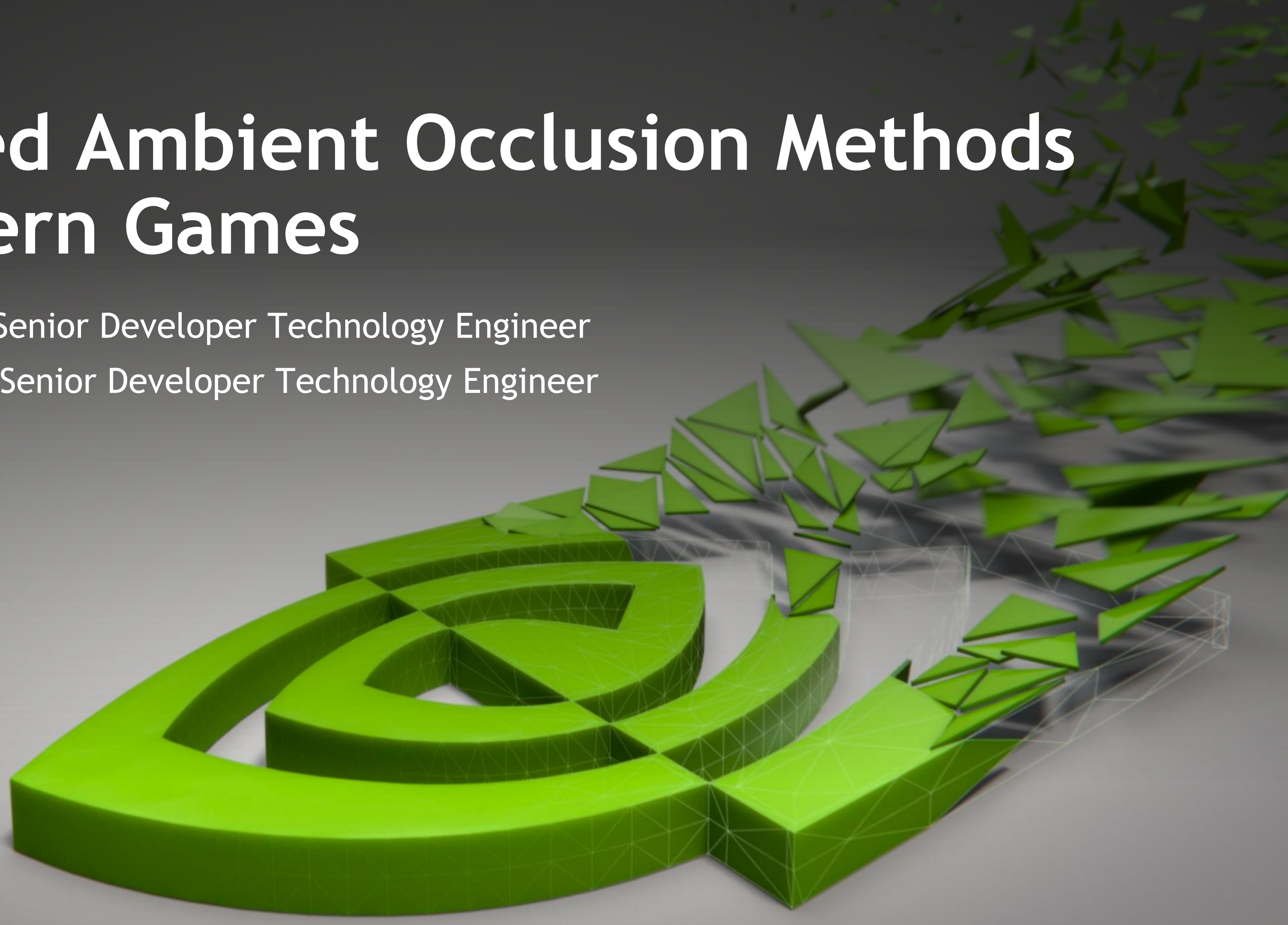
Advanced Ambient Occlusion Methods for Modern Games

Andrei Tatarinov, Senior Developer Technology Engineer

Alexey Panteleev, Senior Developer Technology Engineer

NVIDIA



Outline

- What is AO and why is it SS?
- Is screen space enough?
- HBAO+ Ultra
- Voxel Ambient Occlusion
- VXAO integrations

Screen Space Ambient Occlusion

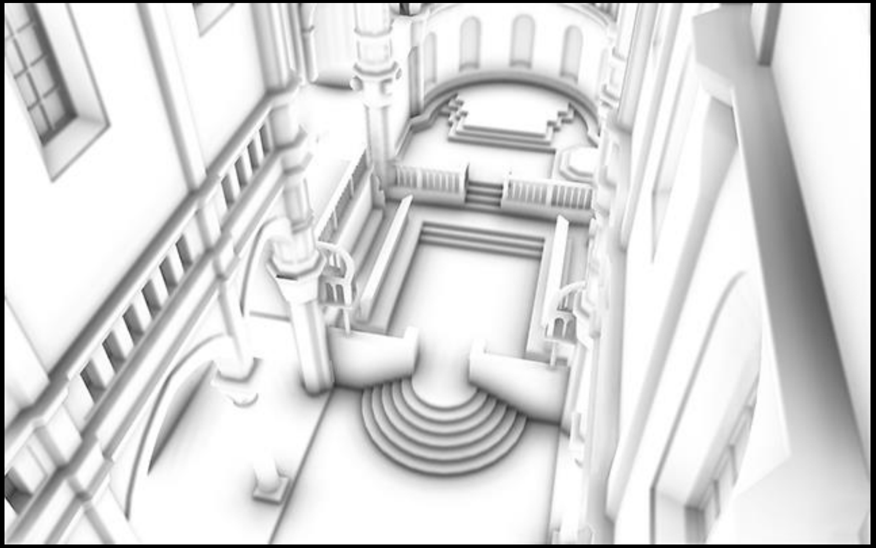
Screen Space Ambient Occlusion

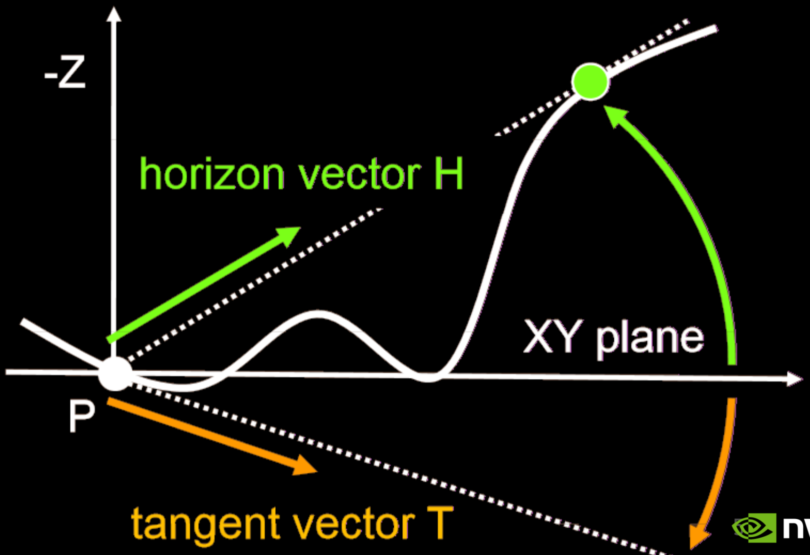
- A rendering technique for efficiently approximating ambient occlusion in the games
- Independent from scene complexity
- Minimal data pre-processing
- Easily integrated into any modern rendering pipeline

© Wikipedia

Horizon-Based Ambient Occlusion +

- Bavoil, L., Sainz, M., Image-Space Horizon-Based Ambient Occlusion, Siggraph 2008
- HBAO+ improves upon existing Ambient Occlusion techniques to add richer, more detailed, more realistic shadows around objects that occlude rays of light
- Compared to previous techniques, HBAO+ is faster, more efficient, and significantly better





Is screen space enough?

Is screen space enough?

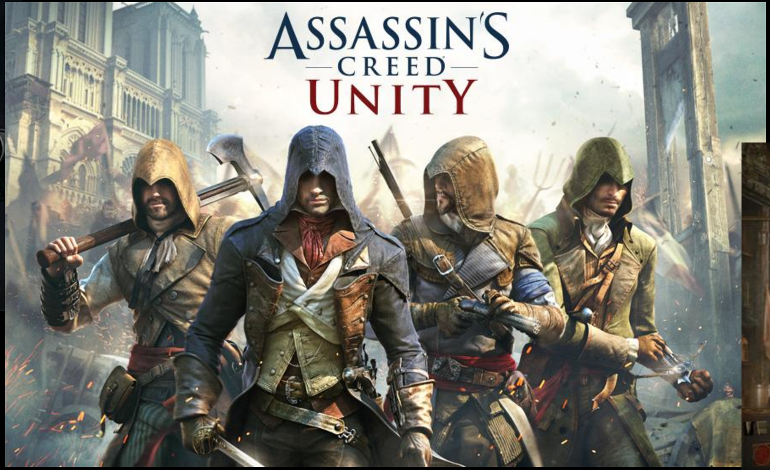
- No.

Is screen space enough?

- No.
- Let’s do an excursion into history…

Assassin’s Creed series

- Strategic partnership between NVIDIA and Ubisoft
- GameWorks is featured in three major installments



Assassin’s Creed IV Black Flag

- Towns look like chaotic collections of buildings
- Objects are fairly far away from each other

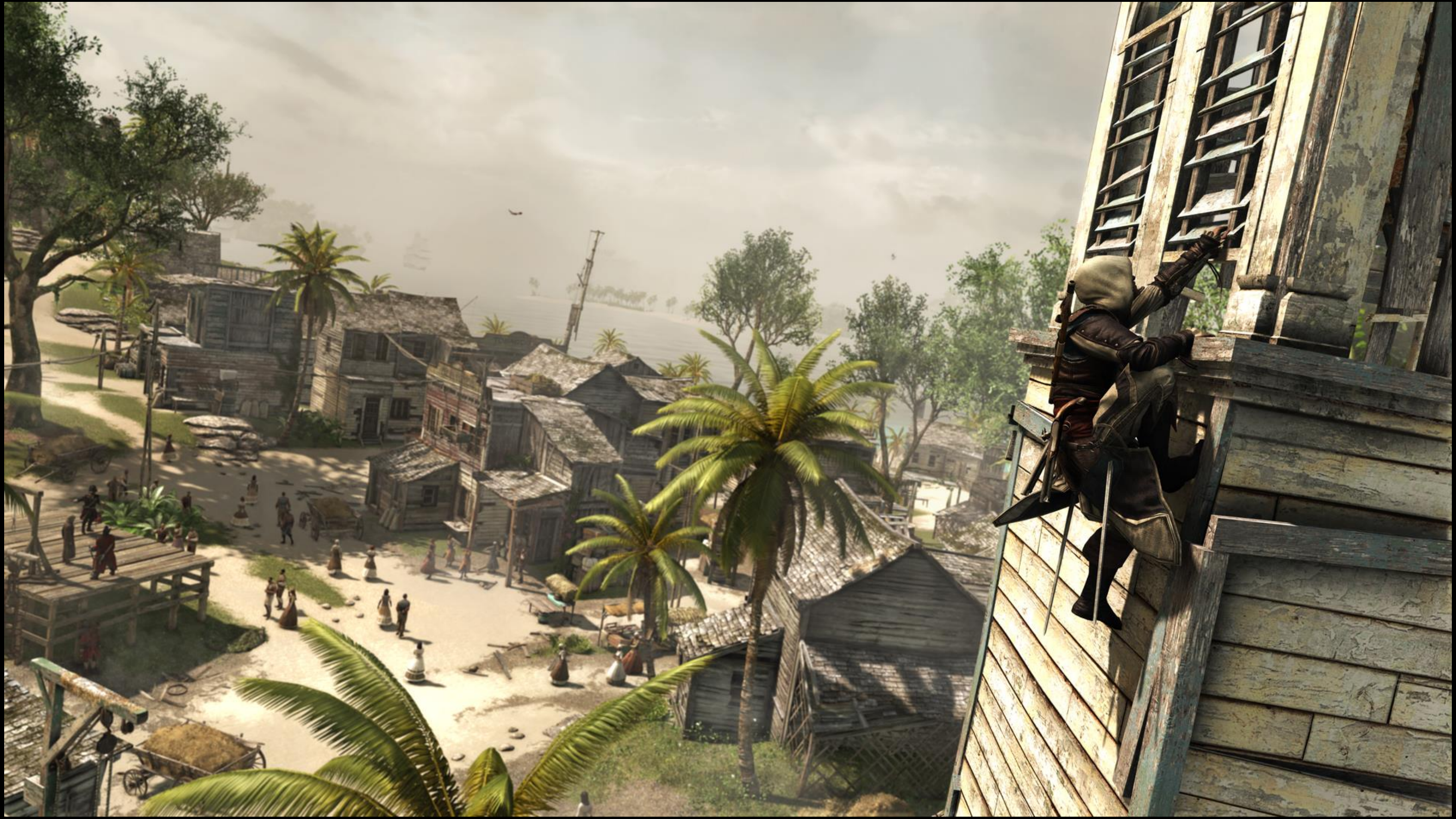


Image courtesy of Ubisoft

Assassin’s Creed Unity

- People are always protesting on the streets and squares

Image courtesy of Ubisoft

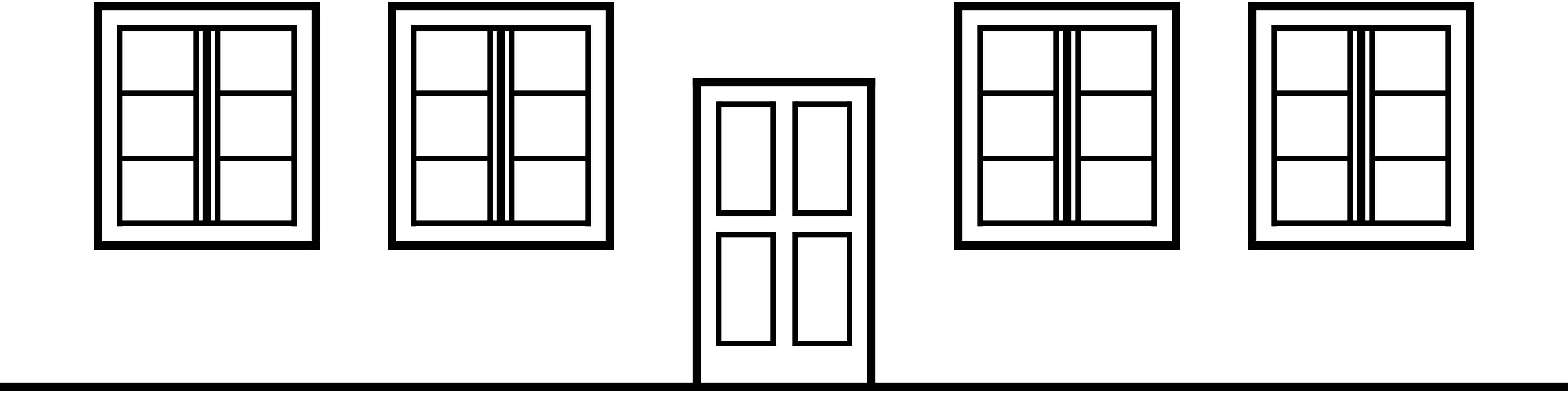
Assassin's Creed Syndicate

- XIX century London brings some order to the crowd

Image courtesy of Ubisoft

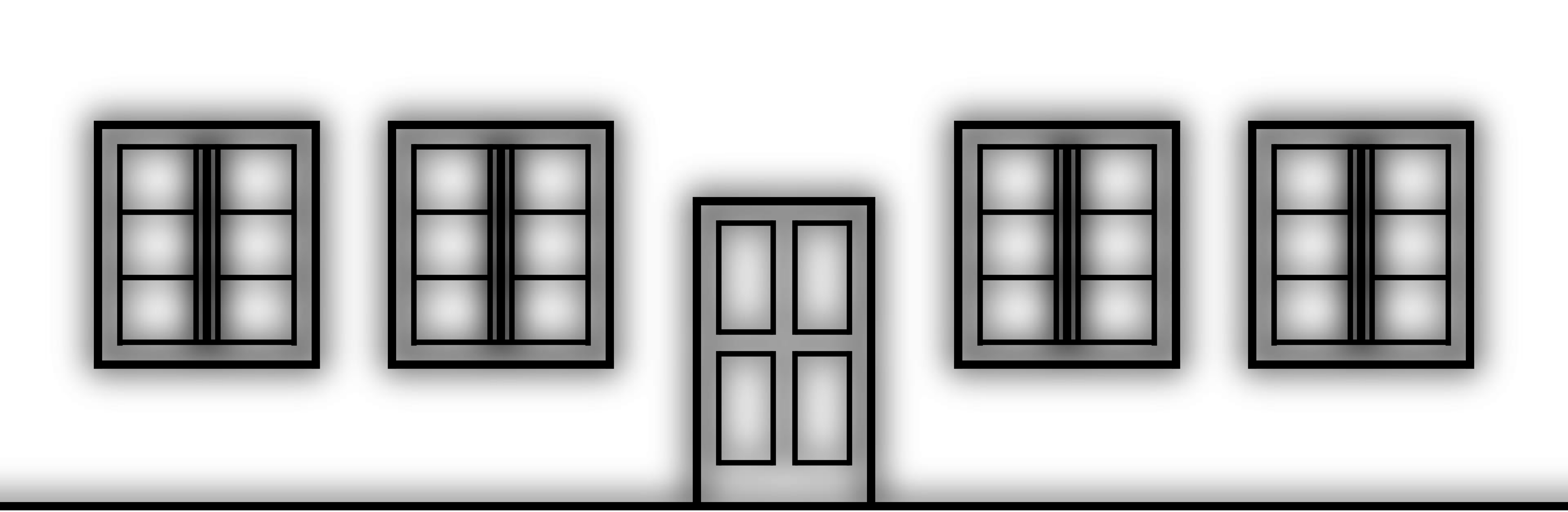
What’s gone wrong?

- Let’s take a simple scene



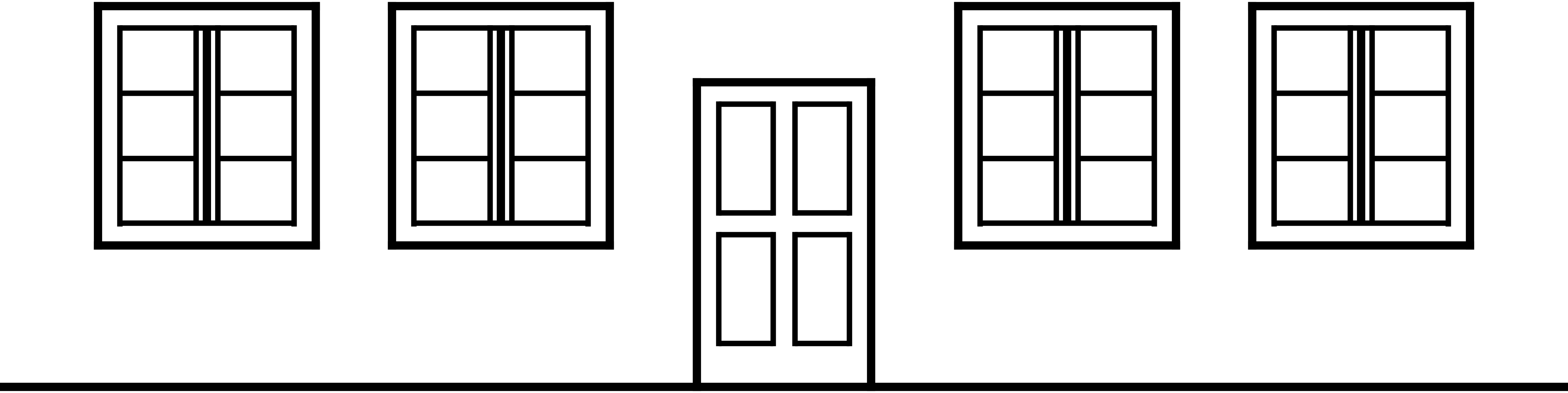
What’s gone wrong?

- This is how AO would look like:



What’s gone wrong?

- Let’s try to add a character walking by:



What’s gone wrong?

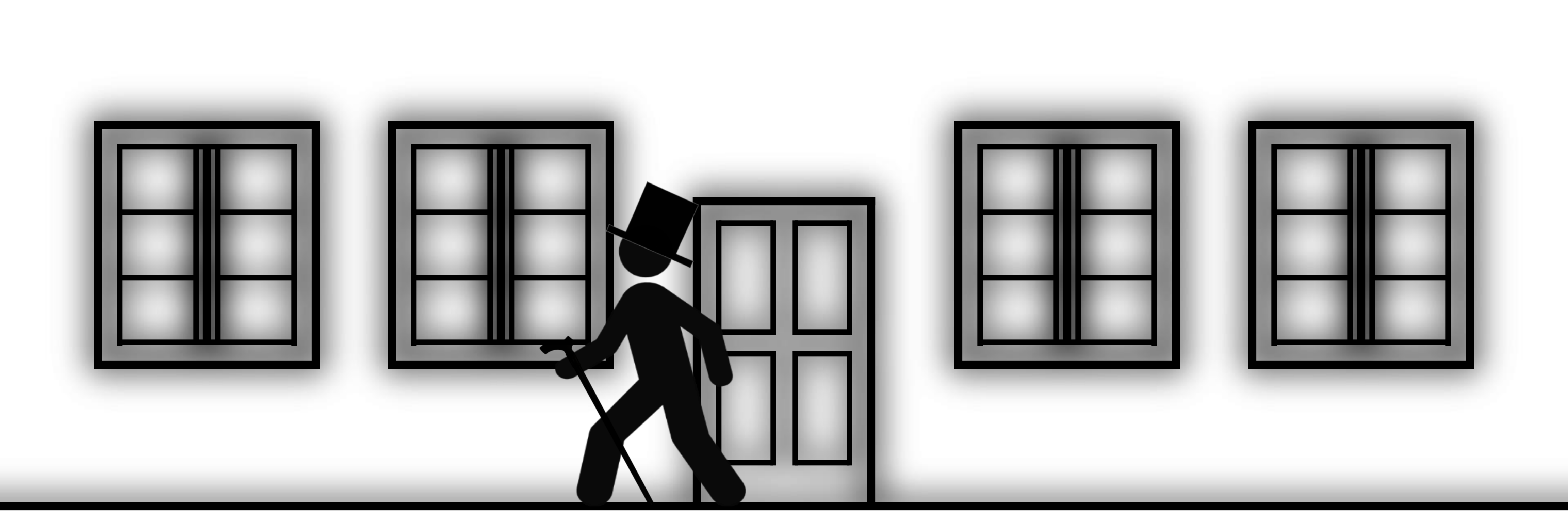
- Let’s try to add a character walking by:

What's gone wrong?

- Let's try to add a character walking by:

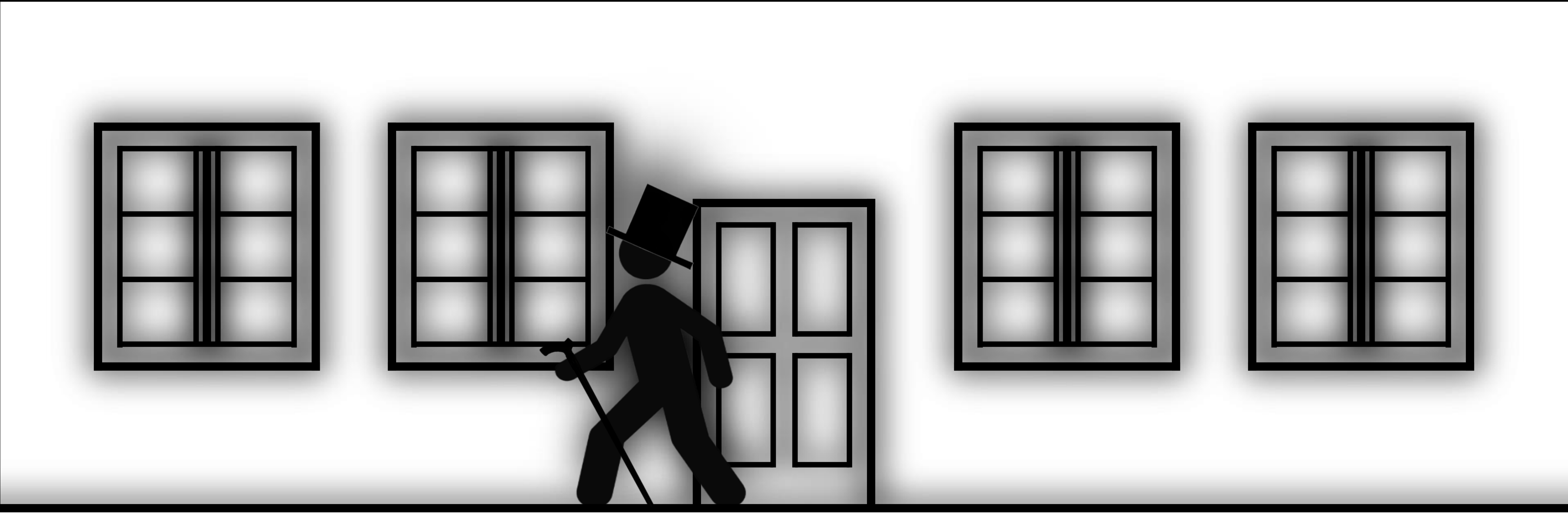
What's gone wrong?

- We want minimal influence of the character on the AO image



What’s gone wrong?

- AO looks correct if character is close enough

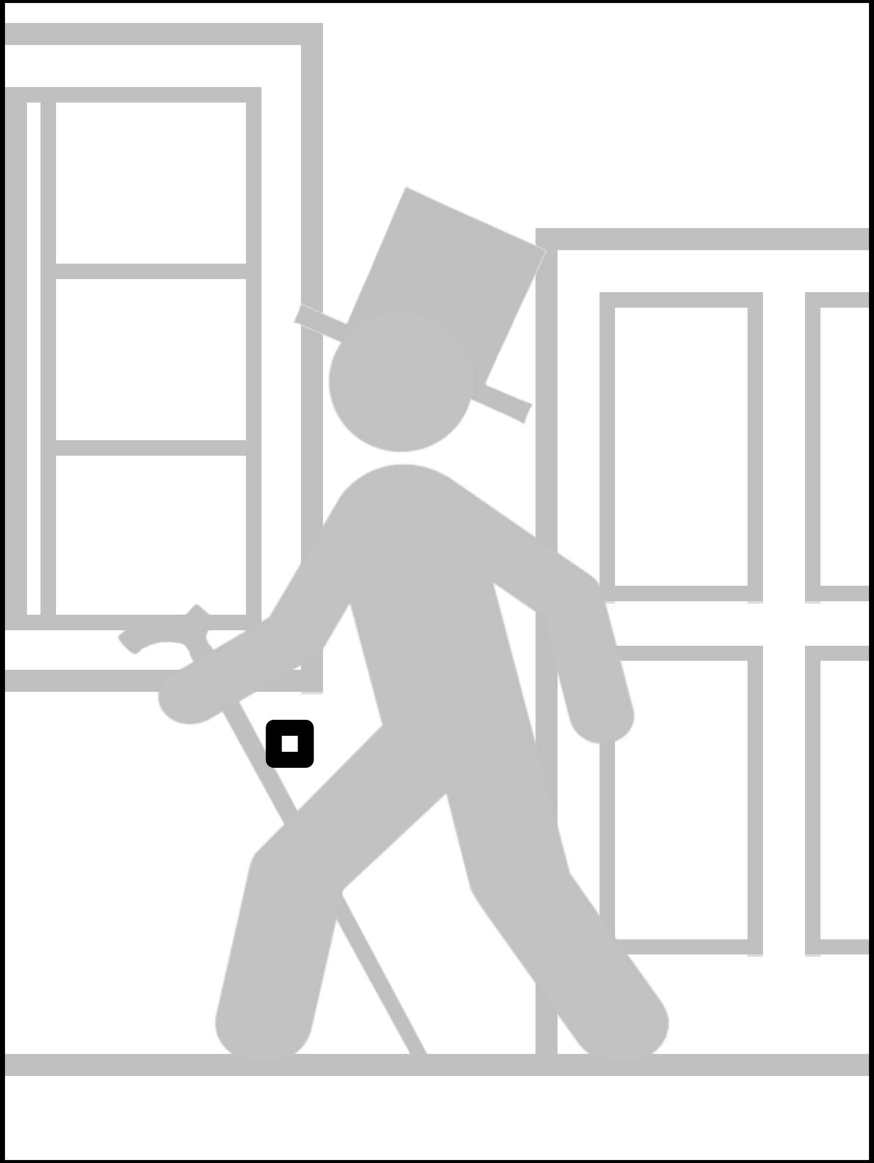


What’s gone wrong?

- HBAO+ detects objects not belonging to a surface
- Samples from these objects are not taken into account
- HBAO+ doesn’t know what is behind the character

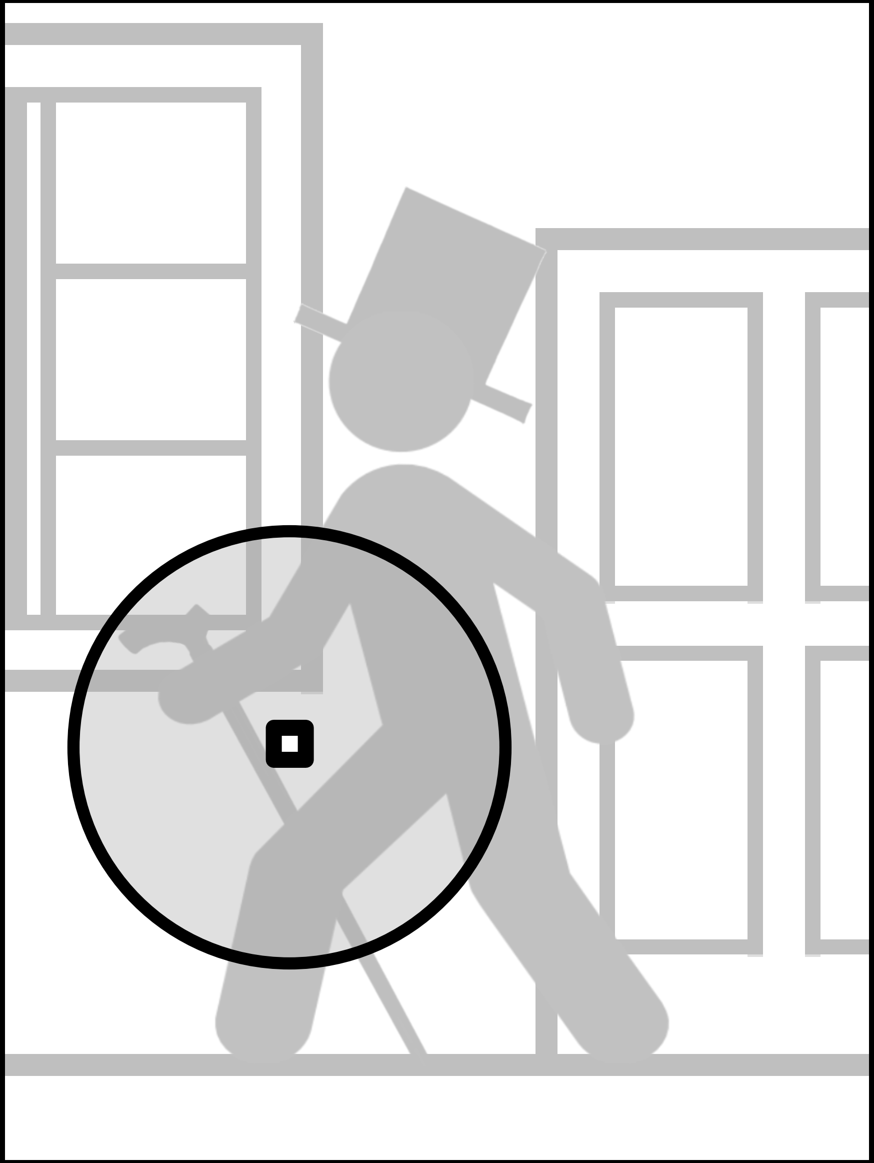
What’s gone wrong?

- HBAO+ detects objects not belonging to a surface
- Samples from these objects are not taken into account
- HBAO+ doesn’t know what is behind the character



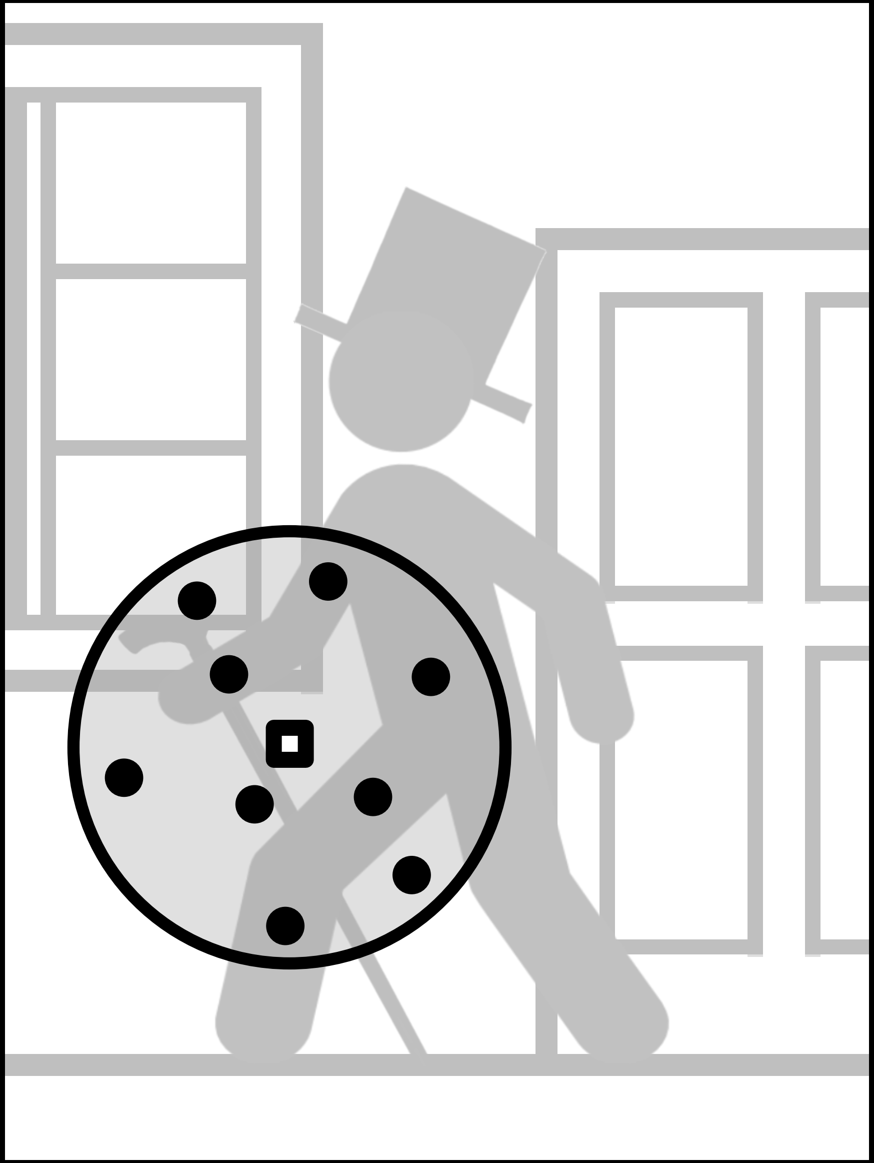
What’s gone wrong?

- HBAO+ detects objects not belonging to a surface
- Samples from these objects are not taken into account
- HBAO+ doesn’t know what is behind the character



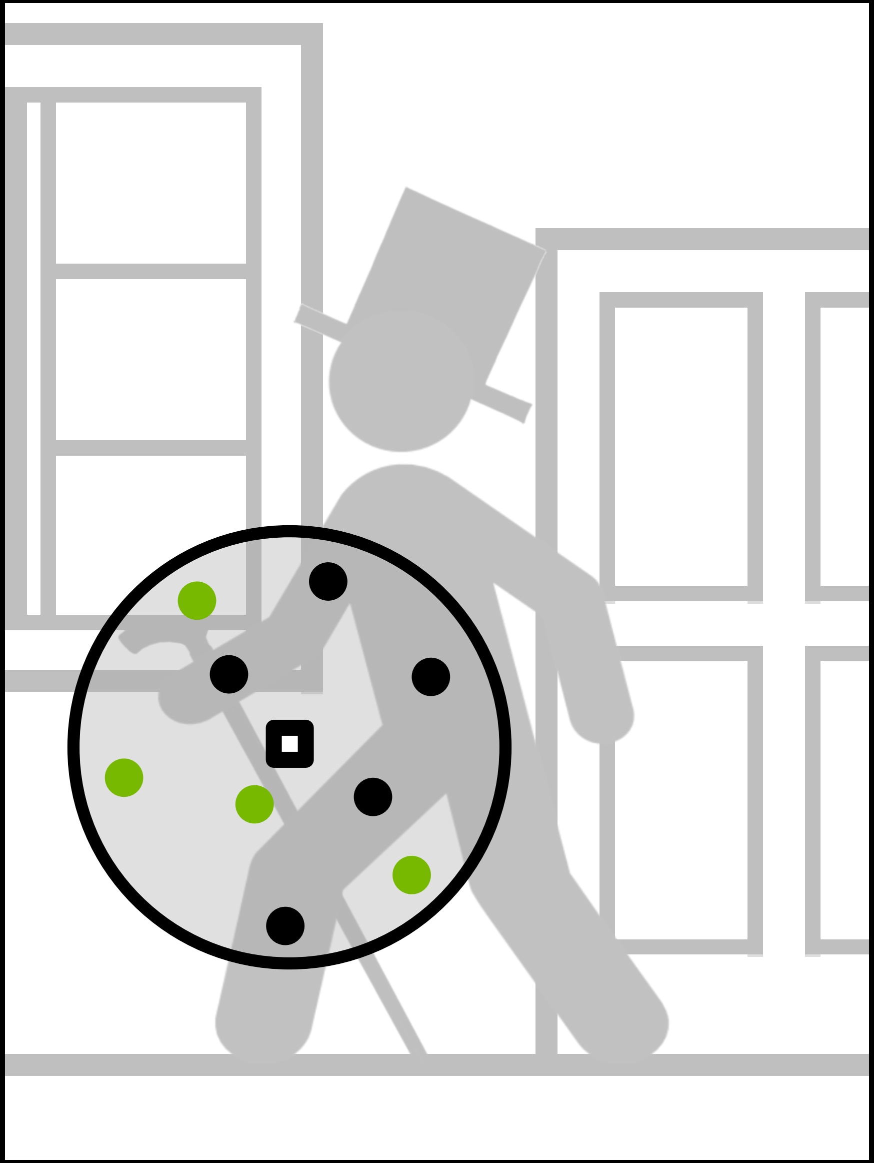
What’s gone wrong?

- HBAO+ detects objects not belonging to a surface
- Samples from these objects are not taken into account
- HBAO+ doesn’t know what is behind the character



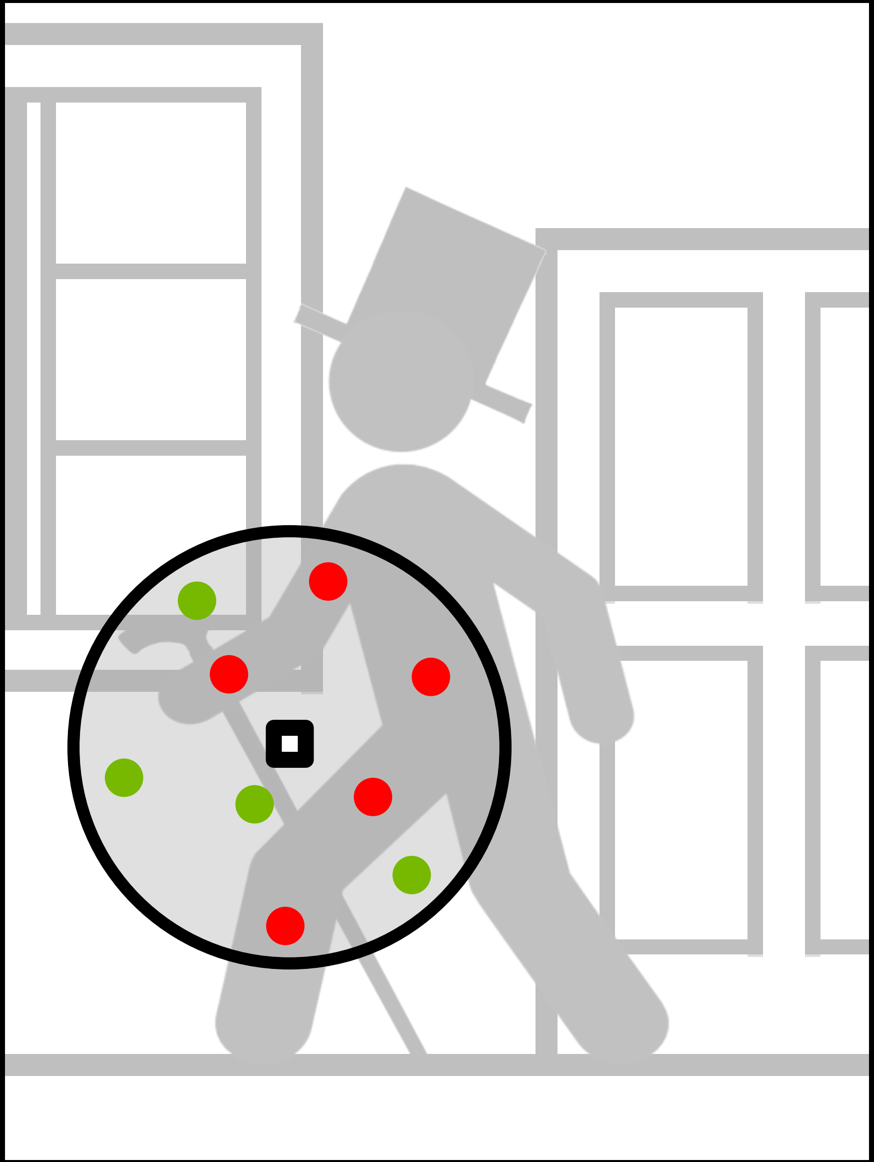
What’s gone wrong?

- HBAO+ detects objects not belonging to a surface
- Samples from these objects are not taken into account
- HBAO+ doesn’t know what is behind the character



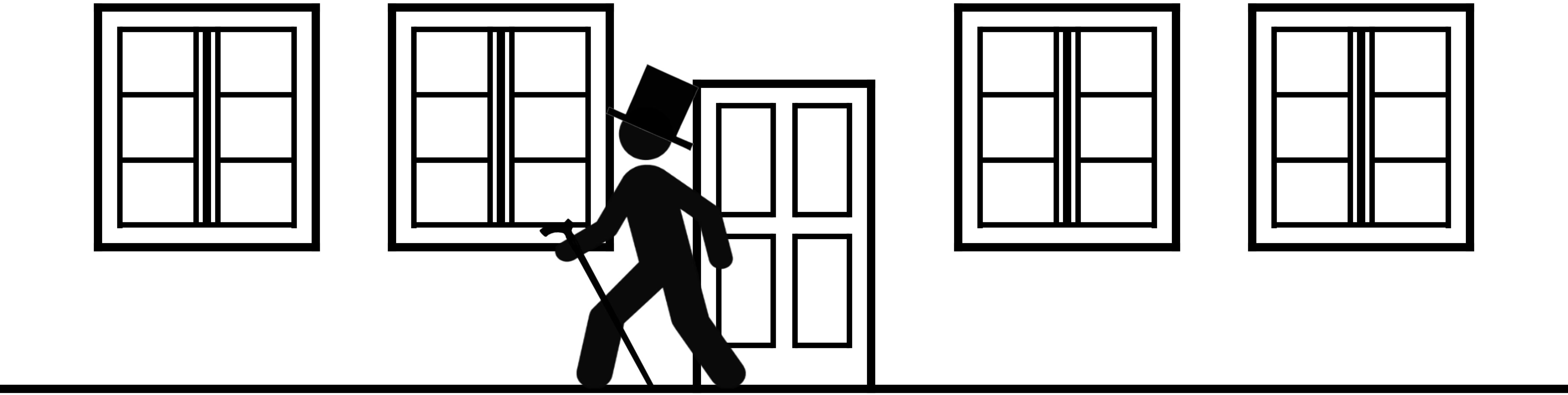
What’s gone wrong?

- HBAO+ detects objects not belonging to a surface
- Samples from these objects are not taken into account
- HBAO+ doesn’t know what is behind the character



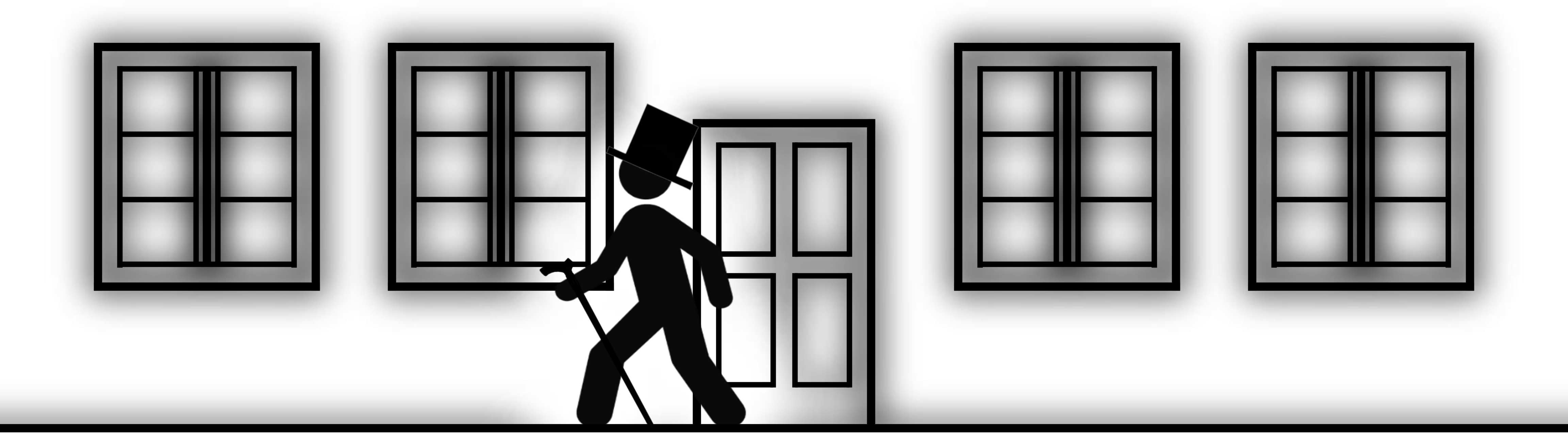
What’s gone wrong?

- This results in visible “halo” around the character



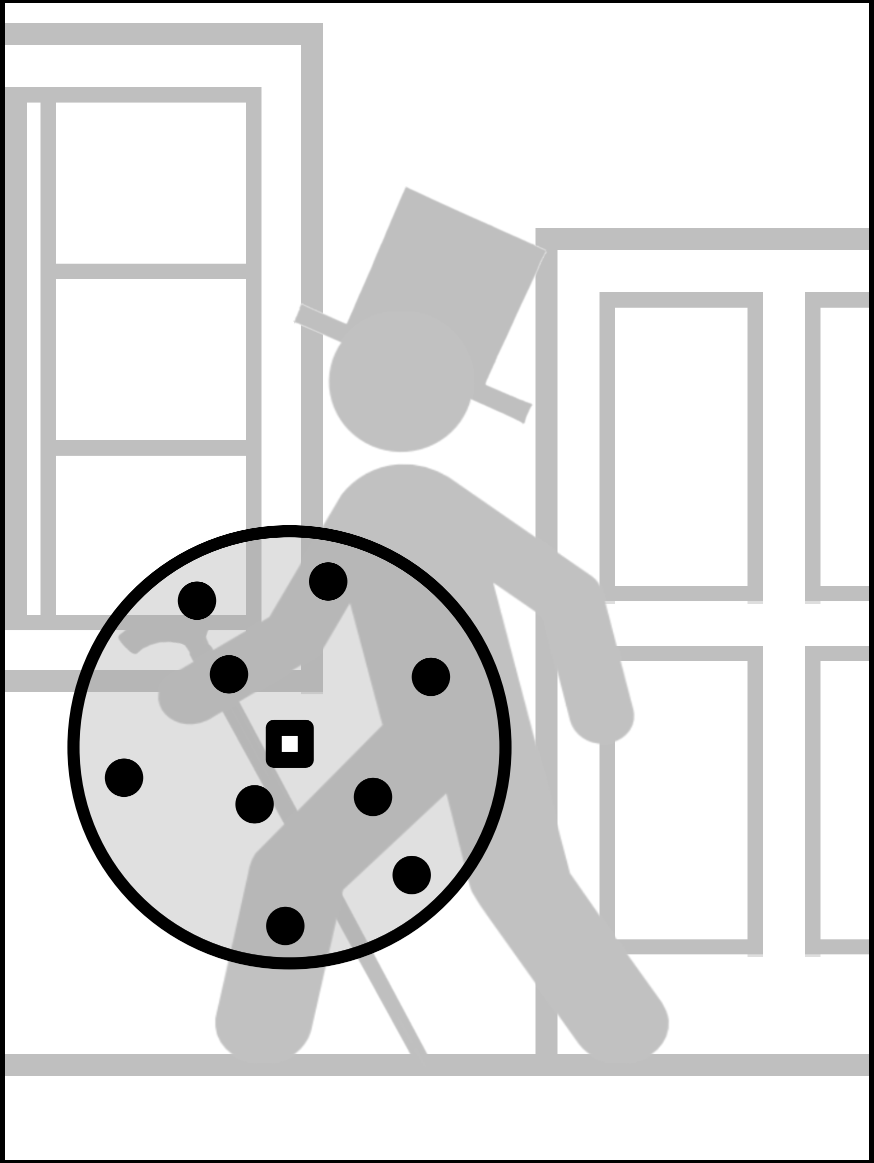
What’s gone wrong?

- This results in visible “halo” around the character



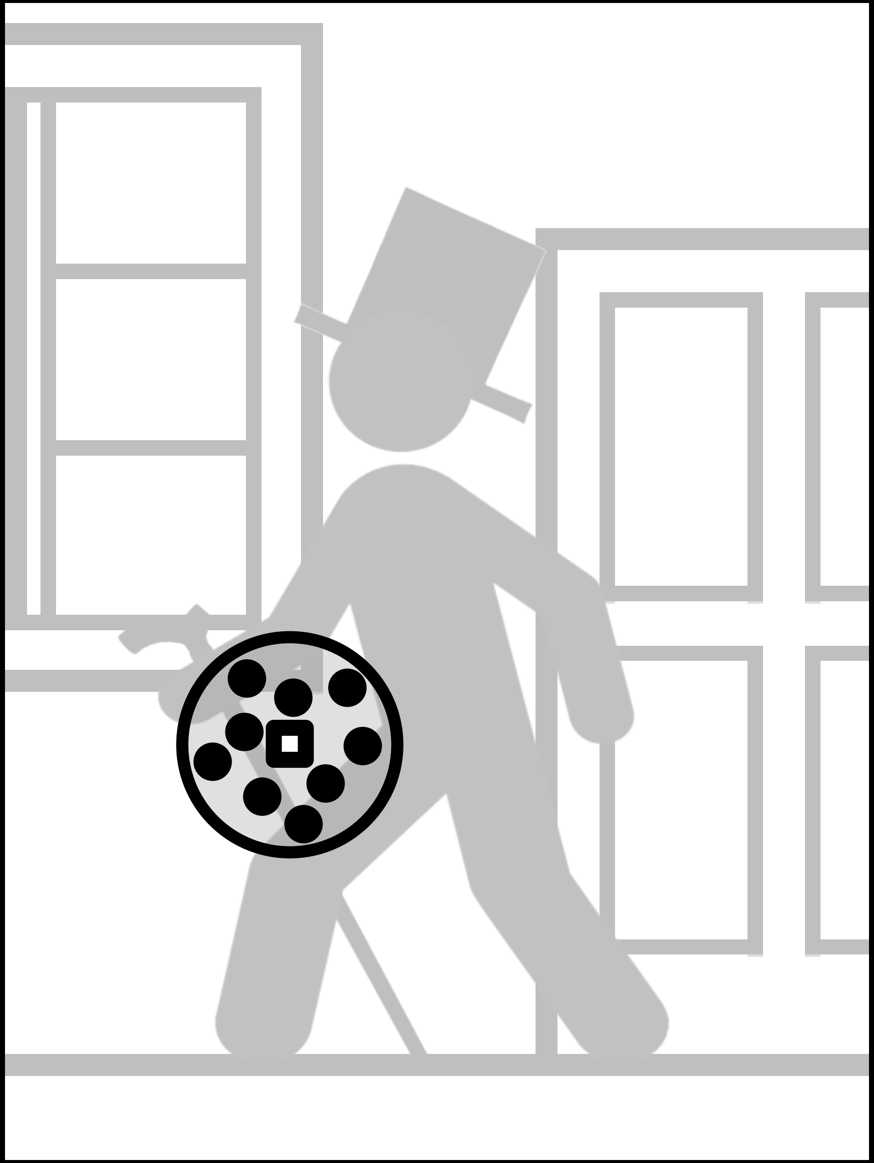
Tune HBAO+

- Use smaller HBAO+ radius
- Minimizes artifacts, not removes them!
- Makes AO look worse



Tune HBAO+

- Use smaller HBAO+ radius
- Minimizes artifacts, not removes them!
- Makes AO look worse



Need a Superman

- Want to teach HBAO+ see through objects

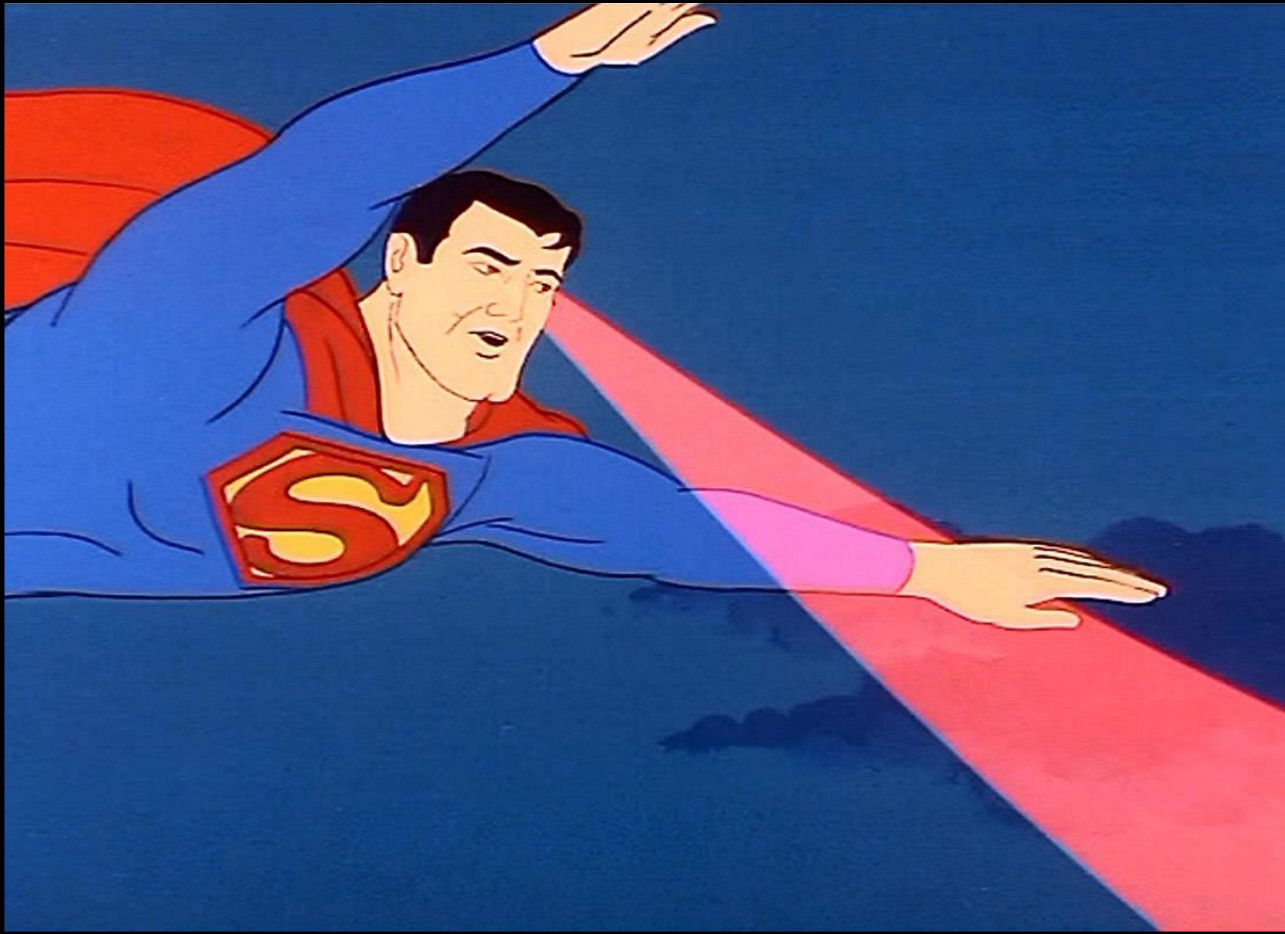


Image from SuperFriends

Need a Superman

- Want to teach HBAO+ see through objects
- Bavoil, L., and Sainz, M., Multi-Layer Dual-Resolution Screen-Space Ambient Occlusion, Siggraph 2009
- Mara, M., McGuire, M., Luebke, D., Lighting Deep G-Buffers: Single-Pass, Layered Depth Images with Minimum Separation Applied to Indirect Illumination, NVIDIA 2013

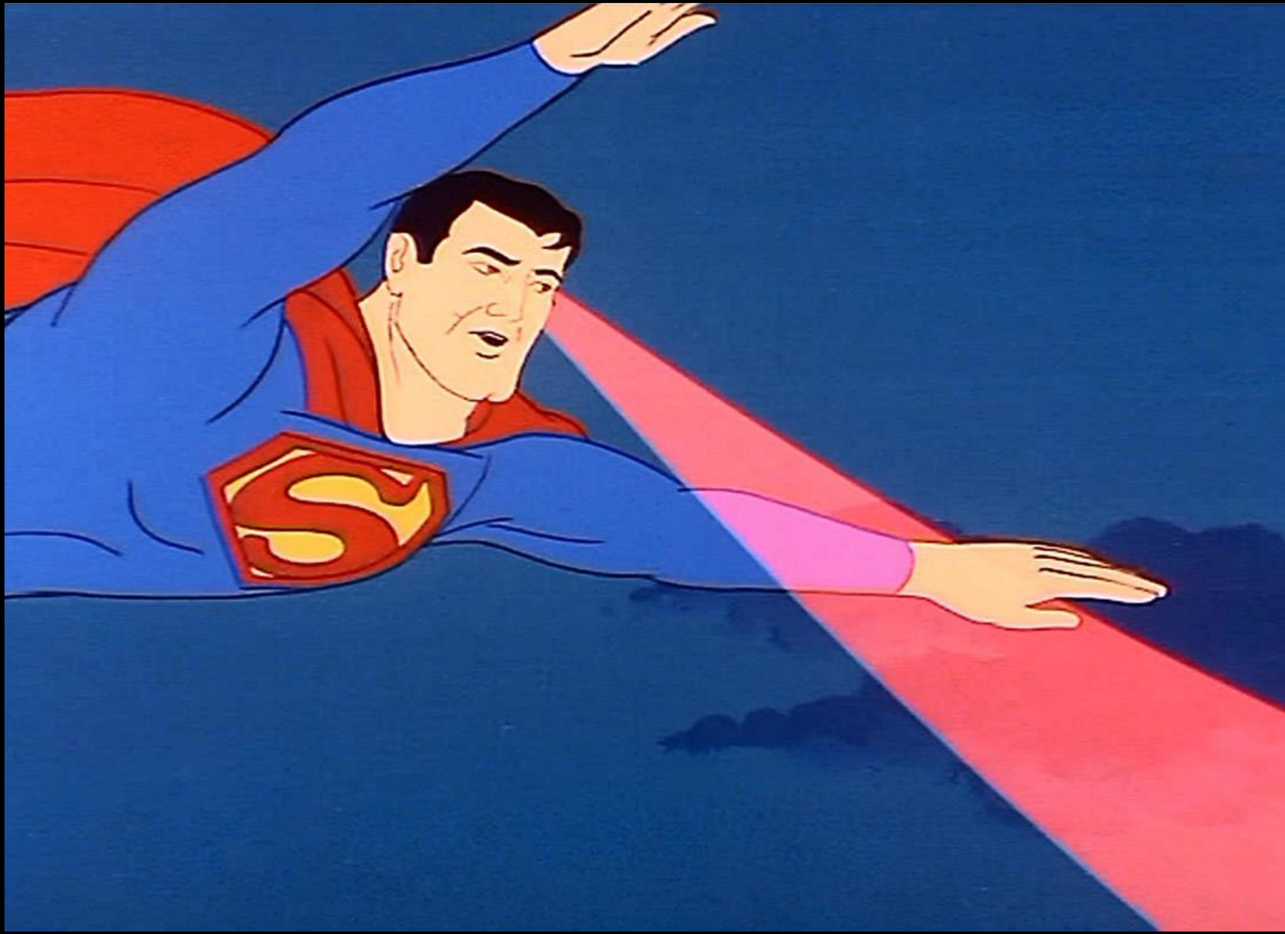


Image from SuperFriends

Depth-peeled AO

- Good solution to a problem

Depth-peeled AO

- Good solution to a problem

- AO shaders become more sophisticated
- Integrating into an engine may be troublesome

Depth-peeled AO

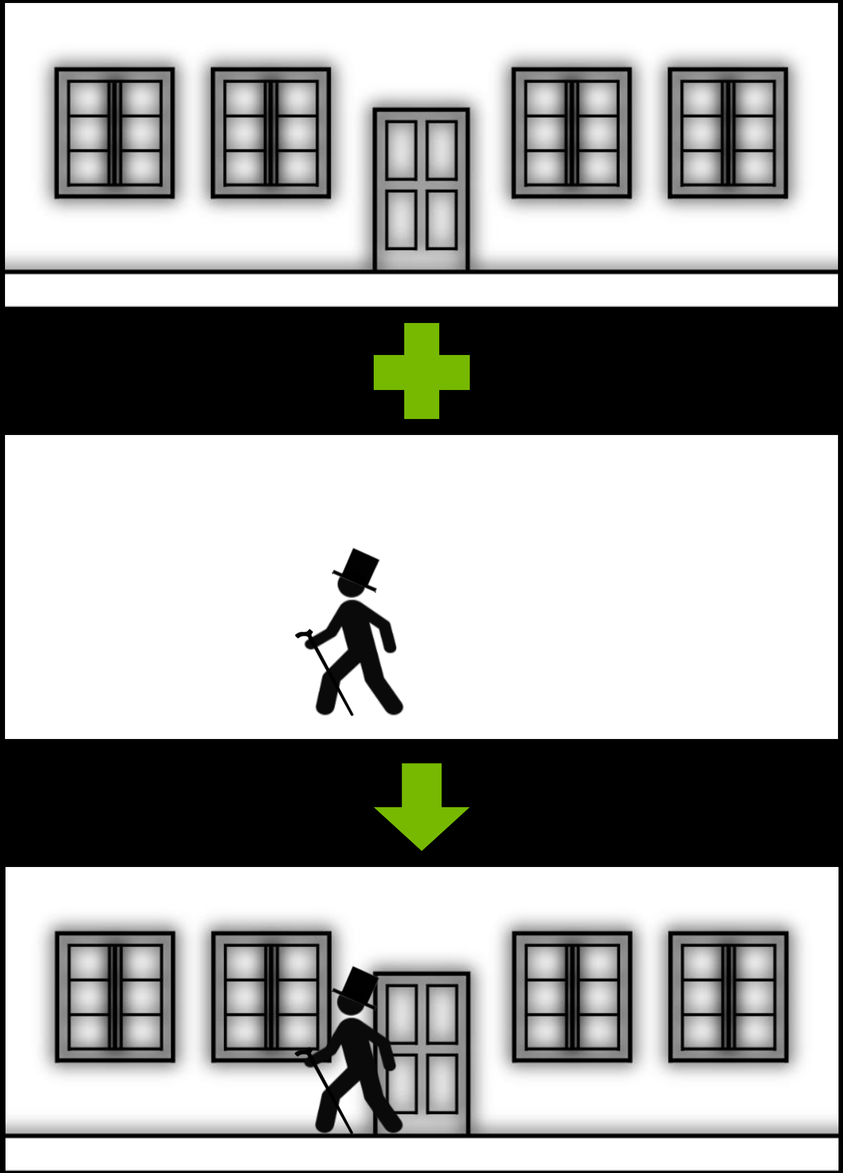
- Good solution to a problem

- AO shaders become more sophisticated
- Integrating into an engine may be troublesome

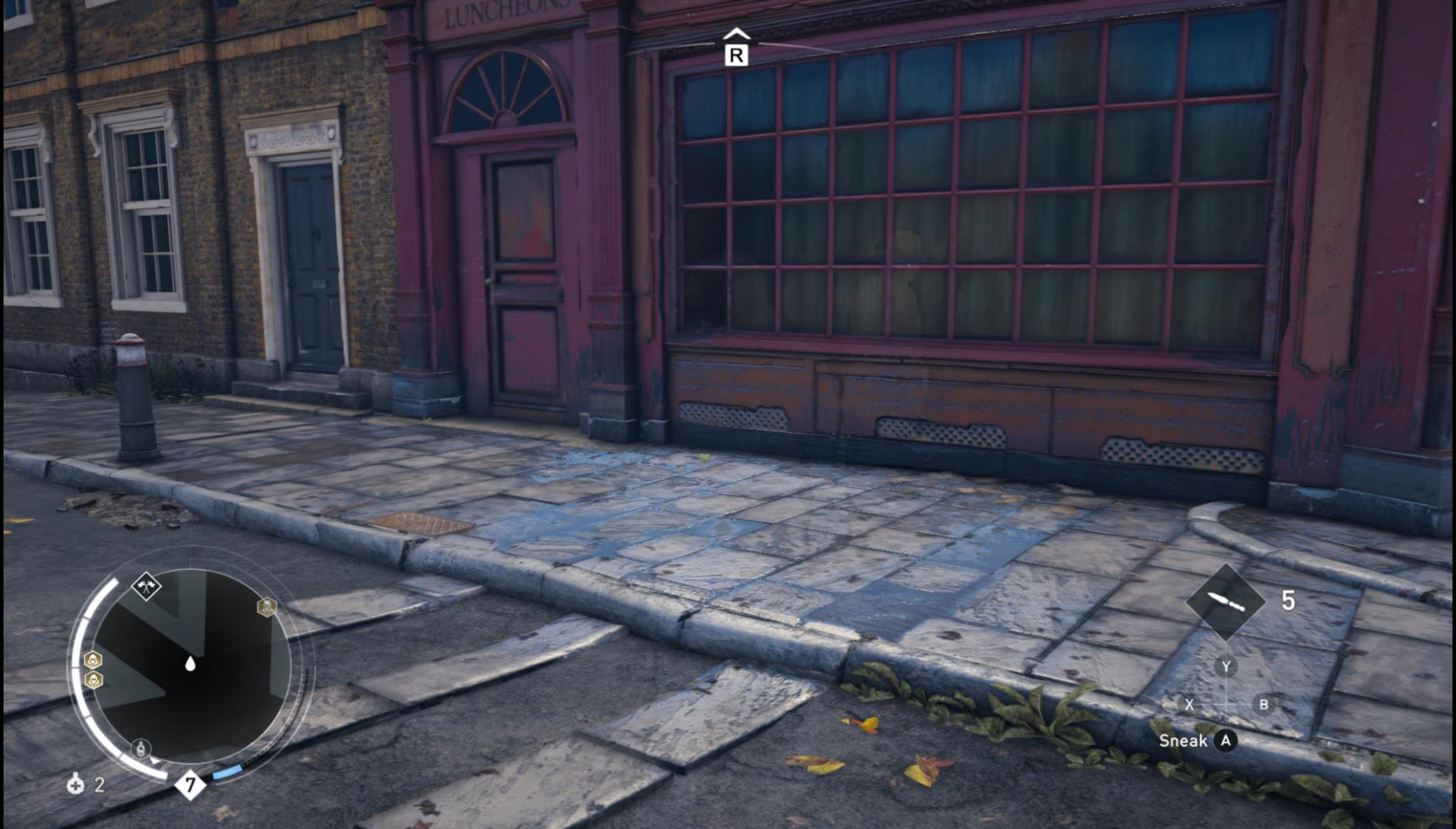
- Do we really need full-scale depth peeling?

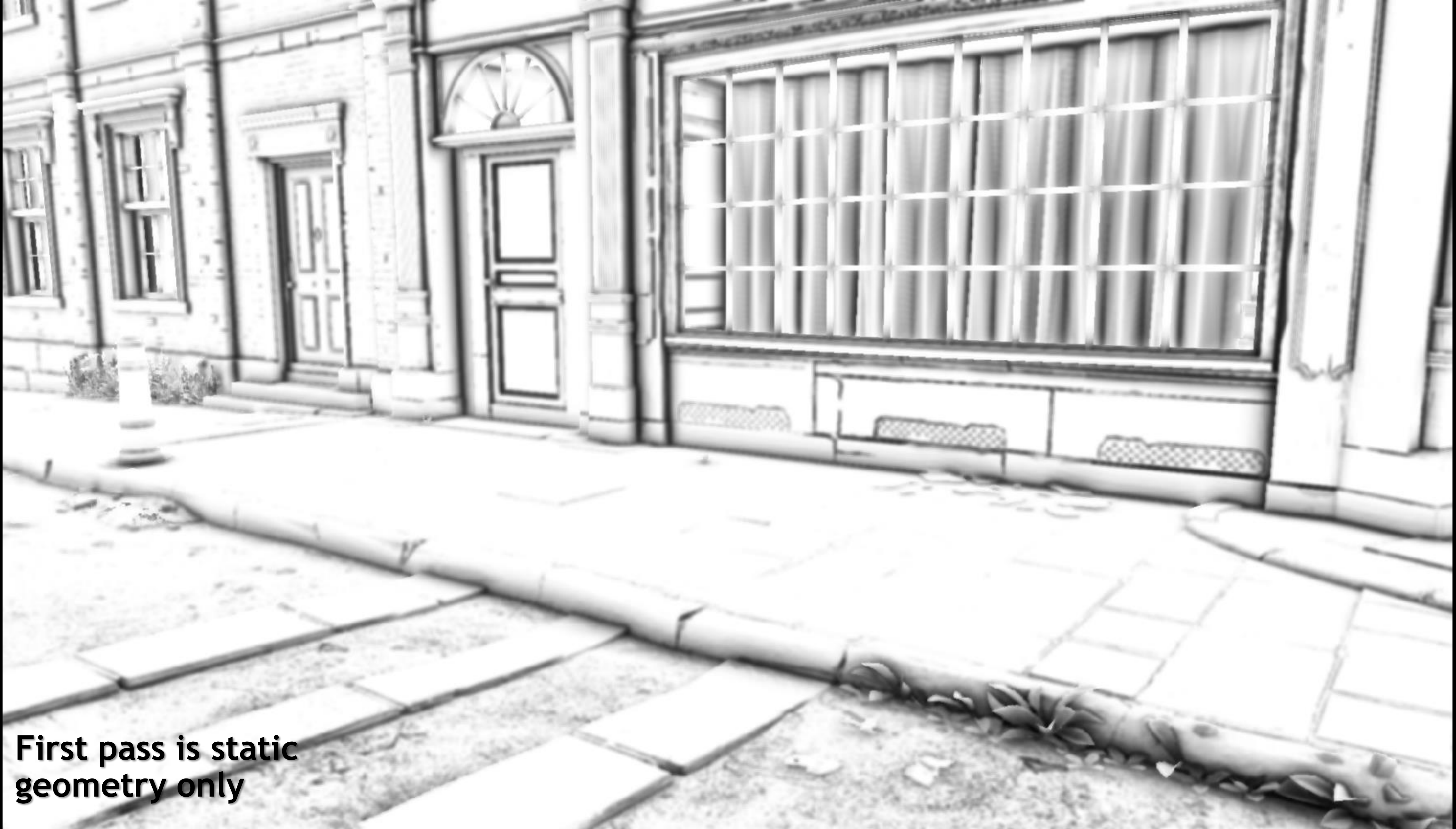
Double-layered AO

- Only moving objects create noticeable artifacts!
- Use two layers to separate statics from dynamics
- AO shaders stay the same

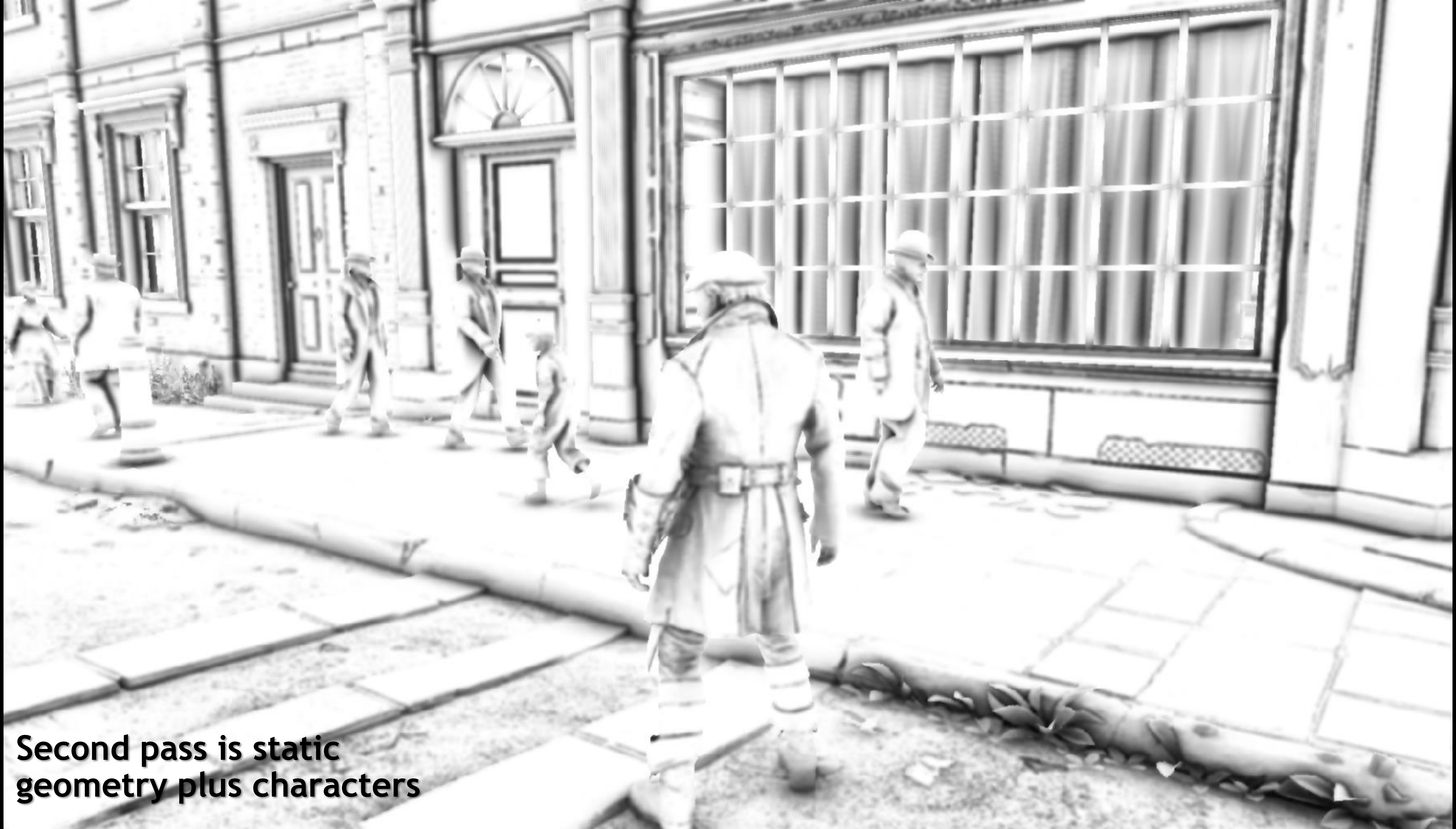


LUNCHEONS
R
5
Y
X
B
Sneak A
2
7

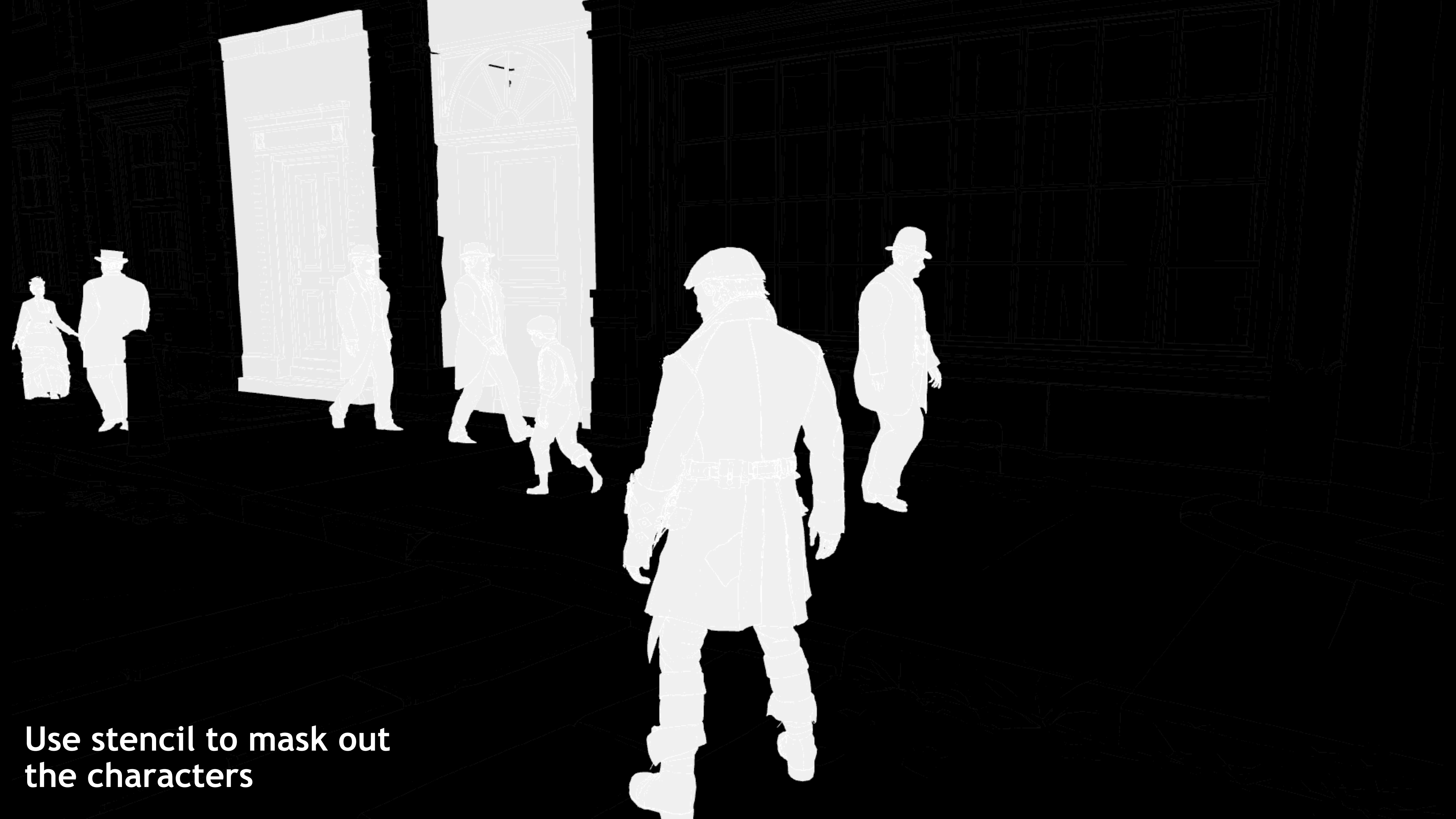

LUNCHEONS
R
5
Y
X
B
Sneak A
2
7



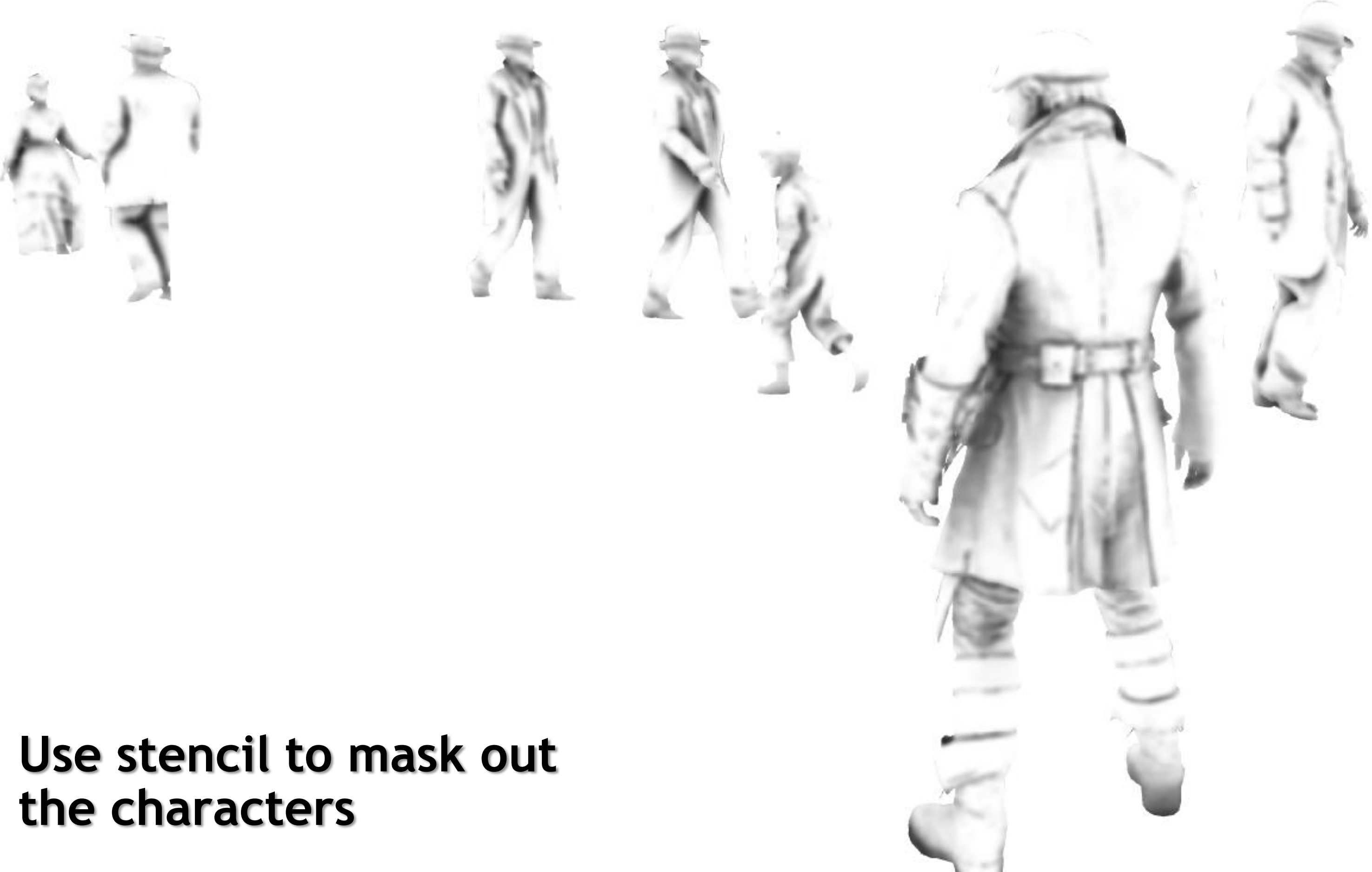
First pass is static geometry only



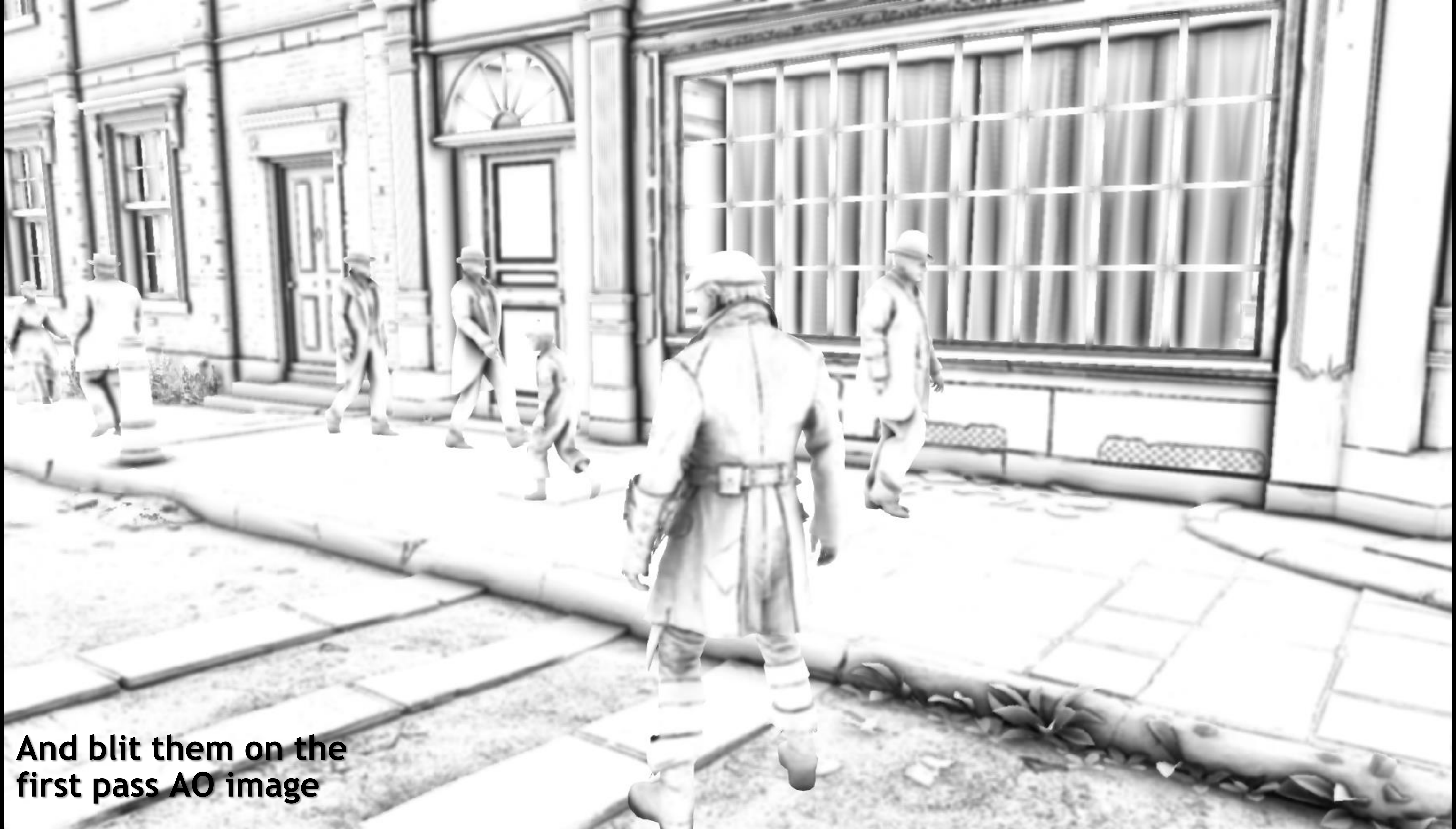
Second pass is static geometry plus characters



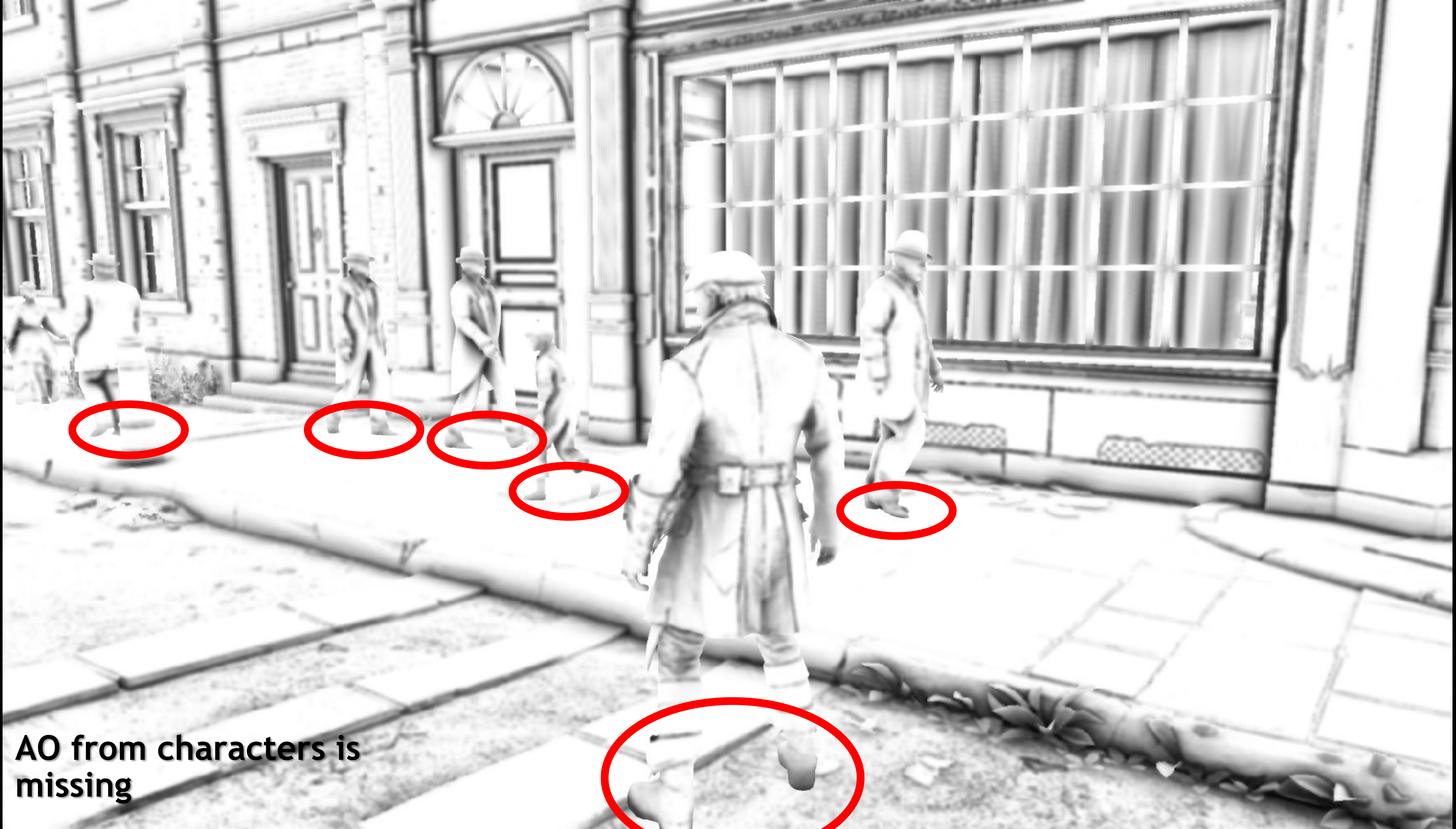
Use stencil to mask out the characters



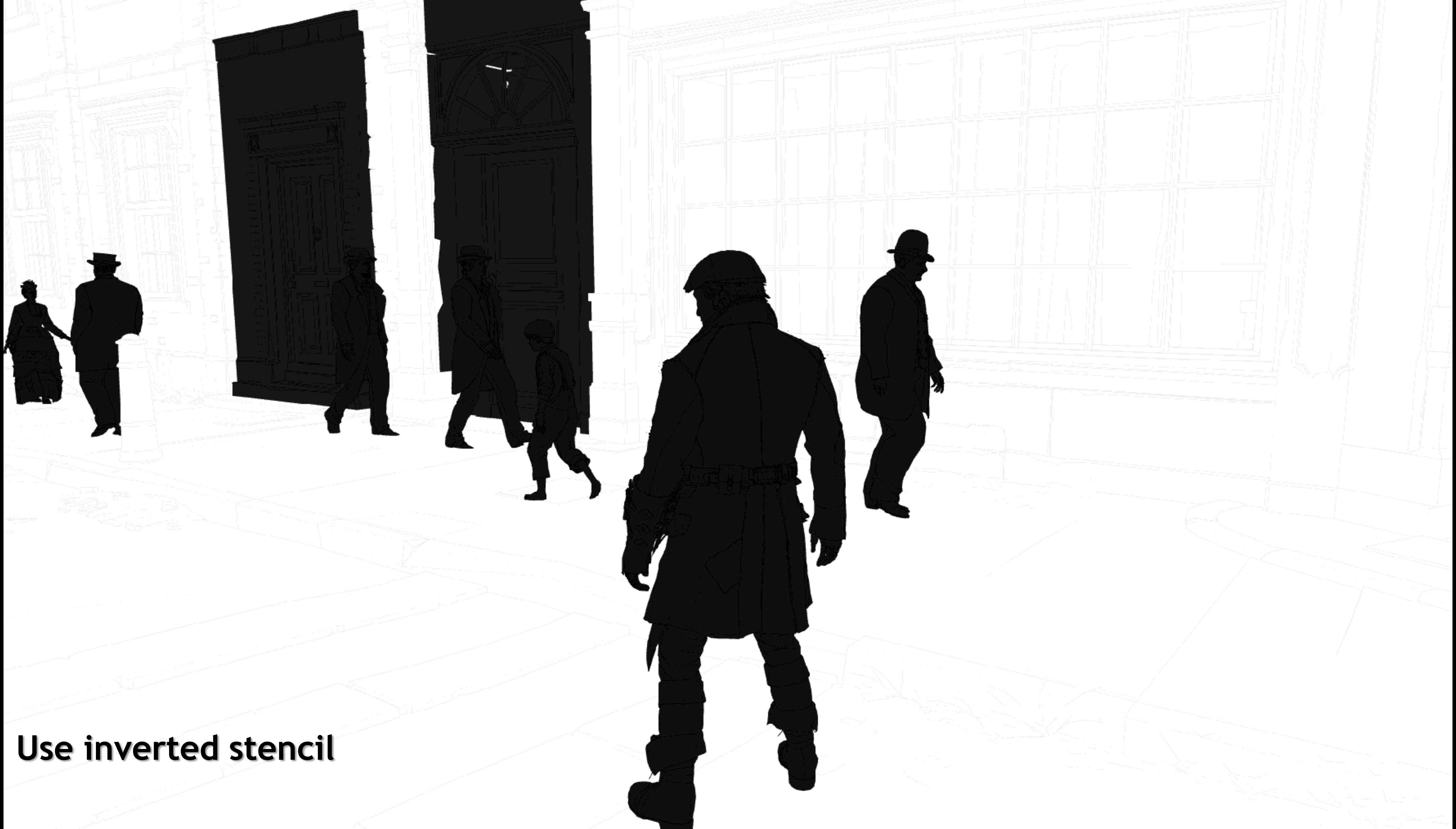
Use stencil to mask out the characters



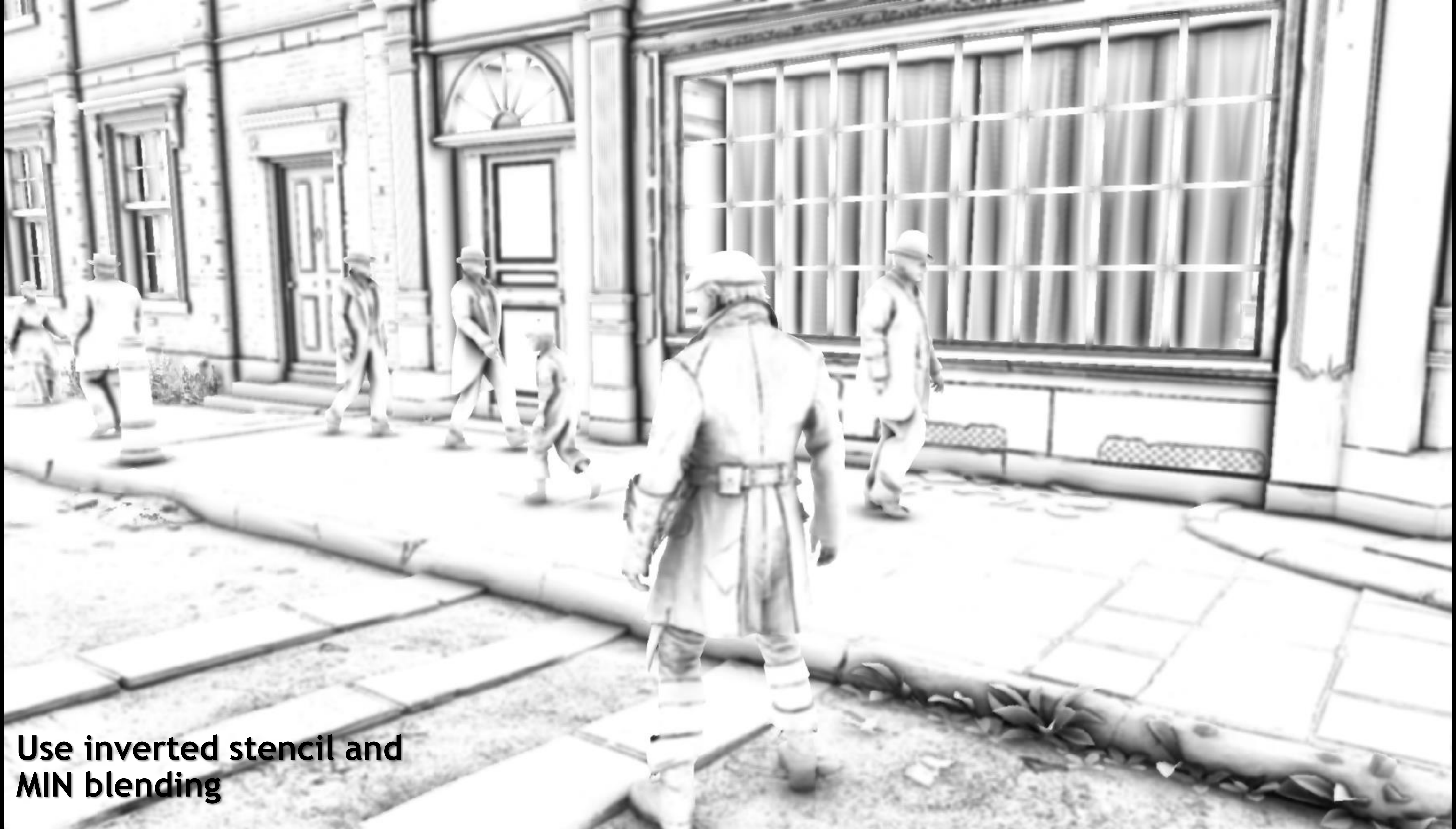
And blit them on the first pass AO image



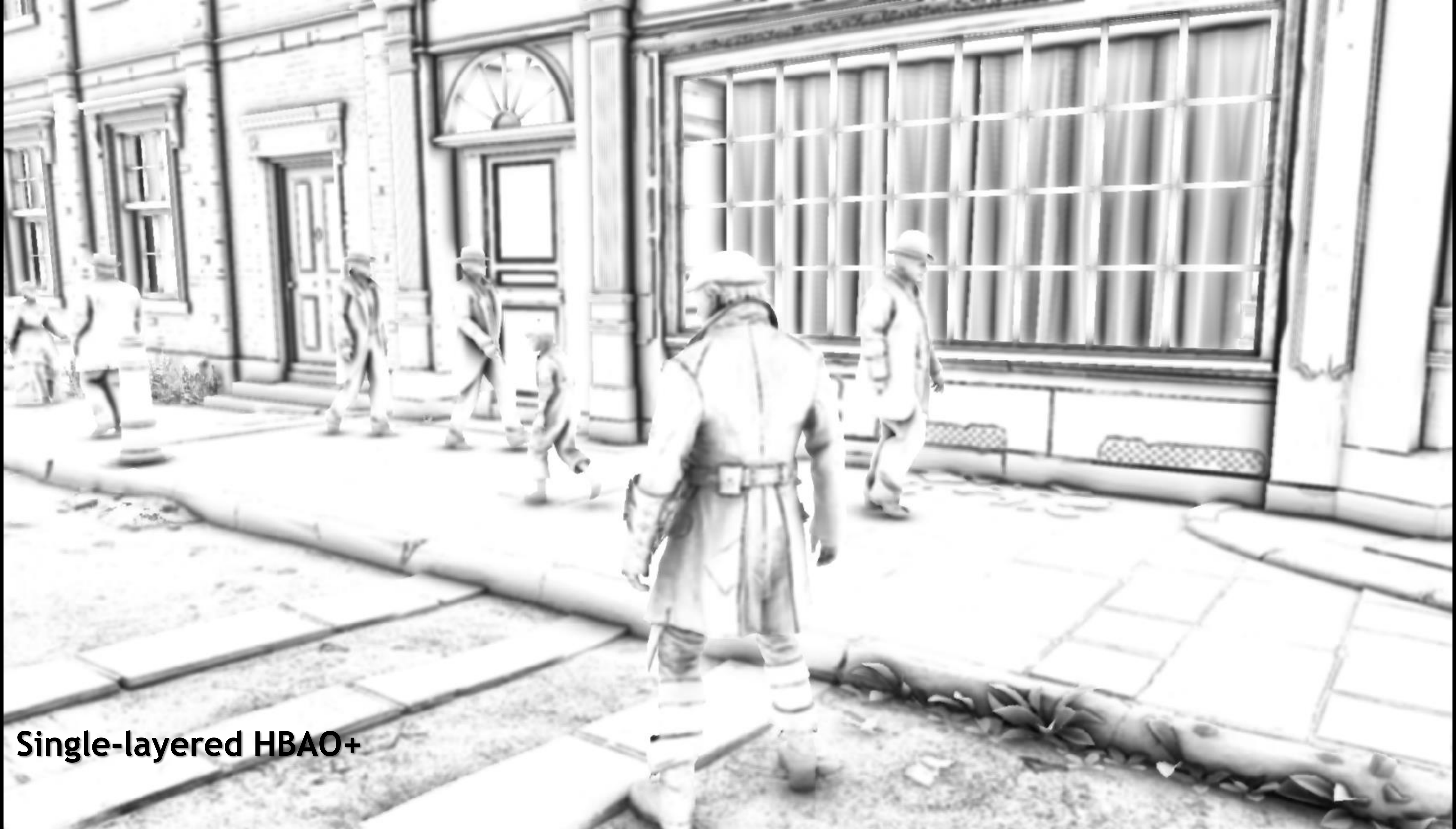
AO from characters is missing



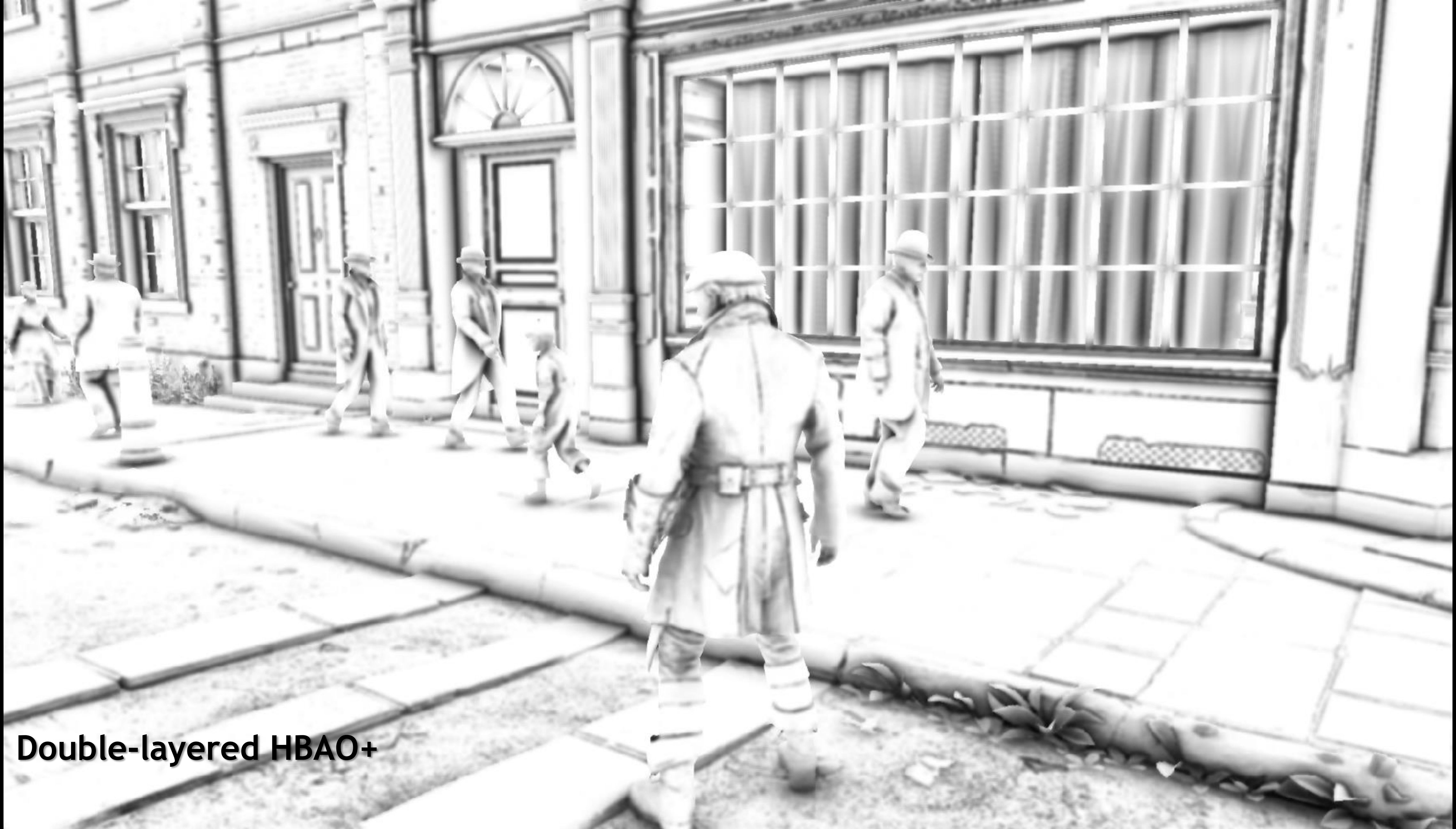
Use inverted stencil



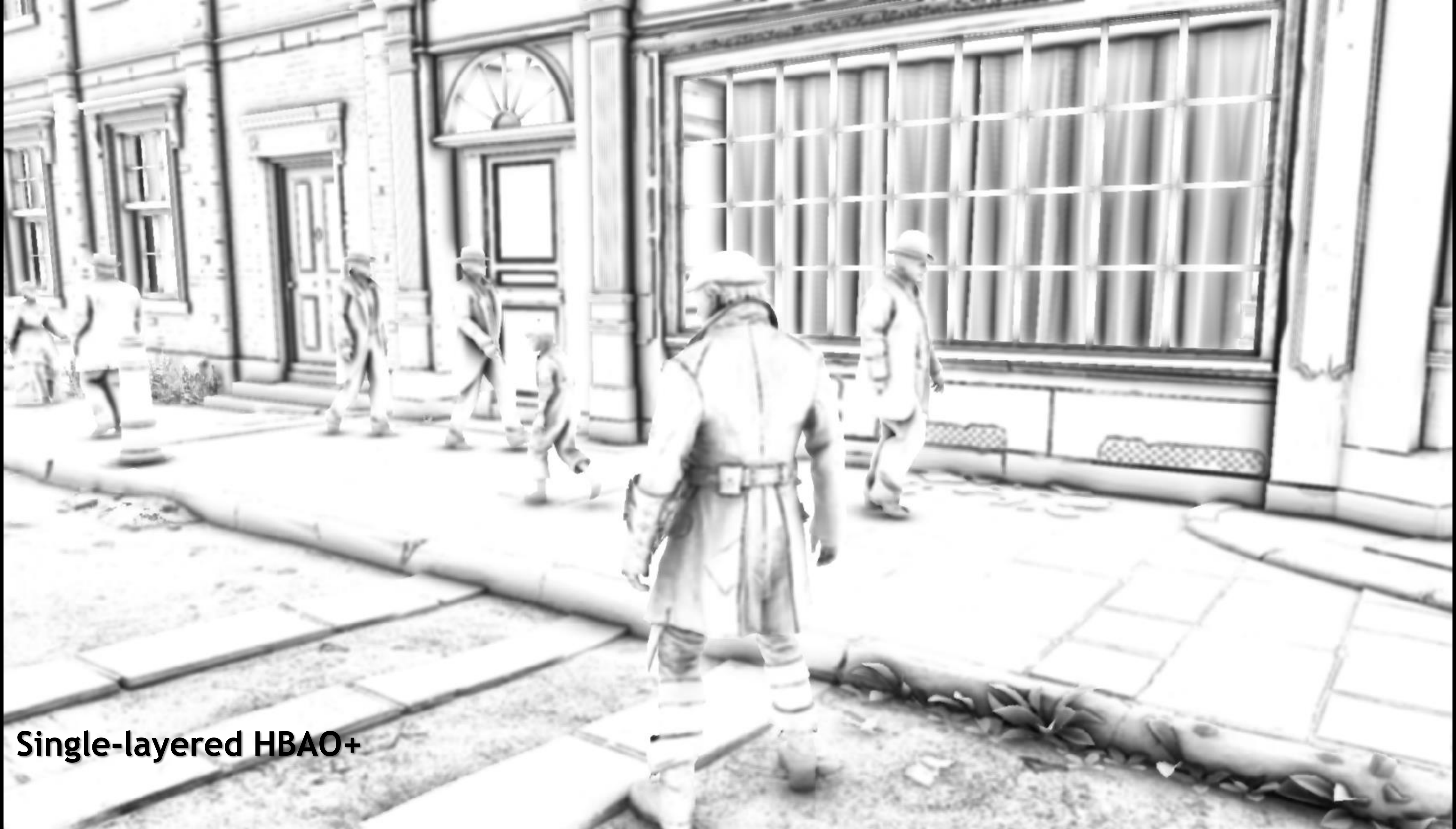
Use inverted stencil and
MIN blending



Single-layered HBAO+



Double-layered HBAO+



Single-layered HBAO+

Integration into Assassin’s Creed Syndicate

- Tune HBAO+ to look good in all cases
- Pick AO radius optimal to handle both small features and large-scale objects
- …and minimize artifacts

Integration into Assassin’s Creed Syndicate

- Tune HBAO+ to look good in all cases
- Pick AO radius optimal to handle both small features and large-scale objects
- …and minimize artifacts

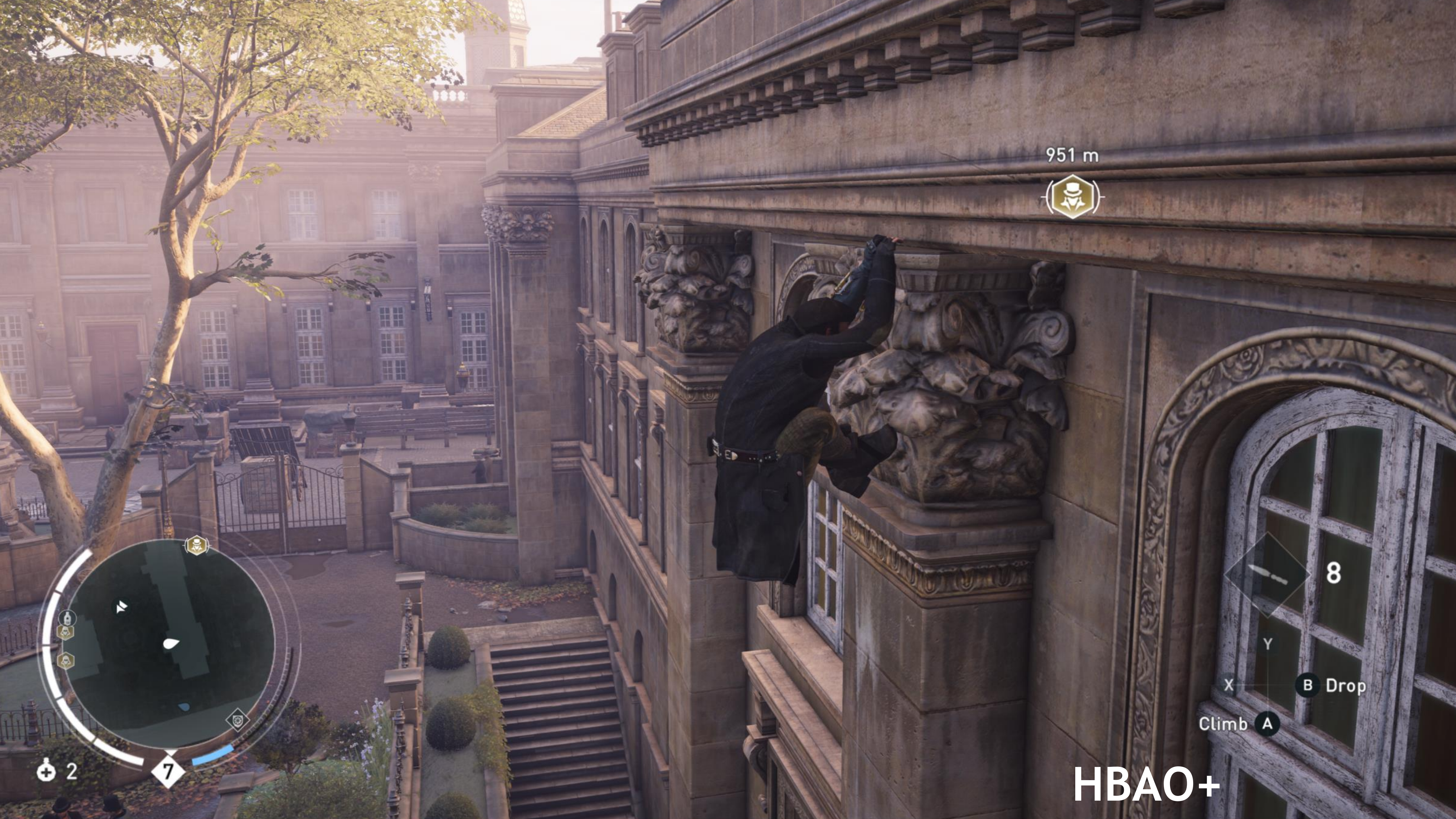
- But hey, we’re running two passes now!

HBAO+ Ultra

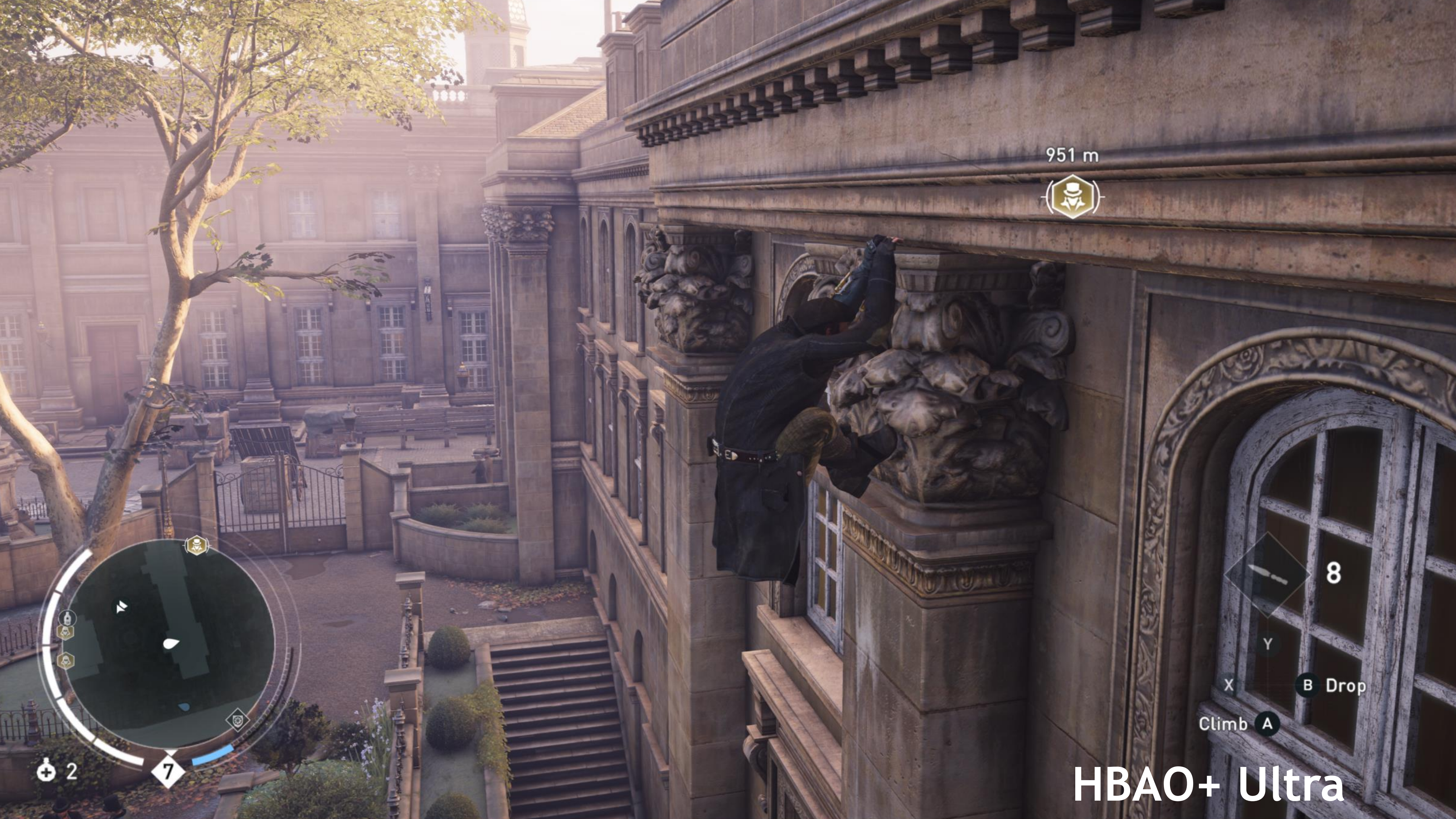
- Double-pass HBAO+ is default in Ultra preset
- Decided to name it “HBAO+ Ultra”
- Added advanced blending functionality to HBAO+

Advanced blending functionality in HBAO+

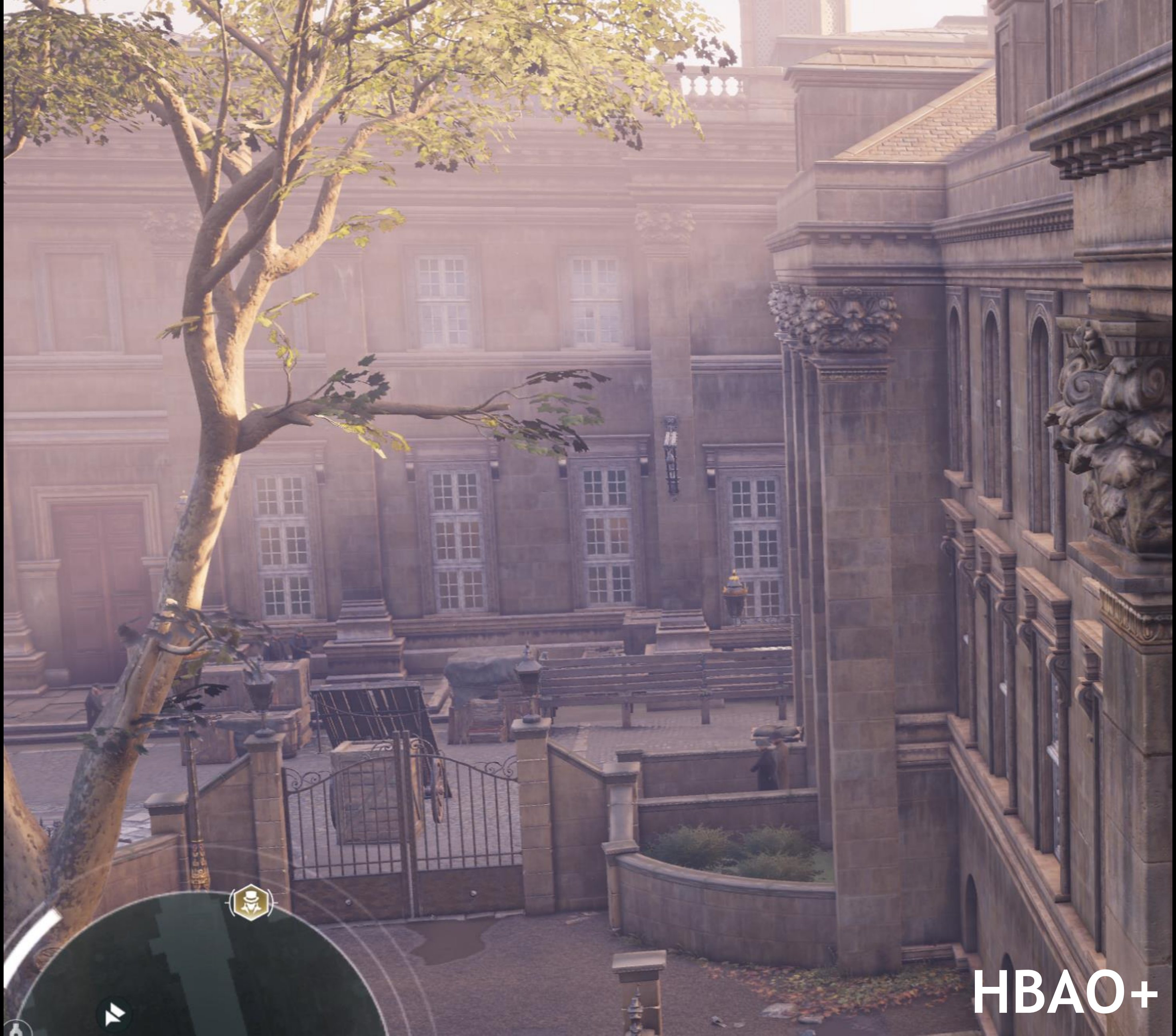
```
struct GFSDK_SSAO_TwoPassBlend_D3D11
{
    // When enabled, overrides any other compositing state
    GFSDK_SSAO_BOOL Enable;

    // Used to mask the pixels in each of the 2 passes
    ID3D11DepthStencilView* pDepthStencilView;

    // Blend & depth-stencil state for the first compositing pass
    GFSDK_SSAO_BlendPass_D3D11 FirstPass;

    // Blend & depth-stencil state for the second compositing pass
    GFSDK_SSAO_BlendPass_D3D11 SecondPass;

    GFSDK_SSAO_TwoPassBlend_D3D11()
        : Enable(false)
        , pDepthStencilView(NULL)
    {
    }
};
```



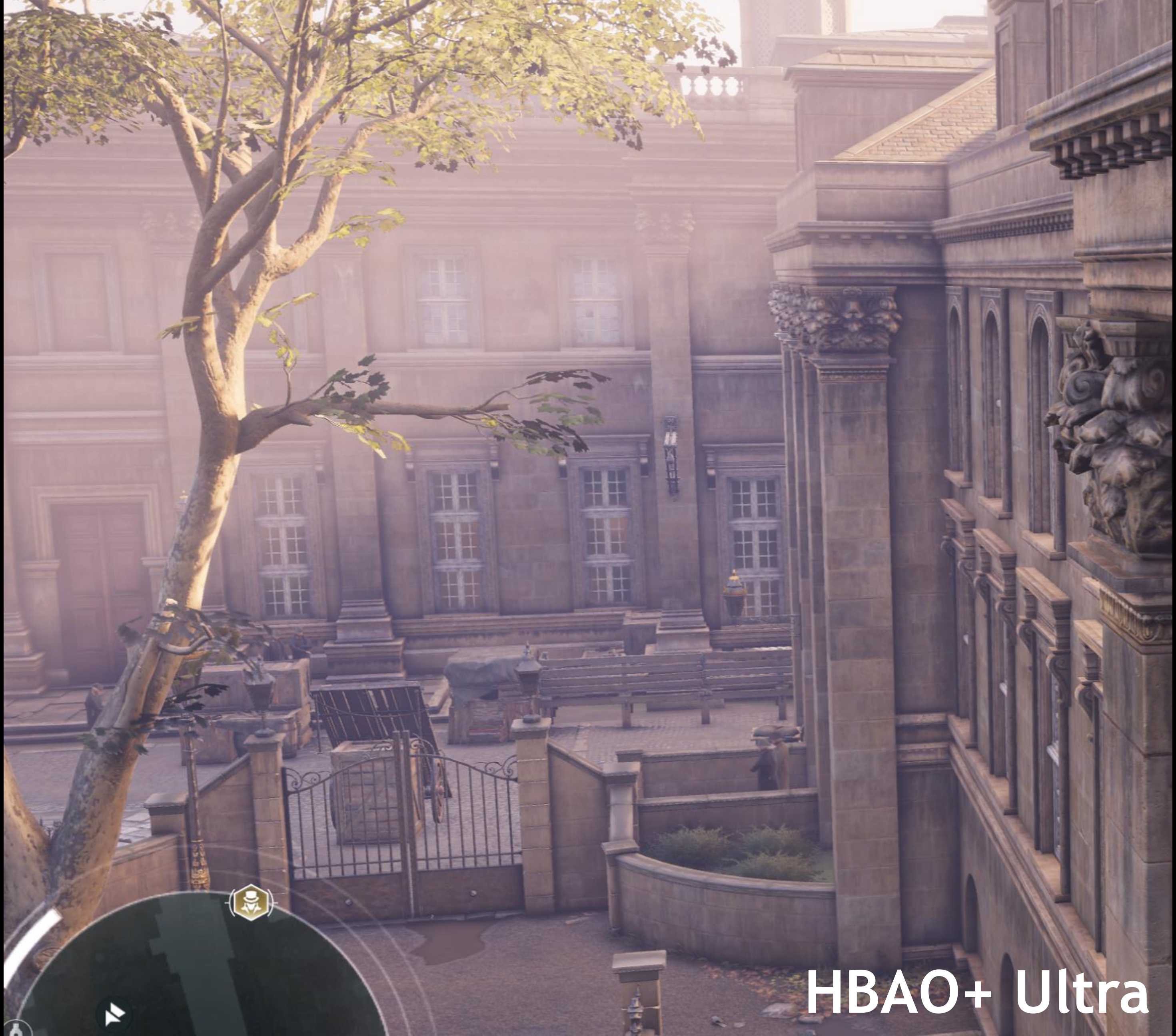
HBAO+



HBAO+ Ultra

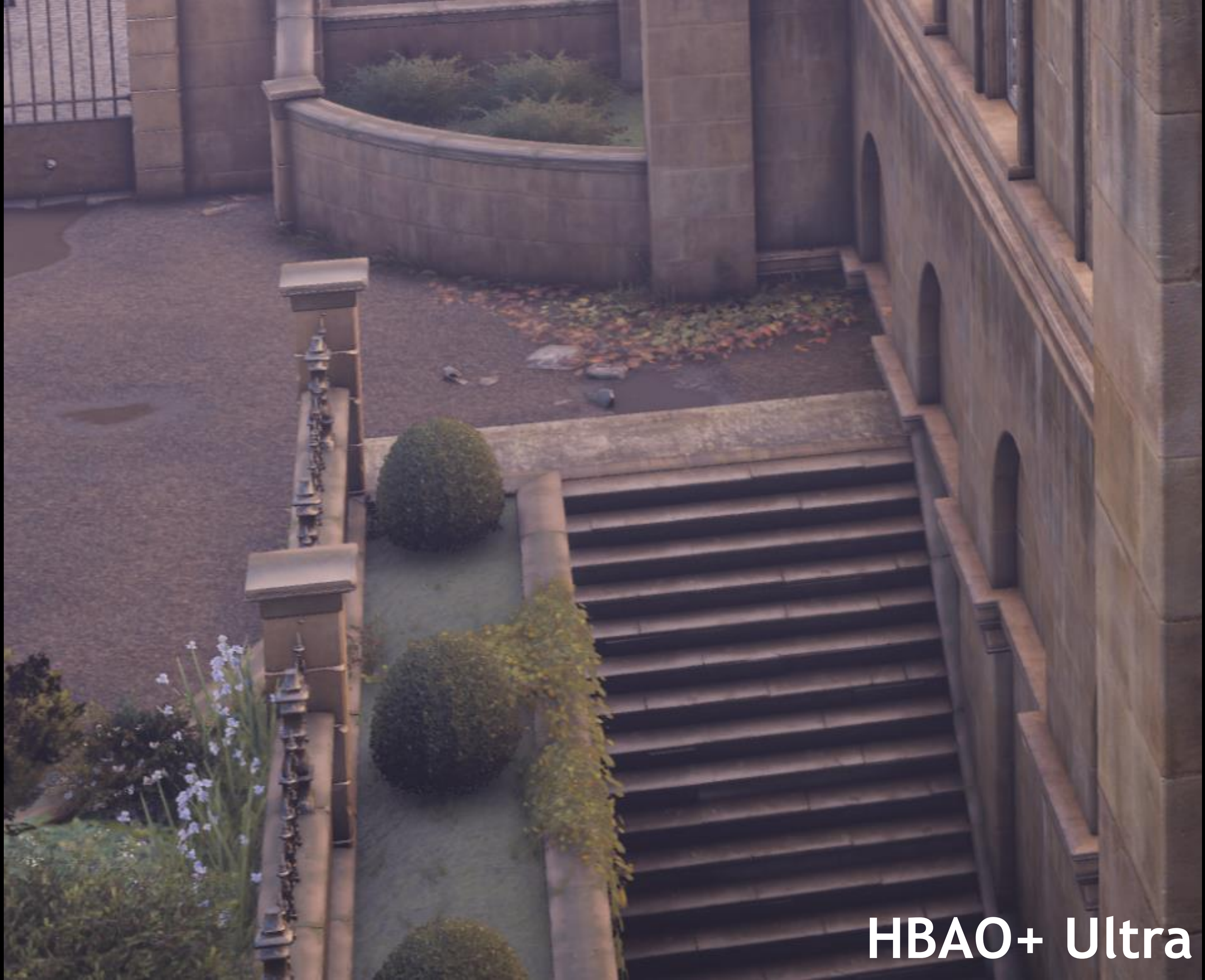


HBAO+

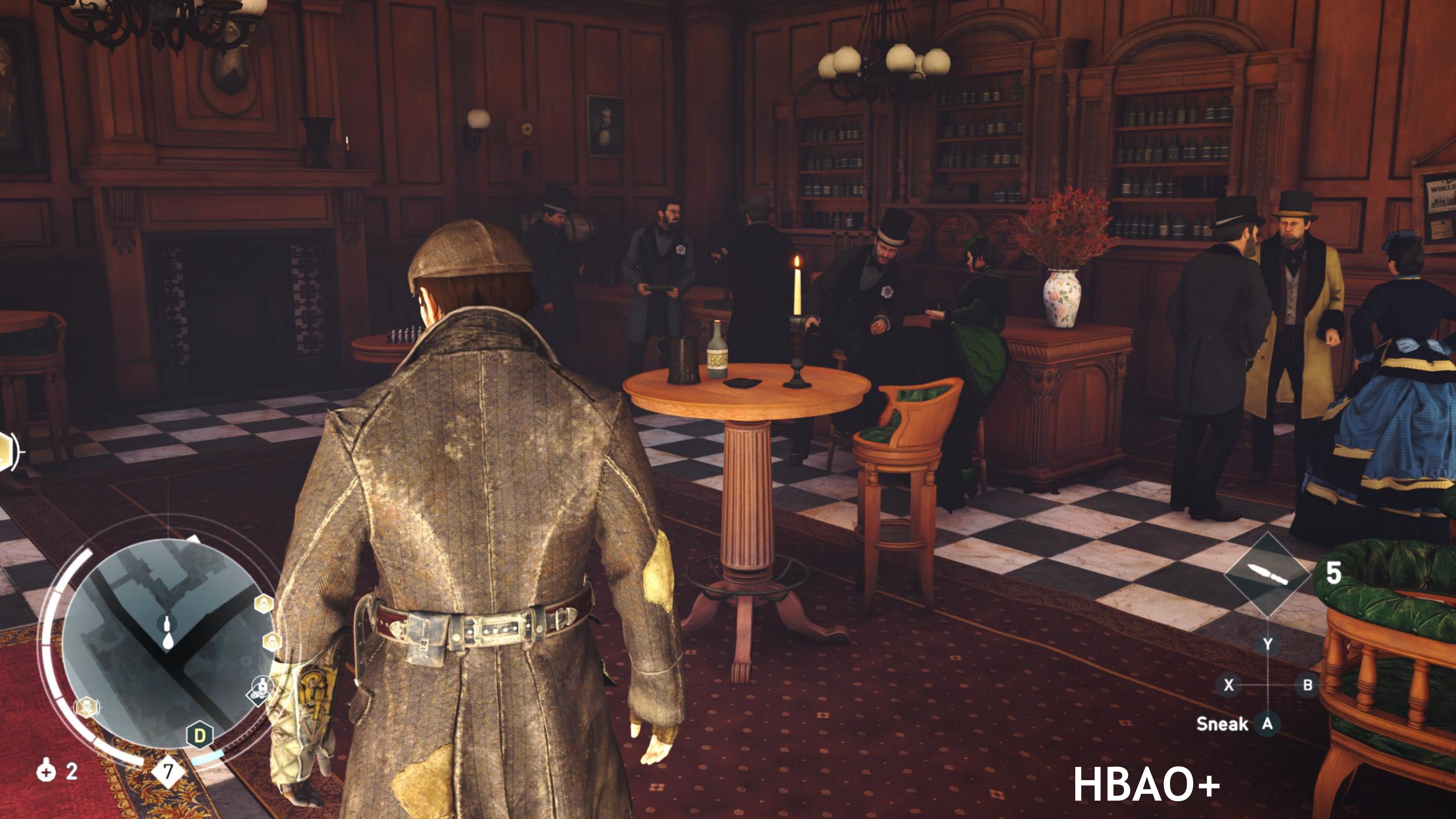


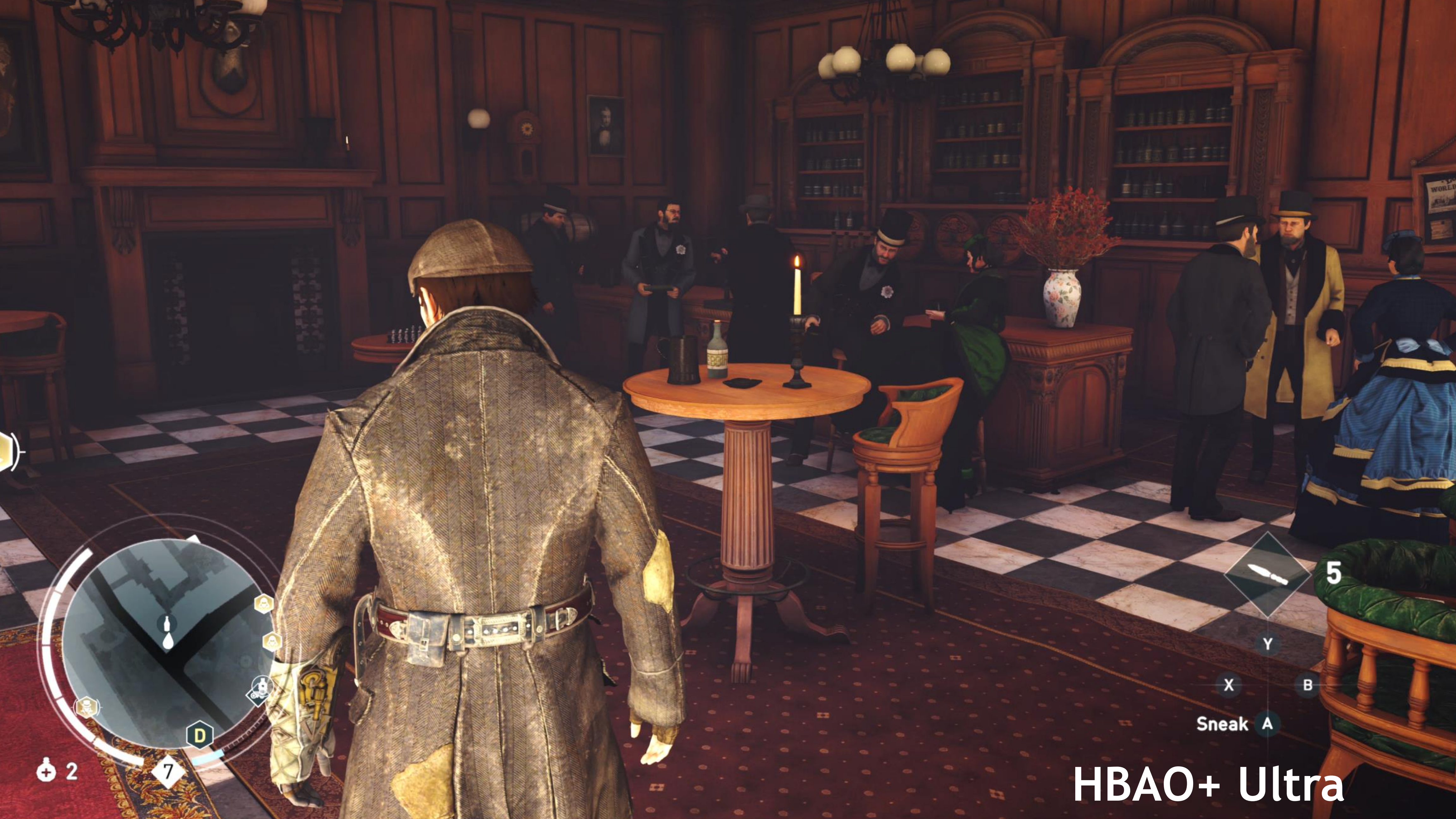
HBAO+ Ultra

HBAO+

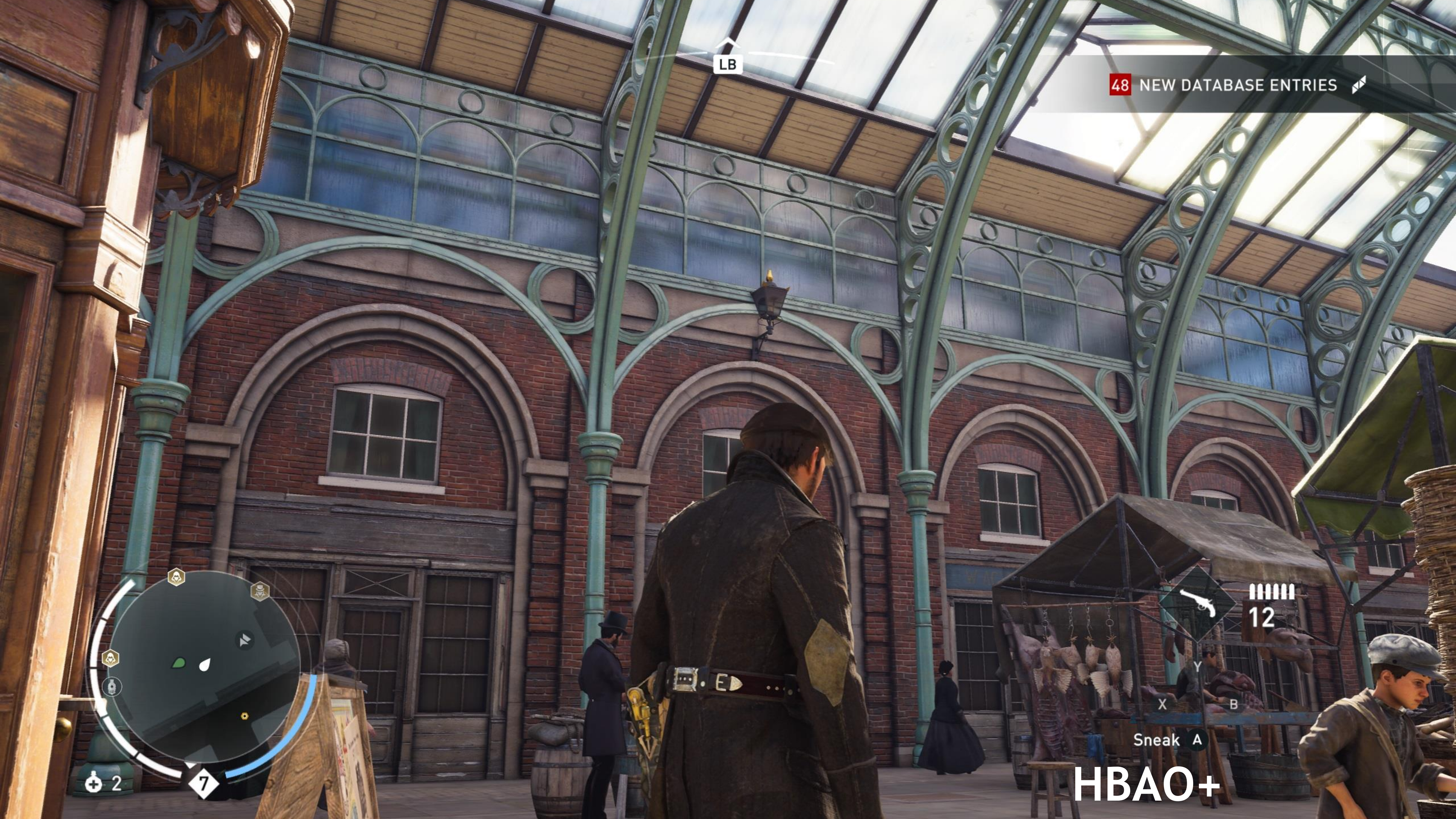


HBAO+ Ultra

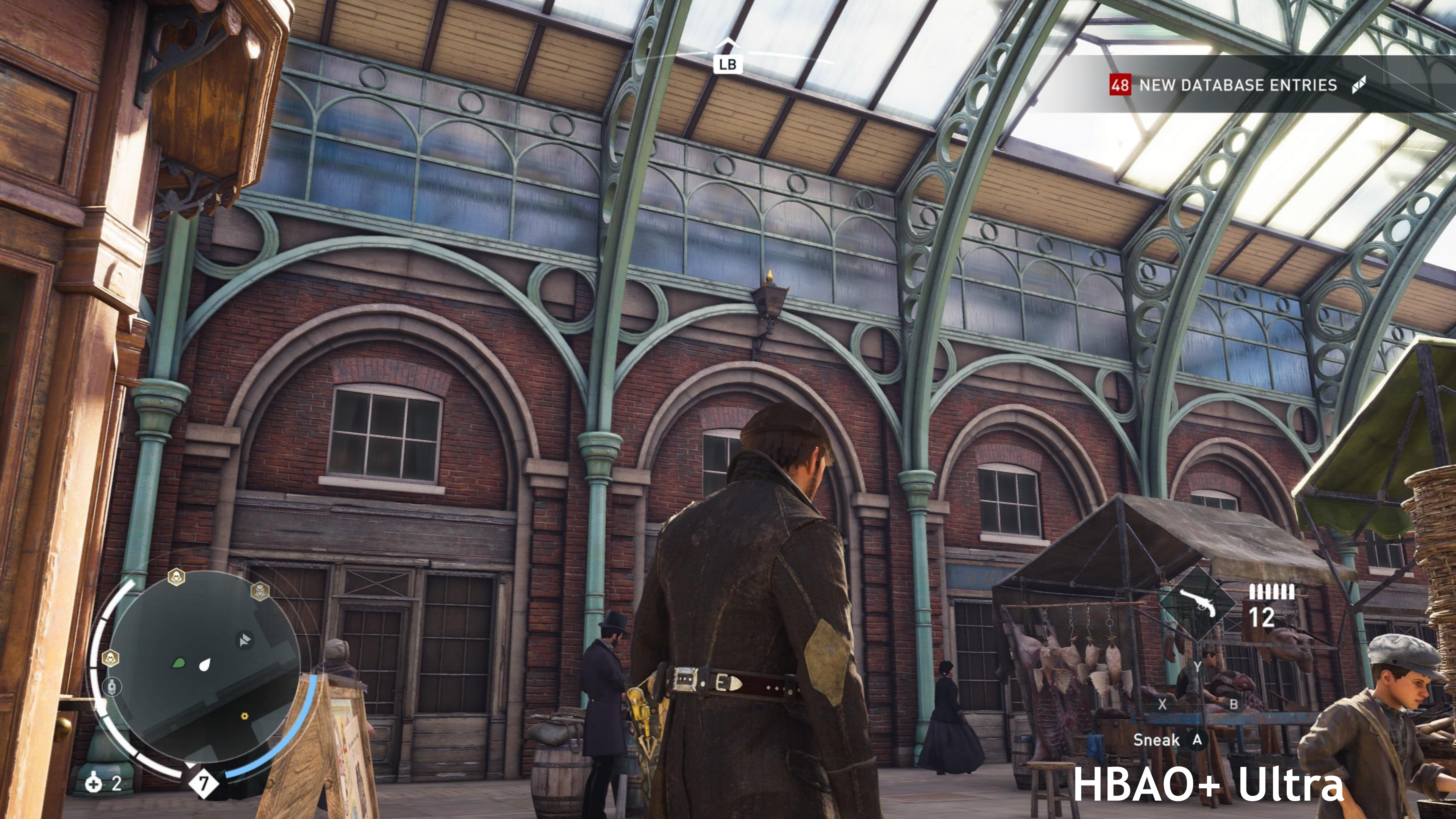

5
Y
X
B
Sneak
A
D
2
7
HBAO+



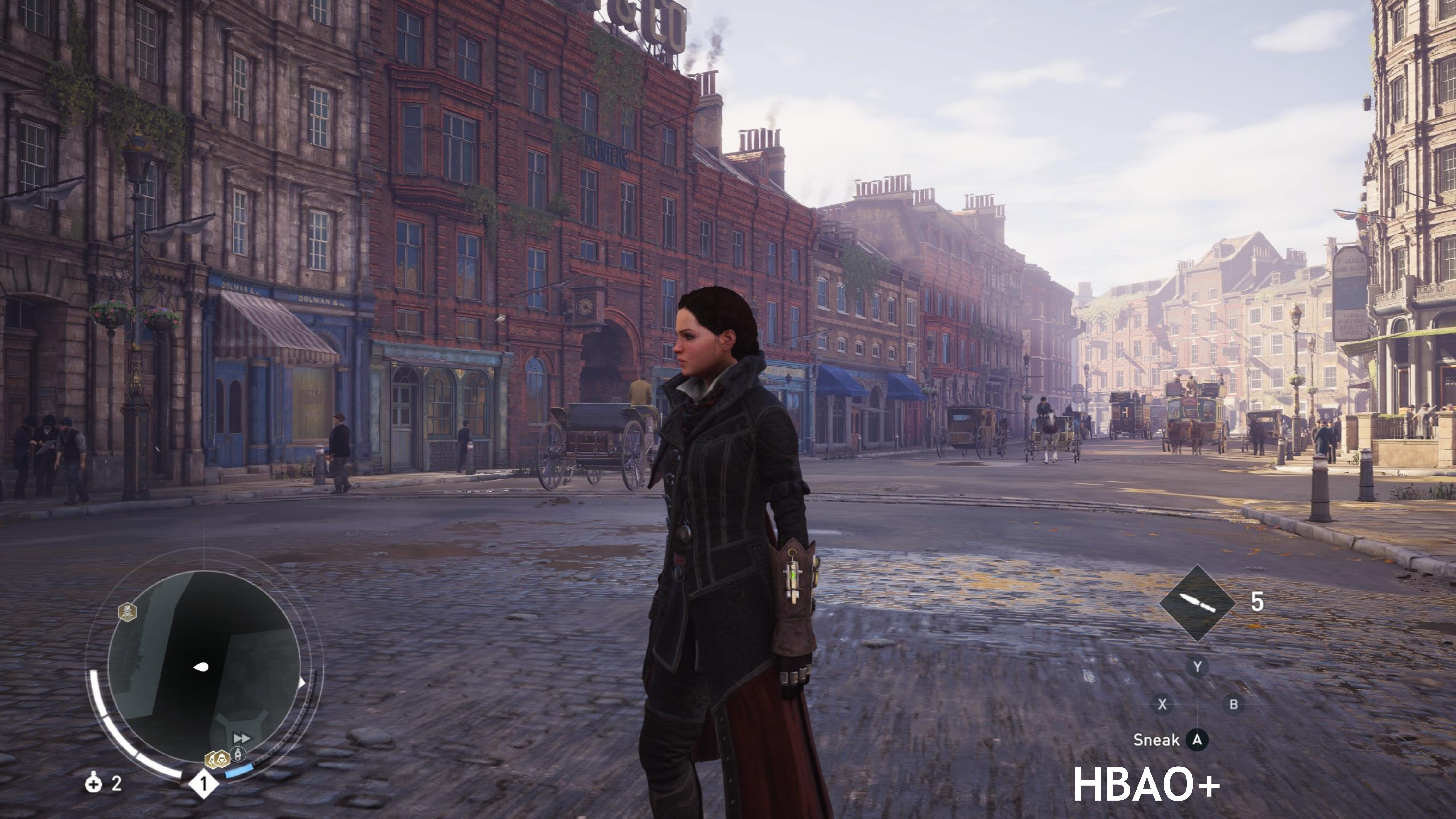
HBAO+ Ultra



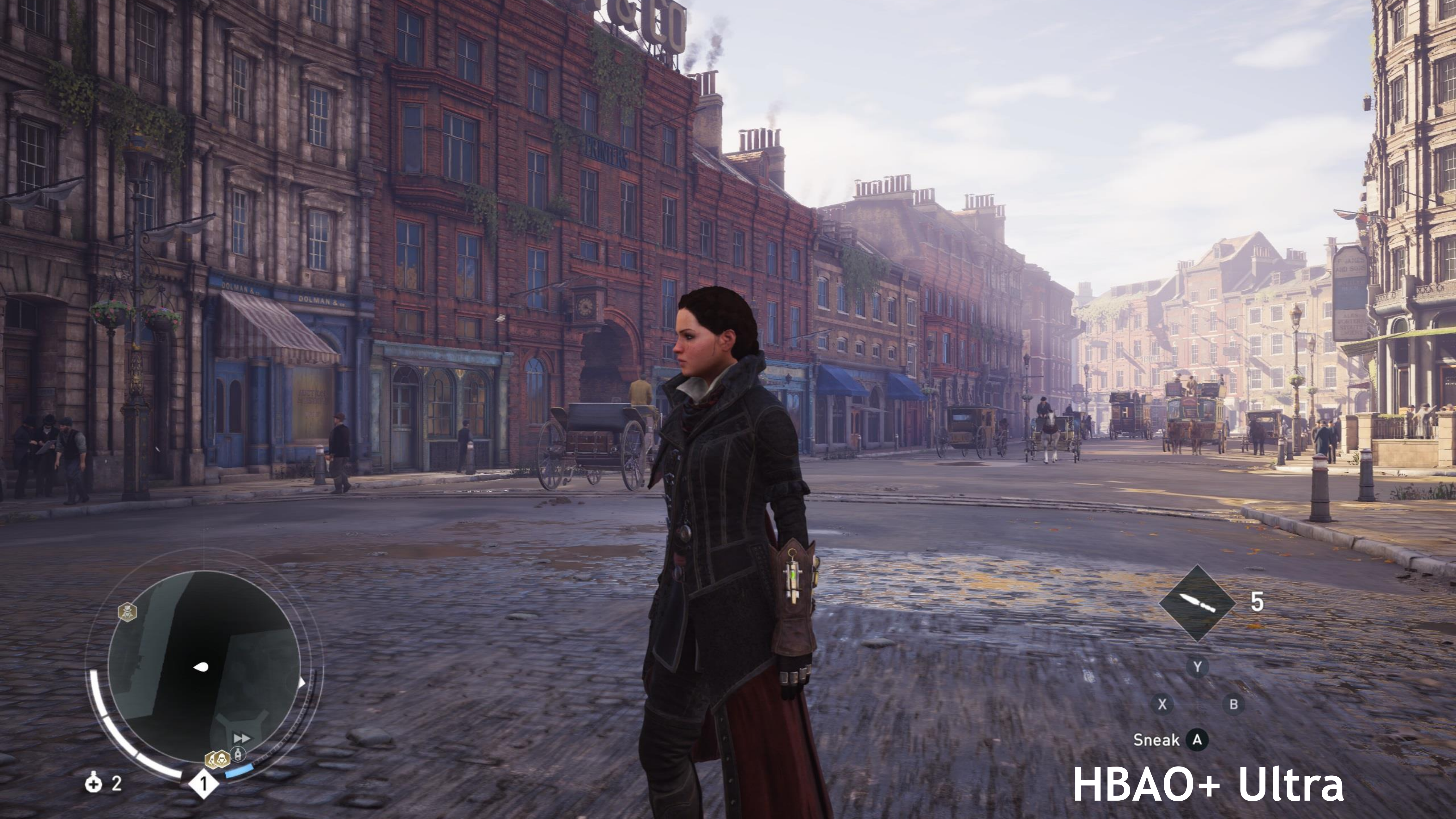
HBAO+



HBAO+ Ultra



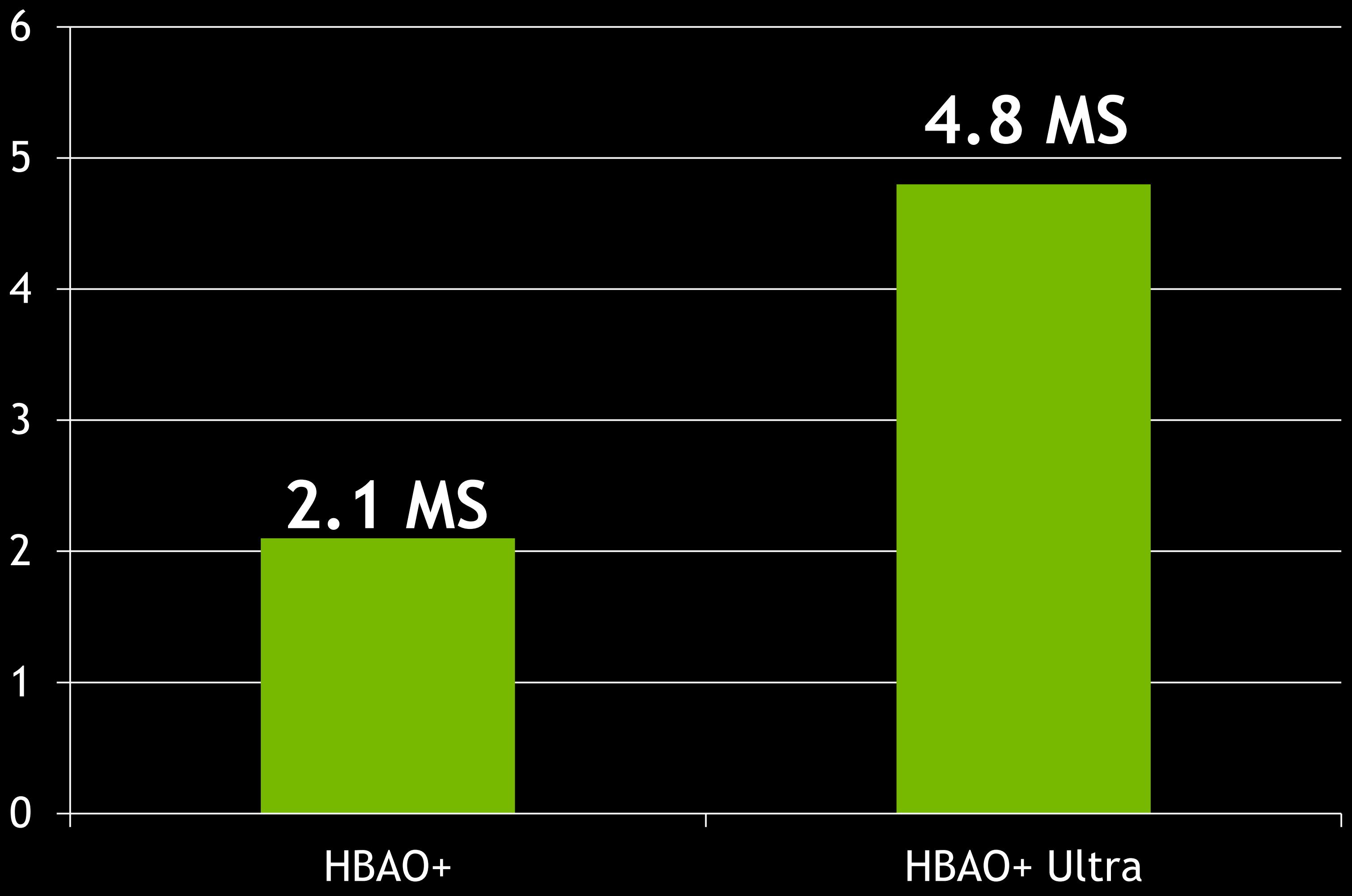
HBAO+



HBAO+ Ultra

Performance

- GeForce GTX 970
- 1920x1080



Summary

- Screen space is not enough for robust AO
- You need just two layers to improve quality
- HBAO+ supports advanced blending modes
- HBAO+ source will be available to registered developers

Huge thanks to

- Louis Bavoil, NVIDIA
- Maksym Rodionov, Ubisoft
- Oleksandr Puchka, Ubisoft
- Andrei Lange, Ubisoft

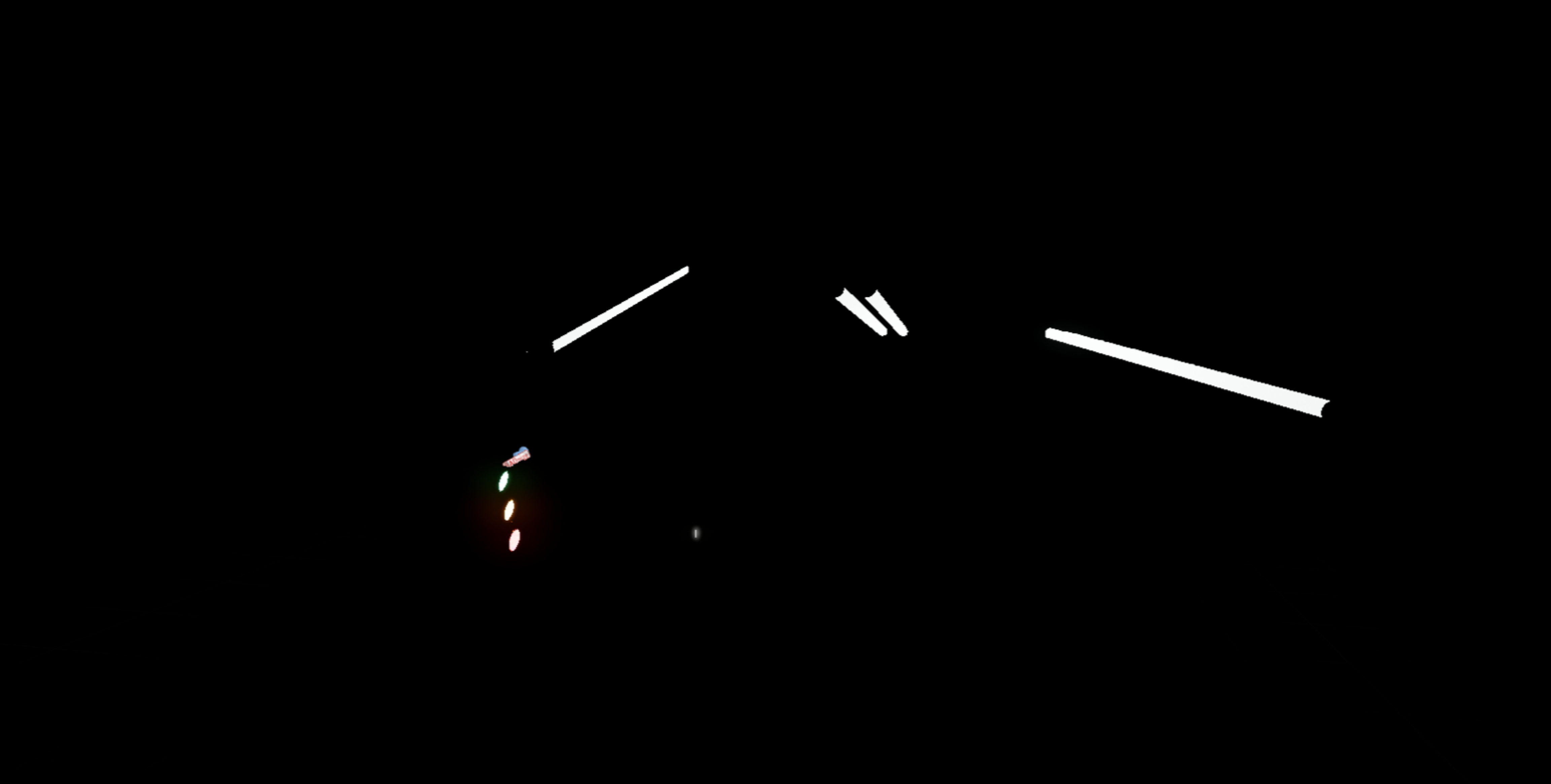
Voxel Ambient Occlusion

Background: NVIDIA VXGI

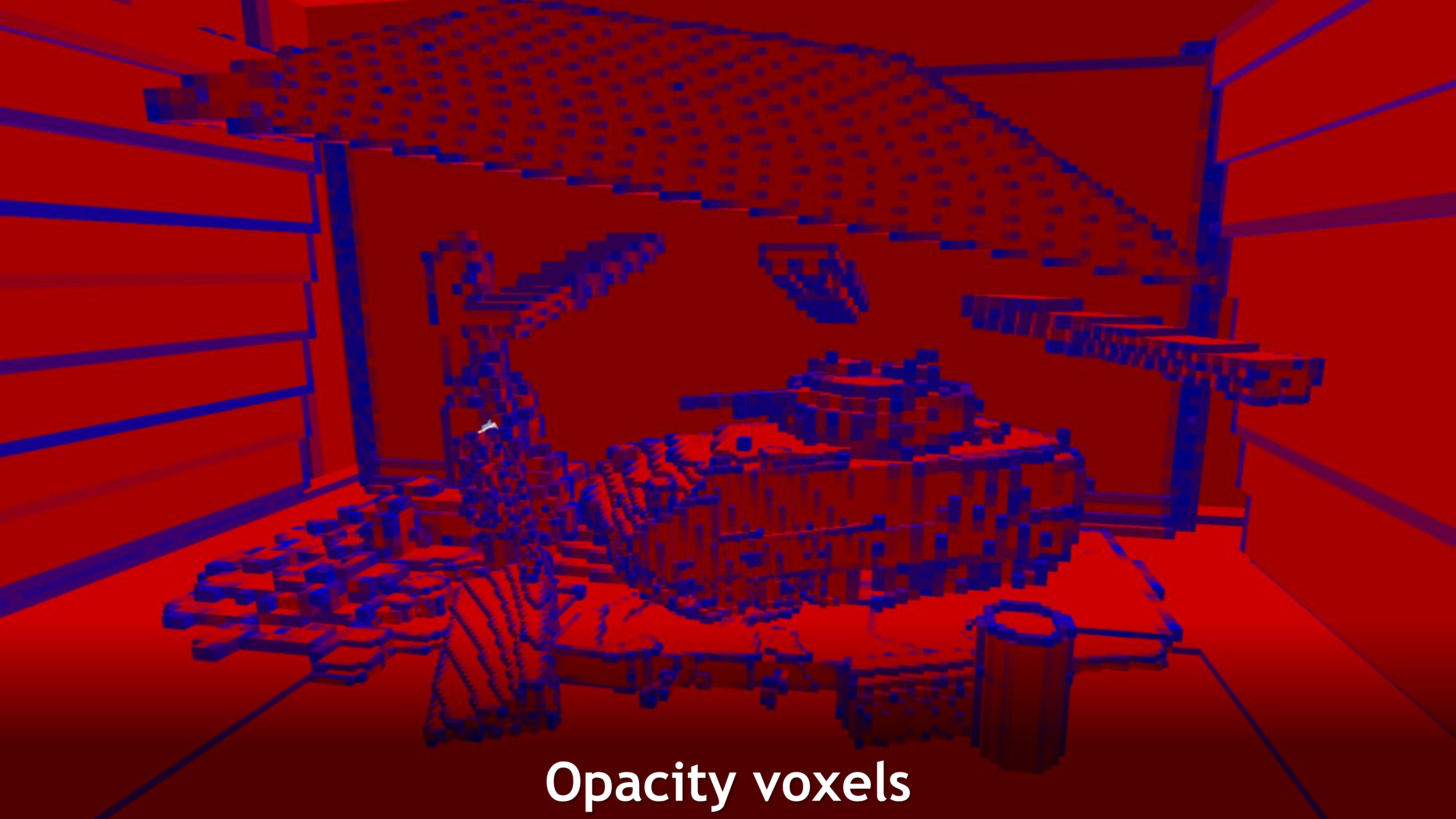
- Voxel Global Illumination

- NVIDIA's new real-time global illumination solution
  - Works by voxelizing geometry on every frame
  - Produces approximate but realistic looking diffuse and specular GI
  - Still too resource intensive to be used in mainstream games

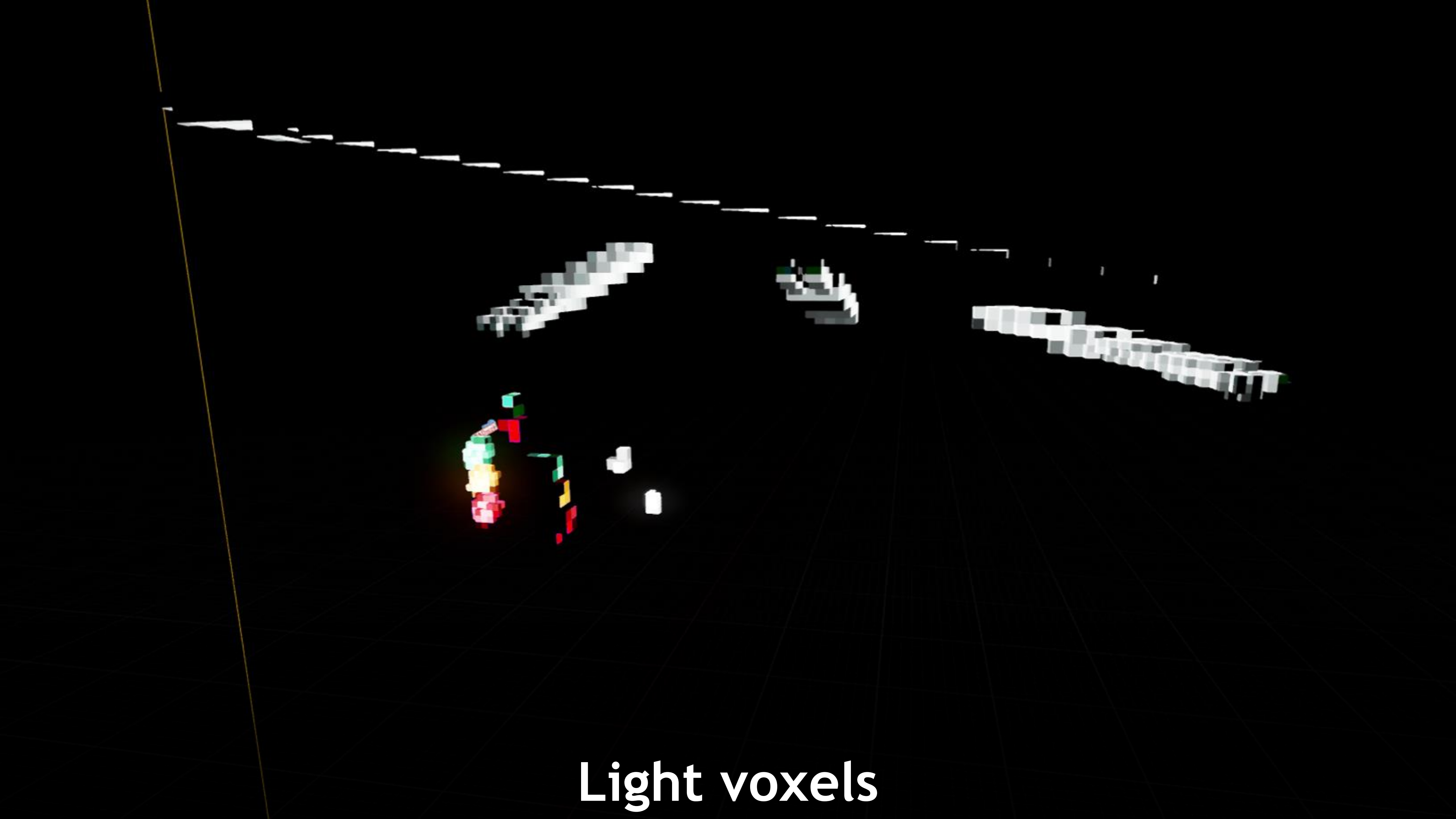
Unlit Scene



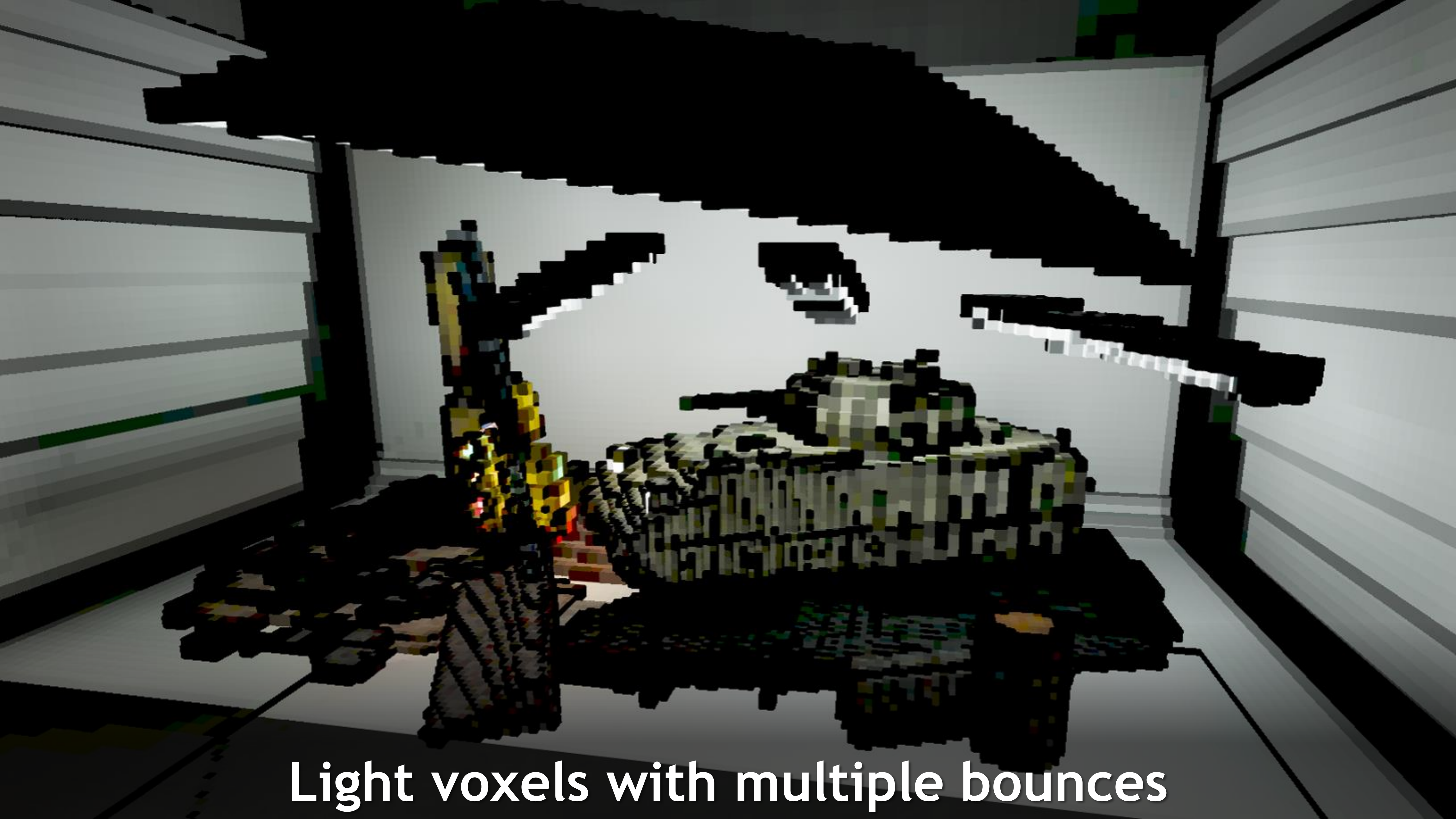
Scene with UE4 lighting



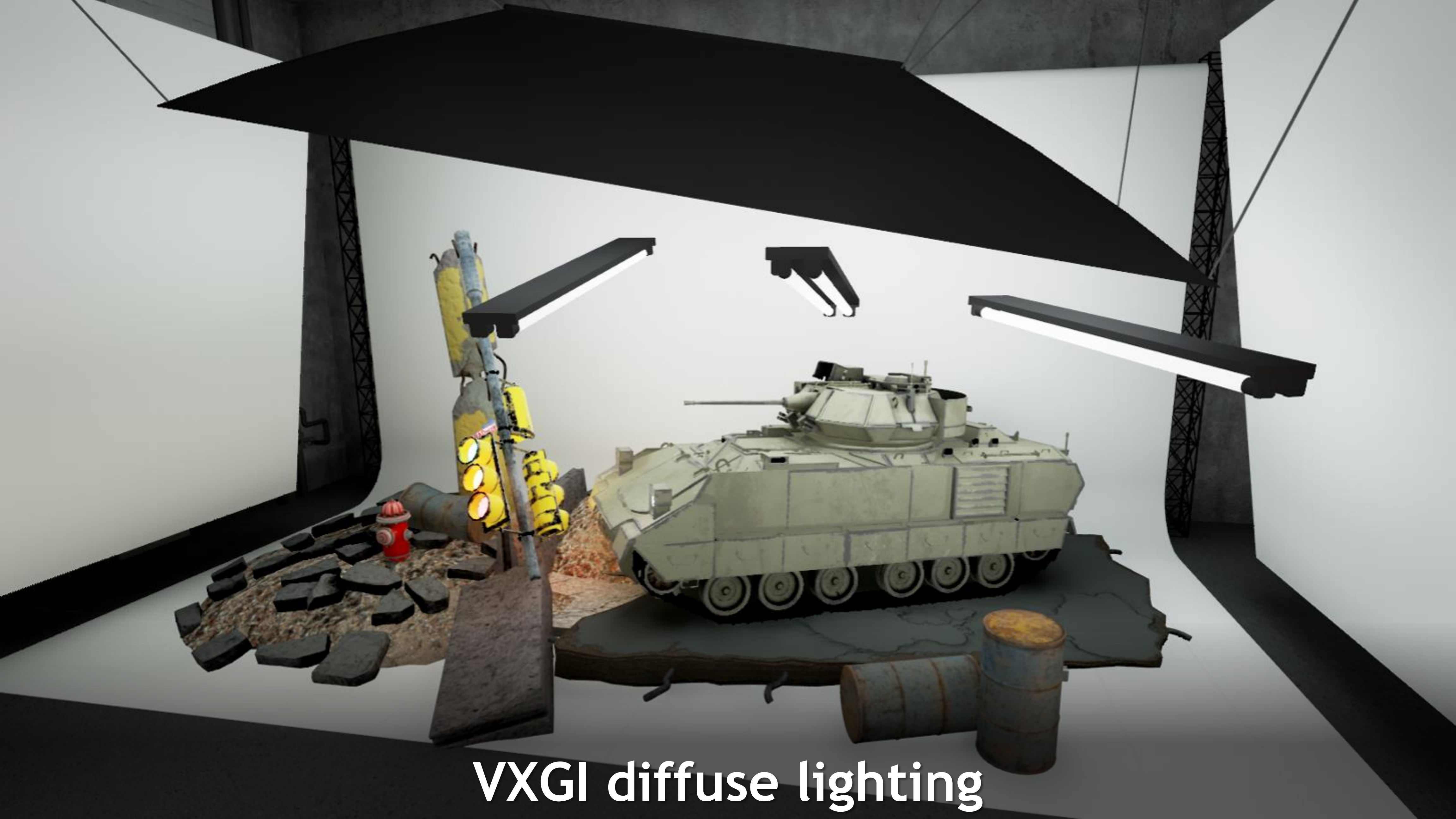
Opacity voxels



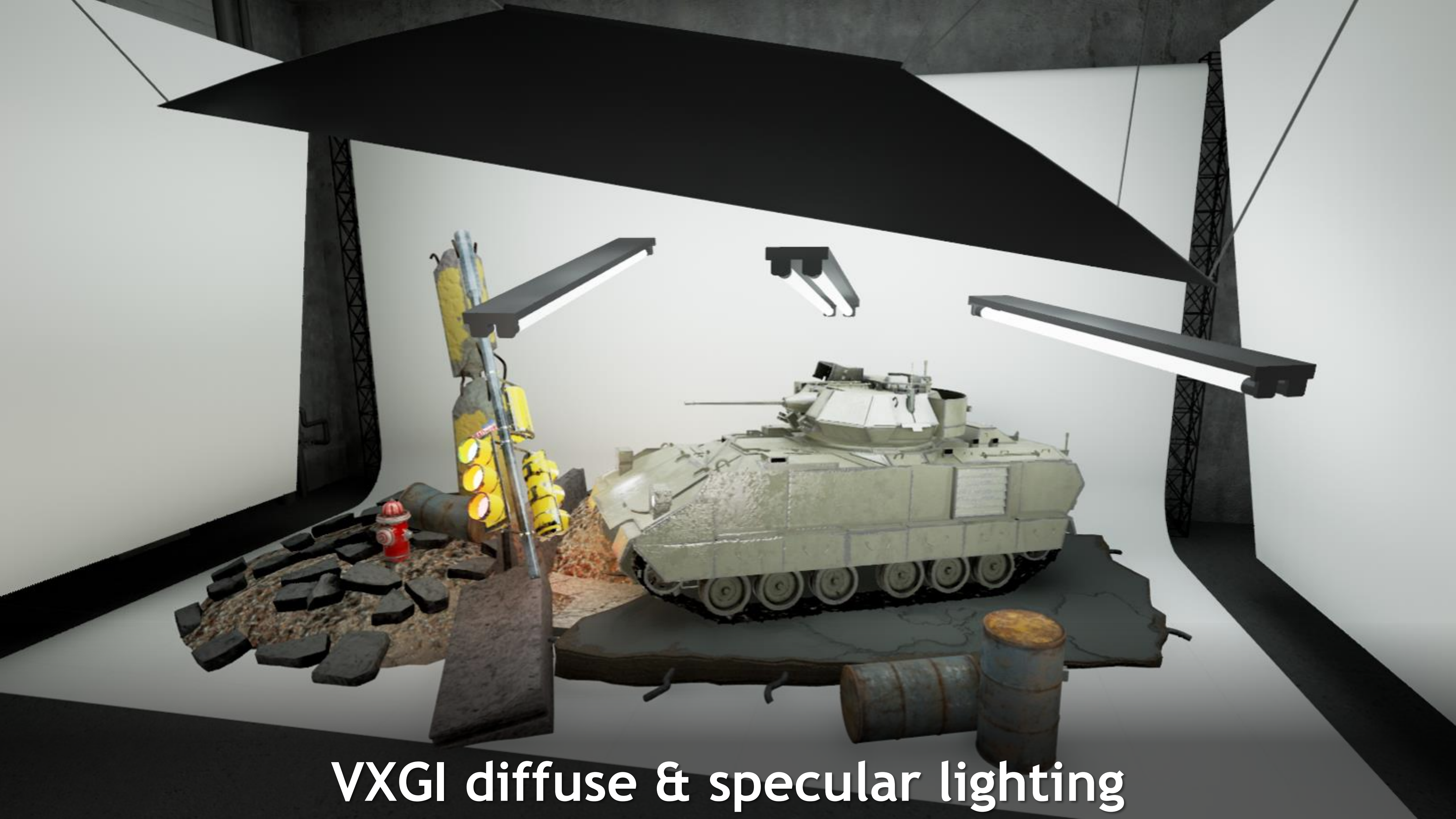
Light voxels



Light voxels with multiple bounces



VXGI diffuse lighting

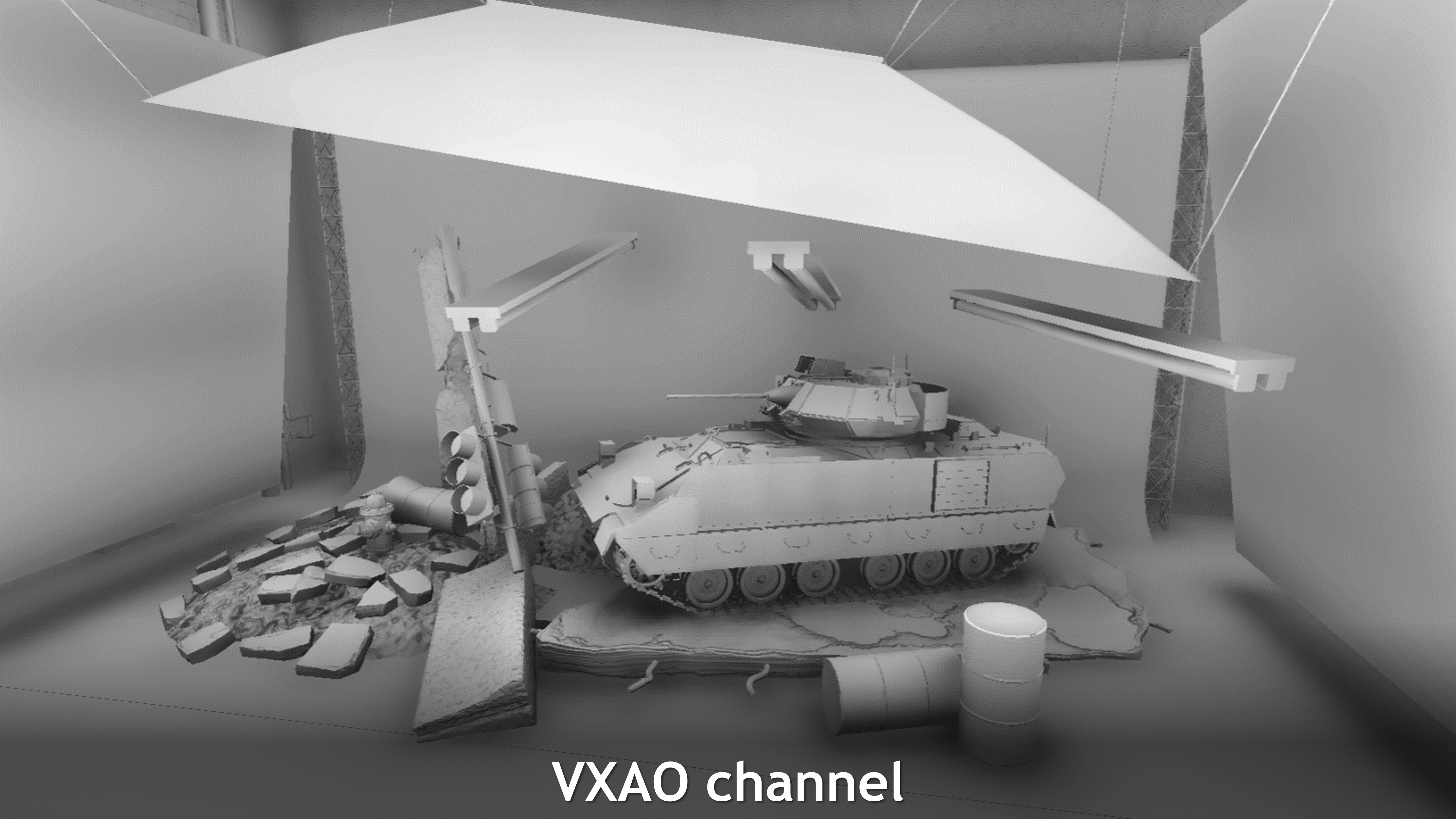


VXGI diffuse & specular lighting

VXGI without Illumination, or VXAO

- Remove the light voxels and rendering passes
- Assume that all space emits uniform light, occluded by opacity voxels
- Use the same diffuse cone tracing pass to compute ambient occlusion

- VXAO works much faster than VXGI
- VXAO engine integration is much simpler than VXGI integration

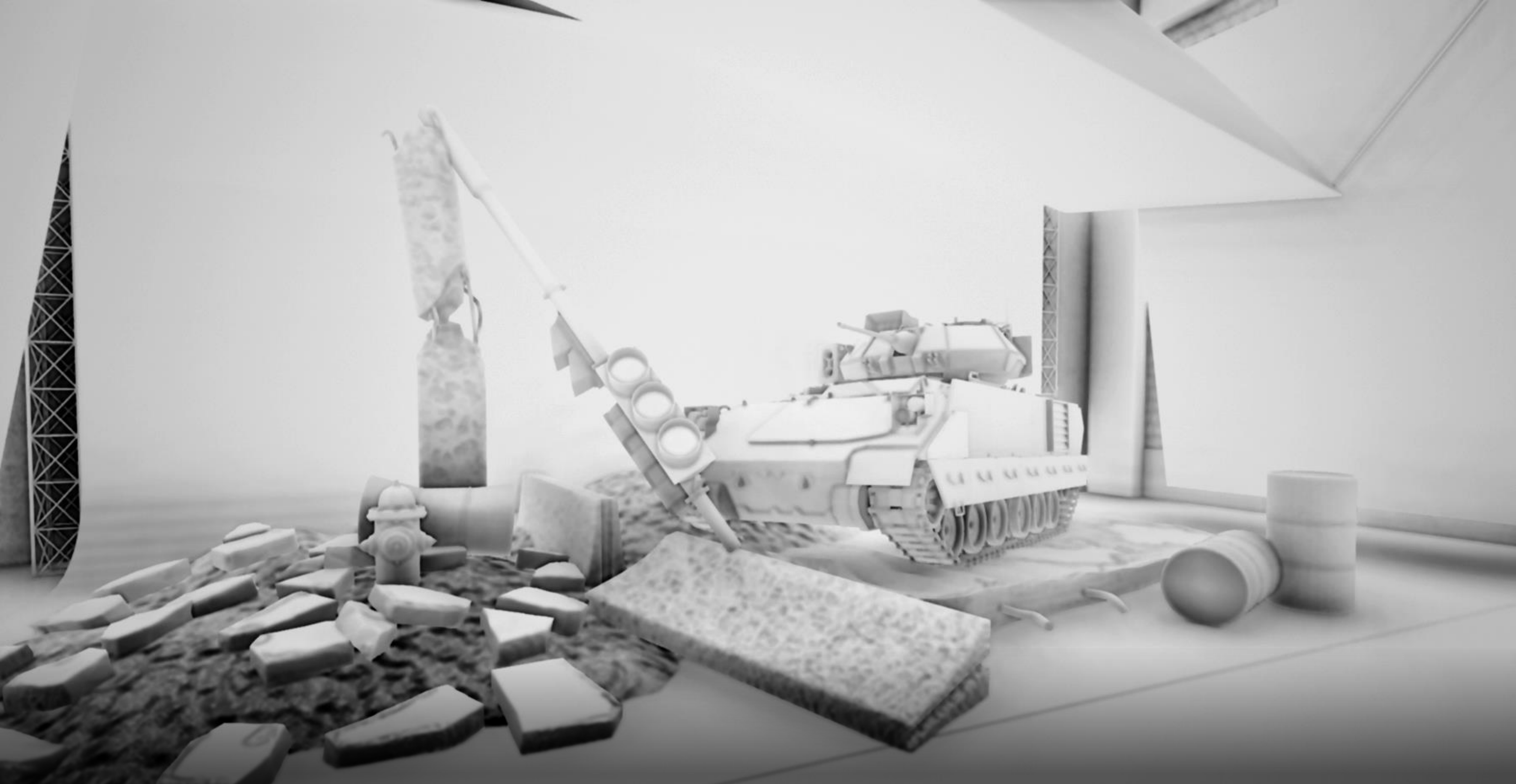


VXAO channel

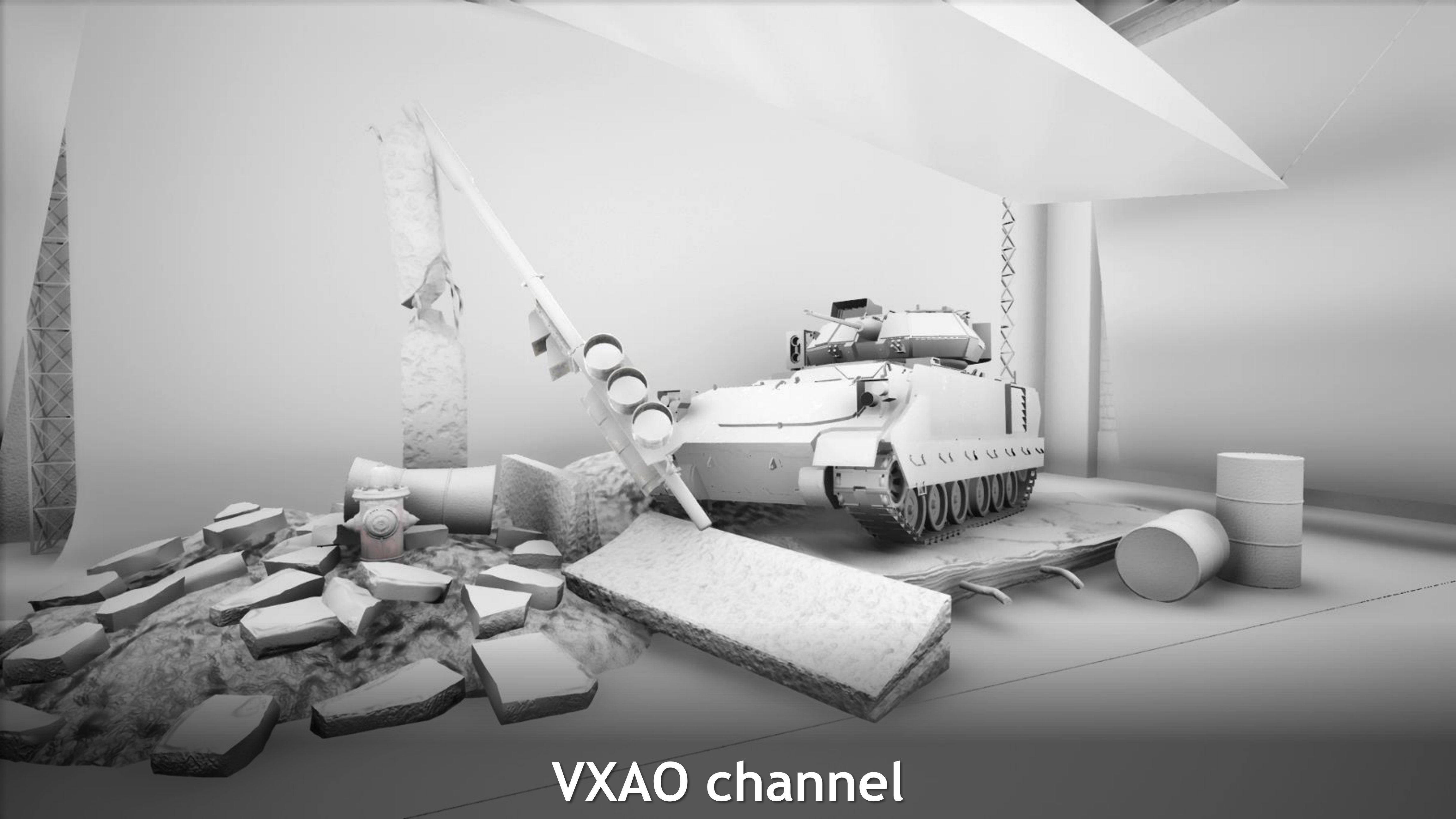
Why VXAO is Better Than SSAO?

- More stable, Large radius effect

- World-space solution: more data available
- Doesn’t lose any hidden or unfortunately oriented occluders
- Occluders can be far away from visible surfaces
- Completely stable under small camera movements
- Completely stable near screen borders



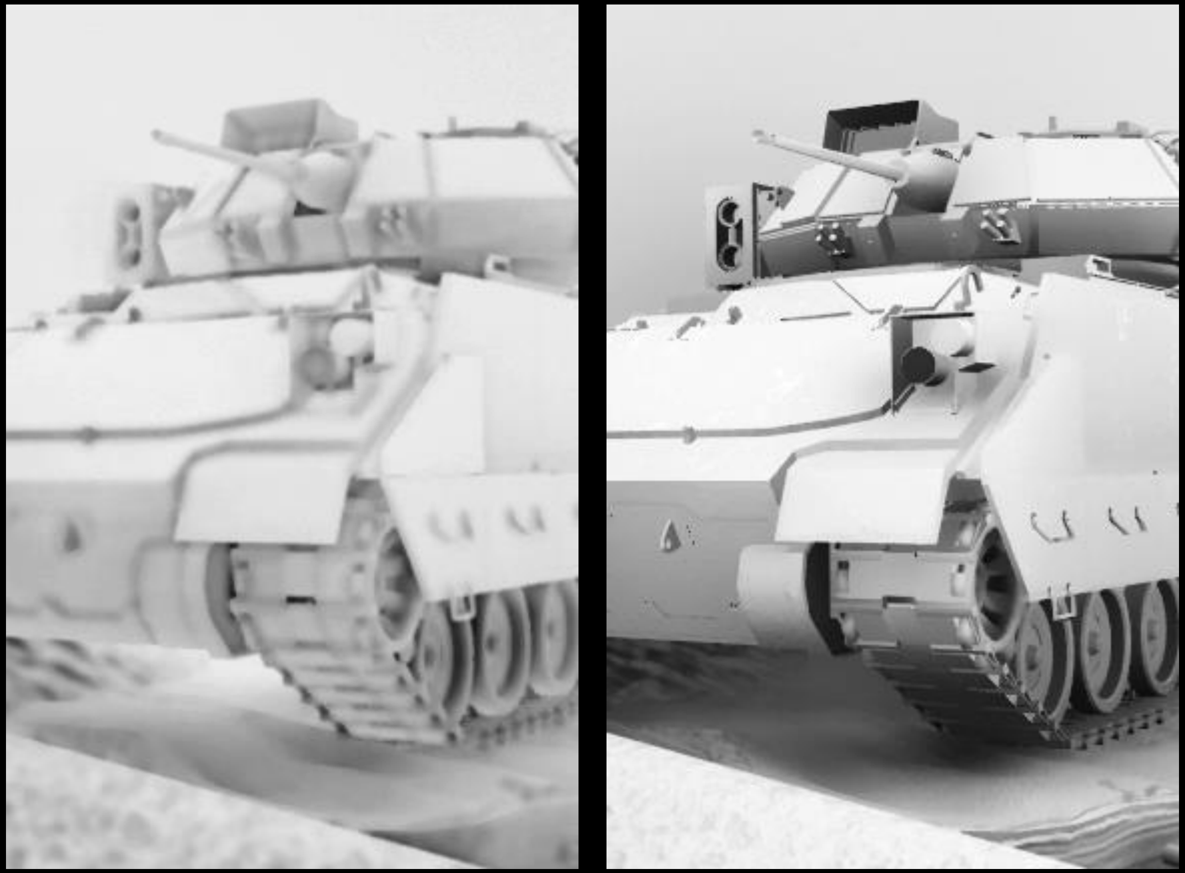
HBAO+ channel



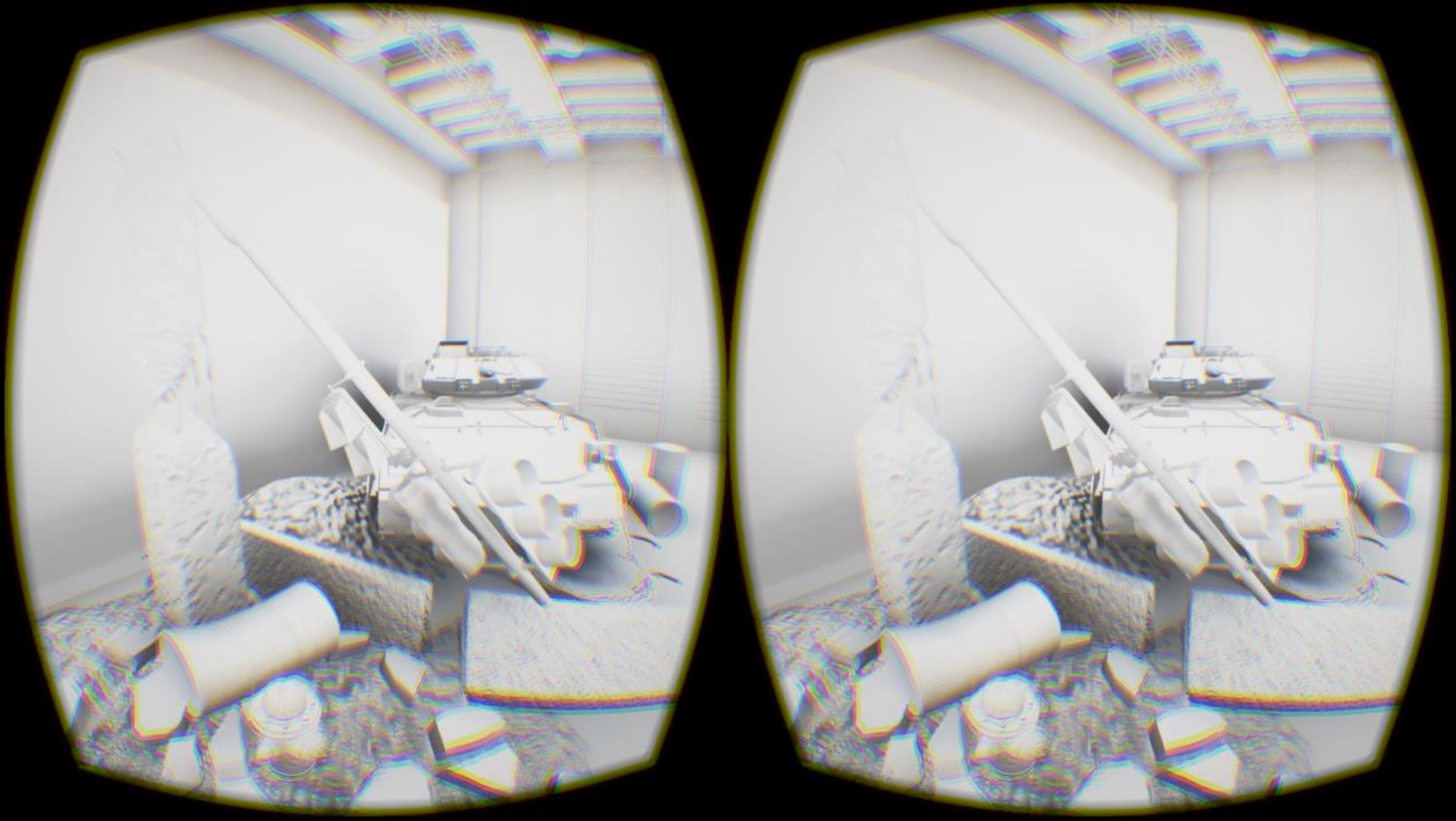
VXAO channel

Image Quality Differences

- HBAO+ vs. VXAO Channels
  - Ground under the tank
  - Bottom part of the tracks
  - Blurriness or lack thereof



VXAO works great with VR!



gameworks.nvidia.com

Handling Dynamic Scenes

- Voxel representation is very expensive to construct or update?
    - Wrong.
    - It takes 1-4 ms to voxelize a full typical game scene on a GTX 980

- Most of voxel data can be preserved between frames to improve performance
- VXGI can clear and update only a set of regions specified by the app

VXAO System Requirements

- Any DX11 class GPU
  - Maxwell GPUs bring some useful hardware features to accelerate VXAO

- 10 – 140 MB of video memory, depending on configuration
- Supported graphics APIs are: DX11, DX12, GL4.5

VXAO Engine Integration

- Engine is responsible for VXGI interaction with the rendering API
    - ❖ Reference API backends are provided for DX11, DX12 and GL

- Engine has to render geometry using VXGI-provided GS, PS and some other state
    - ❖ Be careful not to reset the VXAO state while drawing geometry!

- VXAO needs depth and normal channels of the G-buffer

VXAO in Unreal Engine 4

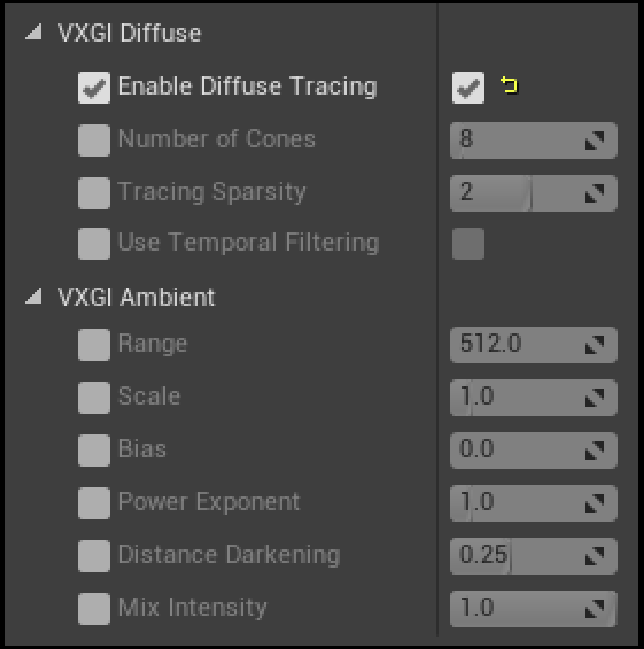
Unreal Engine 4 VXGI Integration

- Available on GitHub since February 2015
- Requires an Unreal Engine 4 subscription
- Set “r.VXGI.AmbientOcclusionMode 1” cvar to switch to VXAO mode

- Tech support on the UE forums:
  - Community / General Discussion / NVIDIA GameWorks Integration

Working with VXAO in UE4

- Create an unbounded PostProcessVolume
- Check “Enable Diffuse Tracing”
  - In “Settings / VXGI Diffuse”
- Tweak the parameters in “Settings / VXGI Ambient”
- VXAO is mixed into SSAO channel
  - Unless “Mix Intensity” is 0
- Use “VXGI Diffuse” channel in post-process materials



VXGI Diffuse

☑ Enable Diffuse Tracing	☑
□ Number of Cones	8
□ Tracing Sparsity	2
□ Use Temporal Filtering	□

VXGI Ambient

□ Range	512.0
□ Scale	1.0
□ Bias	0.0
□ Power Exponent	1.0
□ Distance Darkening	0.25
□ Mix Intensity	1.0

DEMO: VXAO in Unreal Engine 4

VXAO in Rise of the Tomb Raider

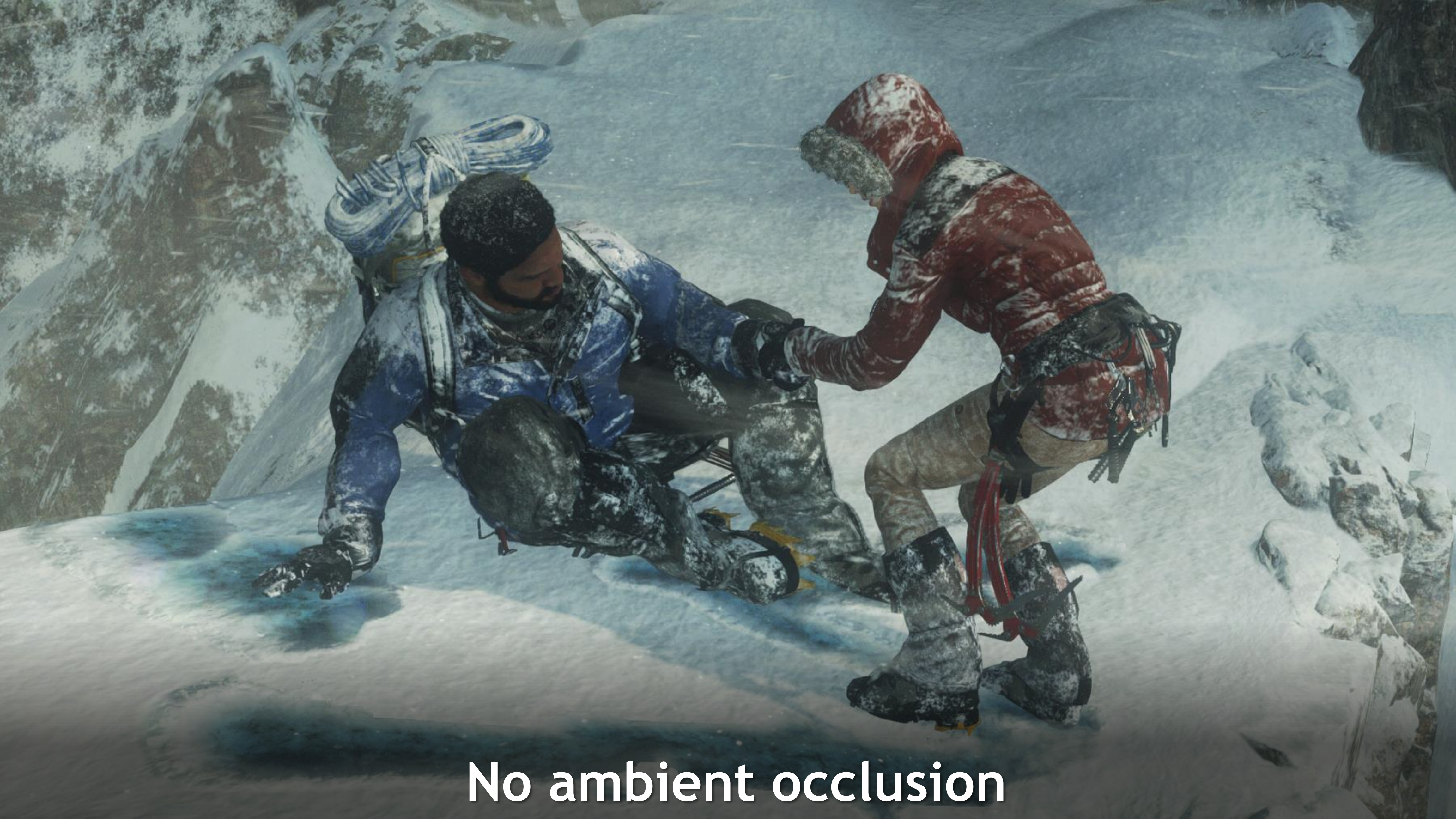
RISE OF THE TOMB RAIDER™

CRYSTAL DYNAMICS

ROTTR Rendering Engine

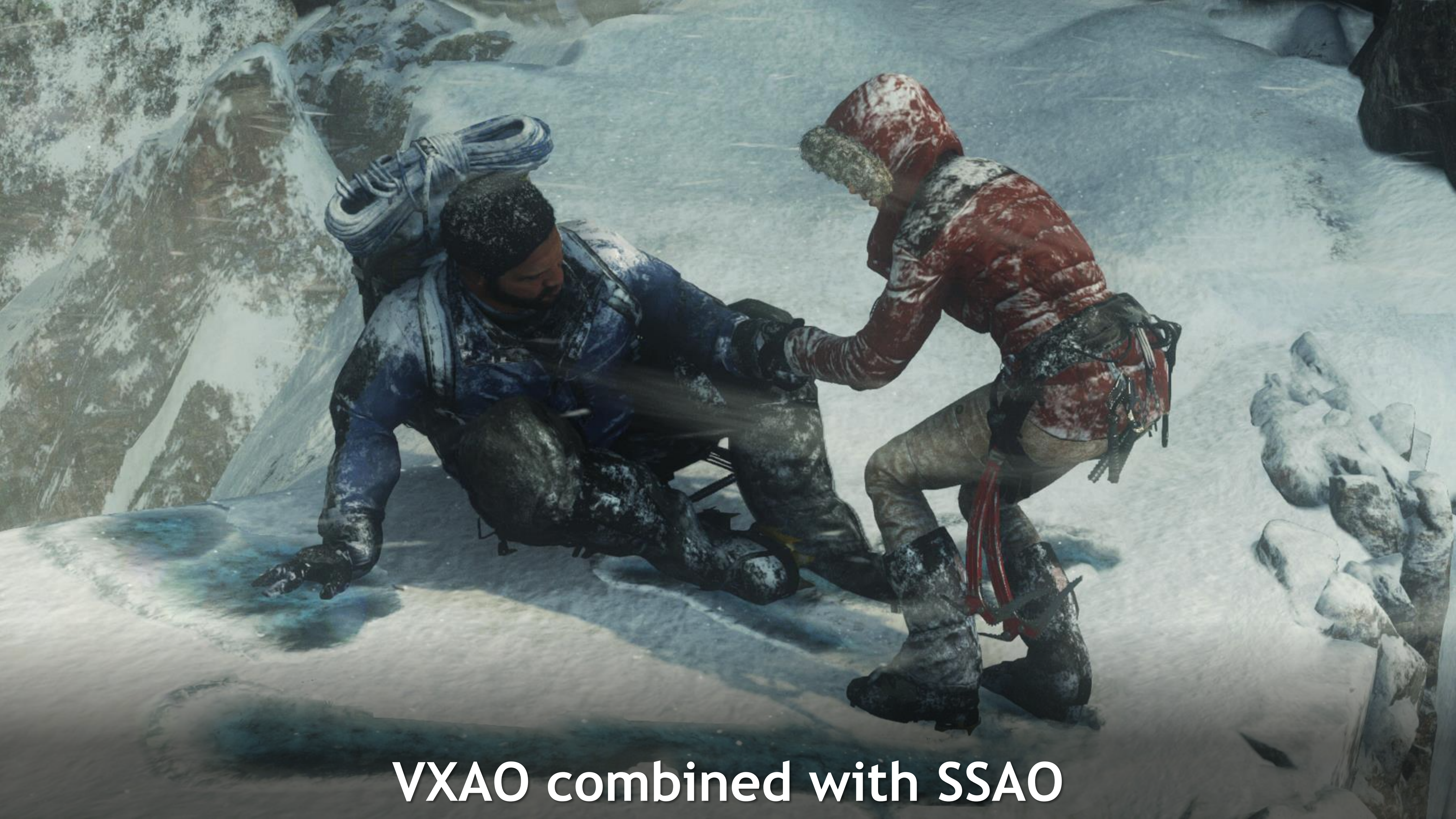
- Foundation Engine by Crystal Dynamics

- Physically based materials
- Image based lighting
- Volumetric lights
- Broad Temporal Ambient Obscurance
- NVIDIA HBAO+ and now VXAO



No ambient occlusion

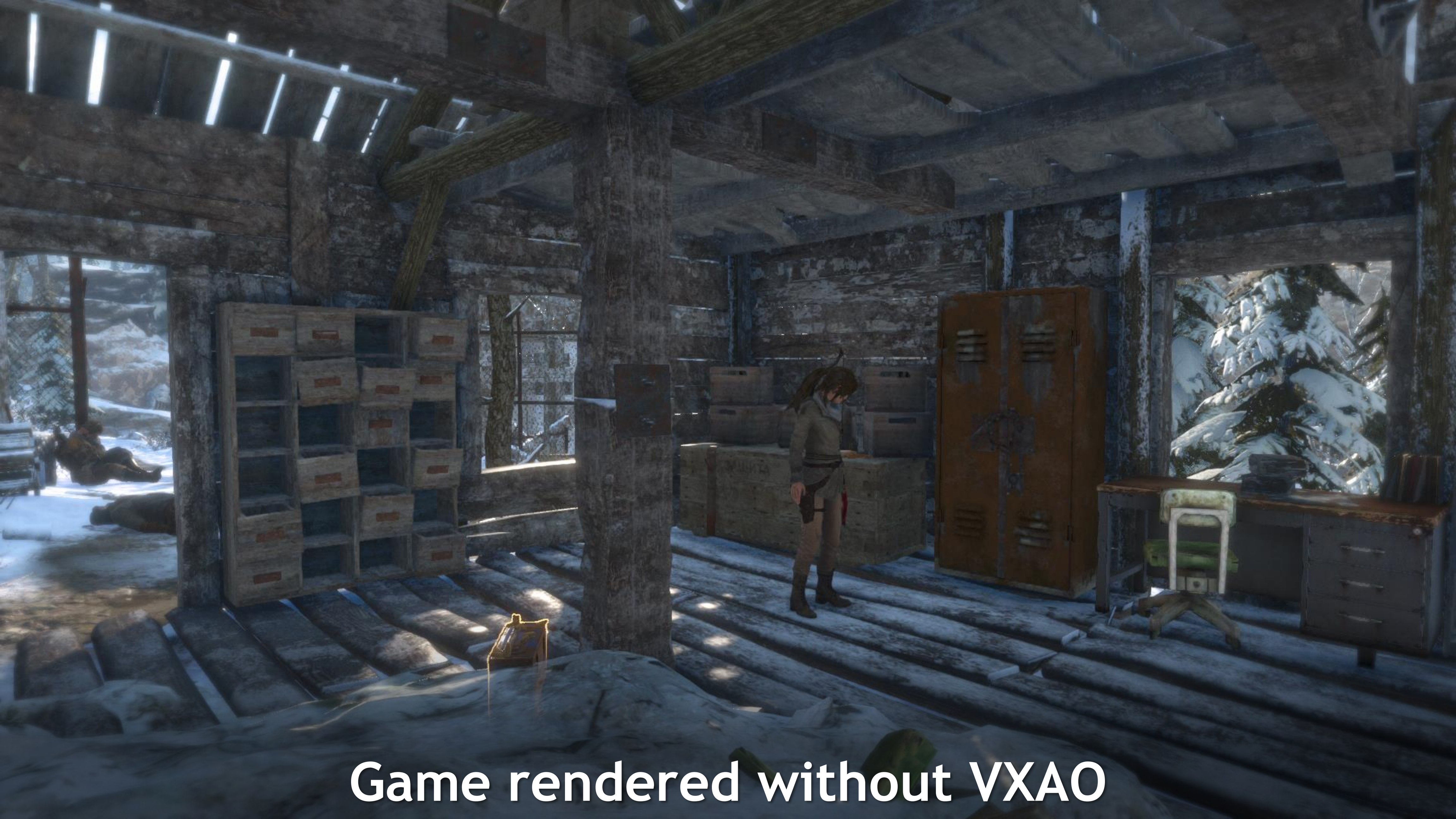
Screen-space ambient occlusion



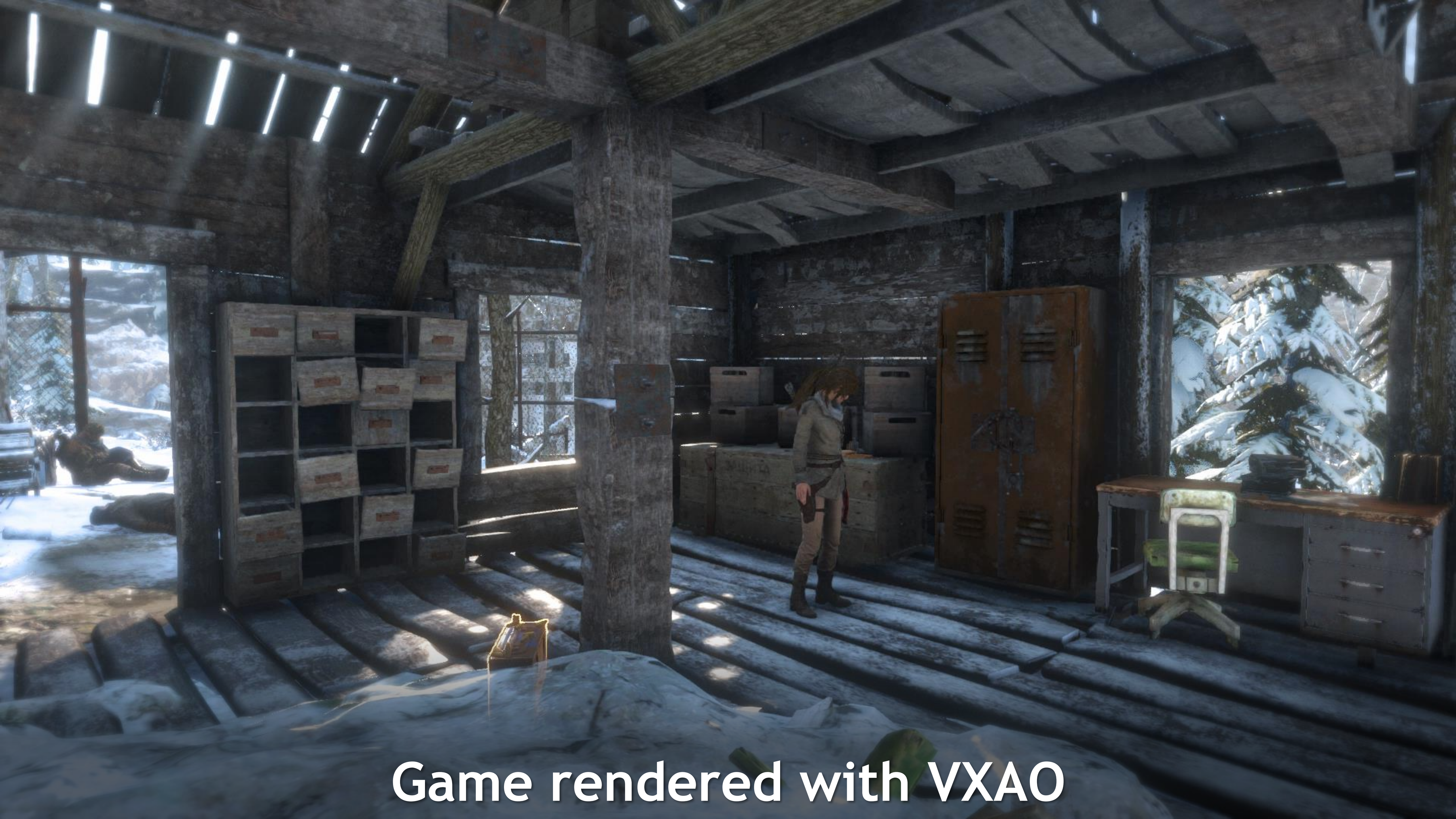
VXAO combined with SSAO

Using VXAO Signal in the Game

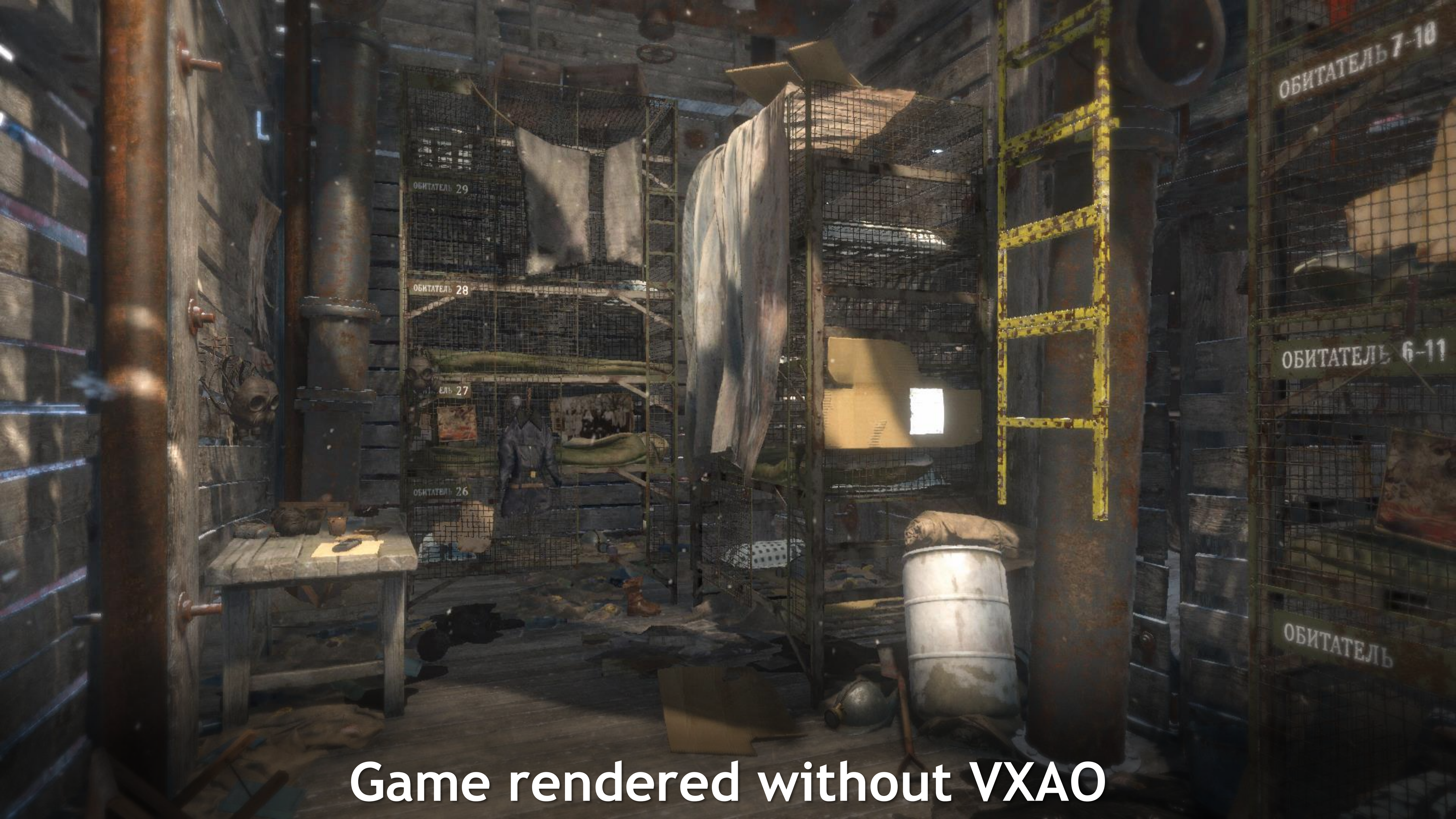
- Separate channels for Ambient Lighting (AL) and Ambient Occlusion (AO)
  - Different materials use these channels differently
  - AO channel is applied on top of direct lights, too: they become dimmer
  - Some materials ignore the AO channel: looks unnatural with VXAO
  - Some materials ignore the AL channel: enabling VXAO shows no difference
- We chose to always multiply VXAO signal into the AL channel
  - Lack of difference is better than unnatural result
  - Some locations start looking much more realistic!



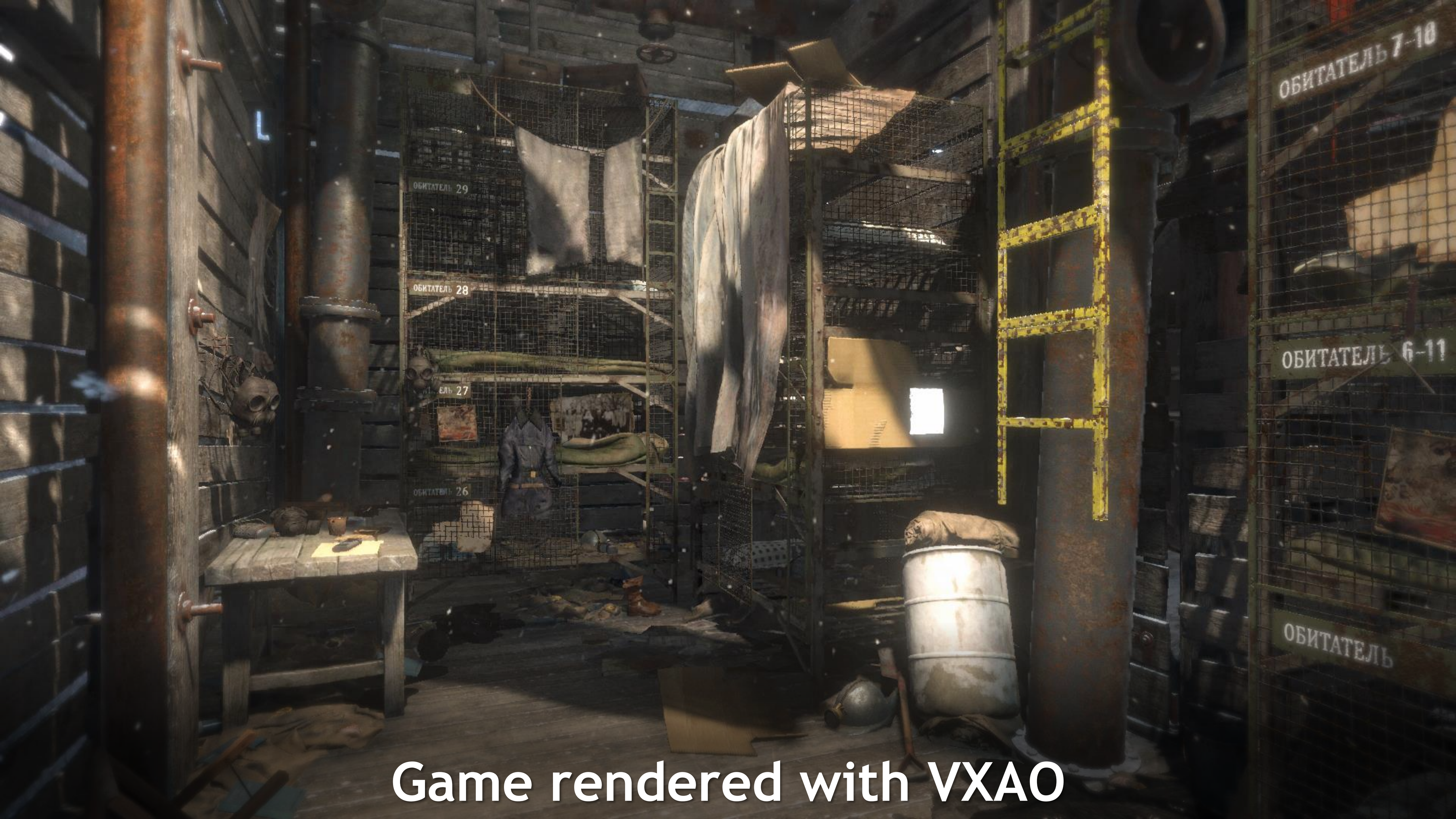
Game rendered without VXAO



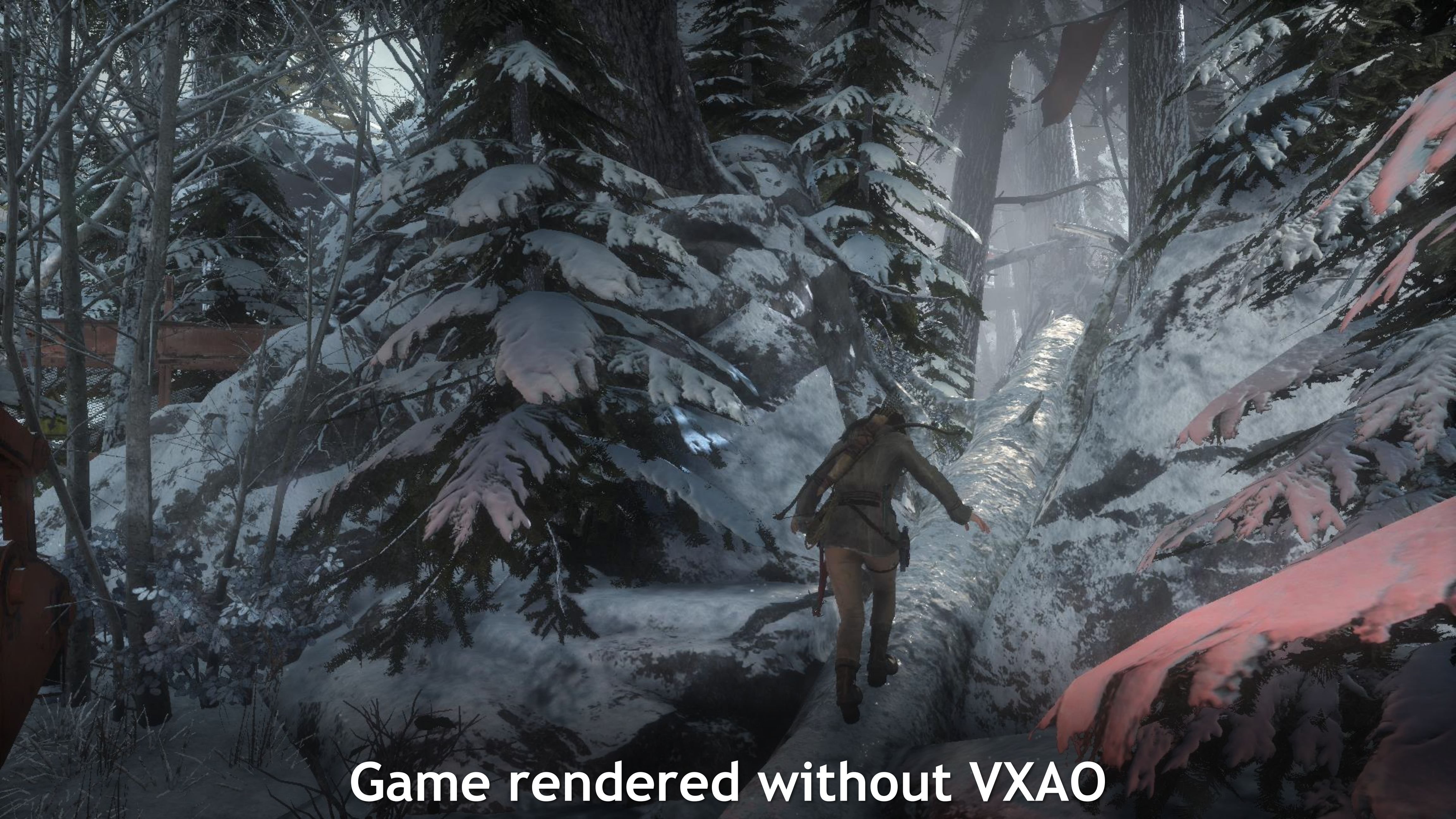
Game rendered with VXAO



Game rendered without VXAO

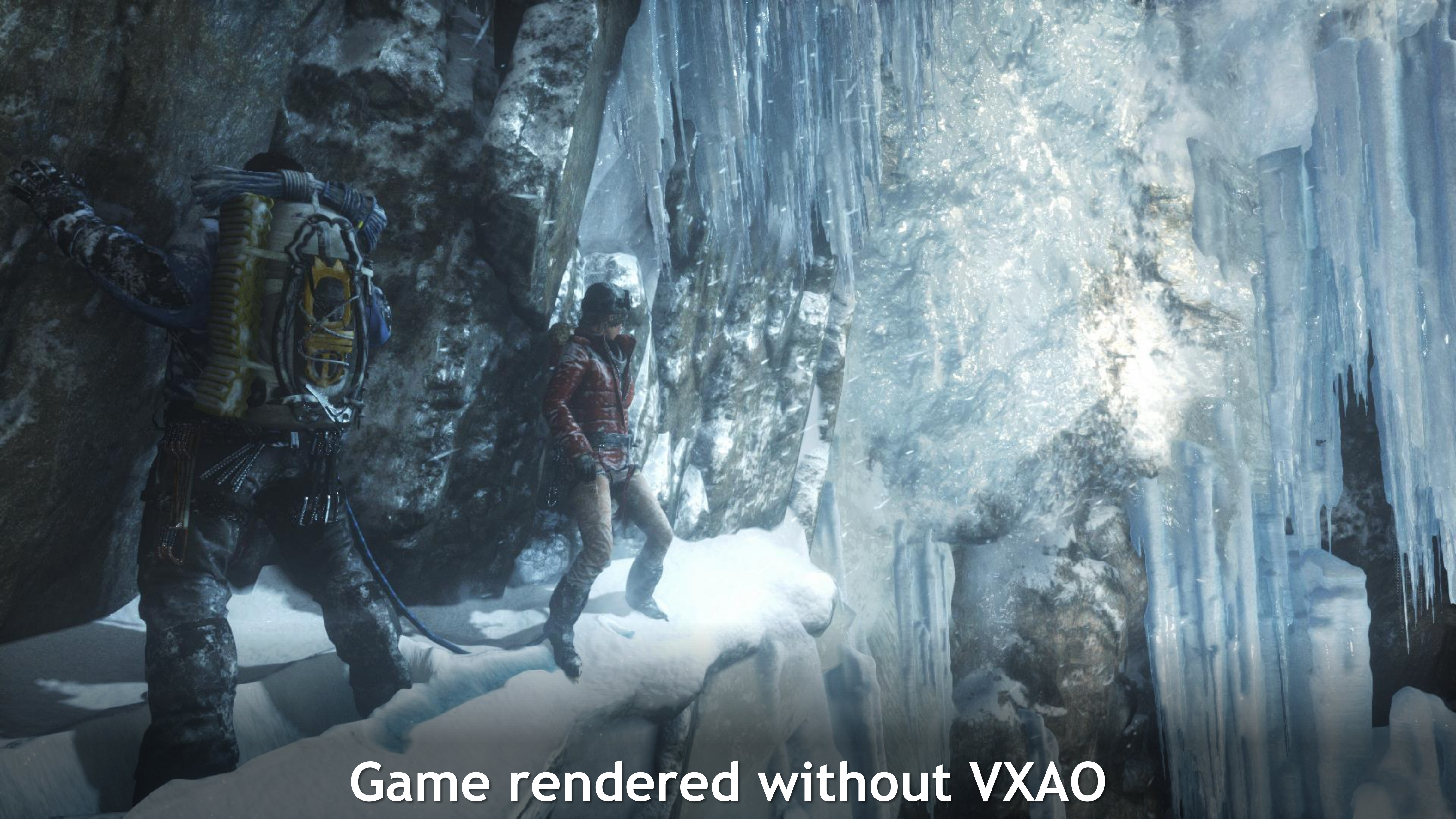


Game rendered with VXAO

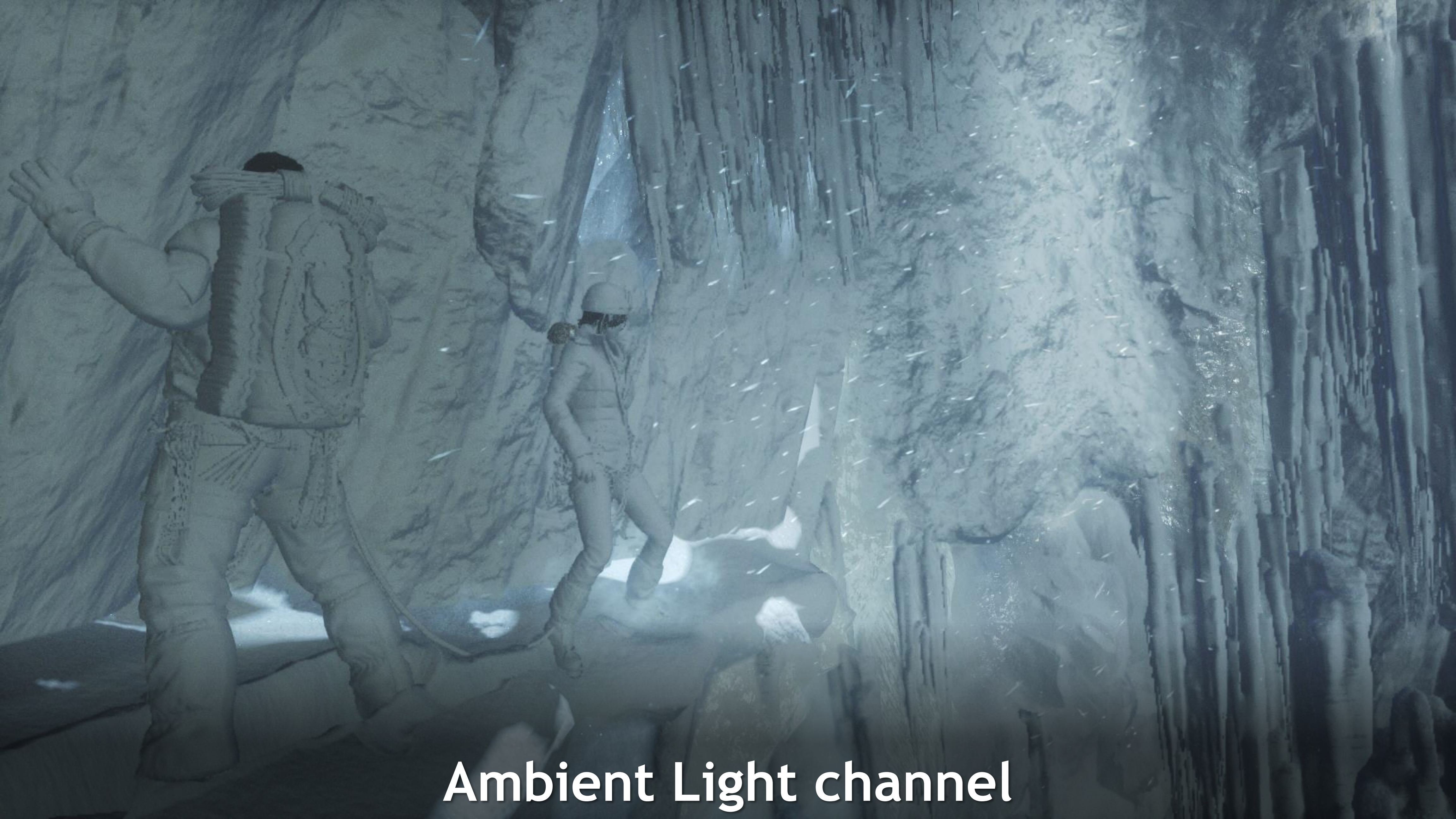


Game rendered without VXAO

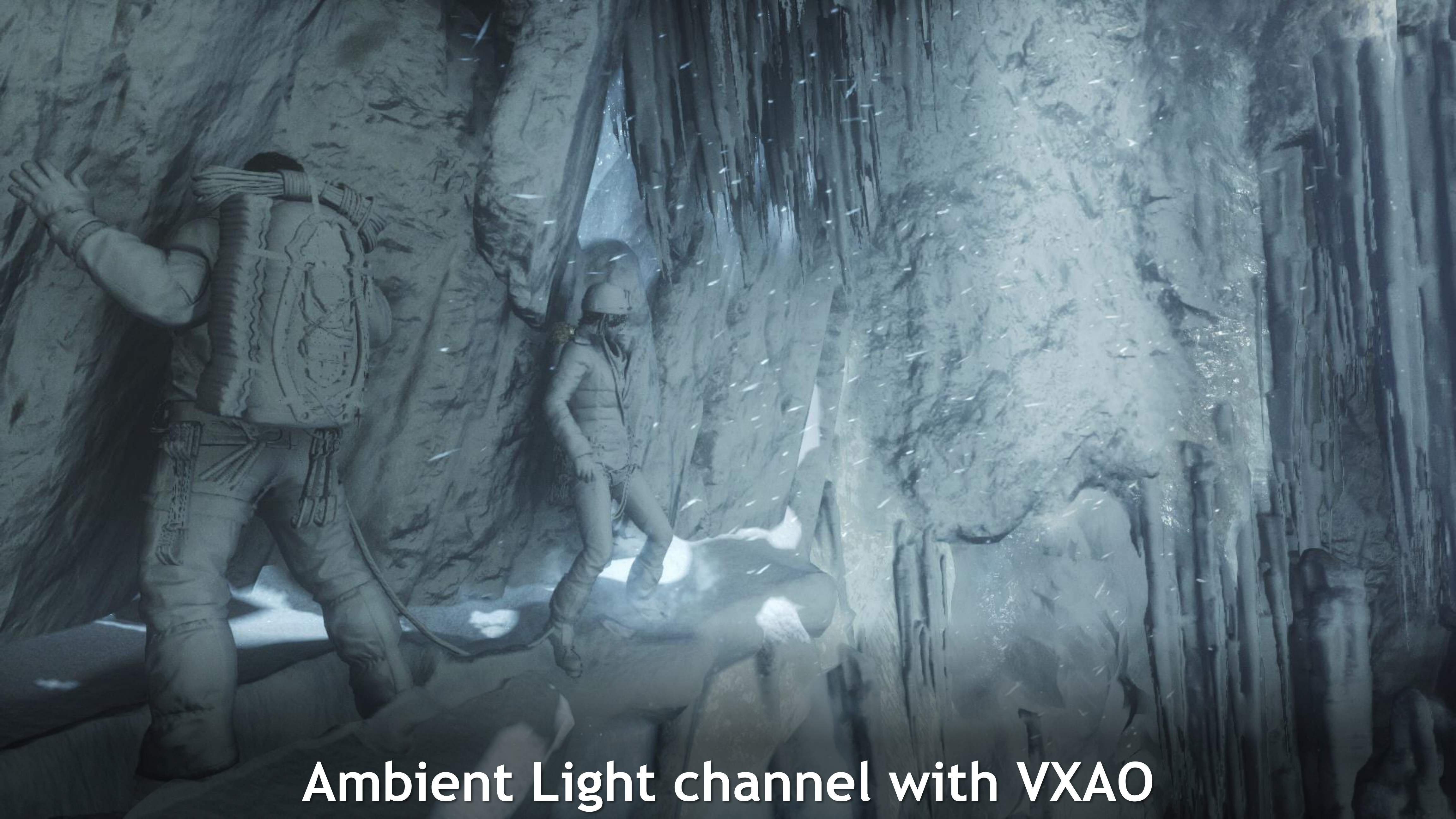
Game rendered with VXAO



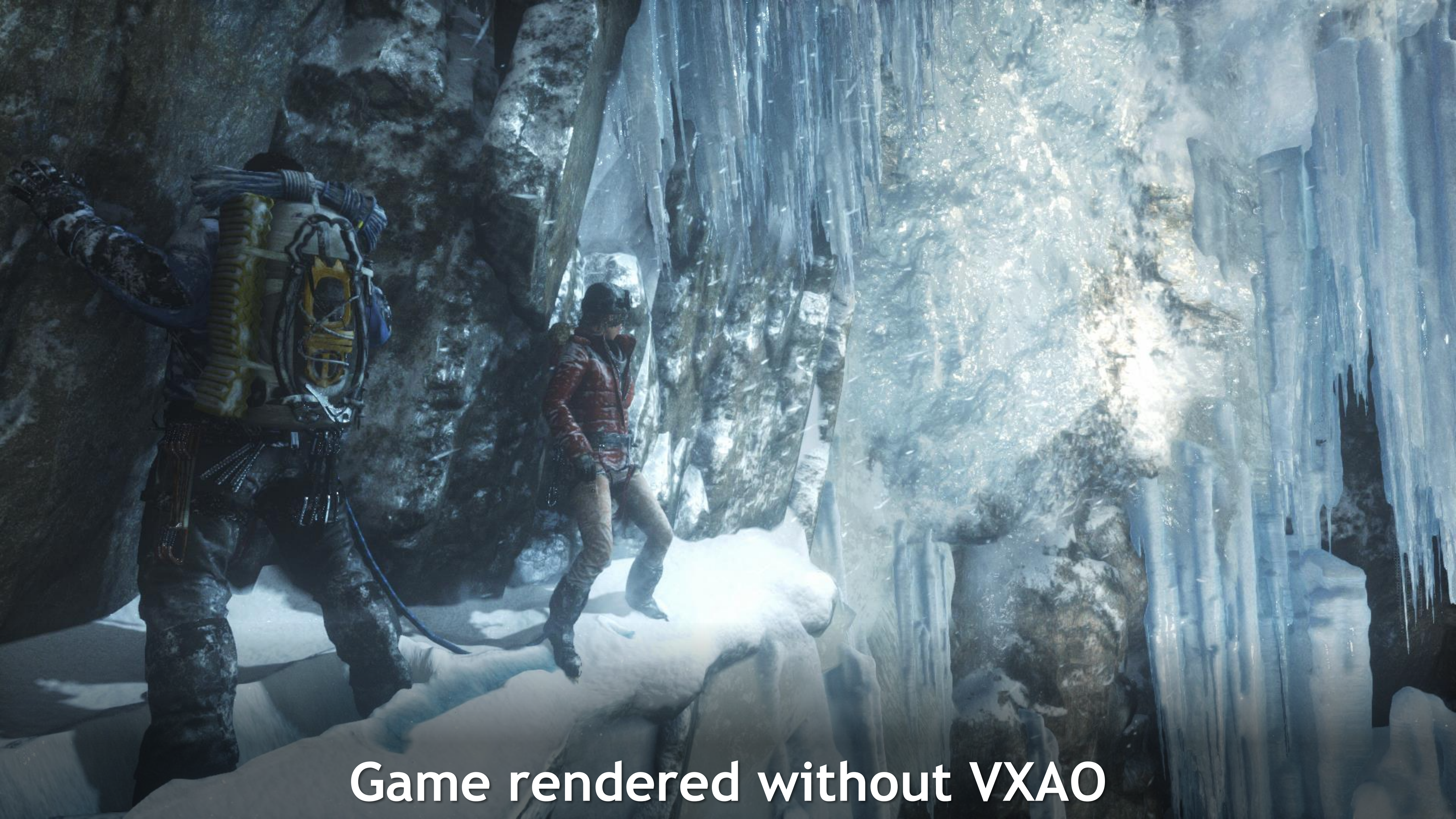
Game rendered without VXAO



Ambient Light channel



Ambient Light channel with VXAO



Game rendered without VXAO

Game rendered with VXAO: very little difference

VXAO Performance in ROTTR

- Heaviest scenes have up to 10 M polygons and over 2000 voxelization draw calls
- Voxelization takes most of the VXAO time, largely depends on the scene
- Overall VXAO time in various scenes [GTX 980, 1920x1200, build 623]:

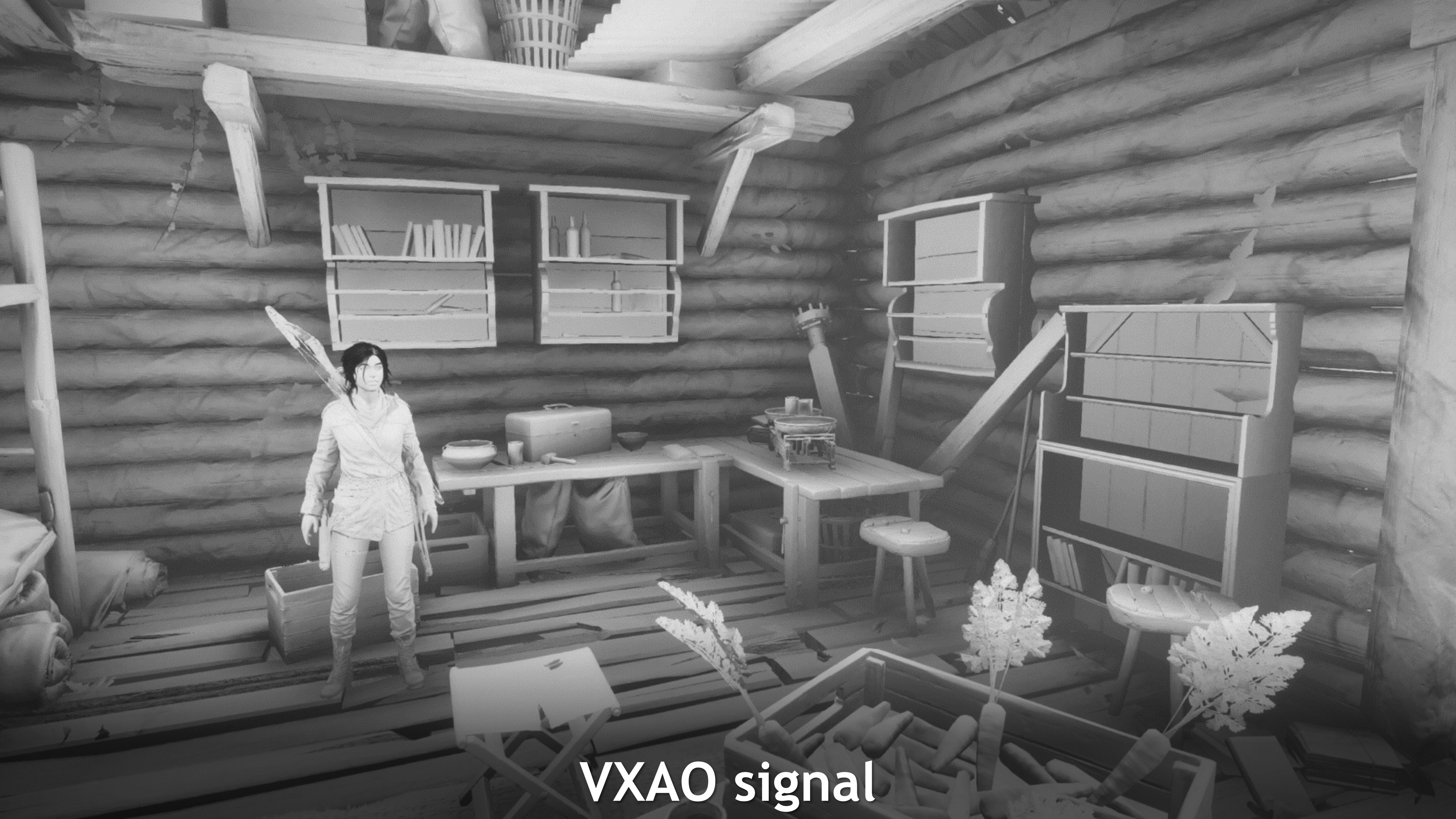
❖ Main Menu:	1.8 ms
❖ Siberian Wilderness:	3.6 - 4.2 ms
❖ The Acropolis:	5.0 - 6.7 ms

- For comparison, HBAO+ time is about 1.3 ms under the same conditions

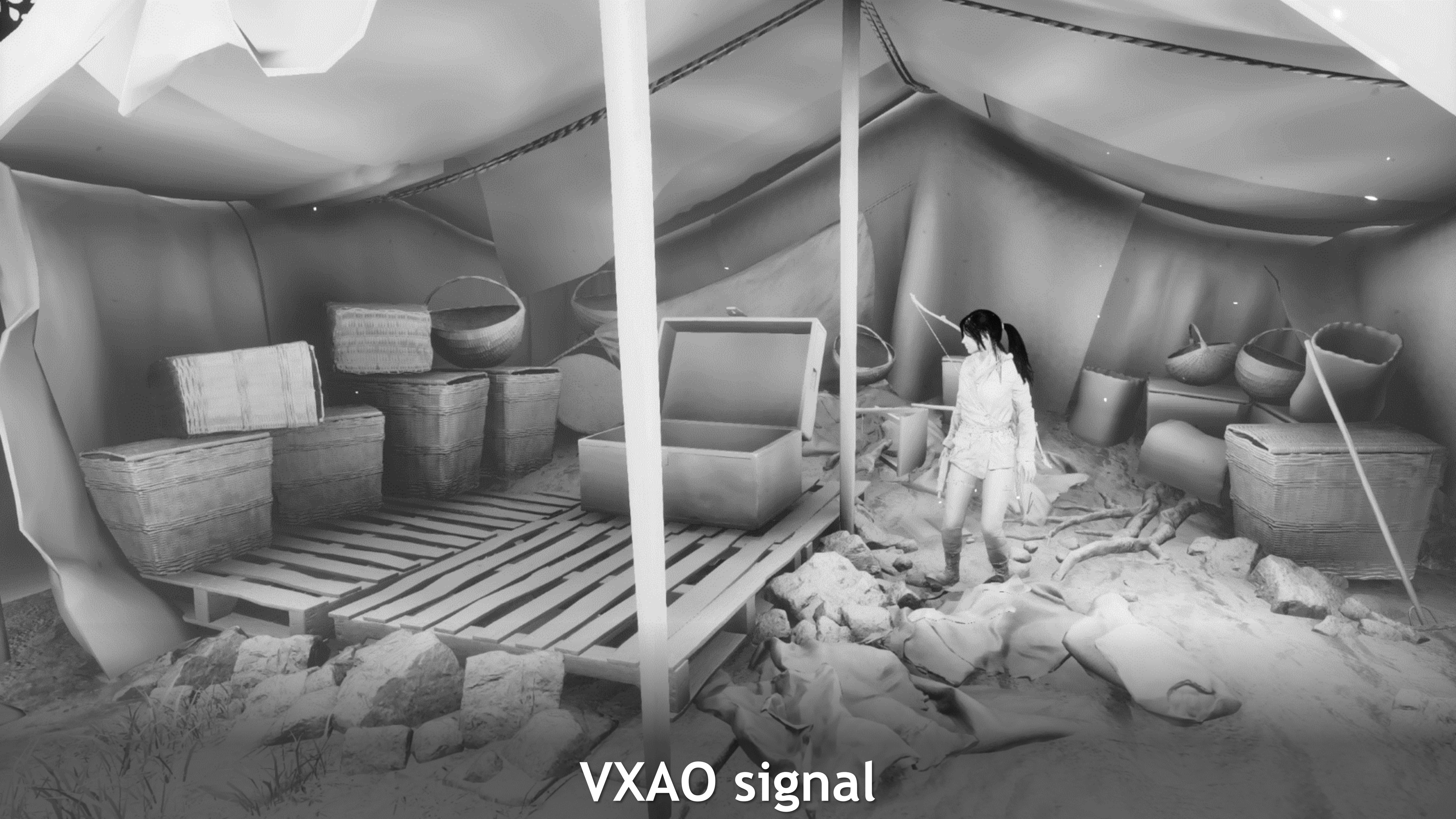
Tuning VXAO Parameters

- Goal: no temporal issues, materials will cover the rest
  - Use low quality tracing settings, won’t be noticeable in the final image
  - Increase tracing offset on Lara to avoid banding when she moves

- Voxelization is the most expensive pass, so make it faster
  - Reduce clip-map range: fewer objects to voxelize, still looks good
  - Skip voxelization of objects smaller than a voxel
  - Use low quality mesh LODs when available



VXAO signal



VXAO signal

Integration Stats

- Integration work started in December 2015
- Performed by one engineer from Nixxes Software with assistance from NVIDIA

- More complicated than HBAO+ integration, but still manageable

- ~100 man-hours of work on Nixxes side
- ~900 VXAO-specific lines of engine code

Integration Takeaways

- Lighting needs to be physically based to highlight the VXAO effect

- VXAO looks best when it's combined with some form of SSAO
  - ❖ VXAO library includes an HBAO-based screen-space AO implementation

- VXAO/VXGI should work through the engine's rendering system
  - ❖ Not using a separate rendering backend that works with D3D
  - ❖ Makes it easier to track state changes during voxelization

References

- HBAO+: https://developer.nvidia.com/shadowworks
- VXGI/VXAO: https://developer.nvidia.com/vxgi
- VXGI in UE4: https://github.com/NvPhysX/UnrealEngine branch VXGI-4.10

- Questions?
  - ❖ atatarinov@nvidia.com
  - ❖ alpanteleev@nvidia.com

NVIDIA