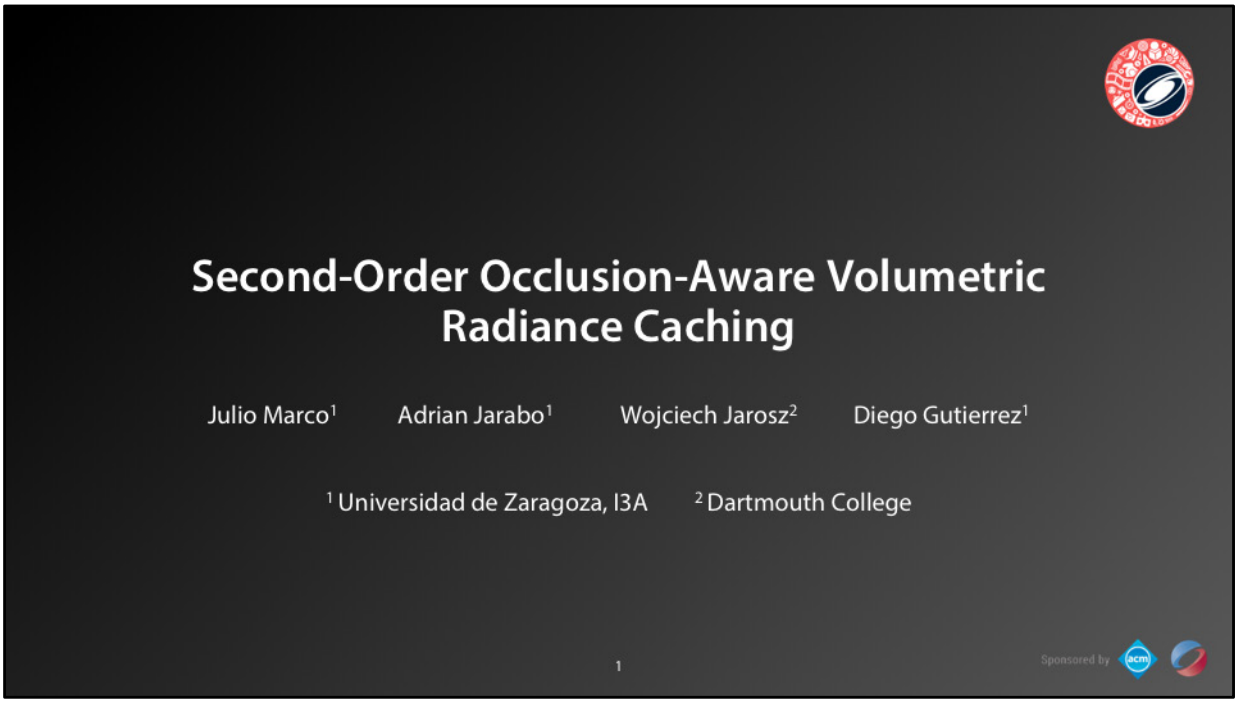
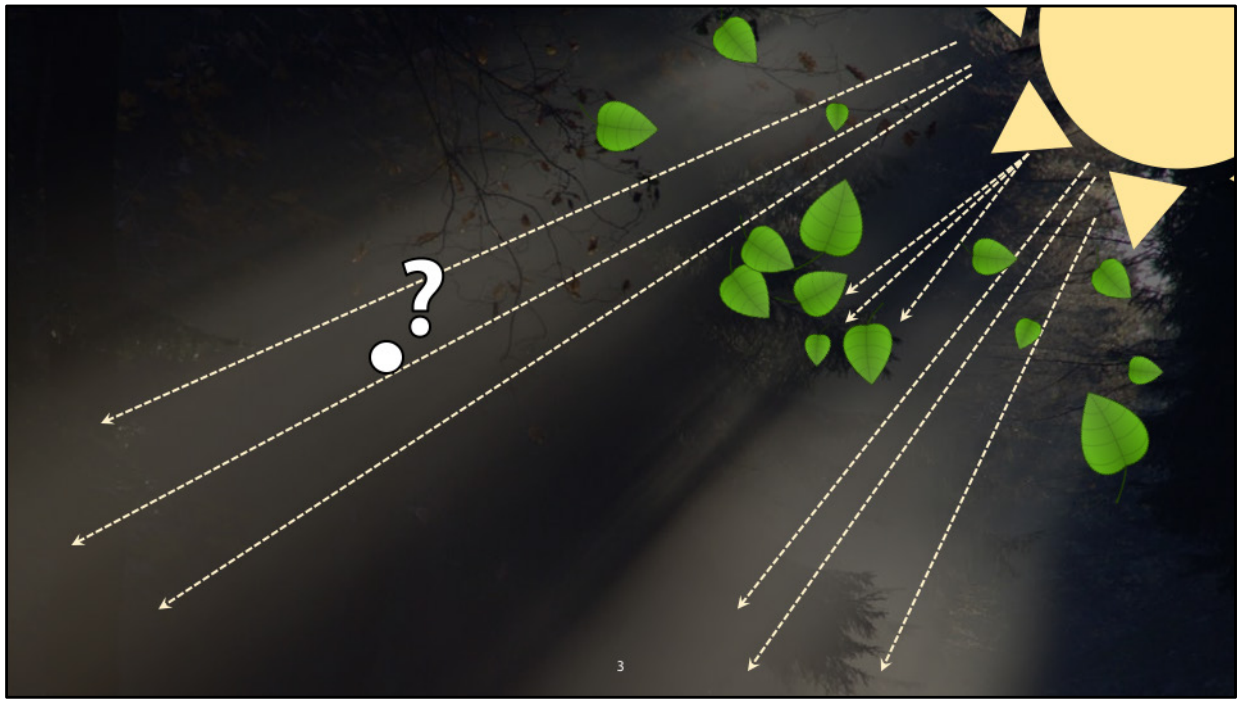


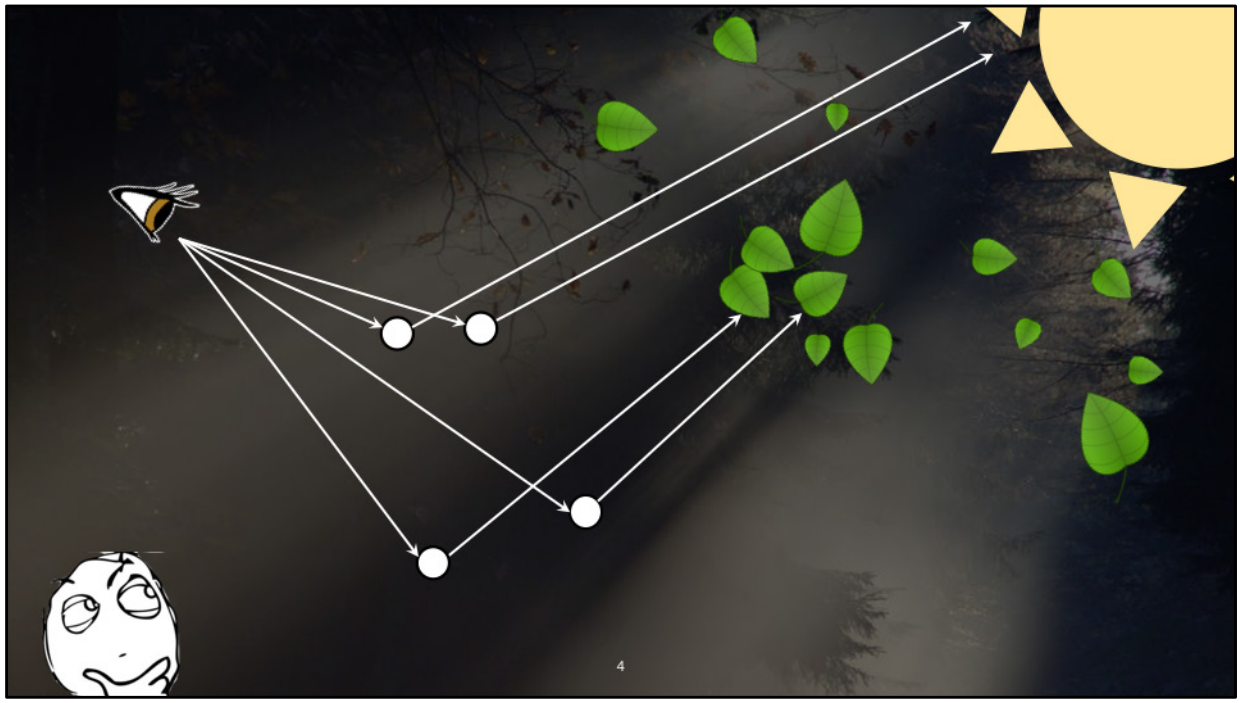
Thanks for the introduction, and thanks everyone for attending

Participating media are responsible of pretty cool lighting effects where geometry can play a big role in how radiance is scattered across the scene.

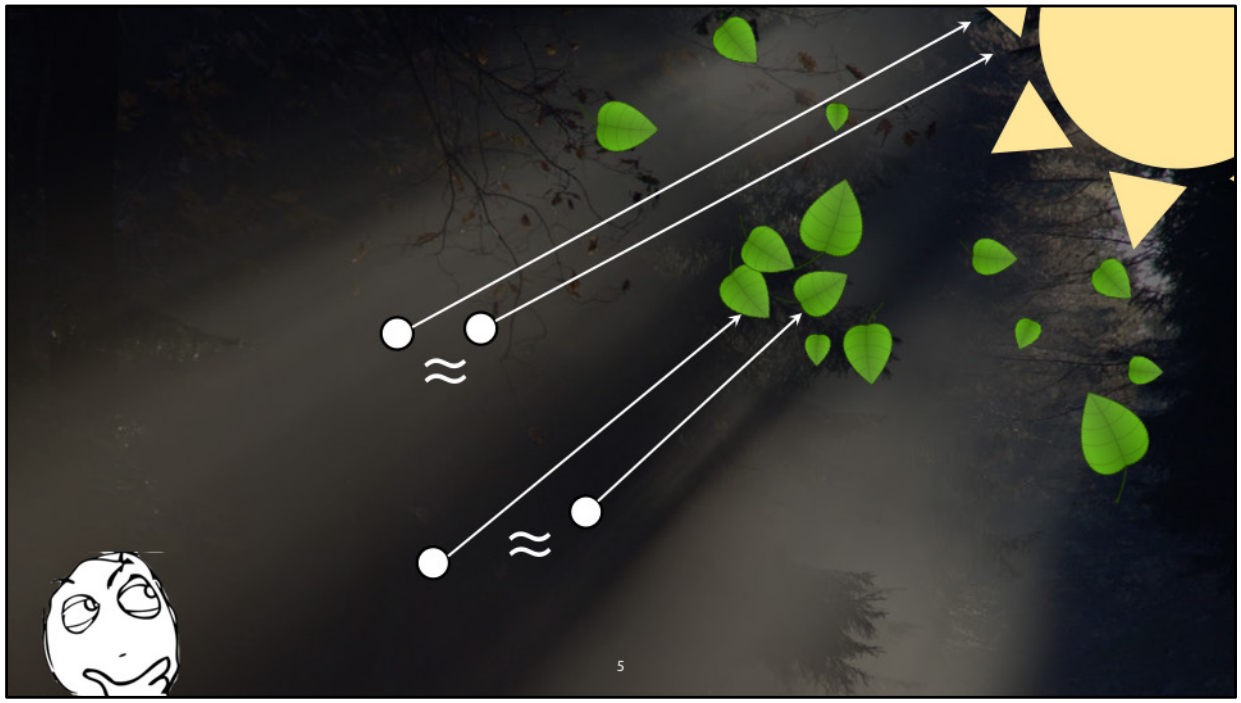
Accurately representing these effects [CLICK]



greatly depends on our ability to represent geometric details during the sampling process, [CLICK] such as the occlusions produced by the leaves of the tree



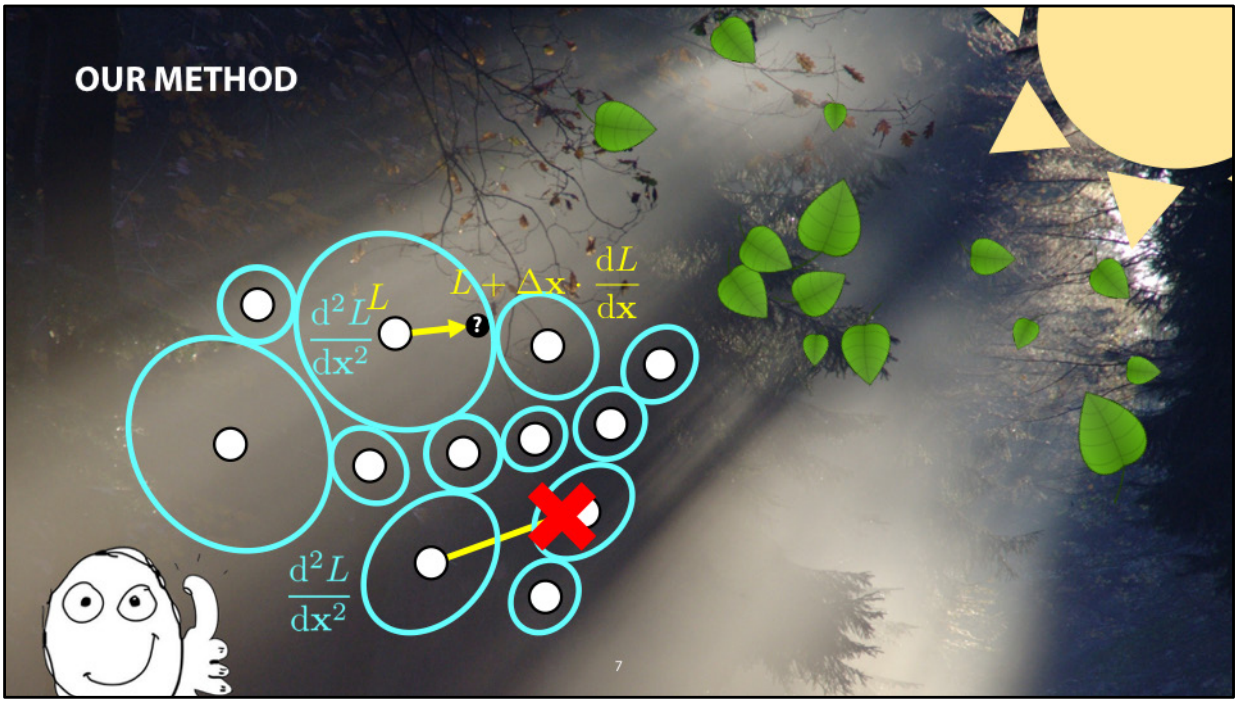
Path tracing based methods are capable of representing many of these effects



but they don’t leverage the smooth radiance variations often found in participating media, leading to many redundant computations

[CLICK]

Inspired by recent work that leverages irradiance Hessians to perform error-bounded extrapolations on surfaces

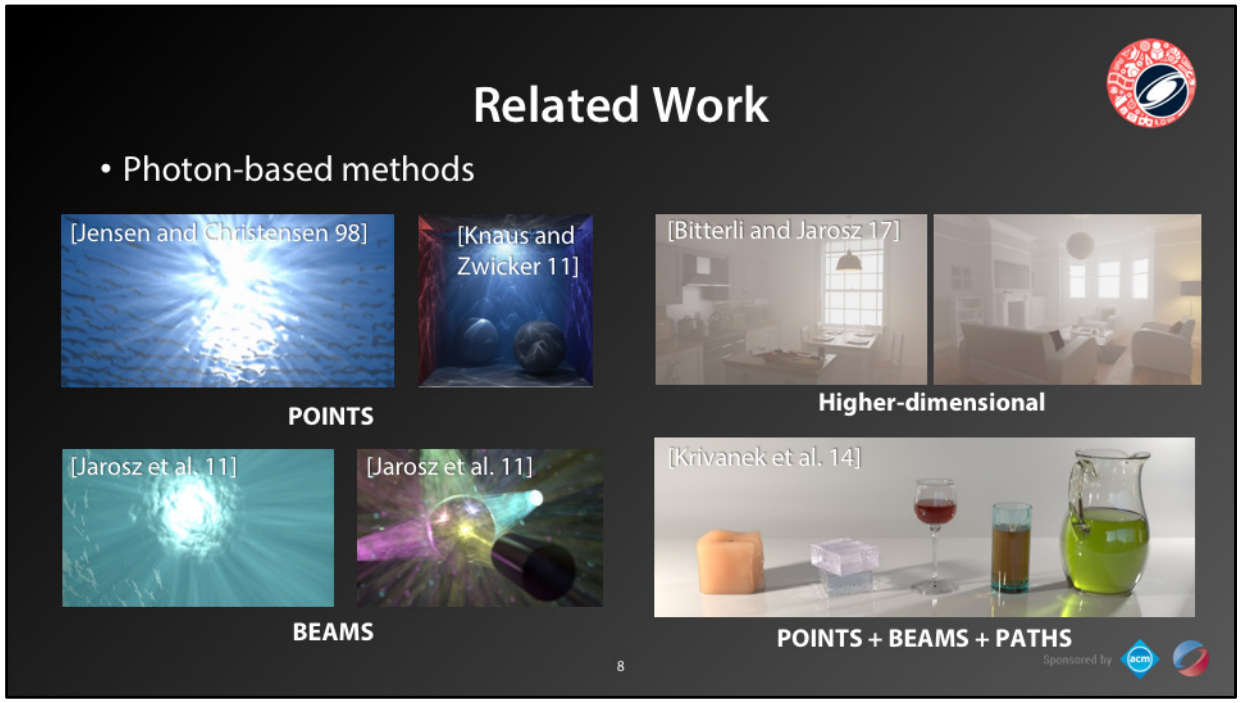


We present a new radiance caching algorithm for participating media that computes occlusion-aware derivatives to improve rendering efficiency

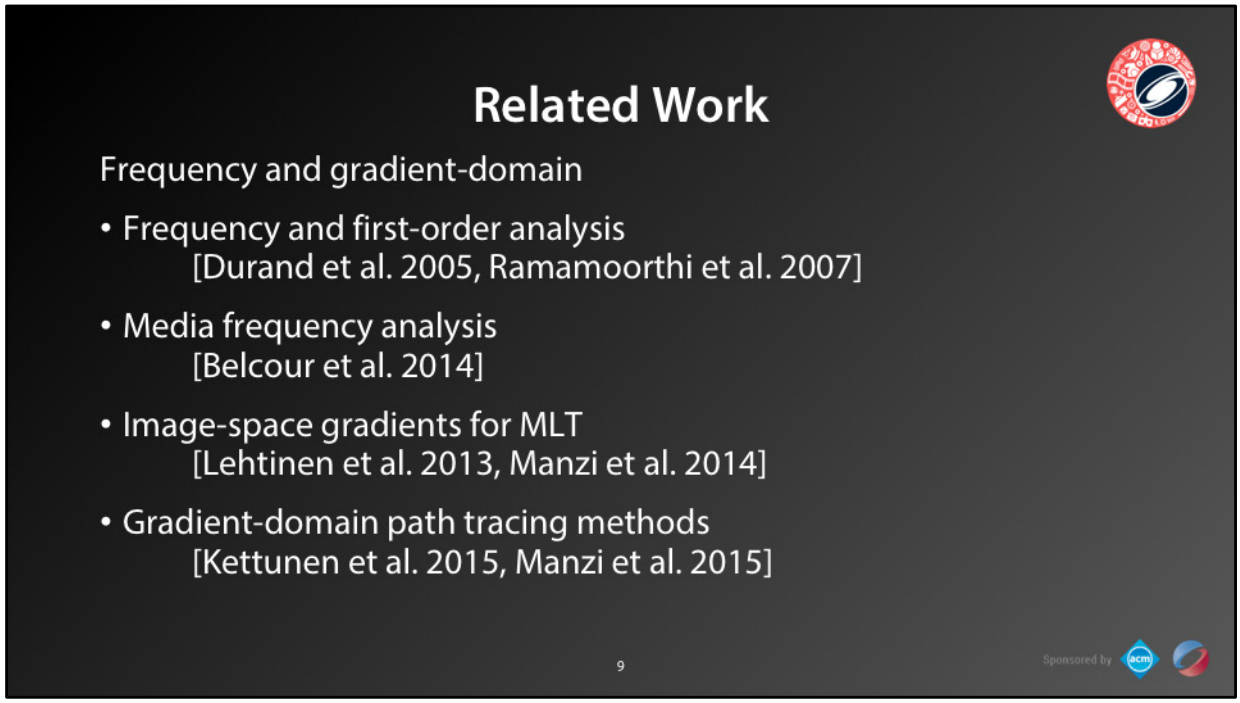
[CLICK] We perform extrapolations across media using first-order derivatives to avoid tracing rays when unnecessary

[CLICK] To make sure we don't extrapolate where we shouldn't, we introduce an error metric based on second-order derivatives [CLICK] that limits the reach of our interpolations, [CLICK] and increase the cache ensity where radiance changes are stronger. [PAUSE]

But before telling how we do this, let's take a look to previous literature

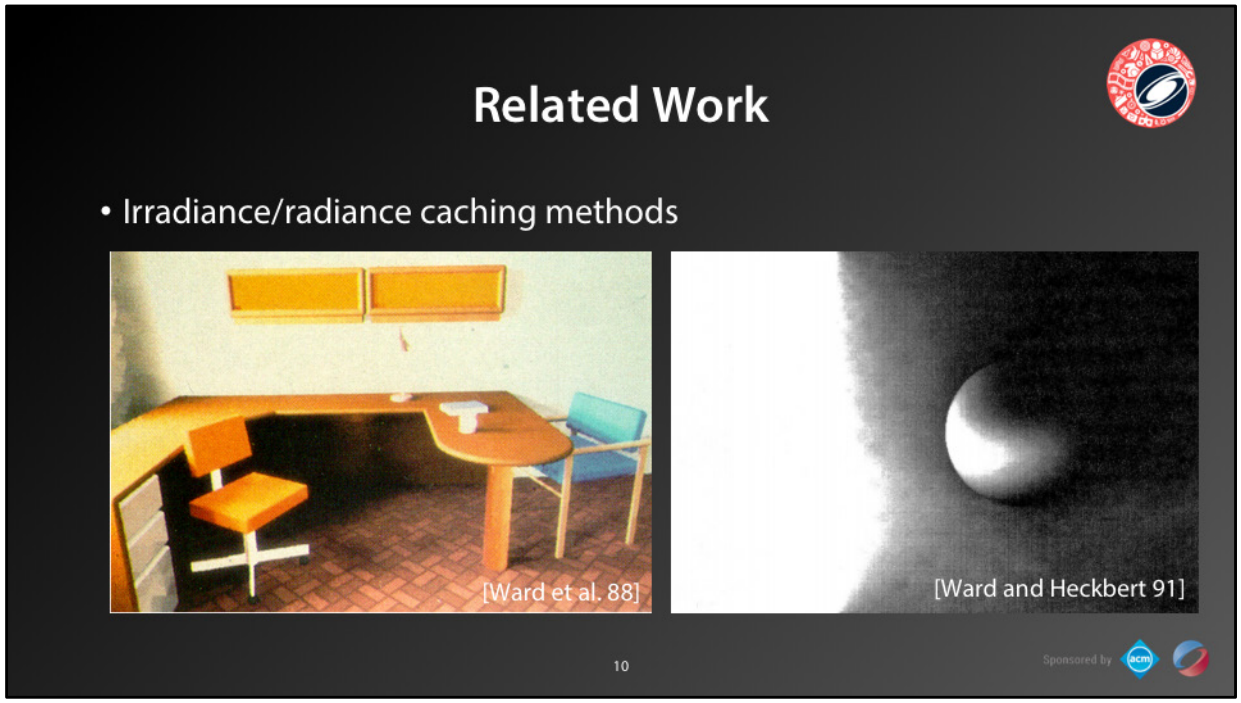


Photon density estimation has fostered a wide number of algorithms for efficient rendering of participating media, either based on [CLICK] points, [CLICK] beams, [CLICK] higher dimensional samples, [CLICK] or even hybrid methods



[CLICK] Frequency and gradient analysis havs also served for adaptive sampling and reconstruction in rendering, or even to improve visual convergence in photon beams.

[CLICK] The use of image-gradients has also helped to improve Metropolis and path tracing methods with Poisson reconstructions in image space



Closer to our work, [CLICK] since the publication of the irradiance caching method in 1988, several works have extended this approach, [CLICK] for example with the inclusion of gradients during extrapolation

Related Work

- Irradiance/radiance caching methods

11

Sponsored by

Some methods have generalized to more complex scenarios by including of neighbor clamping, adaptive sampling, [CLICK] or support for glossy materials.

Related Work

- Irradiance/radiance caching methods

[Jarosz et al. 2008]

- Media gradients → **YES**
- Occlusions → **NO**
- Higher-order → **NO**

12

Sponsored by

And at the core of our work, the first method for radiance caching in participating media was introduced in 2011. [CLICK] This method leverages radiance gradients to adaptively sample and extrapolate radiance in both homogeneous and heterogeneous media.

However, it presents two main limitations:

[CLICK] First, it doesn't account for visibility during gradient computation. This can lead to erroneous interpolations of media radiance when geometry produces significant occlusions

[CLICK] It also ignores higher order derivatives. As we will show later in this presentation, this leads to suboptimal distributions of cached points in the scene.

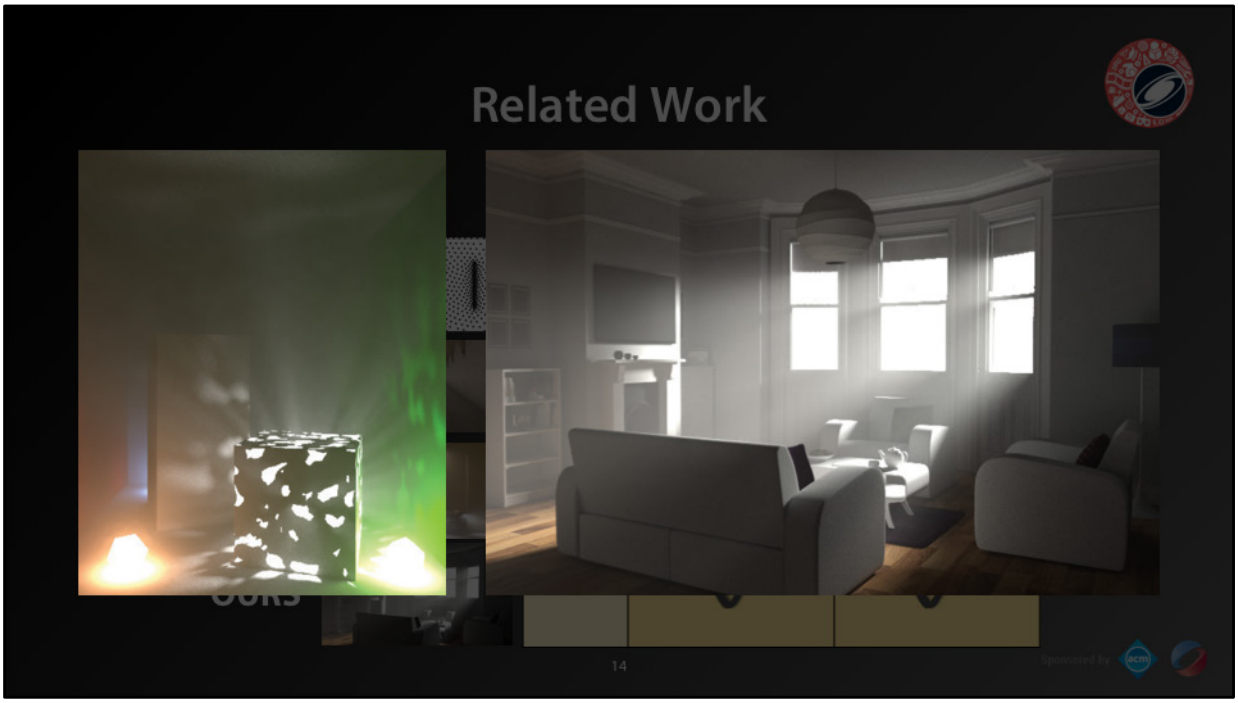
Related Work

[Jarosz et al. 2008]

13

Sponsored by

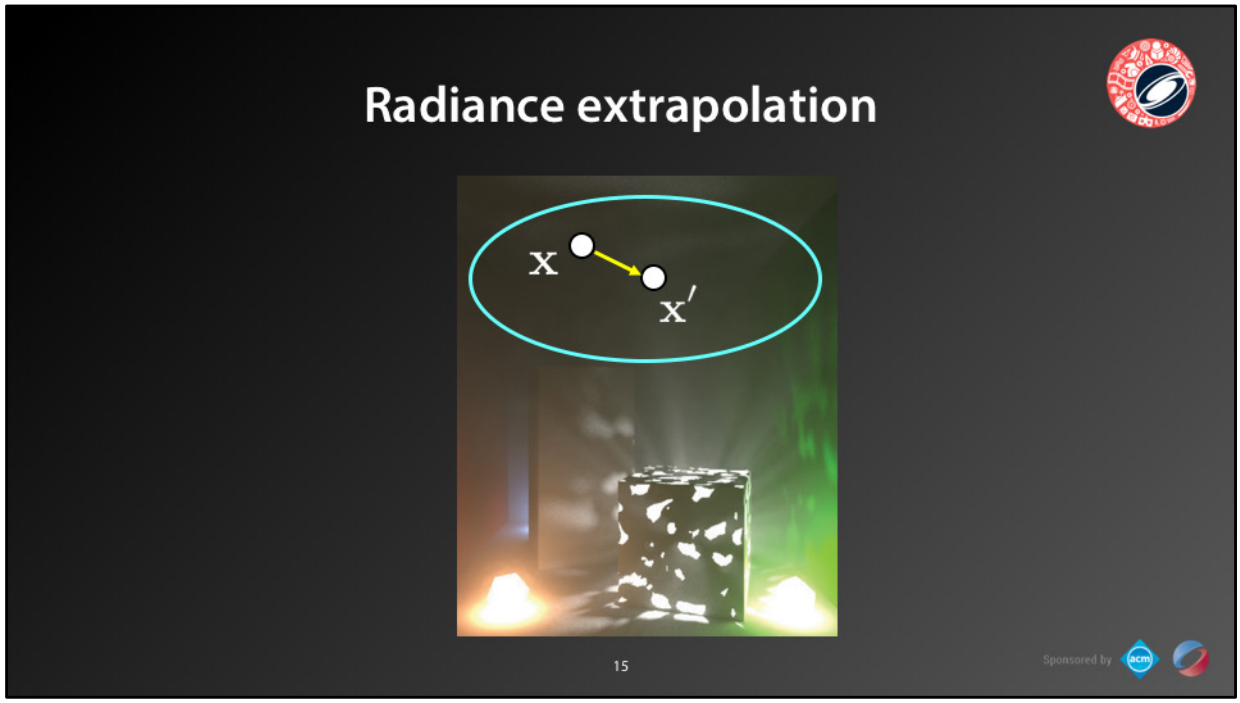
[CLICK]



Several works have improved irradiance caching using second-order error metrics, [CLICK] and proposing practical modifications to support occlusions, but only applicable to surface irradiance.

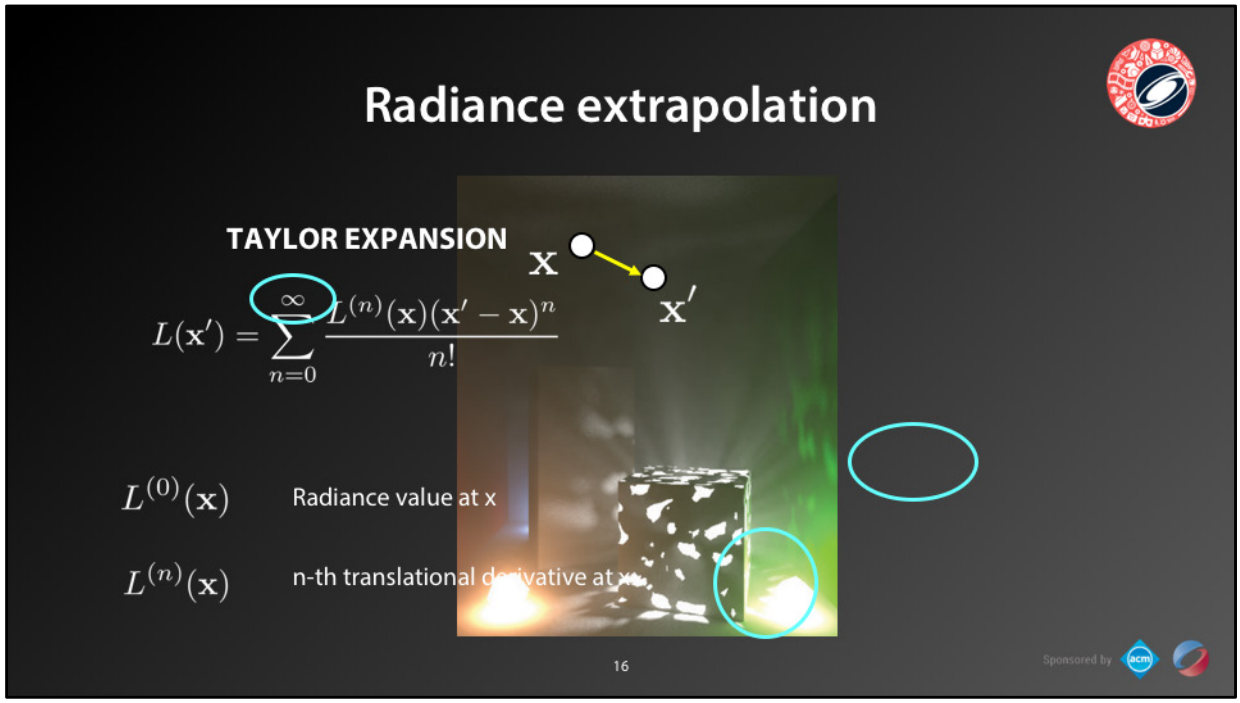
[CLICK] Inspired by surfaces methods, in our work we extend radiance caching to support second-order derivatives of media radiance that account for visibility changes [CLICK] in both single and multiple scattering.

Let's see how we do this [CLICK]



Participating media usually contains regions with very low frequency content [CLICK] due to smooth variations of radiance.

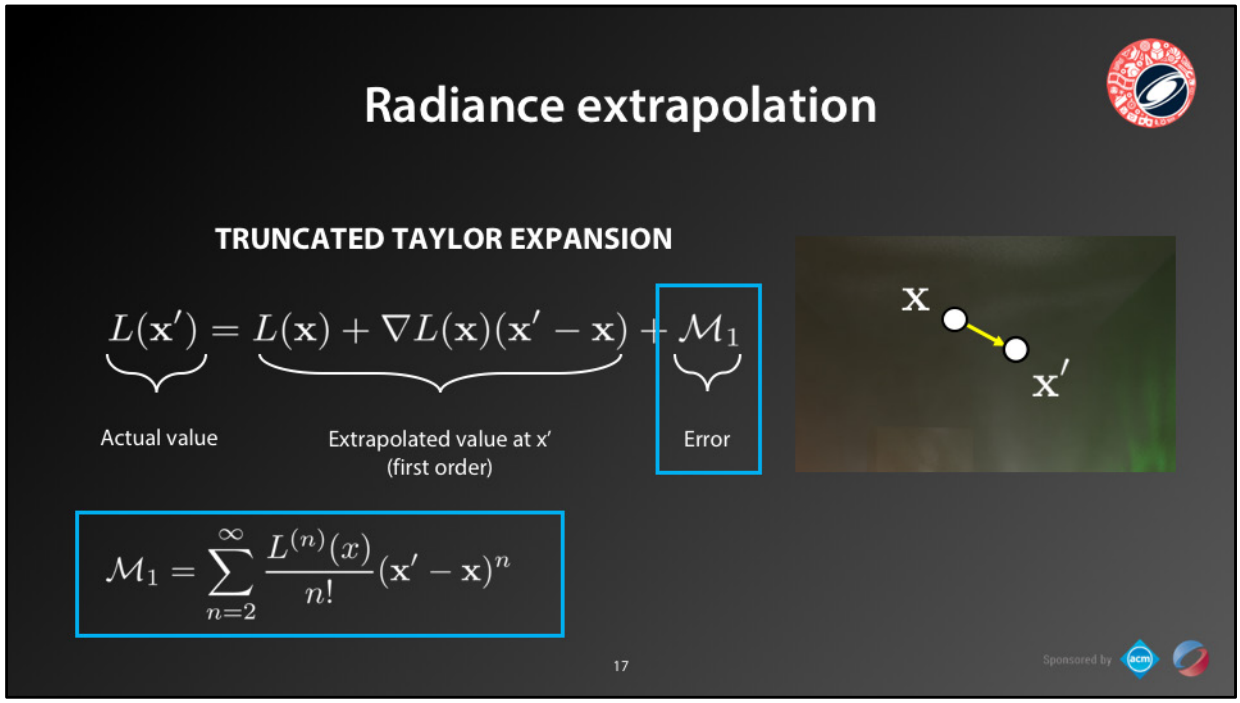
This makes extrapolation very convenient, where sparsely shaded points [CLICK] can be used to predict radiance at nearby locations. This way we avoid path tracing radiance at every shaded point



How is this radiance extrapolated? Ideally, if we knew all translational derivatives at a shaded point, we could compute the radiance at any other point by using a Taylor expansion.

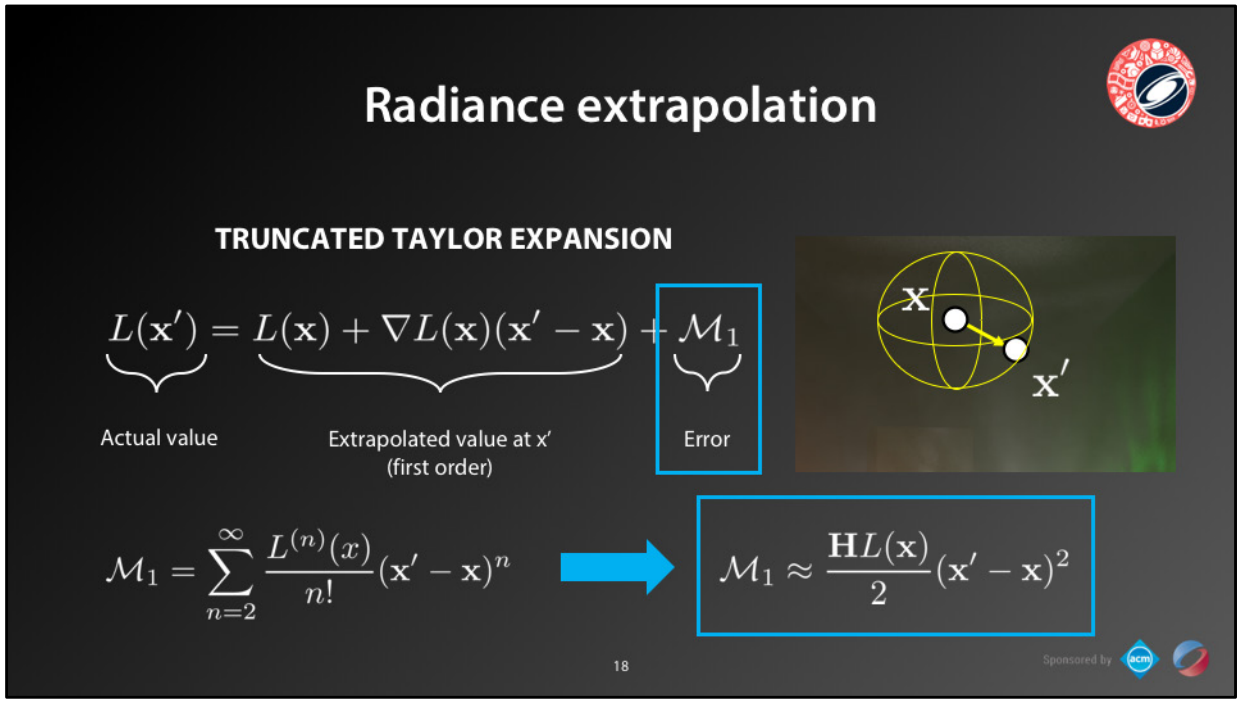
In this Taylor expansion, [CLICK] the zero order derivative is the actual radiance value at the source point, and the remaining terms are the translational derivatives.

However, [CLICK] arbitrary media and geometry make it very challenging to compute accurate derivatives, and we would need an [CLICK] infinite number of them to compute a perfect radiance value, which is unfeasible



To make extrapolation practical, irradiance caching methods in both surfaces and media rely in performing first-order extrapolations, and allow certain error [CLICK] to slip through the image.

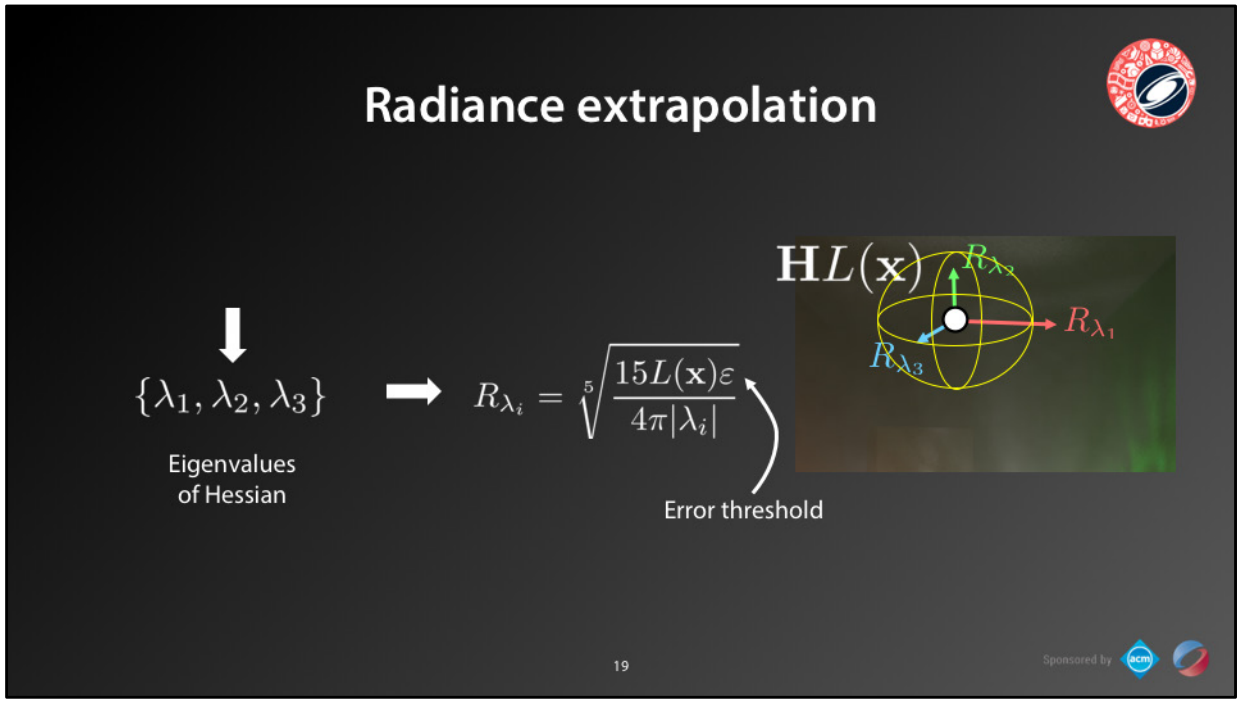
[CLICK] The exact error introduced is determined by the remaining terms of the Taylor expansion, but as we said it's unfeasible to compute all of them.



One practical approach that produced good results in the case of irradiance caching for surfaces, is to use second-order derivatives as an oracle of this error.

Under the assumption that they are that a good estimator of the error of a 1st-order expansion, [CLICK] they can be used to define a bounding region is valid for extrapolation.

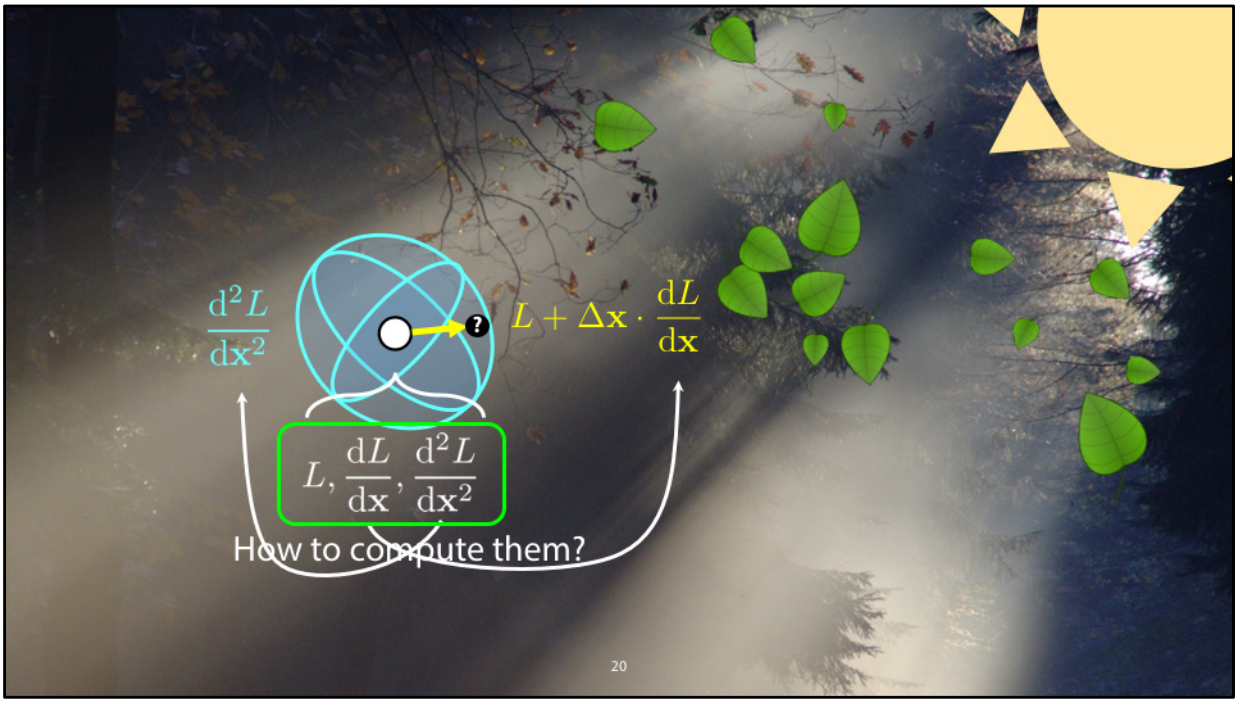
In the case of participating media this bounding region is three dimensional.



Thanks to the eigendecompositon of the Hessian [CLICK], we stretch the region along the axis where radiance gradients are smoother, resulting in an ellipsoidal cache point.

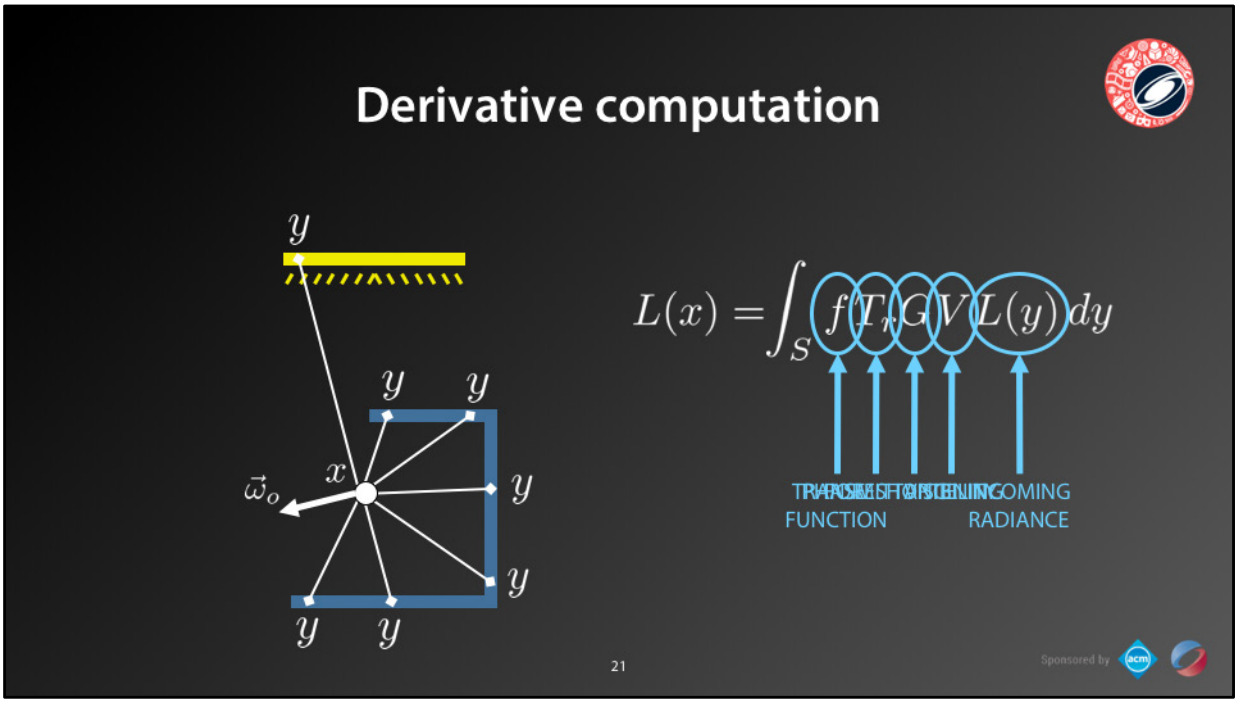
By chosing certain error threshold [CLICK], we can determine the size of this ellipsoid. This will guarantee that the integrated error from extrapolation is under that value.

For the full derivation of the error-driven bounding regions please refer to the paper



So we know how to use derivatives to perform error-bounded extrapolations in a medium: [CLICK] the gradient extrapolates radiance to nearby points, and [CLICK] the Hessian guarantees certain error threshold over an ellipsoidal region of medium.

For that purpose we need to compute derivatives that correctly predict the radiance changes at the sampled cache point.

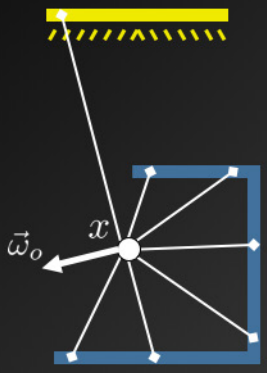


To illustrate it, we can observe the following 2D diagram with an area light source on the top, some geometry in blue, and the point at which we want to compute radiance derivatives

Following a geometric formulation, scattered radiance from surfaces at a medium point corresponds to the [CLICK] integral across all points in light sources and geometry. This integral accounts for [CLICK] the phase function, [CLICK] transmittance, [CLICK] foreshortening, [CLICK] visibility, [CLICK] and the incoming radiance

Derivative computation

[Jarosz et al. 2008]



$$L(x) = \int_S f T_r G V L(y)\, dy$$

Ignored in gradient computation

22

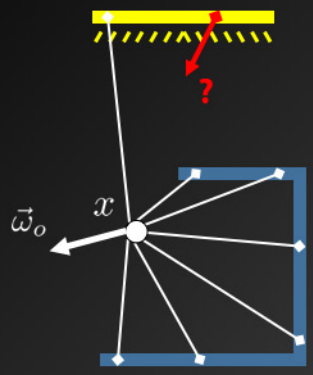
Sponsored by

Previous radiance caching method for participating media assumes point-to-point derivatives, [CLICK] and ignores the visibility term during gradient computation

This implies that, when we extrapolate radiance [CLICK] at a nearby points, radiance changes due to occluders are not accounted for

Derivative computation

[Jarosz et al. 2008]



$$L(x) = \int_S f T_r G V L(y)\, dy$$

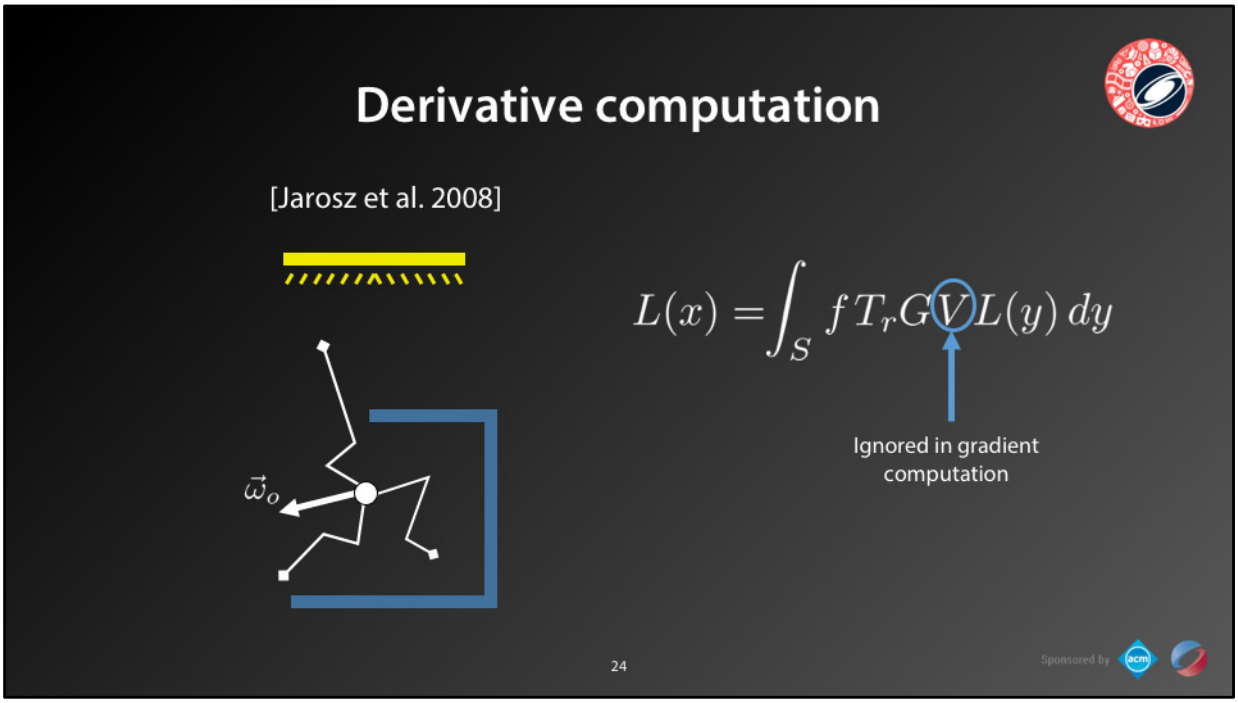
Ignored in gradient computation

23

Sponsored by

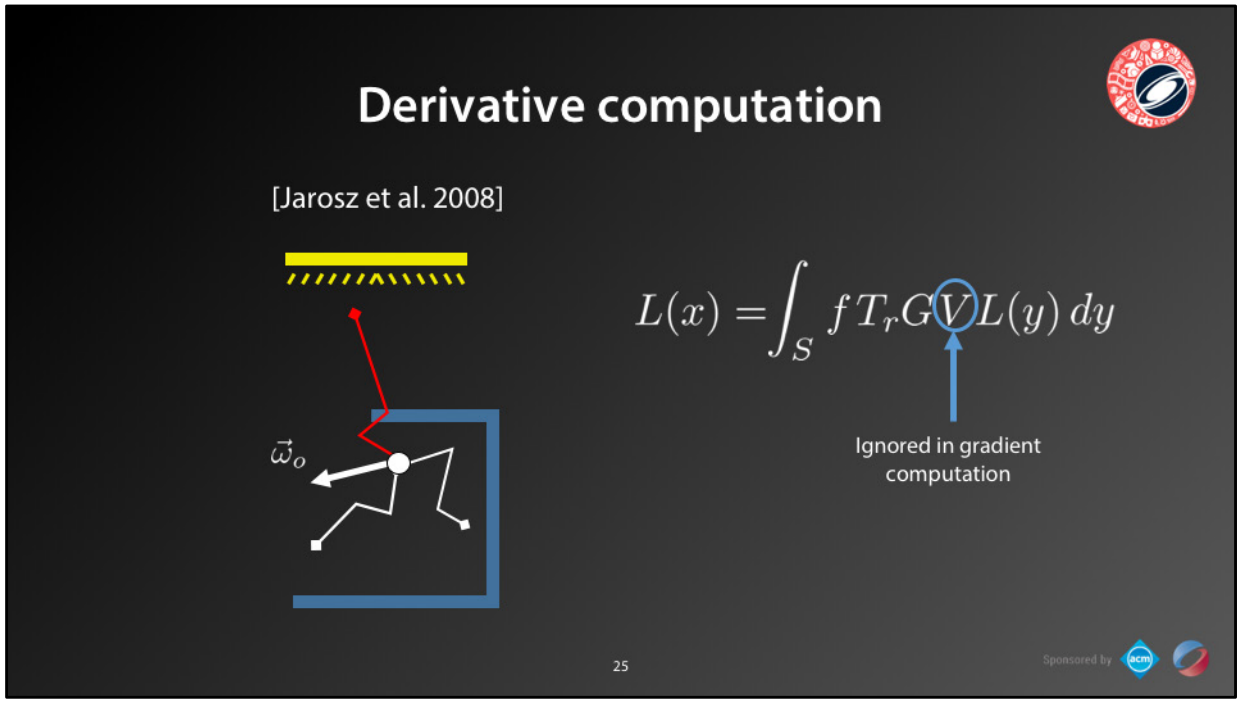
Previous radiance caching method for participating media assumes point-to-point derivatives, [CLICK] and ignores the visibility term during gradient computation

This implies that, when we extrapolate radiance [CLICK] at a nearby points, radiance changes due to occluders are not accounted for



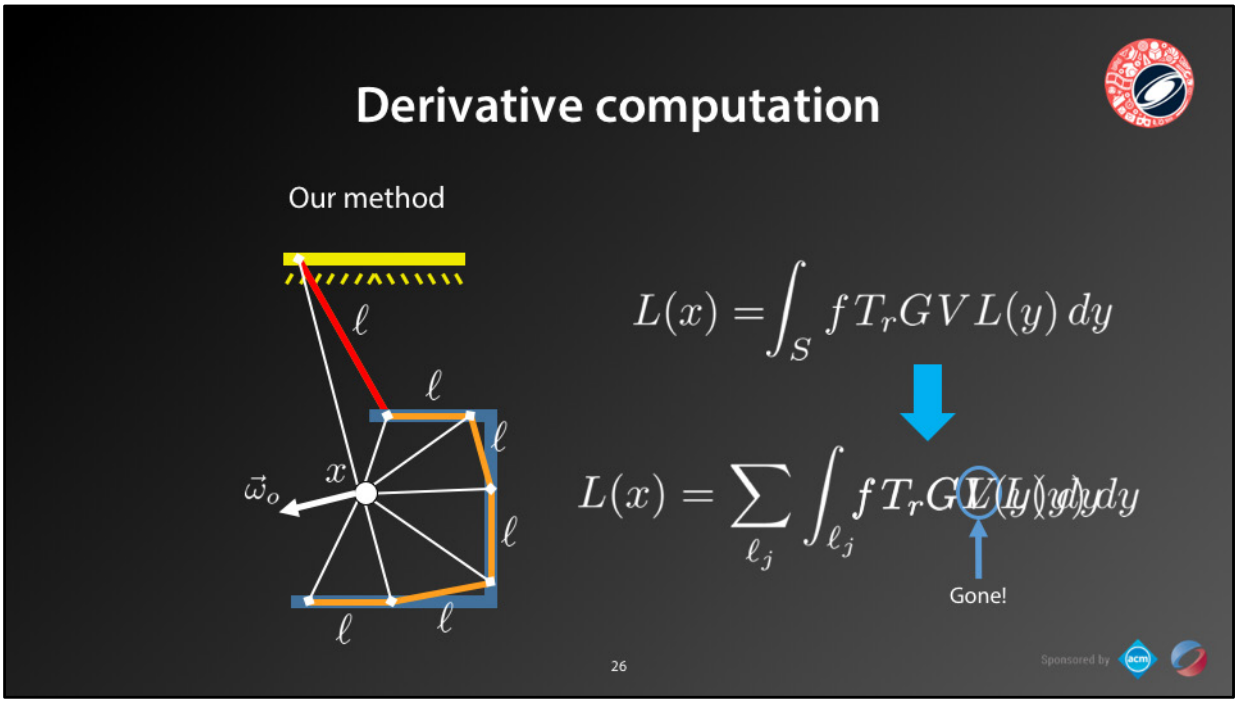
This can also occur in multiple scattering, where for example a path originally contributing to the shaded point [CLICK] gets occluded during extrapolation, but it's not detected by the point-to-point approach

As we will illustrate later in the results section, this leads to erroneous extrapolated values producing noticeable artifacts.



This can also occur in multiple scattering, where for example a path originally contributing to the shaded point [CLICK] gets occluded during extrapolation, but it's not detected by the point-to-point approach

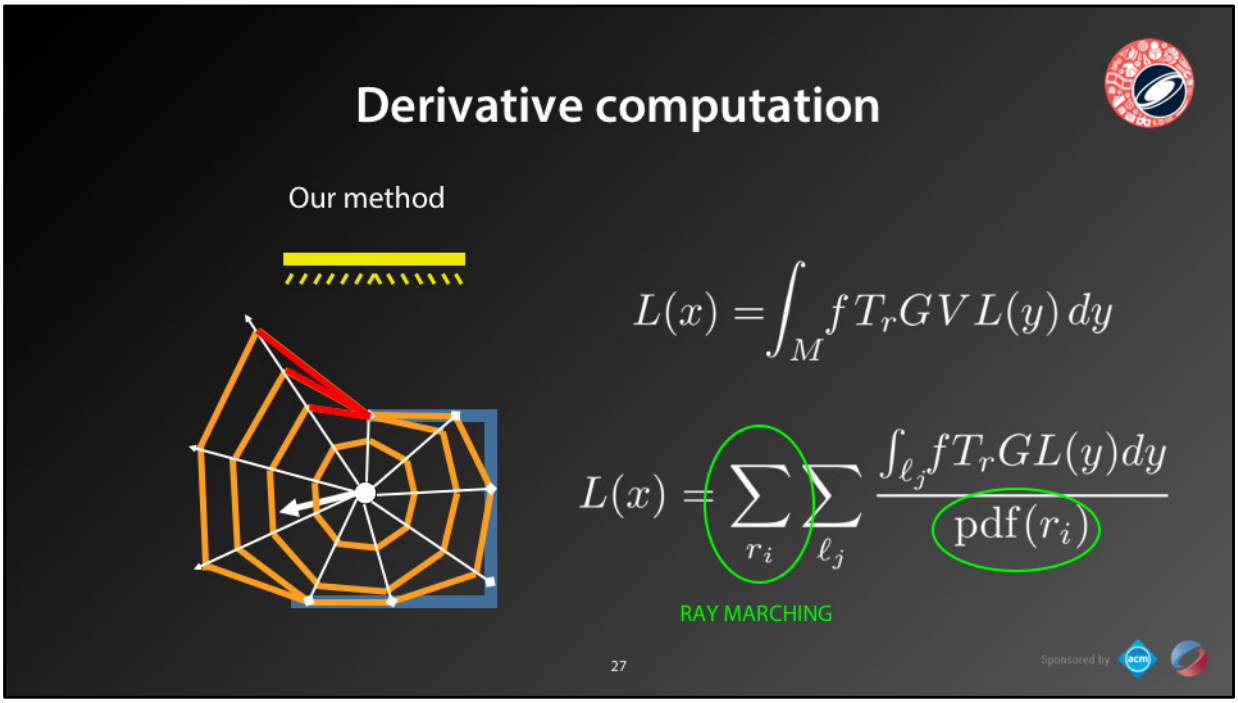
As we will illustrate later in the results section, this leads to erroneous extrapolated values producing noticeable artifacts.



As a solution for this, we propose an alternative approach to account for visibility changes

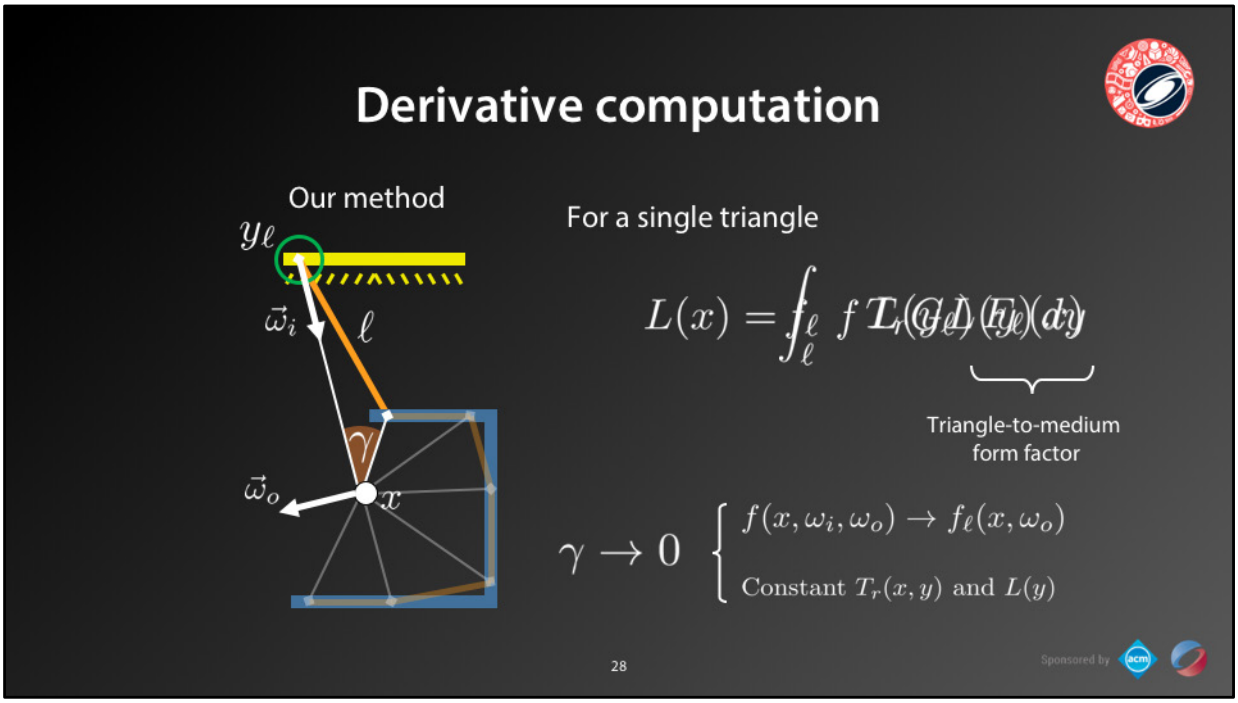
[CLICK] We can use adjacent points during angular sampling to build a virtual triangulation of the surrounding geometry. In this triangulation, [CLICK] potential visibility changes are represented by slanted triangles, as the one seen as a red segment in the 2D example.

[CLICK] Now the integral can be estimated as a piece-wise sum of the integrals of each subdivision. This triangulation is an occlusion-free approximation of the scene as seen by the cached point, [CLICK] and the visibility term can be dropped from the integral.



For multiple scattering, we follow an analogous approach by triangulating media at ray-marched steps [CLICKSSSS].

[CLICK]
Media-to-media visibility changes are now represented by triangles with some vertices lying on geometry found during the ray-marching process



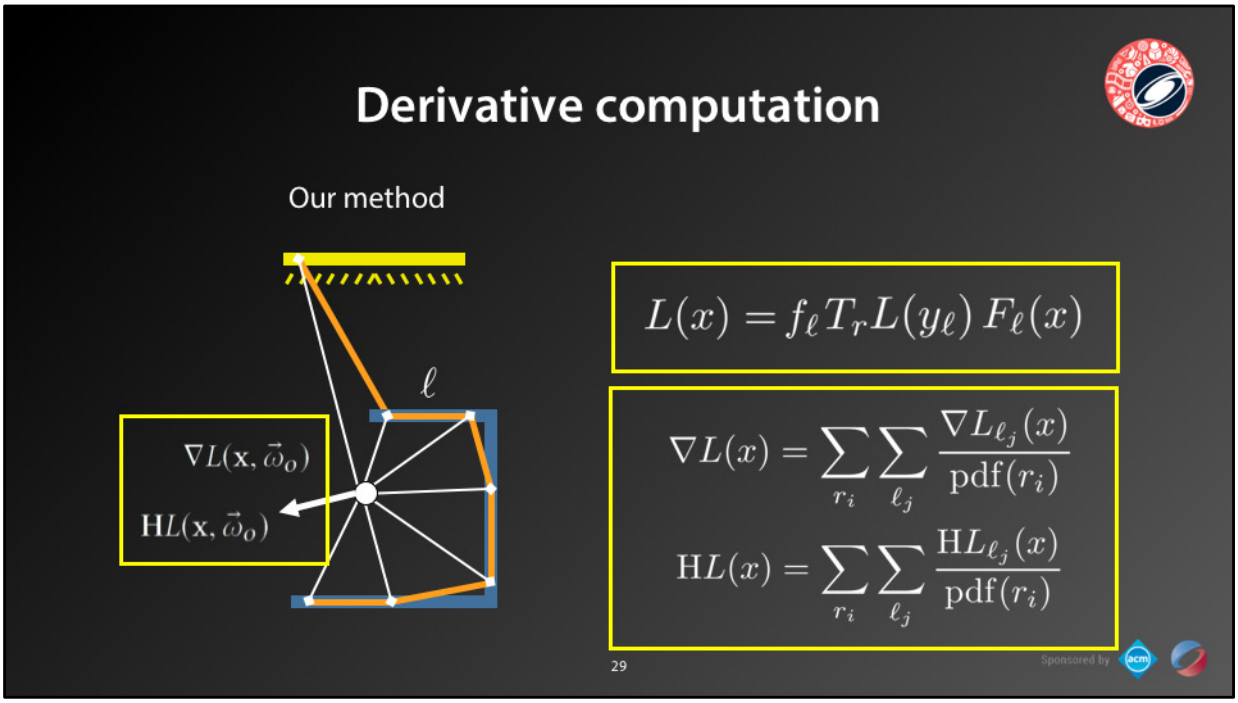
This visibility-free triangulation still requires integrating the radiance coming from all their points.

To alleviate this numerical integration, we introduce the following assumptions:

For a sufficiently fine triangulation [CLICK], incoming direction from a triangle barely changes, and we can assume the phase function is constant [CLICK]

We assume that transmittance and radiance of a subdivision correspond to the furthest point in the triangle [CLICK], which will be the first to change visibility during interpolation. [CLICK] This way, only the geometric term remains within the integral.

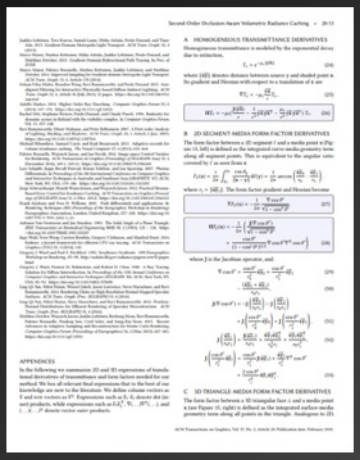
For each subdivision this integral has closed-form solution [CLICK] as the triangle-to-media [CLICK] form factor

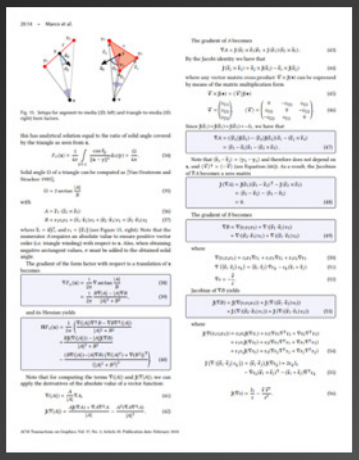


This formulation provides a closed-form solution for every triangle in the subdivision, both in single and multiple scattering

In order to compute the derivatives at the cached point, [CLICK] we can differentiate this expression at every triangle both in single and multiple scattering subdivisions, and sum all the terms. This has a closed form solution by applying the product rule.

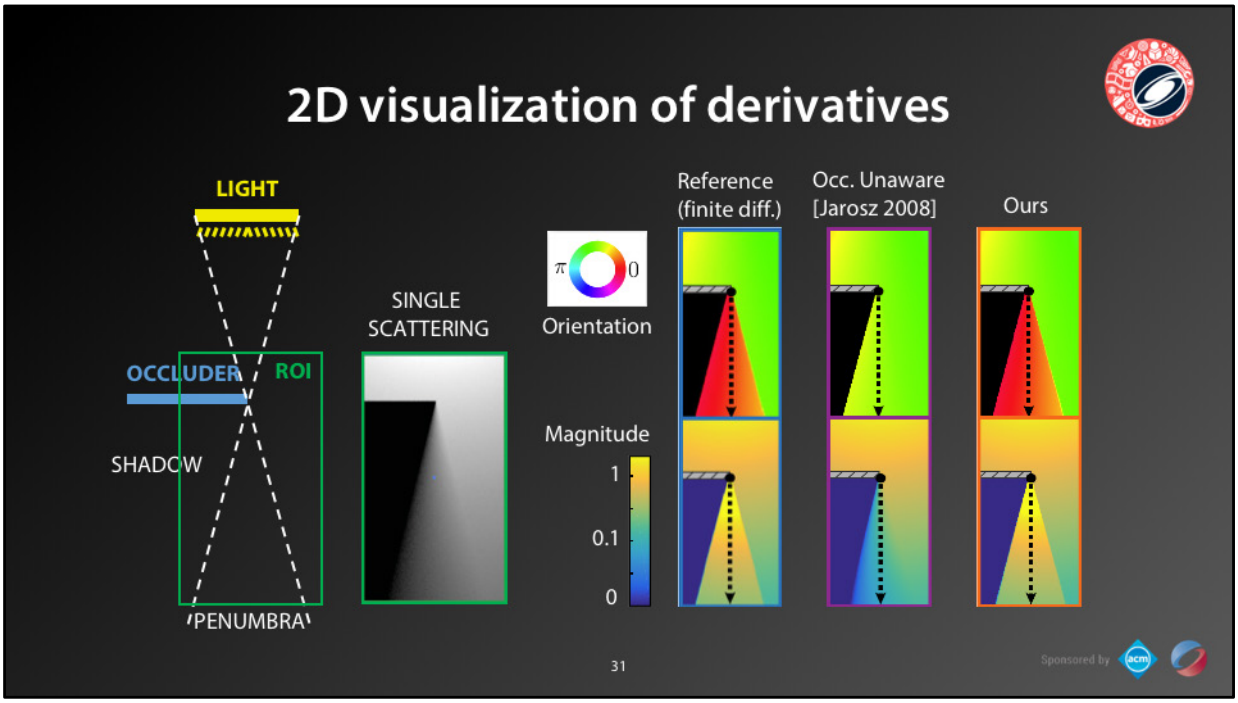
Derivative computation





Sponsored by

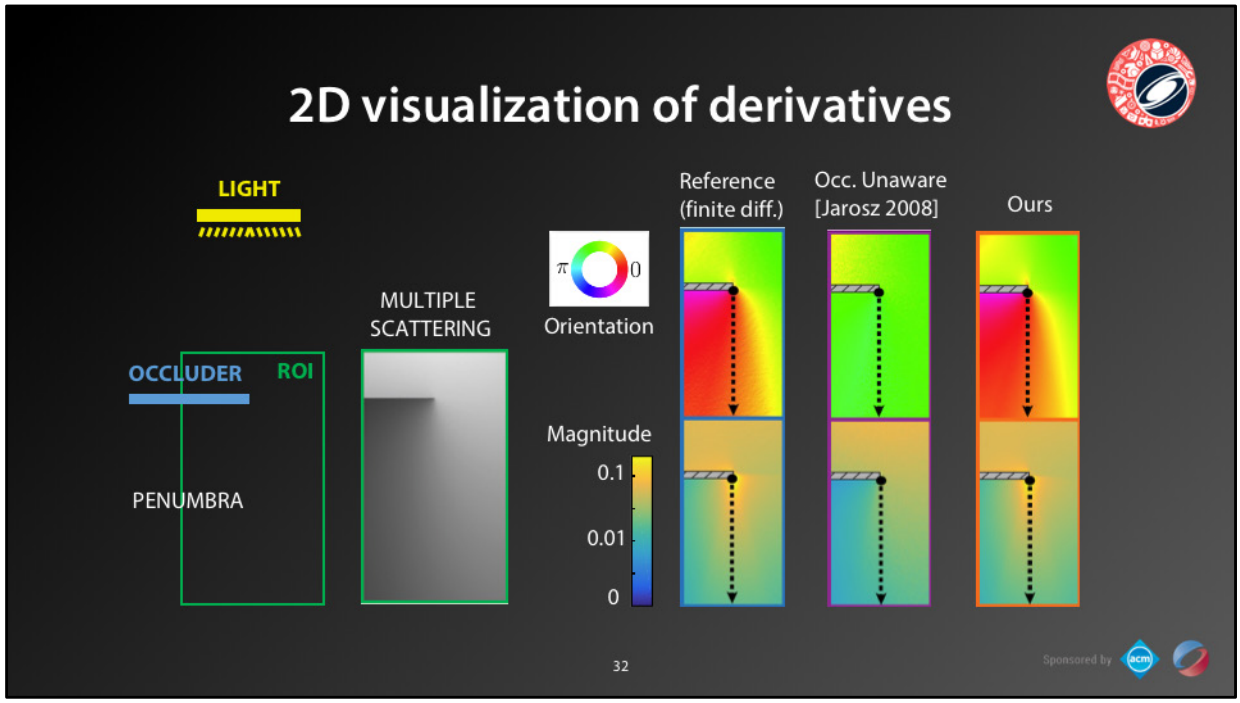
For the full expressions of all the terms in the radiance derivatives, please refer to the paper



To illustrate the improvements of our derivative computation, we choose a toy 2D scene with an area light on top and an occluder below, showing single scattering.

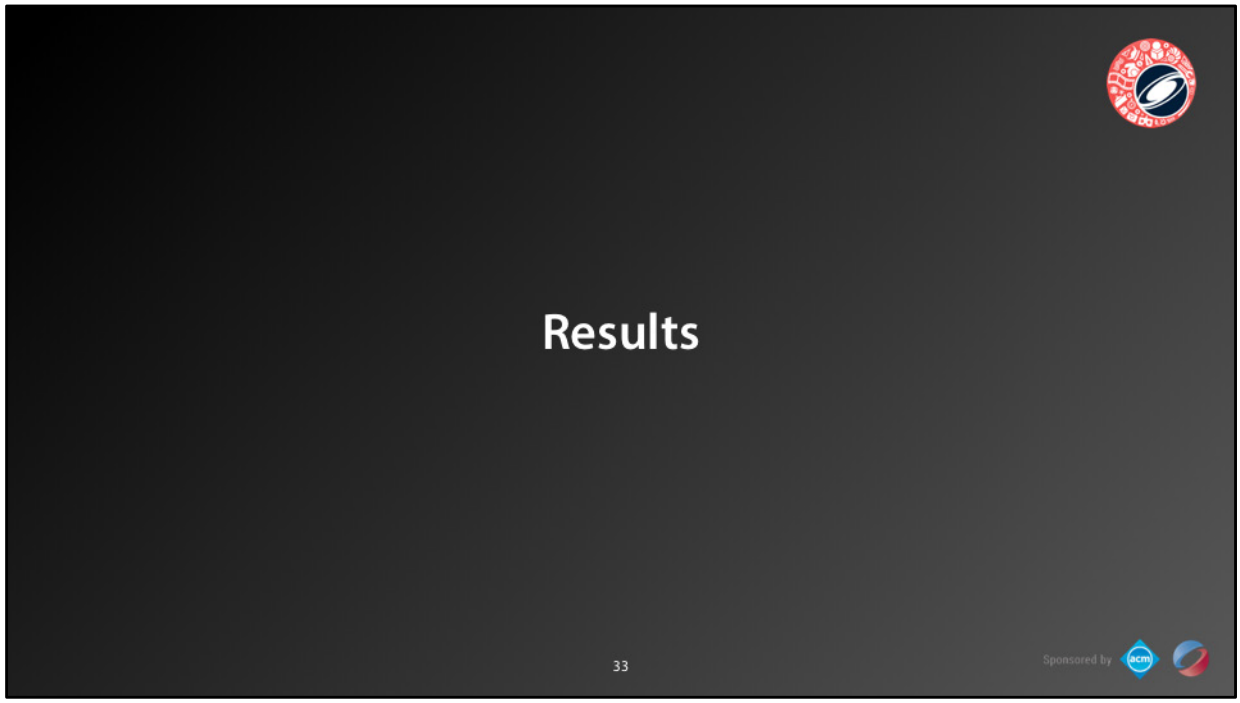
[CLICK] We show the reference gradients computed with finite differences, represented with orientation at the top row and magnitude at the bottom. The penumbra region has stronger gradients near the occluder, oriented towards the right, away from the occluder.

[CLICK] Previous work gradients fail to capture these changes on the penumbra zone, [CLICK] while our occlusion-aware method correctly computes gradients across all medium

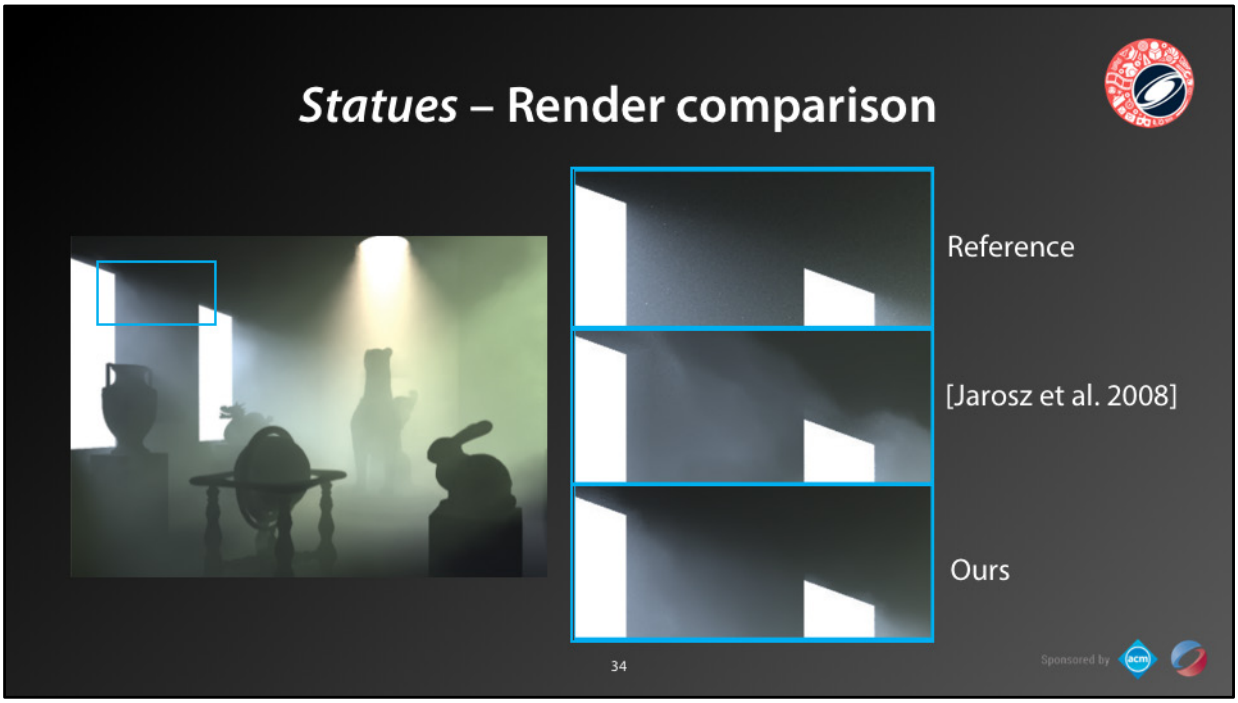


We show also the gradient comparison for multiple scattering, where the penumbra and shadow areas are not well defined anymore.

Again, our method is able to compute the the correct gradients pointing away from the occluder in the penumbra zone, where occlusion-unaware methods fail.

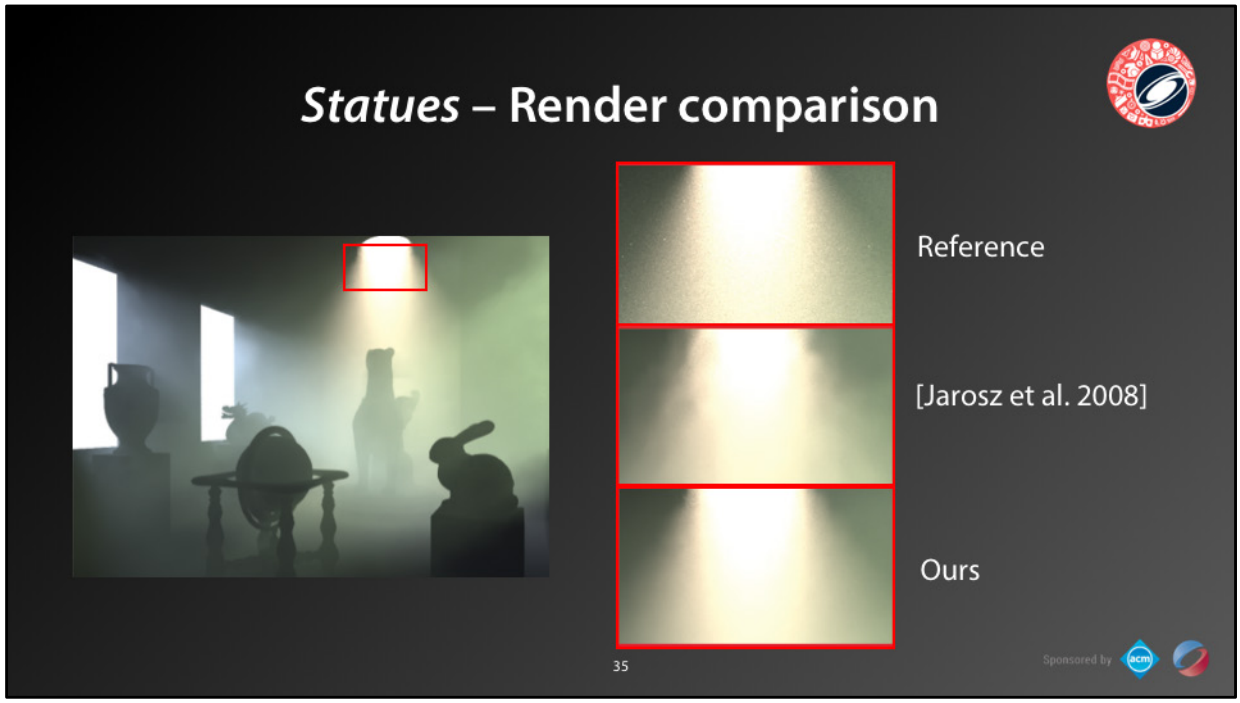


And finally let’s see some results of our method in 3D scenarios

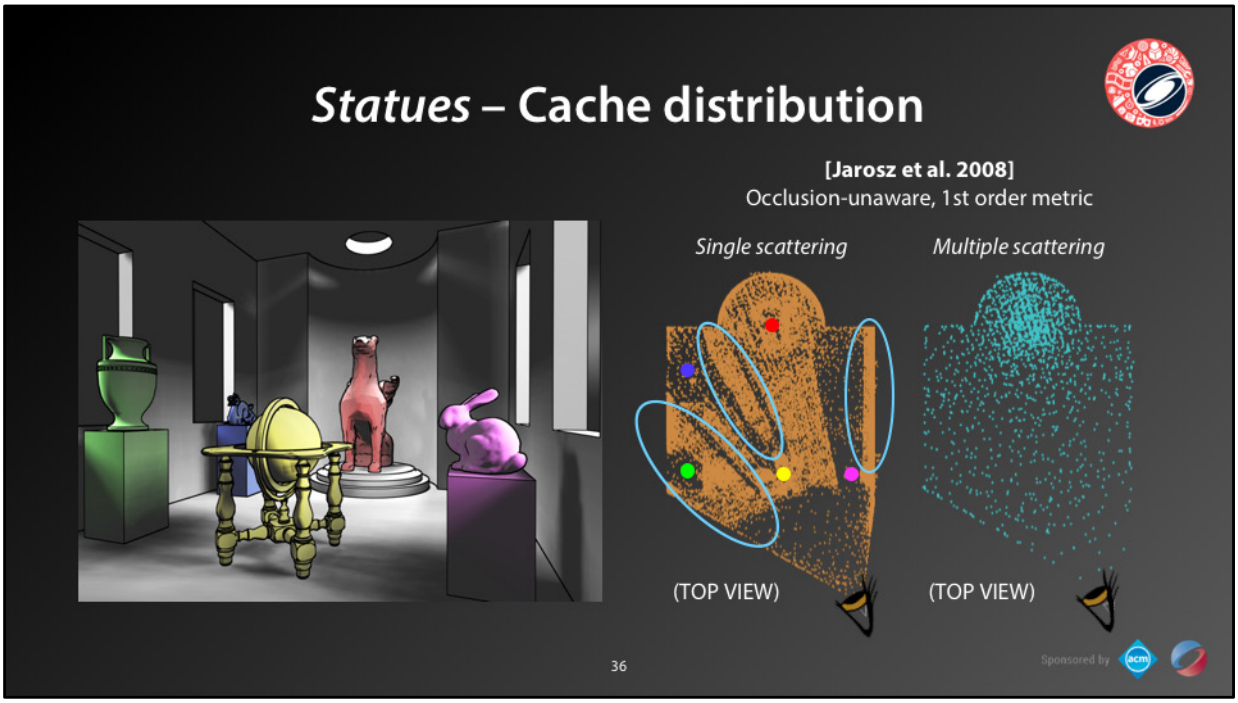


In this first scene we illustrate light shafts coming through the windows in a room with several statues. We compare against previous methods using similar number of cache points.

If we observe the center row of the blue inset [CLICK], the occlusion unaware metric doesn't create enough cache density to capture the boundaries of the shaft produced by a distant light source. Our method on the bottom row detects the occlusion, increasing sampling density on those regions and computing the correct extrapolation.

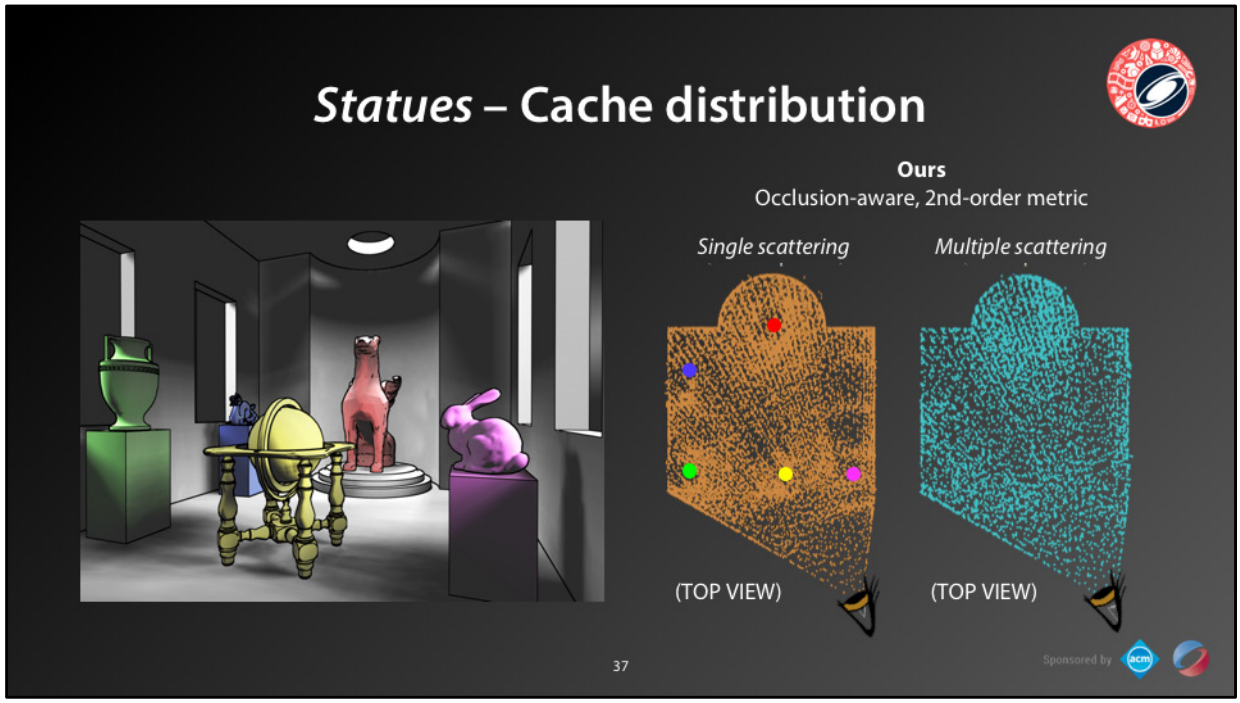


The red inset shows radiance coming through a hole in the ceiling, where the light source is much closer. While previous work's method provides higher cache density in that region, ignoring visibilty gradients still introduces significant artefacts during extrapolation. Our method is able to provide both enough cache density and correct extrapolation thanks to our occlusion-aware derivatives.

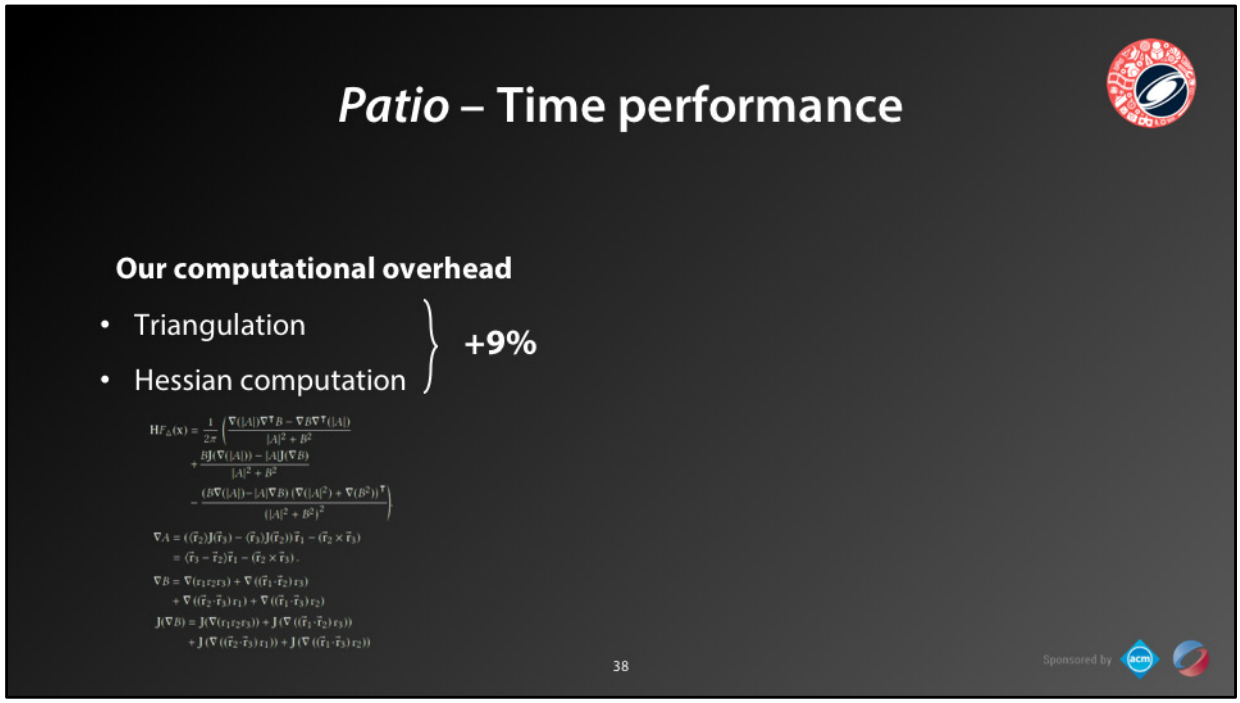


Here we show a top view of the cache generated by the previous method, and a depiction of the geometry of the scene. We map the colored statues in the cache distributions for reference.

While decreasing the error threshold in previous methods increases cache density, [CLICK] the distributions on the right show how it tends to oversample regions near surfaces. Setting a minimum cache radius barely helps, and still the extrapolations would be incorrect.

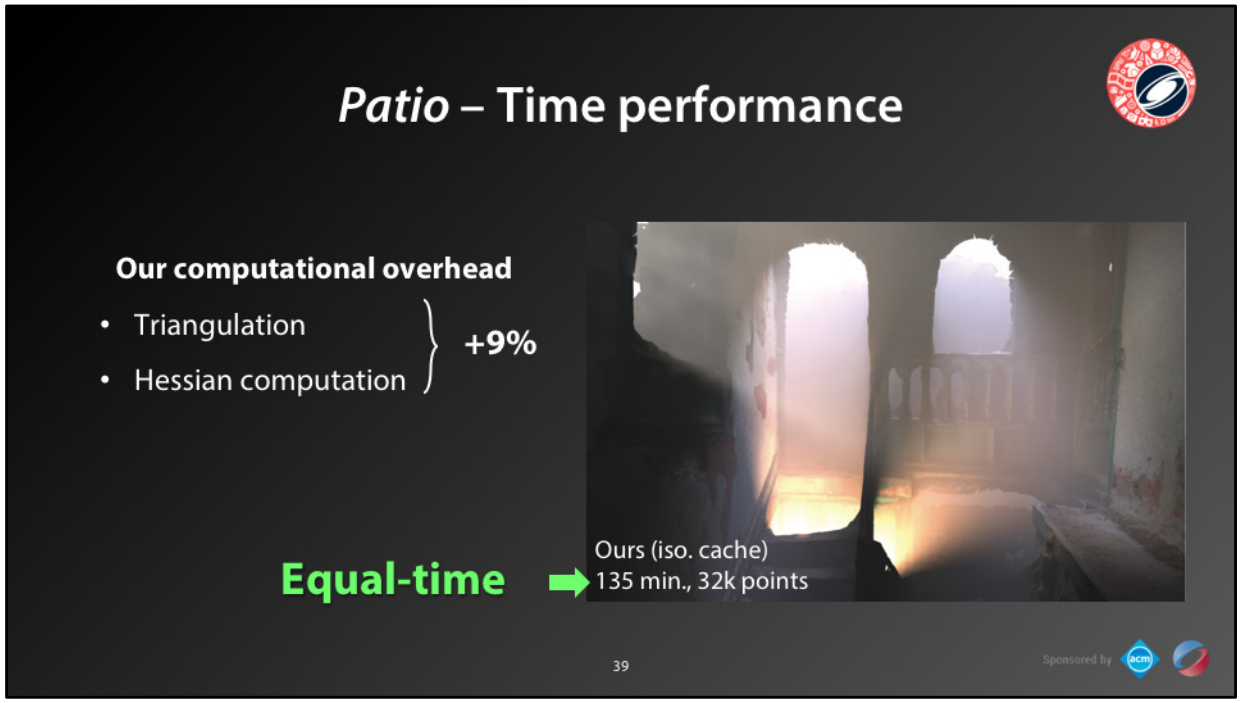


Using a similar number of points, our method is able to place cache points following to the frequency distribution of media radiance in the scene, avoiding oversampling issues



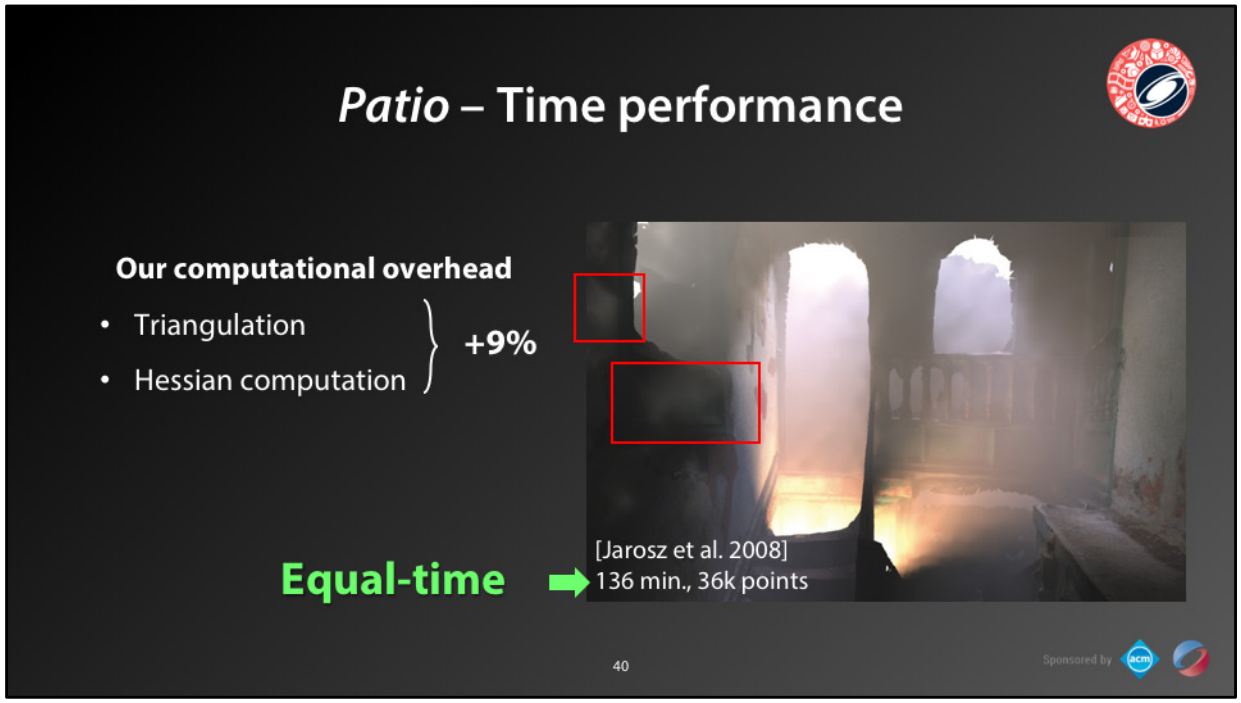
Nevertheless our method introduces certain overhead. Each cache point requires constructing a virtual triangulation, and computing radiance Hessians, [CLICK] which involve long expressions with 3x3 matrices. The computational overhead of our method with respect to the previous work is around an extra [CLICK] 9% during cache computation.

Still our method is able to provide better results in equal-time comparisons using fewer cache points. We show a comparison between our result for this scene [CLICK] and rendering with occlusion-unaware gradients in a similar time.



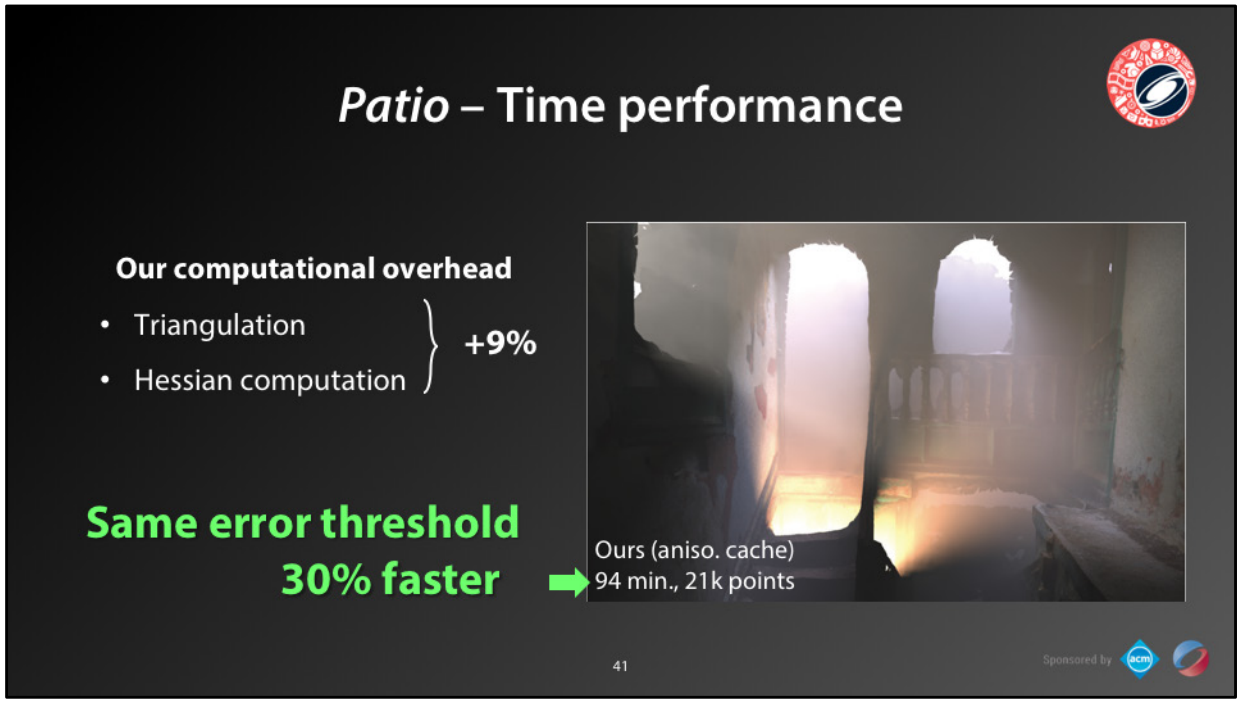
Nevertheless our method introduces certain overhead. Each cache point requires constructing a virtual triangulation, and computing radiance Hessians, [CLICK] which involve long expressions with 3x3 matrices. The computational overhead of our method with respect to the previous work is around an extra 9% during cache computation.

Still our method is able to provide better results in equal-time comparisons using fewer cache points. We show a comparison between our result for this scene [CLICK] and rendering with occlusion-unaware gradients in a similar time which introduces several artifacts as we can see in the red insets

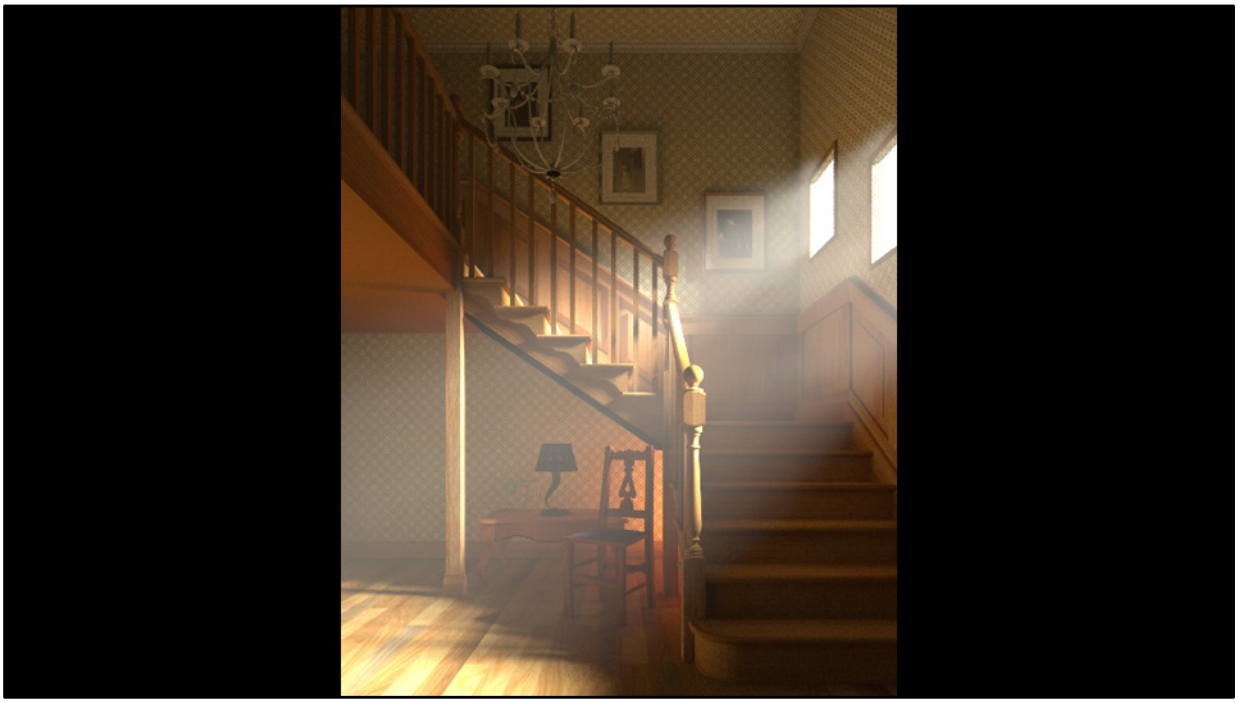


Nevertheless our method introduces certain overhead. Each cache point requires constructing a virtual triangulation, and computing radiance Hessians, [CLICK] which involve long expressions with 3x3 matrices. The computational overhead of our method with respect to the previous work is around an extra 9% during cache computation.

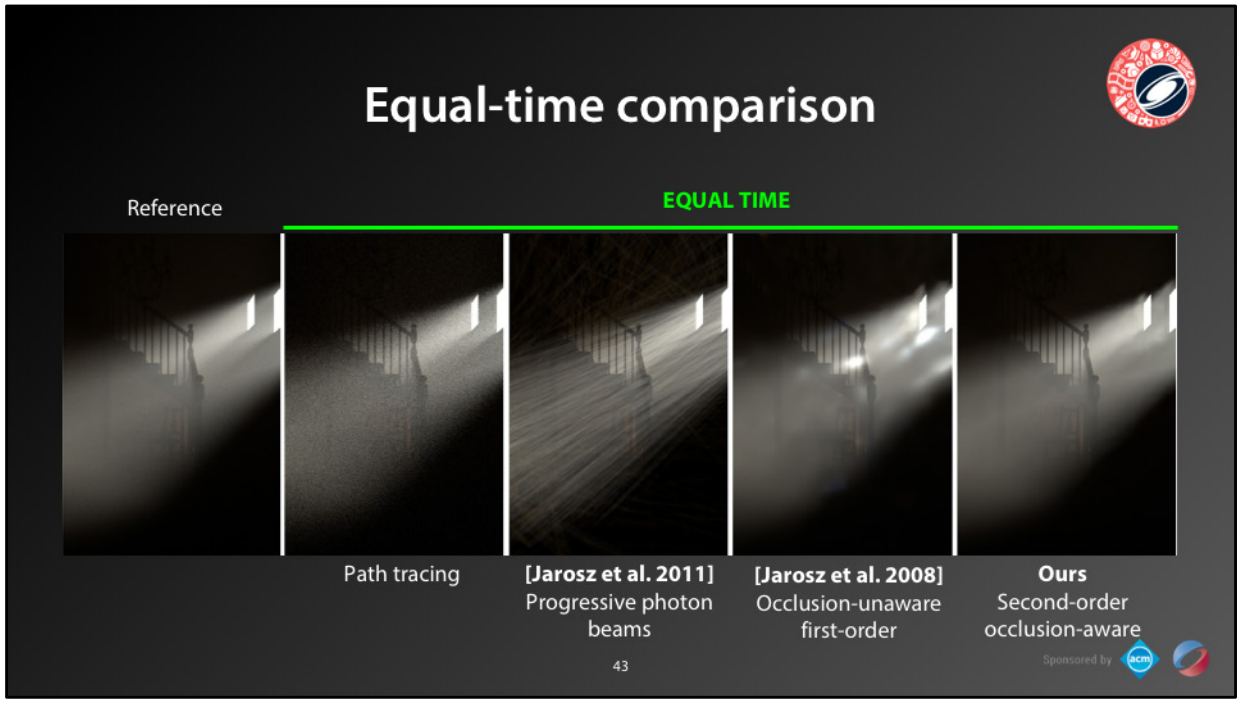
Still our method is able to provide better results in equal-time comparisons using fewer cache points. We show a comparison between our result for this scene [CLICK] and rendering with occlusion-unaware gradients in a similar time which introduces several artifacts as we can see in the red insets



Additionally, we can stretch the cache points along the axes of lower change based on the eigendecomposition of the Hessians. This allows us to provide results under the same error threshold reducing the computational cost by a 30%.

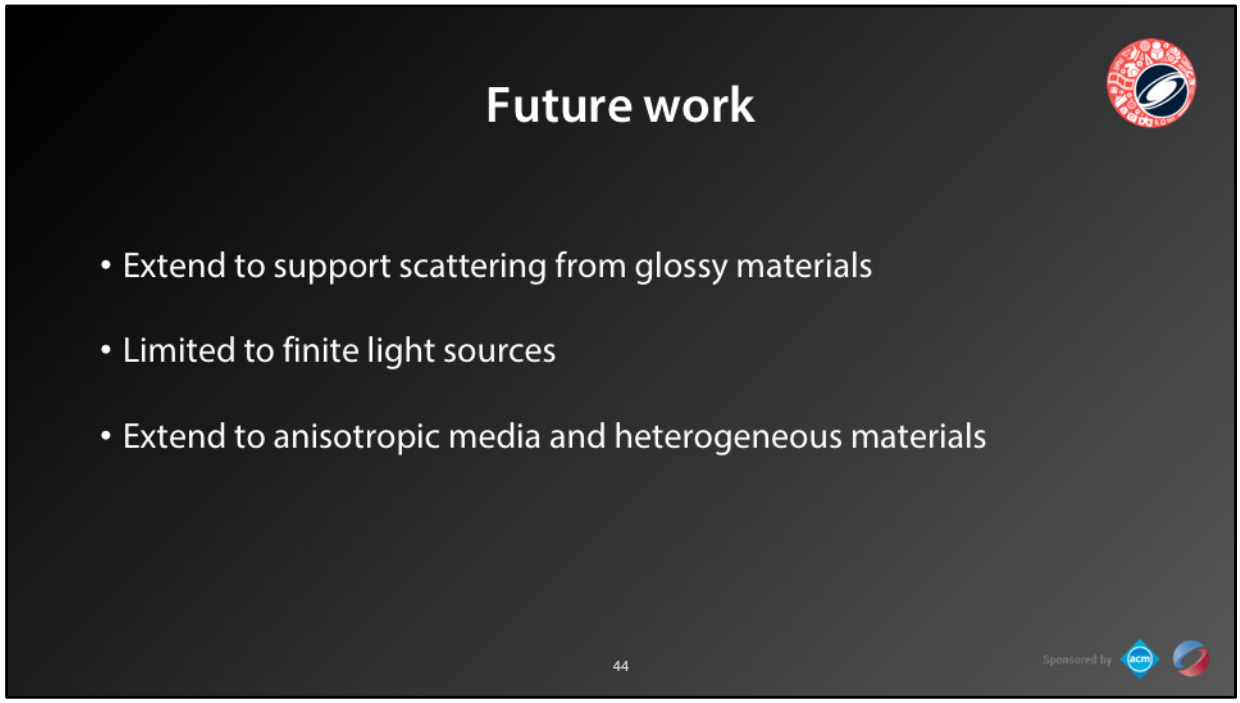


Finally we illustrate a staircase scenario with distant lighting coming through the window.



In this equal-time comparison of several methods, we can see how path tracing still shows significant noise. Progressive photon beams fails to score beams through the window from distant light sources, introducing variance in the results.. Previous method of radiance caching still present artifacts due to not accounting for visibility changes during gradient computation.

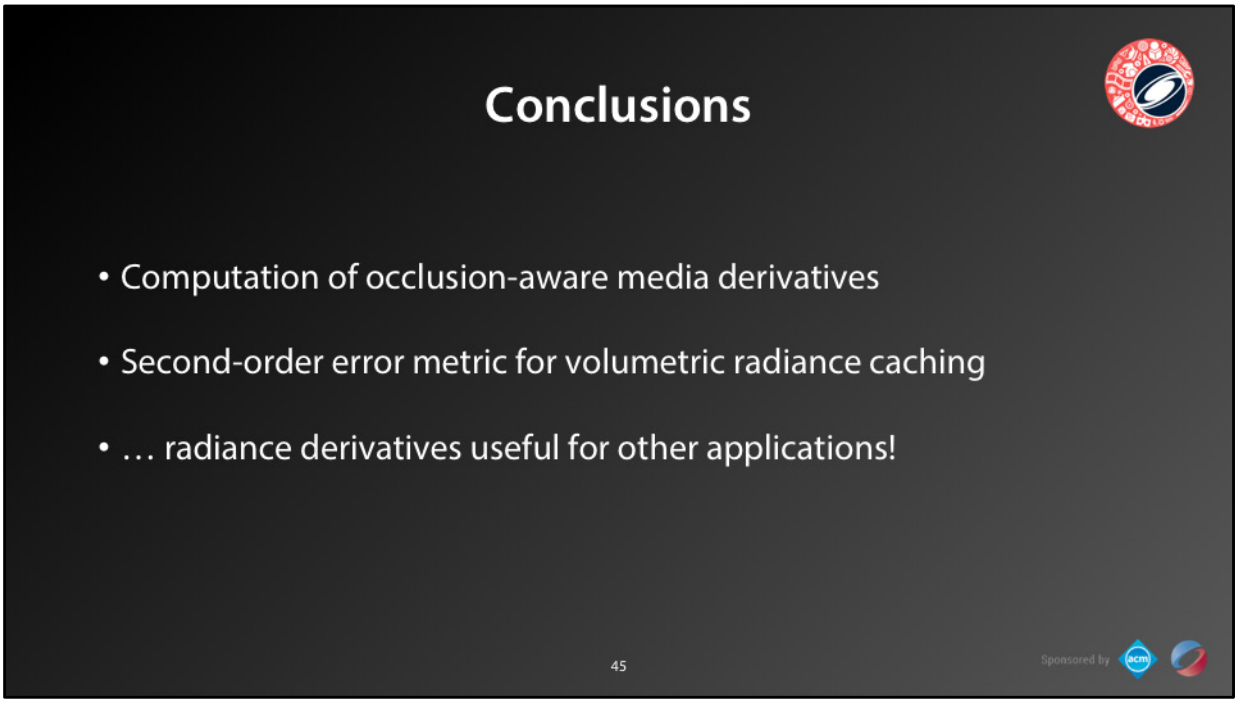
Our method on the far right produces a noise-free solution and while representing shadow details of mixed light shafts through the windows.



Our method is not free of limitations and chances for future improvement.

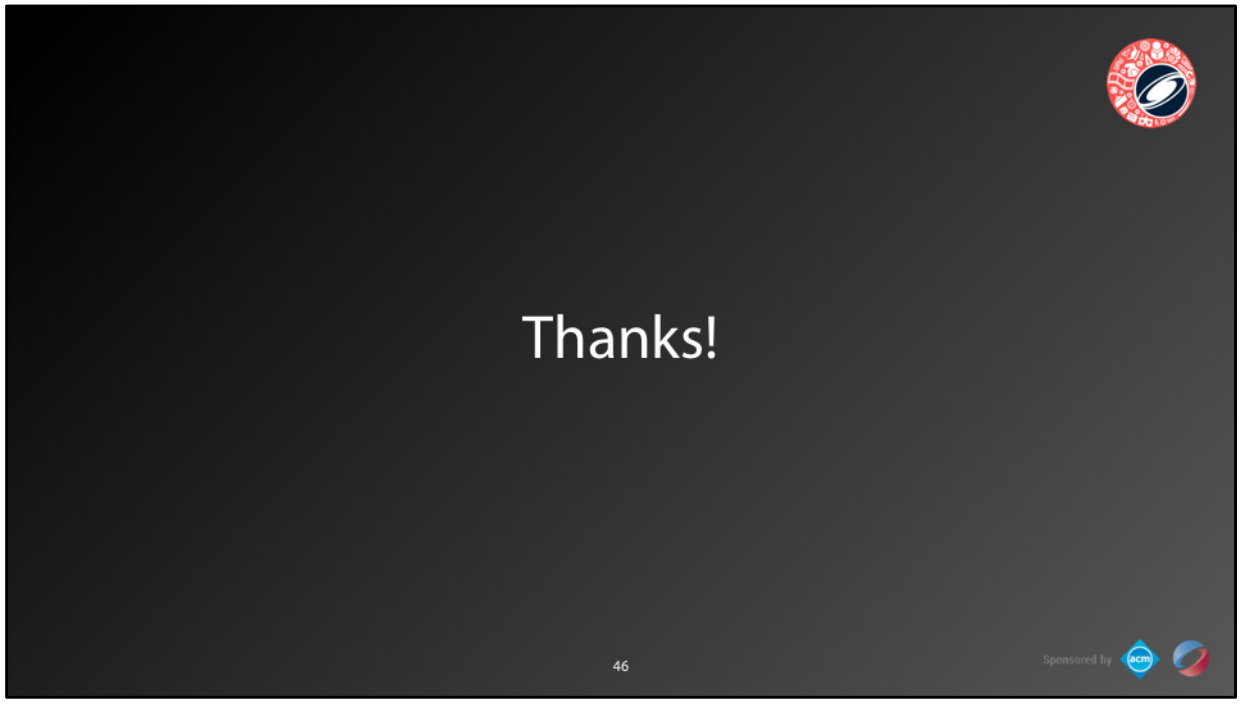
First of all, our gradient formulation based on form-factors is limited to diffuse surfaces, so adding support for glossy and specular materials would be interesting, since they tend to generate high frequency light patterns

Making our method work in anisotropic media would require revising the assumptions made for closed-form triangulation derivatives.



In conclusion, we have introduced a method to compute participating media derivatives that account for occlusion changes, and used them to provide a second-order error metric in radiance caching.

Finally, our radiance derivatives are not limited to radiance caching, but could also be applied for example to compute optimal kernels in density estimation, or improving quadrature methods.


Thanks!
46
Sponsored by