Sensitivity as an Arbitrary Output Variable for Differentiable Rendering

Linas Beresna
Simon Fraser University
Burnaby, BC, Canada
linas_beresna@sfu.ca

Eugene Fiume
Simon Fraser University
Burnaby, BC, Canada
eugene_fiume@sfu.ca

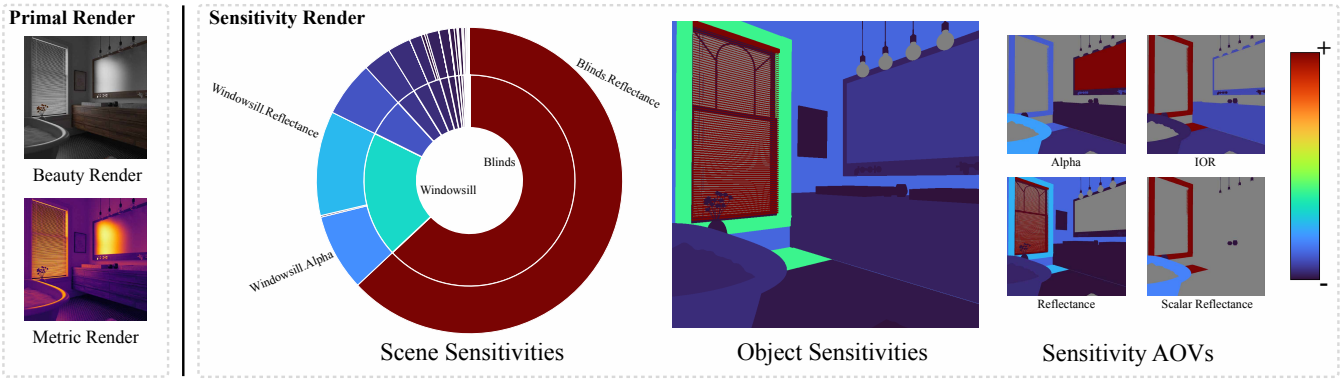


Figure 1: Sensitivity AOVs. Beside the primal outputs (left), one reverse-mode pass yields the derivative of a scalar objective with respect to every scene parameter (right). One buffer supports several views: a sunburst ranking the most influential parameters, a projection onto primary visibility locating them in the image, and per-type channels, each signed, blue where increasing the parameter lowers the objective and red where it raises it.

Abstract

Differentiable renderers expose the derivative of any scalar objective with respect to every scene parameter, yet unlike the primal image, which decades of arbitrary output variables (AOVs) have taught us to decompose, inspect, and composite, these derivatives have no established representation for human inspection. We introduce the *sensitivity AOV*, a render output carrying the sensitivity of an objective to the scene parameters that influence it. A single reverse-mode pass populates a *sensitivity buffer* over the scene's parameter hierarchy, from which many views are read rather than re-differentiated, in direct analogy to deferred shading: image-space sensitivity at object and parameter-type granularity, projections onto a freely navigable scene from any inspection viewpoint, and, for spatially varying parameters, per-texel fields carried to the surface through texture coordinates. We separate the fixed camera that defines the objective from the free camera used to inspect the result, and position reverse-mode attribution against its forward-mode dual. Our aim is not a single algorithm but a scaffolding that establishes derivative outputs as first-class render products alongside the primal image.

This work is licensed under a Creative Commons Attribution 4.0 International License.
SA Technical Communications '26, Kuala Lumpur, Malaysia
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2841-9/2026/12
https://doi.org/10.1145/3829339.3847832

CCS Concepts

• **Computing methodologies** → **Ray tracing**.

Keywords

differentiable rendering, arbitrary output variables, gradient visualization, scene attribution, adjoint methods

ACM Reference Format:
Linas Beresna and Eugene Fiume. 2026. Sensitivity as an Arbitrary Output Variable for Differentiable Rendering. In *SIGGRAPH Asia 2026 Technical Communications (SA Technical Communications '26), December 01–04, 2026, Kuala Lumpur, Malaysia.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3829339.3847832

1 Introduction

Production renderers rarely output only the final picture. Alongside the beauty pass, an *arbitrary output variable* (AOV) exposes some intermediate quantity of the light transport, such as depth, surface normals, albedo, or object and material identity, so that artists can inspect, relight, denoise, and composite [Industrial Light & Magic 2026]. Decades of practice have made these outputs first-class: a compositor expects a normal pass, and an identity pass such as cryptomatte [Friedman and Jones 2015] is a standard deliverable.

Differentiable renderers [Li et al. 2018; Nimier-David et al. 2020, 2019; Vicini et al. 2021] expose a fundamentally new quantity: the derivative of any scalar objective evaluated on the image with respect to *every* scene parameter. A single reverse-mode pass returns, for each material, light, and geometric parameter, how sensitive

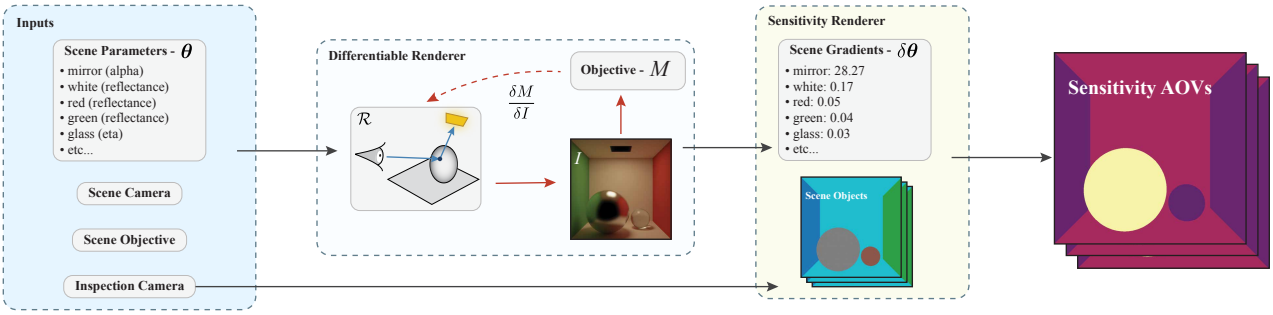


Figure 2: The sensitivity AOV pipeline. The inputs (left) are the scene parameters θ, the camera and objective defining the differentiated quantity, and an *inspection camera* used only for viewing. The renderer (centre) forms $I = \mathcal{R}(\theta)$ and evaluates M; one reverse-mode pass propagates $\partial M/\partial I$ back through light transport to yield a sensitivity for every parameter (right). That populated set is the sensitivity buffer: binding it to primary visibility gives the *Scene Objects* pass, selecting subsets over it the *Sensitivity AOVs*.

the objective is to that parameter. To date this quantity has been consumed almost exclusively by optimizers, in inverse rendering and design. Yet the derivative is also informative on its own: it tells a human which parts of a scene matter for a given goal. Previous work [Beresna and Fiume 2026] established that these gradients are interpretable and that the resulting rankings are objective-specific, but treated each map as a one-off visualization; what the quantity still lacks, and what is supplied here, is the output representation the primal image has long possessed, a reusable buffer and a taxonomy of the views read from it.

Treating the gradient of an output as an attribution over its inputs is by now standard in neural network interpretability [Selvaraju et al. 2017]. We apply the principle to a different differentiable function: the inputs become scene parameters and the output an arbitrary scalar objective, so the attribution lands in the scene rather than in a feature map.

We propose the *sensitivity AOV*: a render output carrying the sensitivity of an objective to the scene parameters that influence it. Where a forward AOV is *scene-intrinsic* (depth is depth, independent of any goal), a sensitivity AOV is *objective-conditioned*, so the same scene yields different sensitivity AOVs under different objectives. Forward AOVs answer *what is here*; sensitivity AOVs answer *what would change this*.

Our contribution is a scaffolding rather than a single algorithm. We: (i) define the sensitivity AOV along four axes: selection, binding, aggregation, and encoding; (ii) observe that one reverse-mode pass populates a *sensitivity buffer* over the scene's parameter hierarchy, from which many views follow without recomputation, in direct analogy to deferred shading; (iii) give a taxonomy of those views, distinguishing where a parameter *lives* from where its influence *lands*; and (iv) provide a reference implementation. Our aim is to establish derivative outputs as first-class render products alongside the primal image.

2 The Sensitivity Buffer

Let $\mathcal{R}$ be a differentiable renderer producing an image $I = \mathcal{R}(\theta)$ from scene parameters θ, and let M be a scalar objective evaluated on I. A single reverse-mode pass yields

$$\mathbf{g} = \frac{\partial M(\mathcal{R}(\theta))}{\partial \theta} = \frac{\partial M}{\partial I}\,\frac{\partial I}{\partial \theta}, \tag{1}$$

a vector assigning to every scene parameter a scalar sensitivity. Crucially, $\mathbf{g}$ is computed *once*, at the cost of one backward pass whose cost is largely independent of the number of parameters. We call the populated parameter set the *sensitivity buffer*. Every visualization we describe is a re-projection of this buffer, never a recomputation, the same relationship deferred shading has to the G-buffer [Saito and Takahashi 1990]: the adjoint sweep happens once, and the user then explores many views over the stored result.

Scene parameters are naturally *hierarchical*. Renderers expose them through structured keys such as `floor.bsdf.reflectance`, so the buffer is a tree: leaves are individual parameters, interior nodes are materials and objects.

3 From Buffer to Image

The buffer of Section 2 is a tree of scalar sensitivities, one per parameter component. A *sensitivity AOV* is what results from reading that tree at a chosen granularity and drawing it: we select a subtree, aggregate its components to a single scalar per group, bind that scalar to pixels, and encode it for display. The first two steps decide *what* is shown, the last two *where* and *how*.

Selection and aggregation. Selection is a query over the tree, and two cuts are natural. A *vertical* cut groups by object, collecting every parameter of one material into a single value, and answers which material matters (the *Object Sensitivities* of Fig. 1). A *horizontal* cut groups by parameter type across the whole scene, yielding the reflectance, roughness, and index-of-refraction channels shown as the *Sensitivity AOVs*. Either way a group must reduce to one scalar. We use the ℓ_2 norm of the group's value-weighted gradients, which measures the total response to perturbing the whole group. Because parameters differ in dimensionality, a single reflectance versus a million-texel texture, this total can instead be taken per component when the intent is to compare across parameter types rather than total leverage; the choice is part of the encoding and is recorded with the channel (Fig. 3).

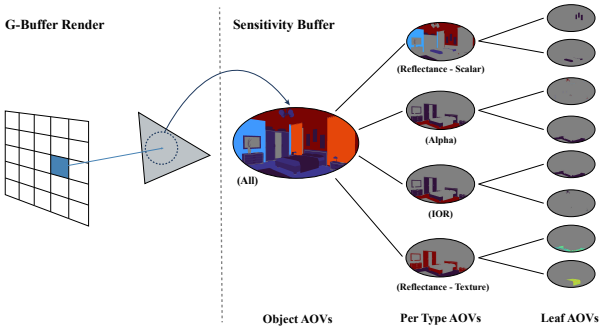


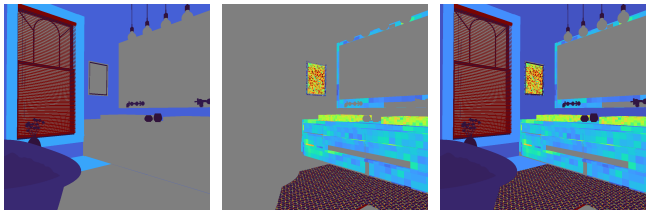
Figure 3: The sensitivity buffer as a hierarchy. Left: a primary ray from a pixel identifies the visible surface, the G-buffer step that binds sensitivities to the image. Right: the buffer read at three granularities, leaf parameters aggregating into per-type channels, which aggregate into the object-level view. Each level is a coarser query over one buffer.

Binding. An aggregated value acquires a pixel location through primary visibility: we trace one ray per pixel, identify the object hit, and paint that object's sensitivity. Because this step touches only primary rays and reuses the stored buffer, it carries no gradient computation, so the same sensitivities can be re-bound from any *inspection camera* without a new adjoint pass. The objective is fixed by one camera; the camera used to look at the result is free, and moving it does not change the objective (Fig. 6). A spatially varying parameter binds differently: a texture has its own domain, so its per-texel gradient can be carried to the surface through the hit UVs and shown in place. This exposes a choice a flat parameter never forces, since the same texture can be reduced to one object value or painted per texel, and the two do not share a scale, a single texel's influence being small beside a whole material's. We treat them as distinct views of one buffer (Fig. 4) rather than forcing them into a single image.

Encoding. A sensitivity can be shown as a signed quantity or a magnitude, under linear or compressive tone mapping, against a chosen normalization. The most consequential choice is what a unit of the parameter means: a raw gradient $\partial M/\partial\theta$ is expressed per unit of θ, but a unit differs for a reflectance in $[0, 1]$ and an unbounded emitter radiance, so weighting by the value, $\theta\,\partial M/\partial\theta$, instead reports the response to a proportional perturbation. Because a sensitivity AOV is objective-conditioned, two maps are comparable only if they share this encoding, so we record it explicitly with each channel.

4 Forward and Reverse

The buffer is a reverse-mode object: one objective propagated to every parameter at once, dense in parameters and sparse in objectives. Its transpose is forward mode, which pushes one chosen parameter through to every pixel, $\partial I/\partial\theta_i$. The two answer different questions, reverse for attribution (which parameters matter for this objective) and forward for prediction (what a single parameter does to the image), and the second cannot be assembled from the first without repeating the computation per parameter. Figure 5 places both side by side: forward mode is not an alternative to the buffer but its dual, suited to the question the buffer does not answer.



(a) Per object (b) Per texel (c) Combined

Figure 4: Viewing a spatially-varying parameter. A textured reflectance aggregated to one object value, painted per texel, and under a joint normalization. The two levels use separate scales (Section 3).

5 Example Objective and Results

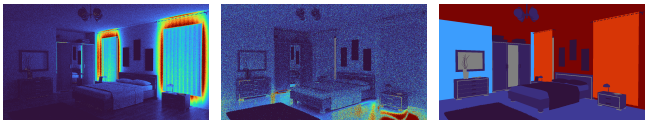
To keep the objective an *example* rather than a contribution, we use the simplest scalar on the image, its mean photometric response, $M = \frac{1}{|P|}\sum_{p\in P} Y(I_p)$, where $Y(I_p)$ is the photometric response of pixel p. We choose it because its sensitivities are easy to reason about, which makes the buffer and its views straightforward to check; the pipeline is agnostic to M, and a perceptual or learned objective enters at exactly the same point, changing only $\partial M/\partial I$.

Figure 1 shows the result on an interior scene. Read at the object level, the sensitivity AOV attributes the objective across the scene's materials, including surfaces whose influence is carried by indirect transport and is not evident in the primal image. Read as per-type channels it separates which *kind* of parameter drives the objective, and the hierarchy would surface parameters such as a global exposure that never appear in the frame at all. The forward channels of Figure 5 and the free-viewpoint projection of Figure 6 come from this same single pass.

Many objectives at once. Because each objective conditions its own sensitivity AOV, a scene can be inspected under several at once, one channel per objective, so that a user selects among objective-conditioned derivative passes exactly as they select among forward AOVs today. Objectives differ only in the image-space adjoint $\partial M/\partial I$ that seeds the backward pass, but they do not share that pass for free: each requires its own adjoint propagation unless the renderer can carry several adjoint channels through one sweep, and each populates its own buffer, so storage grows linearly in the number of objectives inspected together.

6 Implementation

We realize sensitivity AOVs on a physically based path tracer with reverse-mode differentiation [Jakob et al. 2022; Vicini et al. 2021]. Evaluating the objective and backpropagating populates the buffer via Equation 1, a single pass that all placement views then share. Binding a uniform parameter uses the object visible at each pixel, recovered from a one-sample-per-pixel primary-visibility pass that also yields the UVs and object identities the other views reuse. A



(a) frwd | reflectance (b) frwd | alpha (c) Reverse

Figure 5: Reverse and forward modes are duals. Forward mode (a, b) pushes a single parameter to all pixels; reverse mode (c) gives one objective's sensitivity to all parameters at once (Section 4).

spatially varying parameter carries its per-texel gradient to the surface through these UVs; because the gradient and the visible texel are sampled independently, we pool the per-texel field before projection to fill gaps and suppress fireflies, which is the main practical cost of the textured case. Hierarchy views are read directly from the tree and need no image-space pass at all. The forward-mode view is the one output not derived from the buffer: it requires its own forward-derivative evaluation, seeding a single parameter and propagating to the image, and is therefore recomputed per parameter shown. Everything else re-projects the one reverse pass.

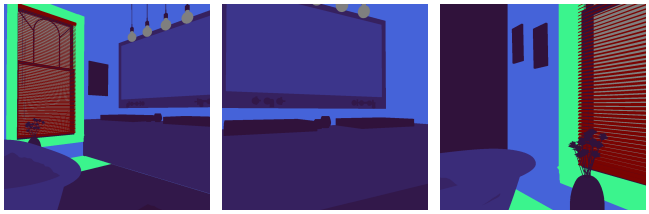
7 Limitations and Future Work

The principal open view is the *transport* view: rather than painting a parameter where it lives, painting it where its influence *lands*, so that a wall's reflectance colors the table it brightens through interreflection. Unlike the placement views, this cannot re-project a finished buffer: it must accumulate contributions during the adjoint sweep and is tied to a single viewpoint. We leave it, together with a fuller treatment of per-texel fields, which is the least explored case in our current implementation, to future work.

Several further limitations bound the present method. Our placement views read interior gradients and do not treat visibility discontinuities, so a boundary-correct account of geometric sensitivity is out of scope here. Sensitivities are local first-order quantities and do not capture interactions between parameters. Like all Monte Carlo derivatives they inherit path-tracing variance, so buffers computed at low sample counts may need denoising, and denoising a sensitivity field for faithful inspection, without reordering which parameters dominate, is itself an unstudied problem distinct from beauty-image denoising. Finally, our objectives are deliberately simple; richer perceptual and neural objectives fit the same framework and are natural to explore.

8 Conclusion

Differentiable renderers already compute, as a byproduct of every optimization, a quantity that answers which parts of a scene matter for a given goal. We have argued that this quantity deserves the same first-class output status as the primal image. The sensitivity AOV names the output, the sensitivity buffer reduces each further view to a primary-visibility pass over a stored result rather than a second adjoint sweep, and the selection, binding, and encoding choices determine how those derivatives are shown to a human. Forward AOVs describe what the scene is; sensitivity AOVs describe what would change it. Much of the space they open, the transport view, per-texel fields, denoising for human inspection, remains unexplored, and we offer this as scaffolding for it.



(a) Objective camera (b) Inspection view 1 (c) Inspection view 2

Figure 6: One buffer, many viewpoints. The buffer is computed once by an adjoint pass fixed by the *objective camera* (a). The *inspection camera* then moves freely (b, c): each view re-runs only primary visibility and re-projects the *same* sensitivities, with no gradient recomputation. All three share one colour scale.

Acknowledgments

We acknowledge with gratitude the funding provided for this research by Simon Fraser University as well as a Discovery Grant provided by the Natural Sciences and Engineering Research Council of Canada. Additionally, we thank Benedikt Bitterli [Bitterli 2016] for providing the scene used in our figures.

References

Linas Beresna and Eugene Fiume. 2026. Scene Parameter Saliency via Differentiable Light Transport. *arXiv preprint arXiv:2607.21562* (2026).

Benedikt Bitterli. 2016. Rendering resources. https://benedikt-bitterli.me/resources/.

Jonah Friedman and Andrew C. Jones. 2015. Fully automatic ID mattes with support for motion blur and transparency. In *ACM SIGGRAPH 2015 Posters* (Los Angeles, California) *(SIGGRAPH '15)*. Association for Computing Machinery, New York, NY, USA, Article 47, 1 pages. doi:10.1145/2787626.2787629

Industrial Light & Magic. 2026. OpenEXR. https://openexr.com. Accessed: 2026.

Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, and Delio Vicini. 2022. Dr.Jit: A Just-In-Time Compiler for Differentiable Rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)* 41, 4 (July 2022). doi:10.1145/3528223.3530099

Tzu-Mao Li, Miika Aittala, Frédo Durand, and Jaakko Lehtinen. 2018. Differentiable Monte Carlo ray tracing through edge sampling. *ACM Trans. Graph.* 37, 6, Article 222 (Dec. 2018), 11 pages. doi:10.1145/3272127.3275109

Merlin Nimier-David, Sébastien Speierer, Benoît Ruiz, and Wenzel Jakob. 2020. Radiative Backpropagation: An Adjoint Method for Lightning-Fast Differentiable Rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)* 39, 4 (July 2020). doi:10.1145/3386569.3392406

Merlin Nimier-David, Delio Vicini, Tizian Zeltner, and Wenzel Jakob. 2019. Mitsuba 2: A Retargetable Forward and Inverse Renderer. *Transactions on Graphics (Proceedings of SIGGRAPH Asia)* 38, 6 (Dec. 2019). doi:10.1145/3355089.3356498

Takafumi Saito and Tokiichiro Takahashi. 1990. Comprehensible rendering of 3-D shapes. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques* (Dallas, TX, USA) *(SIGGRAPH '90)*. Association for Computing Machinery, New York, NY, USA, 197–206. doi:10.1145/97879.97901

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.

Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2021. Path Replay Backpropagation: Differentiating Light Paths using Constant Memory and Linear Time. *Transactions on Graphics (Proceedings of SIGGRAPH)* 40, 4 (Aug. 2021), 108:1–108:14. doi:10.1145/3450626.3459804