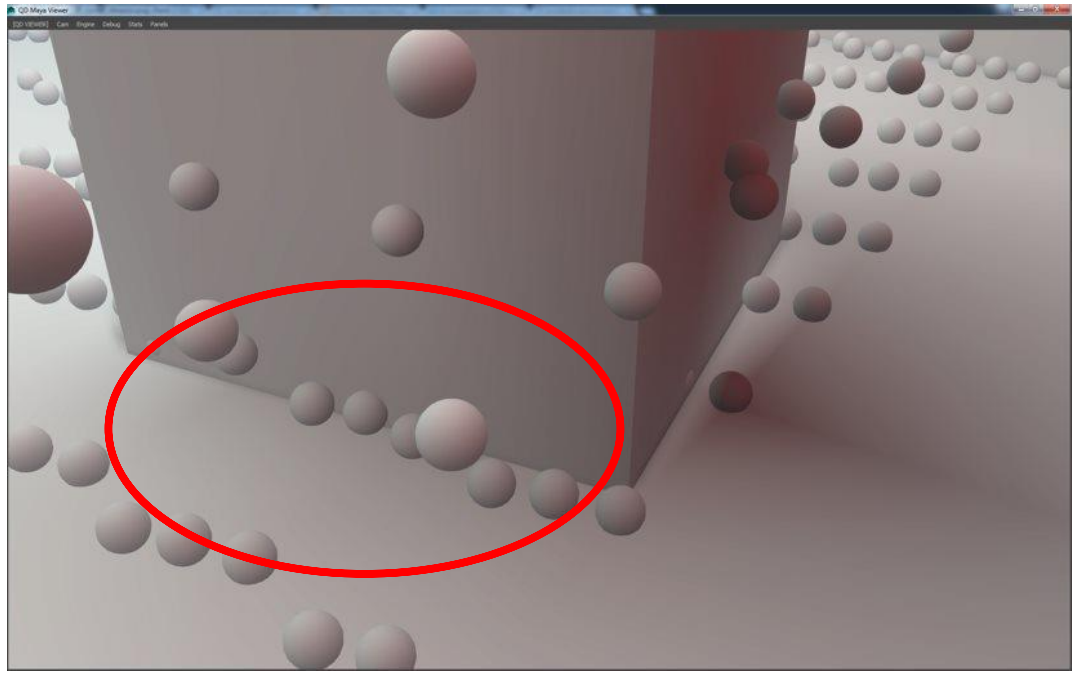




Probes repulsor

- Closed mesh that repulse probes inside volume

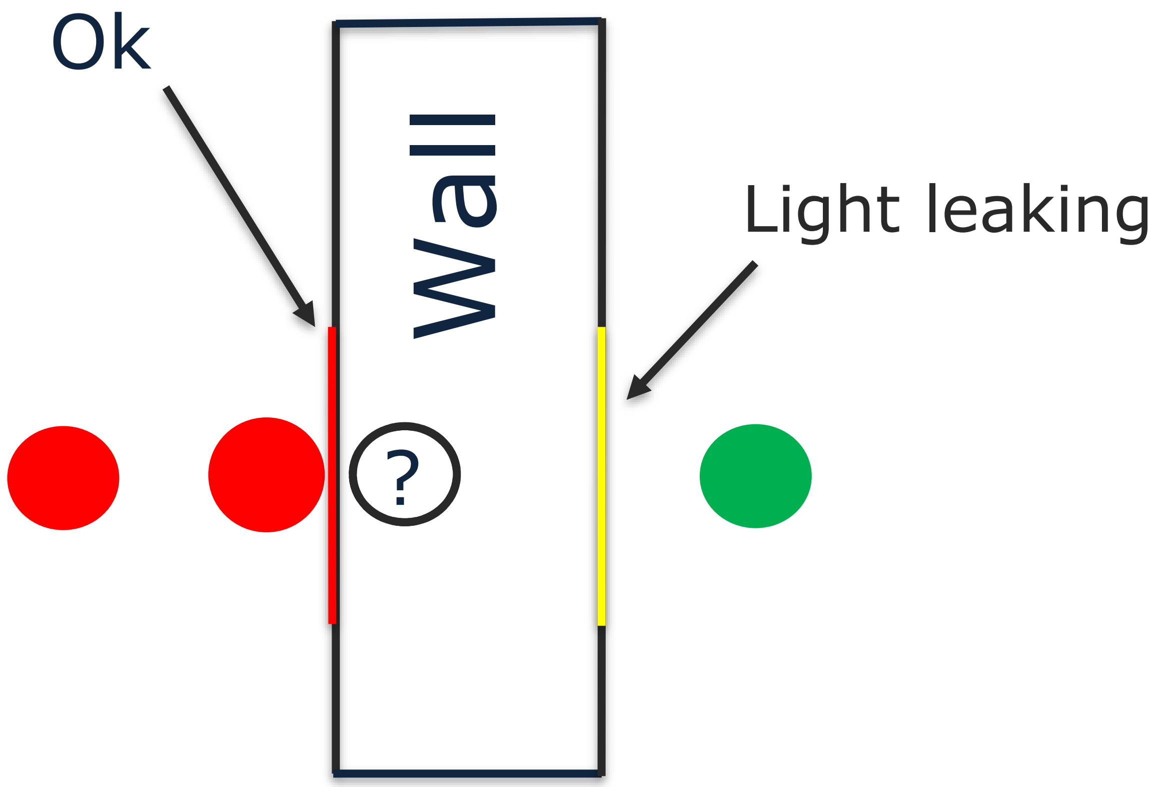




Probes offset

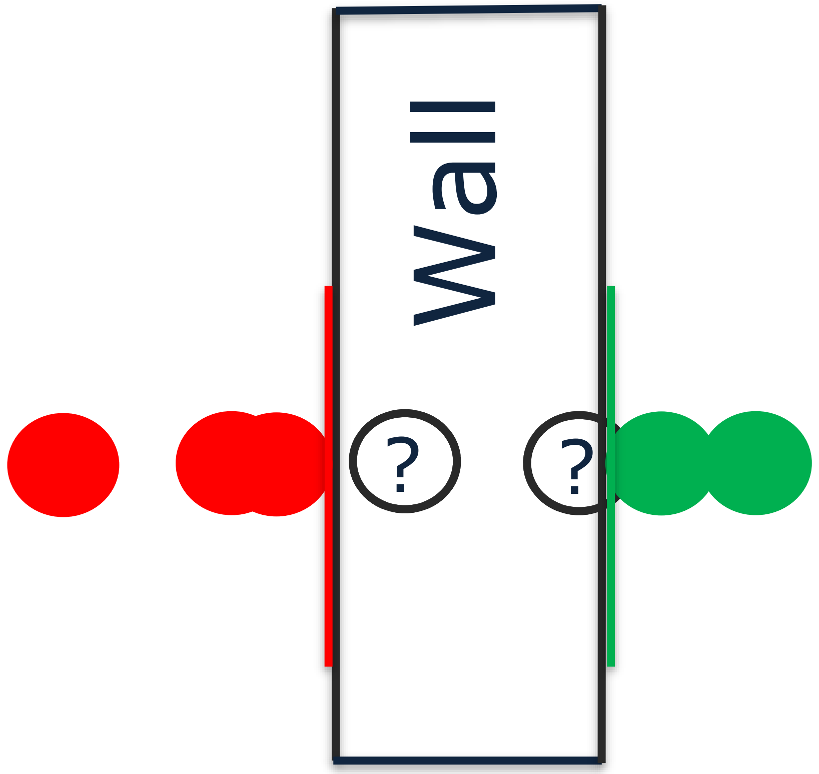
- Solves majority of cases
- Virtual offset during capture / baking
- Evaluation based on original grid positions

- Some cases are more difficult



Probes offset

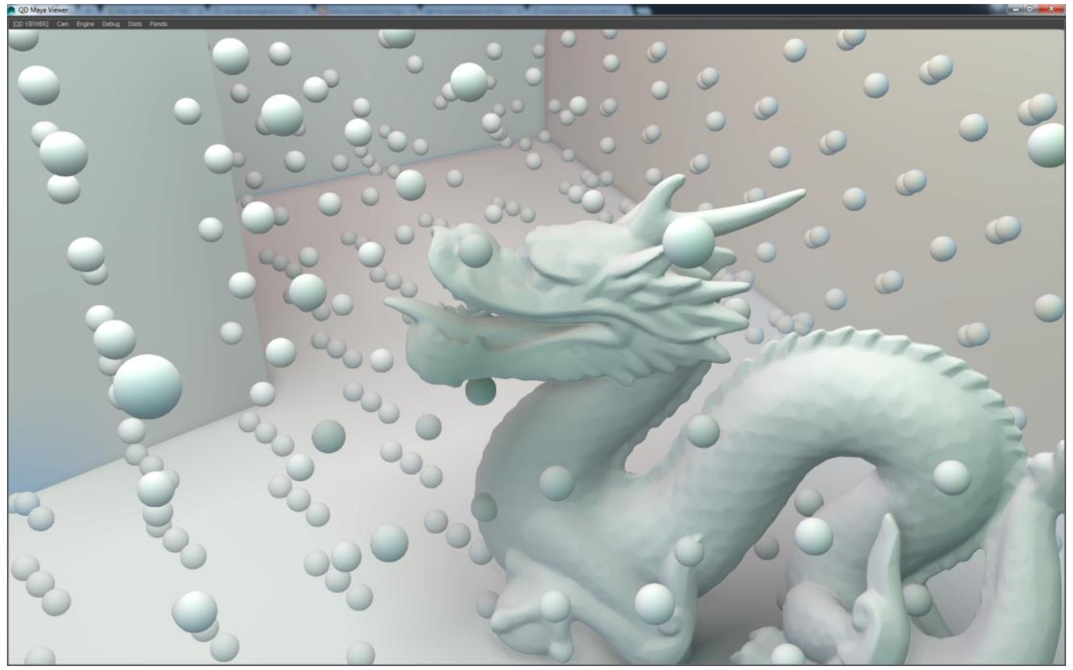
- If leaking on walls or obstacles, subdivide and offset
- Or try to minimize error
  - Bright leaks are more noticeable than dim leaks

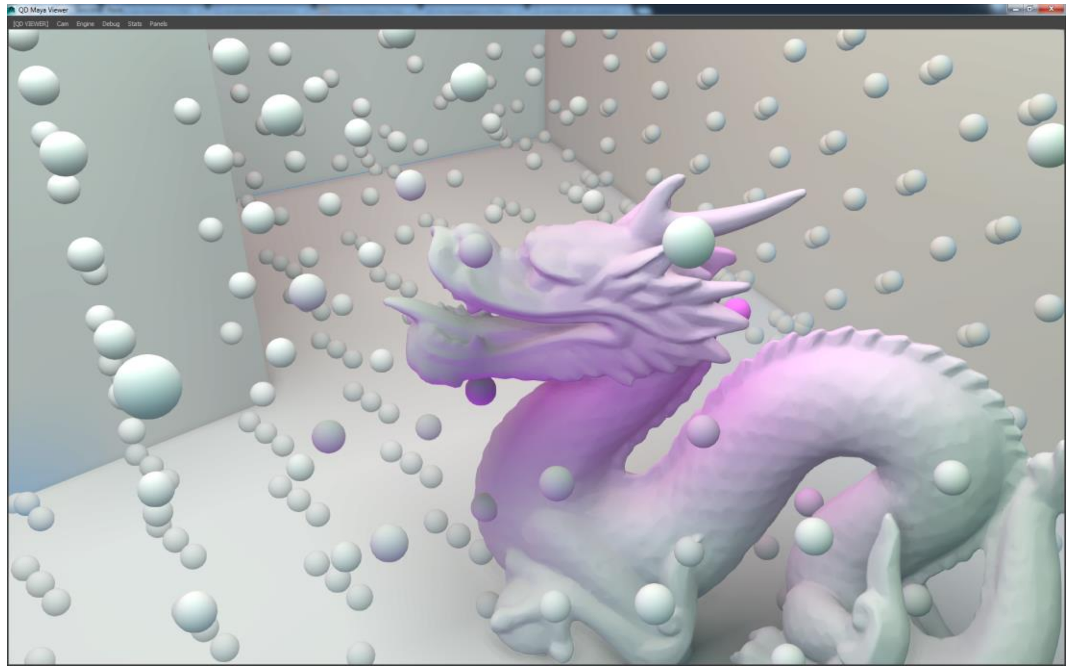


No leaking

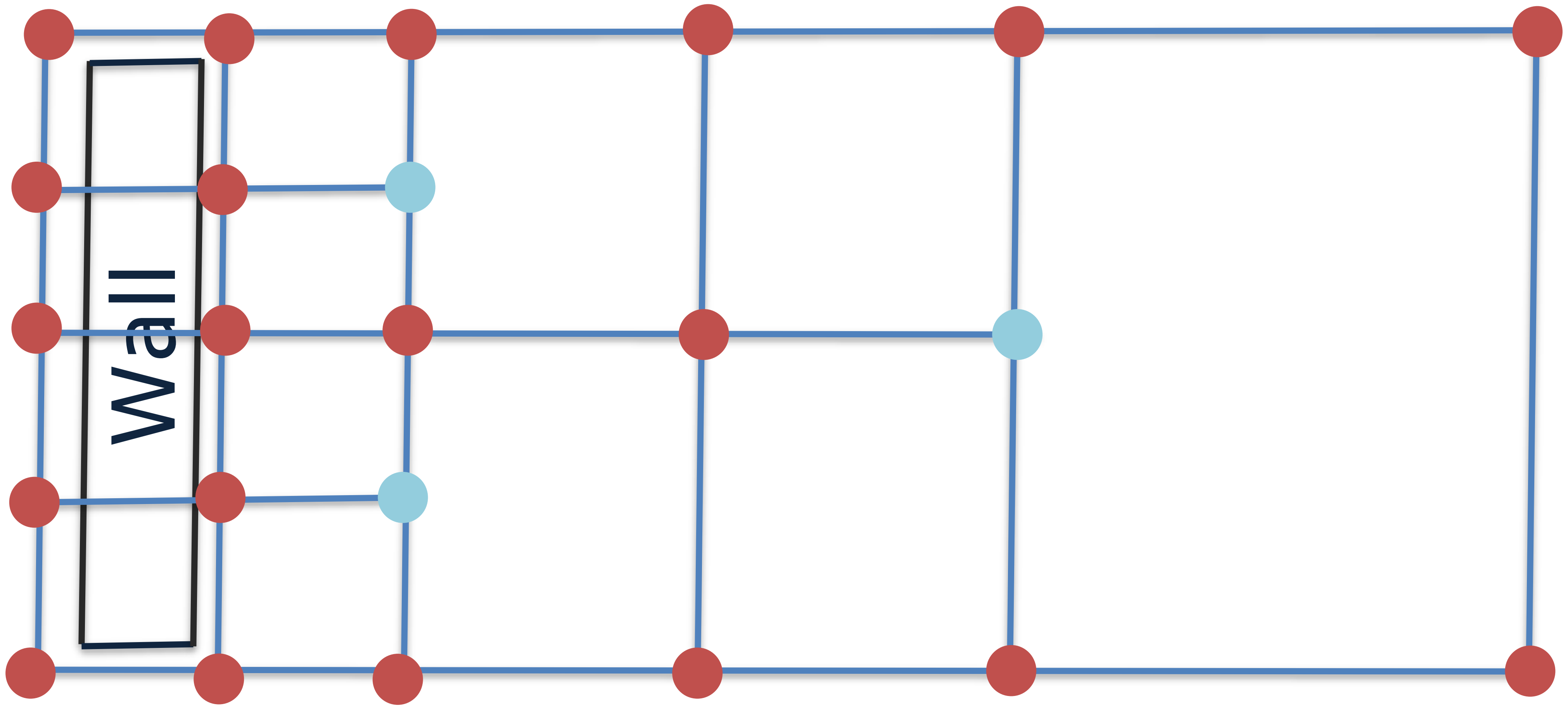
Probes tweaking

- Probes color modifiers for (last resort) manual modifications





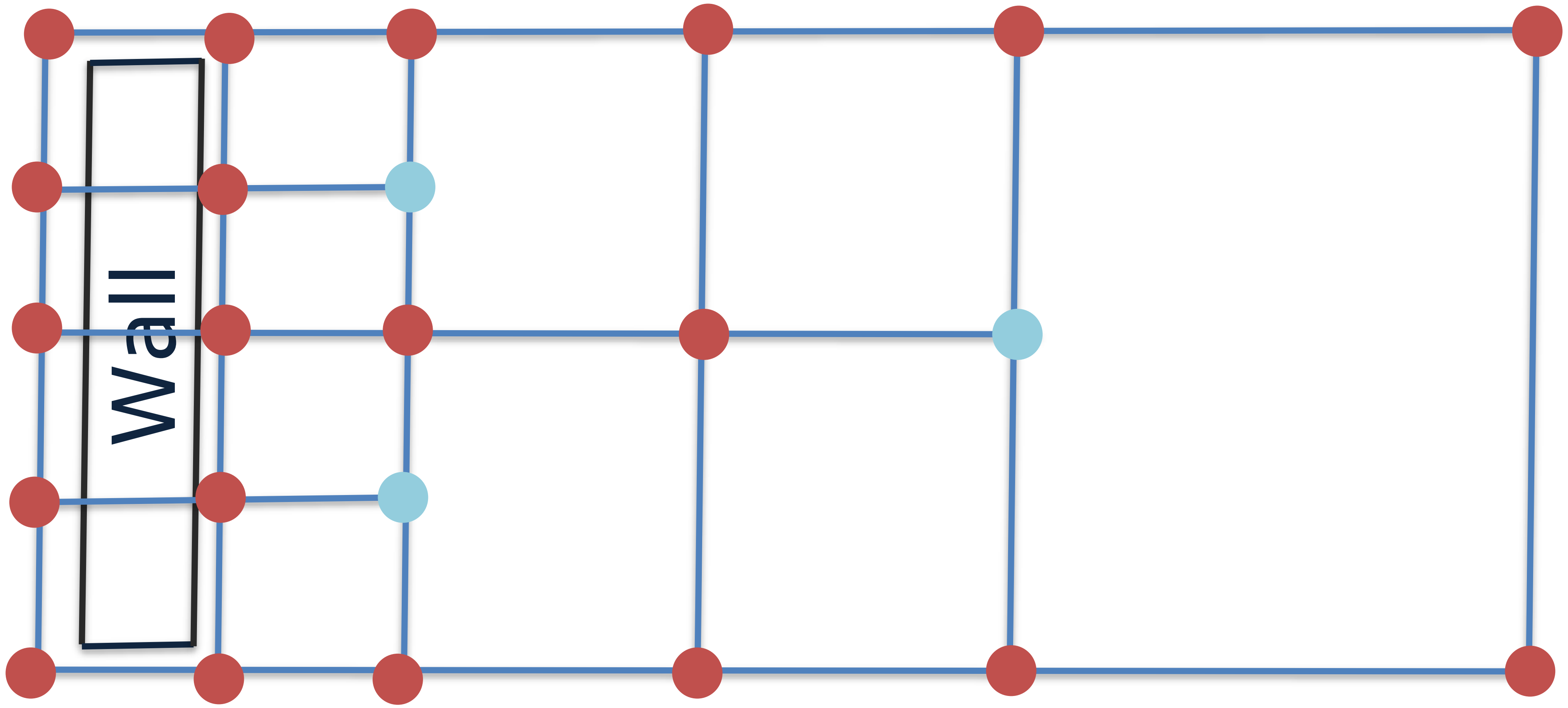
Octree level discontinuity

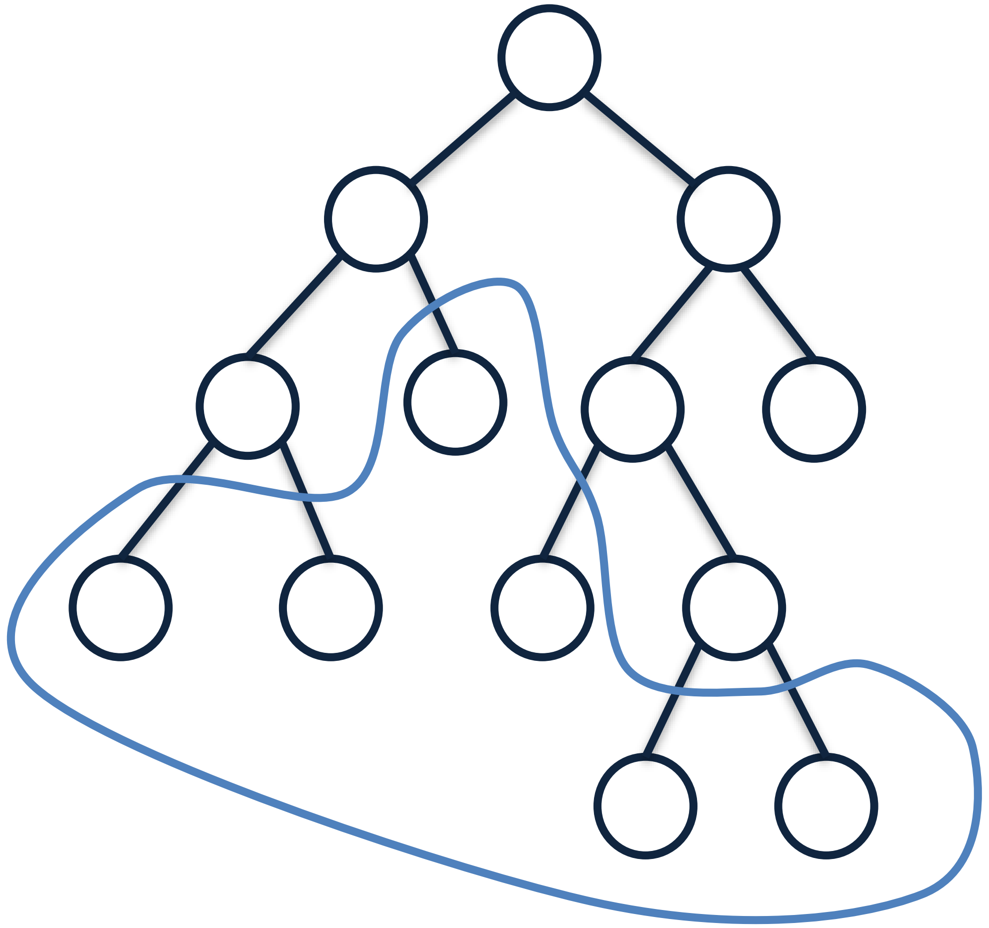


● Computed probes

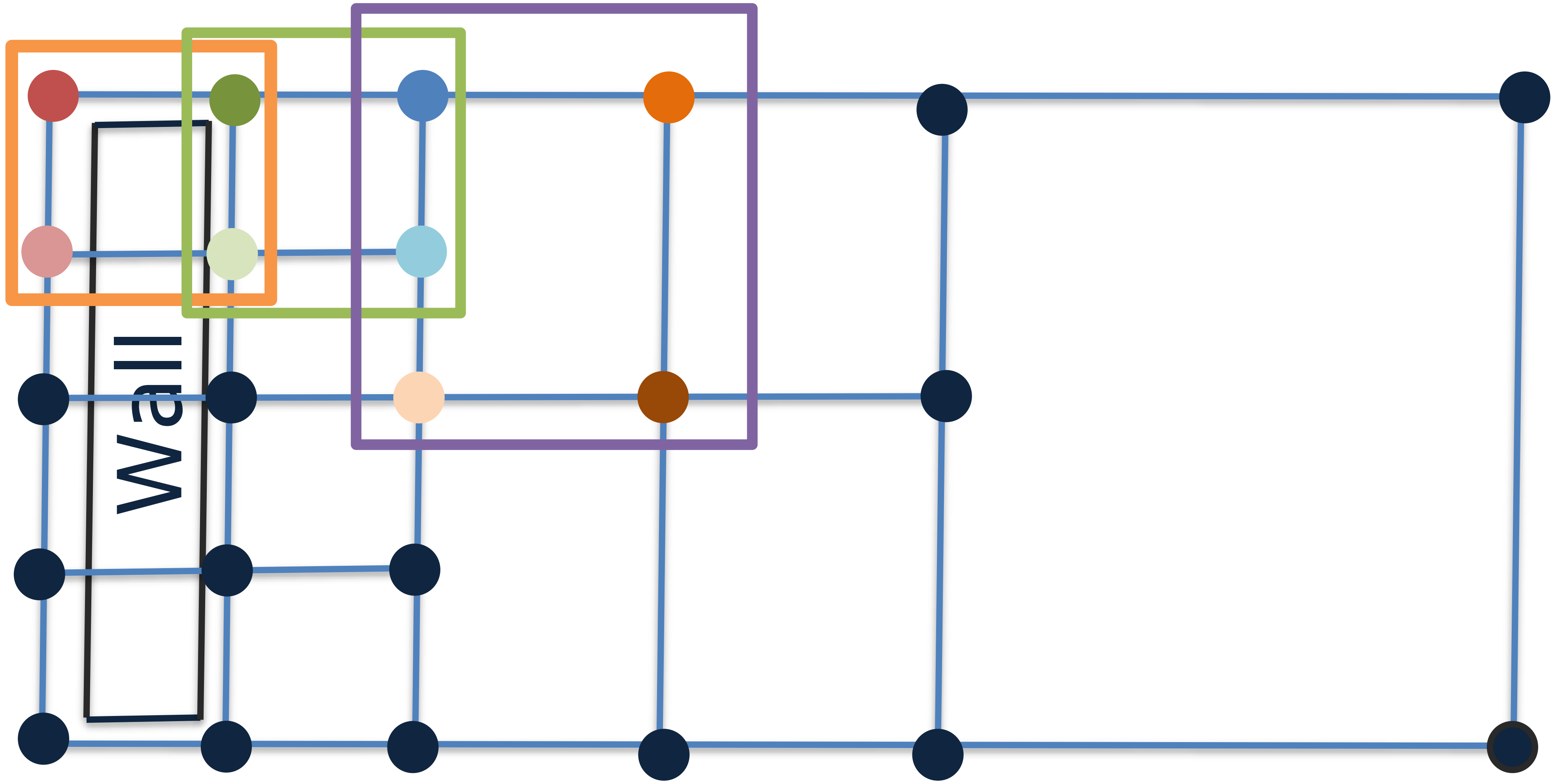
● Interpolated probes

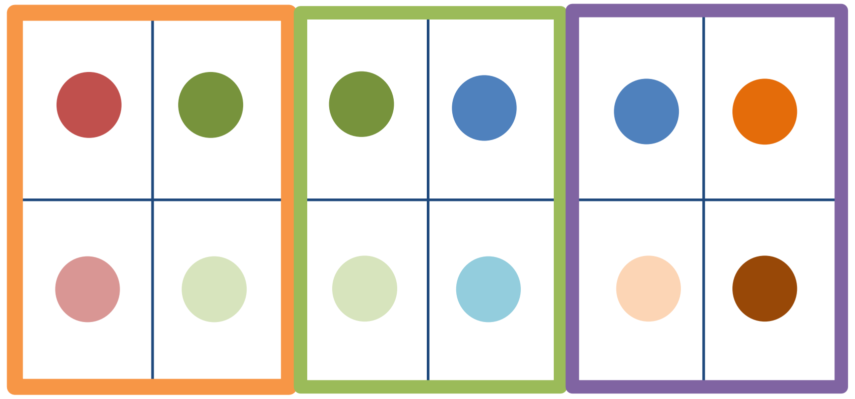
Volume texture Atlas





Volume texture Atlas

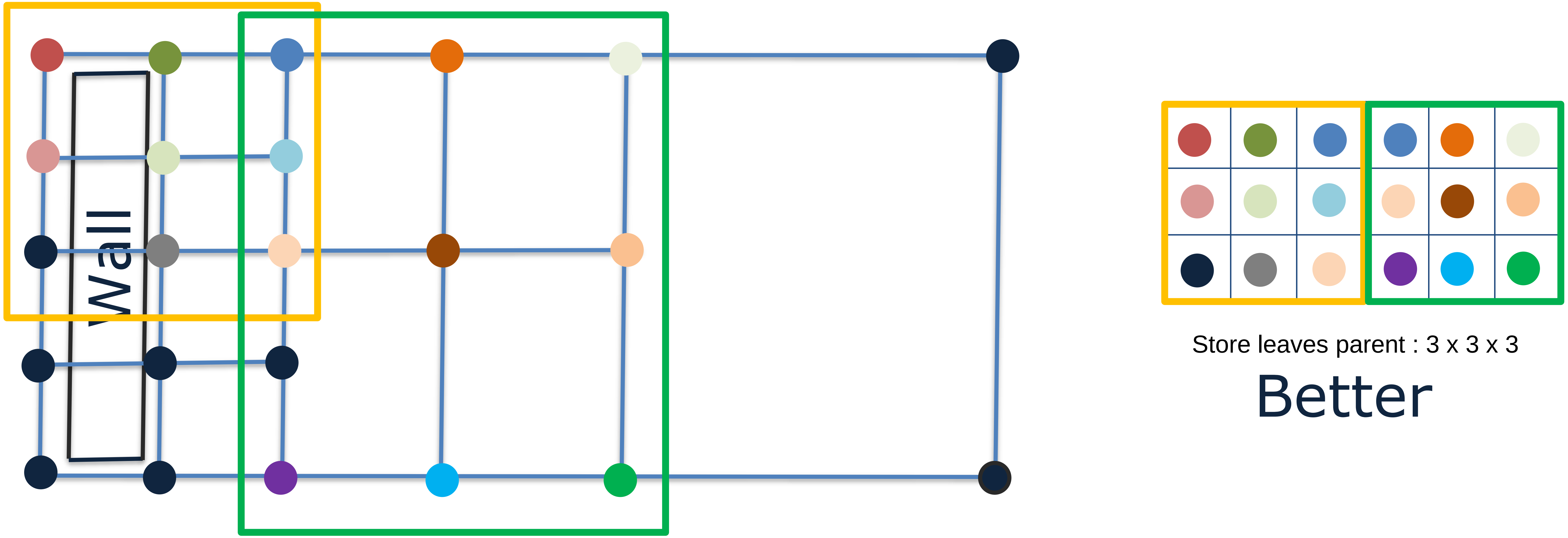




Store every leaves : 2 x 2 x 2

Lot of redundancy

Volume texture Atlas



Store leaves parent : 3 x 3 x 3

Better

Spherical Harmonics

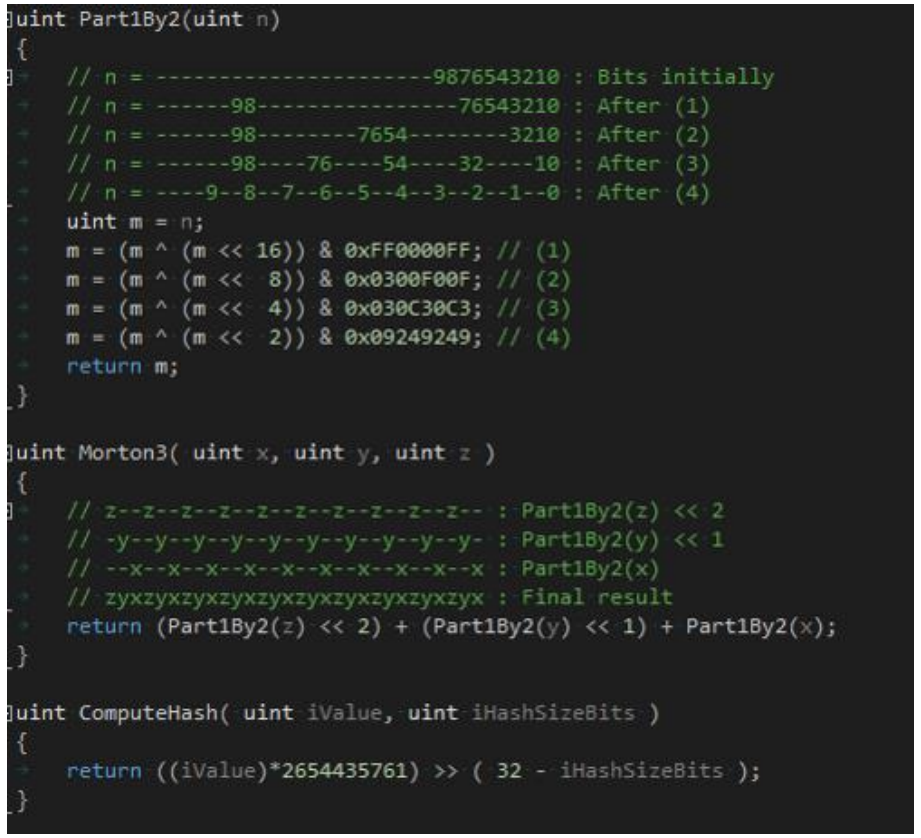
- 2nd order SH
  - 4 coefficients
  - Using Geomerics reconstruction [GEOM15]
- 3 RGBA16F volume textures (R, G, B)
  - 24 055 probes → (105x105x3) x 3 textures ~= 3MB

GI Display

- Never discarding probes allows sampling volumetric texture
- Simply use hardware (3D) bilinear filtering

GI Display

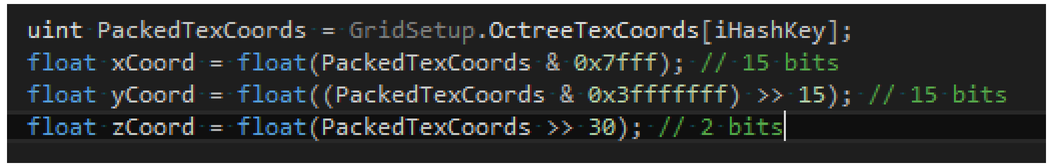
- Octree cell hash key is found from 3d position
    - Morton Key
    - O(1)
    - Restricted to 32 bits
- Use precomputed hash to texcoord buffer
    - X coord coded on 15 first bits
    - Y coord coded on following 15 bits
    - Z coord coded on last 2 bits



```
uint Part1By2(uint n)
{
    // n = ----------------------9876543210 : Bits initially
    // n = ------98----------------76543210 : After (1)
    // n = ------98--------7654--------3210 : After (2)
    // n = ------98----76----54----32----10 : After (3)
    // n = ----9--8--7--6--5--4--3--2--1--0 : After (4)
    uint m = n;
    m = (m ^ (m << 16)) & 0xFF0000FF; // (1)
    m = (m ^ (m <<  8)) & 0x0300F00F; // (2)
    m = (m ^ (m <<  4)) & 0x030C30C3; // (3)
    m = (m ^ (m <<  2)) & 0x09249249; // (4)
    return m;
}

uint Morton3( uint x, uint y, uint z )
{
    // z--z--z--z--z--z--z--z--z--z-- : Part1By2(z) << 2
    // -y--y--y--y--y--y--y--y--y--y- : Part1By2(y) << 1
    // --x--x--x--x--x--x--x--x--x--x : Part1By2(x)
    // zyxzyxzyxzyxzyxzyxzyxzyxzyxzyx : Final result
    return (Part1By2(z) << 2) + (Part1By2(y) << 1) + Part1By2(x);
}

uint ComputeHash( uint iValue, uint iHashSizeBits )
{
    return ((iValue)*2654435761) >> ( 32 - iHashSizeBits );
}
```



```
uint PackedTexCoords = GridSetup.OctreeTexCoords[iHashKey];
float xCoord = float(PackedTexCoords & 0x7fff); // 15 bits
float yCoord = float((PackedTexCoords & 0x3fffffff) >> 15); // 15 bits
float zCoord = float(PackedTexCoords >> 30); // 2 bits
```

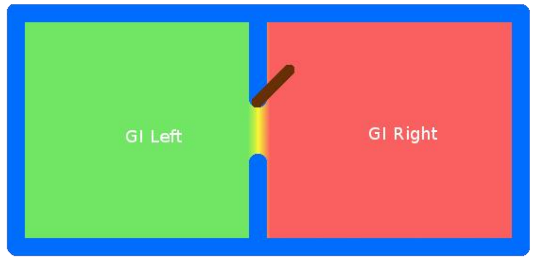
GI Bonuses

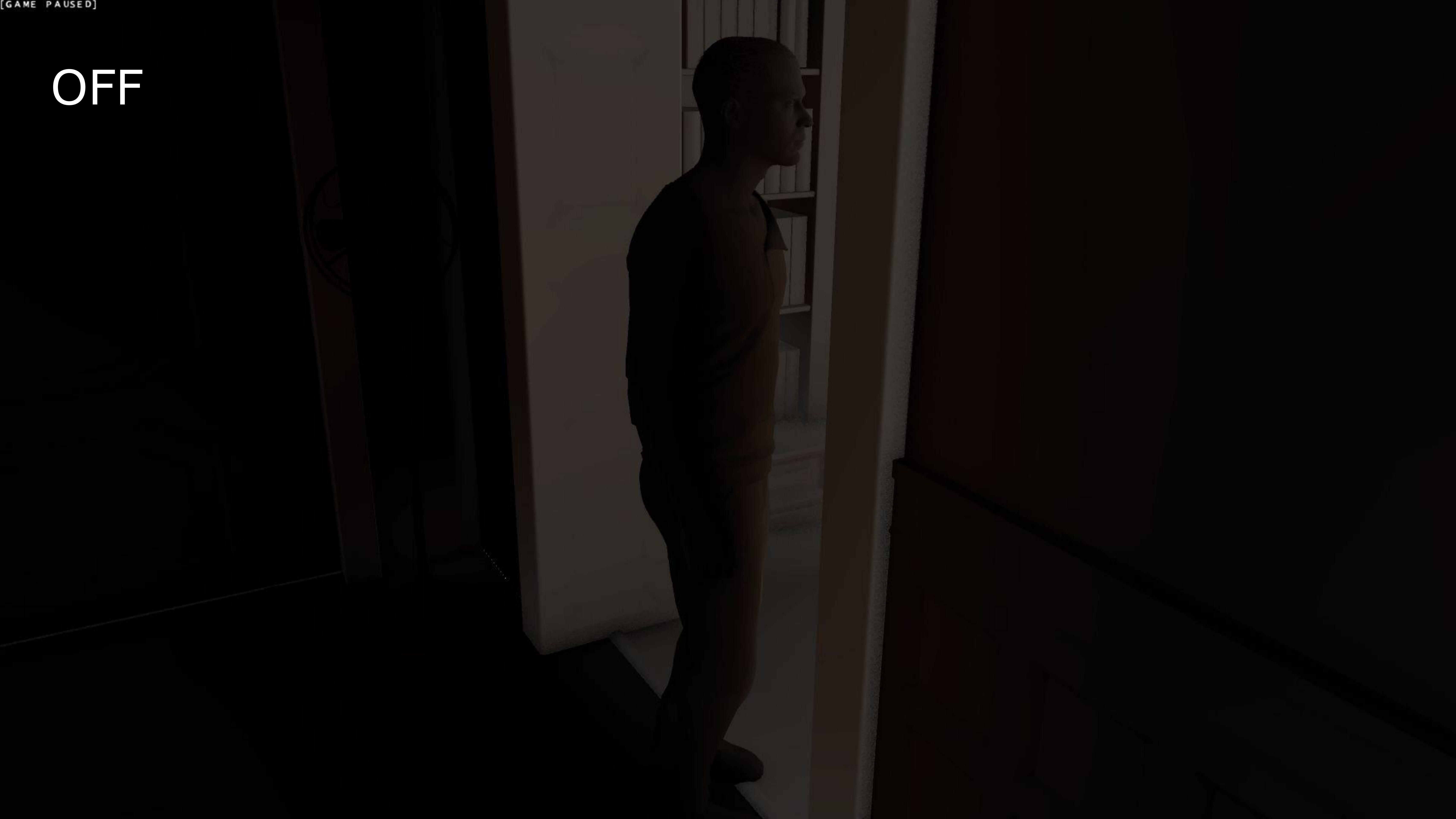
- Trivial to blend multiple GI sets with Compute Shader
  - GI switch (interior light switch, lightning, ...)
  - GI transition (time of day, curtains opening/closing, …)

DRAW

GI transition

- Avoid hard GI transition between Scene Zones
  - Interior ↔ Exterior
  - Setup distance around portal
  - Dynamic objects passing through sample both GIs
  - Based on distance to portal & normal direction





OFF

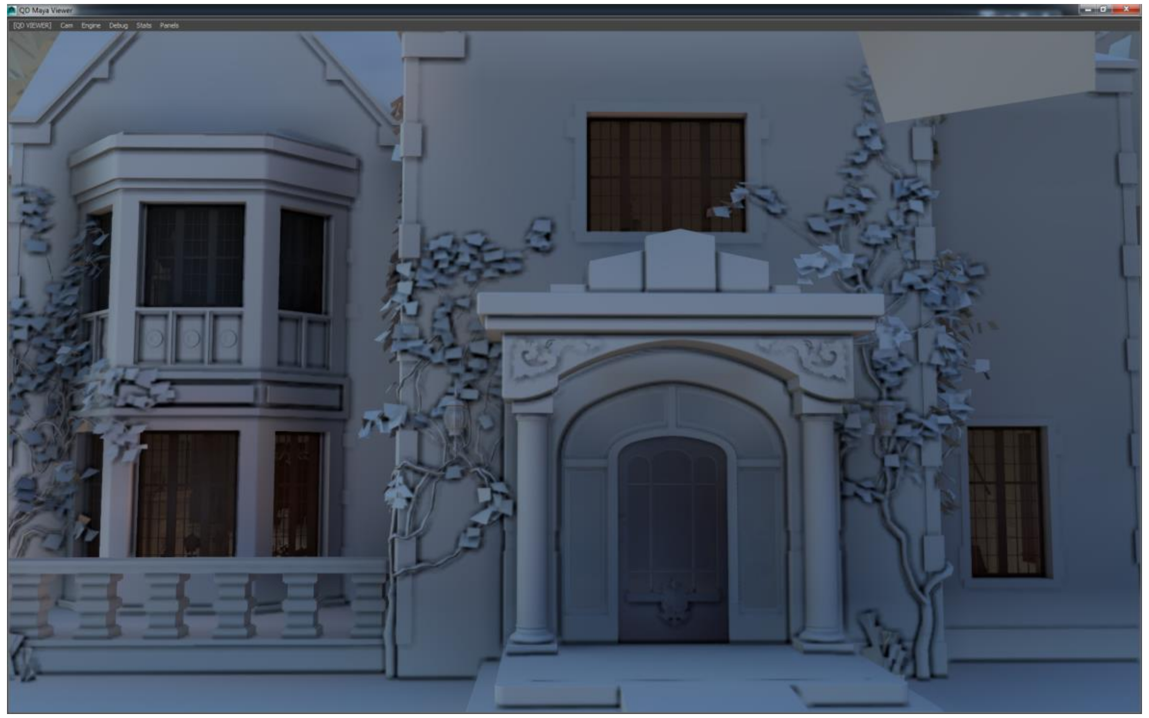
ON

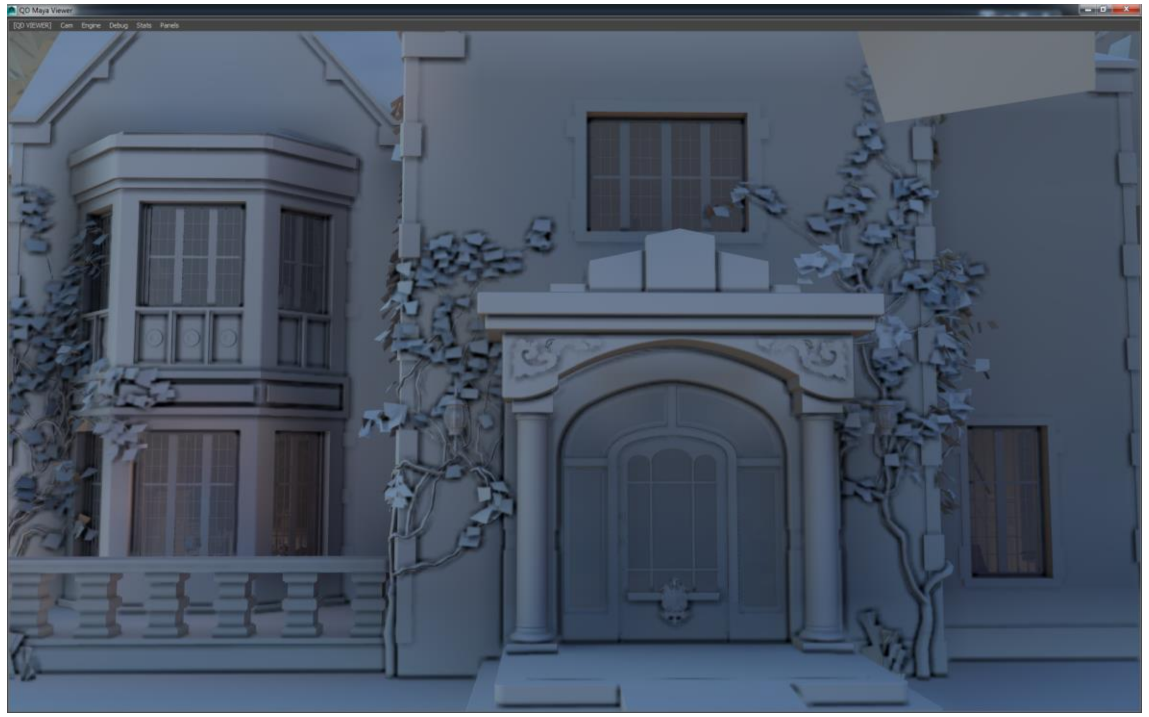
GI transition

- Most static objects are in a unique scene zone
  - Interior walls vs Exterior walls
- What about doors, windows, window frames ?
  - Assigned to a scene zone but also visible from others
  - Not assigned to a scene zone

GI transition

- Manually tag objects to sample both GIs





Conclusion

Conclusion

- The road to PBR never ends
  - Need better energy conservation for lights and materials
  - Materials are not physically accurate enough
  - Being Physically based is good, but **don't forget artists visions and needs**
  - Photometric units provides a good framework to ensure coherent asset production
  - Always use reference environment to validate materials

Conclusion - What’s next

- Area lights and soft shadows everywhere ♥
- Improve BRDF layer stack
- Better handling of energy conservation
- Add dynamic component to our GI
- Volumetric Lighting improvement

Acknowledgements

- Rendering Team
  - Christophe Bonnet
  - Bertrand Cavalié
  - Thibault Lambert
  - Gregory Lecot
  - Eric Lescop
  - Ronan Marchalot
  - Sylvain Meunier
  - Nicolas Vizerie

- Special Thanks
  - Jean-Charles Perrier
  - Antoine Galbrun
  - Julien Merceron

- Indirect Lighting Slides
  - Thibault Lambert

Questions ?

Email : gcaurant@quanticdream.com
Twitter: @GCaurant

References

- [GEOM15] Reconstructing Diffuse Lighting from Spherical Harmonic Data
- [HIL15] Physically Based and Unified Volumetric Rendering in Frostbite
- [HOL11] Saying “Goodbye” to Shadow Acne
- [IWA17] Precomputed lighting in Call of Duty : Infinite Warfare
- [KARIS13] Real Shading in Unreal Engine 4
- [LAG12] Local Image based Lighting with parallax-corrected cubemap
- [LAG14] Moving Frostbite to Physically based rendering
- [LAG16] An Artist-Friendly Workflow for Panoramic HDRI
- [LAN14] Physically Based Shader Design in Arnold
- [MCG17] Real-Time Global Illumination using Precomputed Light Field Probes
- [RAM01] An Efficient Representation for Irradiance Environment Maps
- [TAT07] Irradiance Volumes
- [WAL07] Microfacet models for refraction through rough surfaces
- [WRO14] Volumetric Fog: Unified compute shader based solution to atmospheric scattering