Computational
Photography Lab.

Find the project web page here:
https://yaksoy.github.io/DiffLightComp/

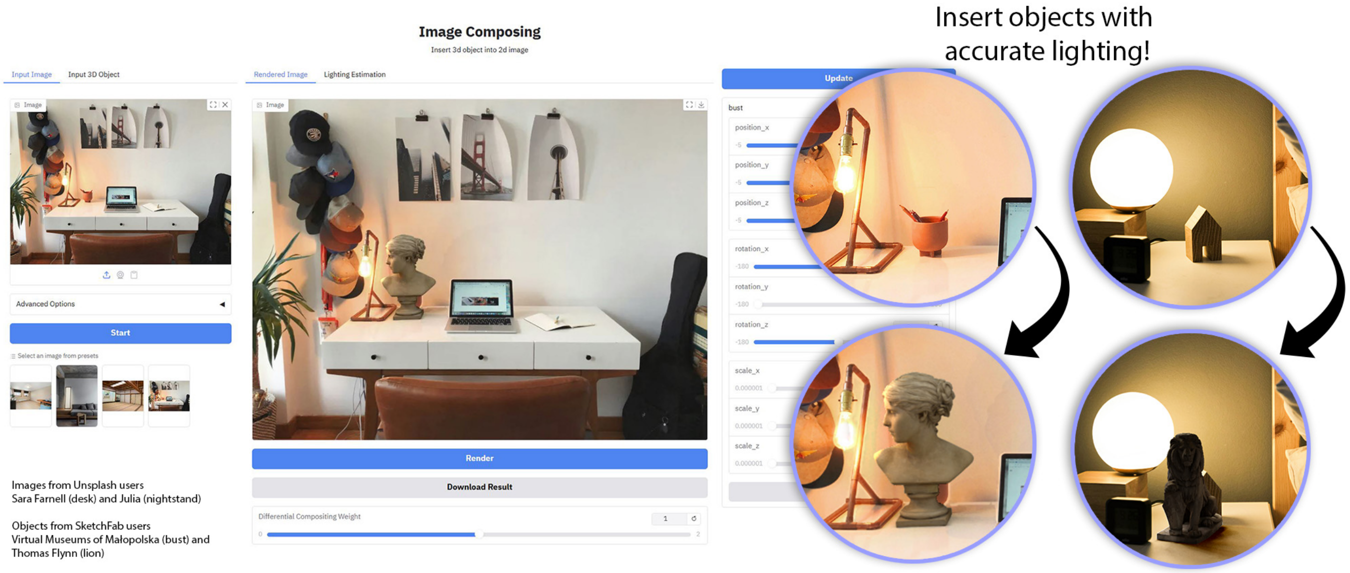
# Interactive Object Insertion with Differentiable Rendering

Weikun Peng*
Simon Fraser University
Canada

Sota Taira*
Simon Fraser University
Canada

Chris Careaga
Simon Fraser University
Canada

Yağız Aksoy
Simon Fraser University
Canada



Differential Compositing Weight: 1 (0 – 2)

bust
position_x: -5
position_y: -5
position_z: -5
rotation_x: -180
rotation_y: -180
rotation_z: -180
scale_x: 0.000001
scale_y: 0.000001
scale_z: 0.000001

Images from Unsplash users
Sara Farnell (desk) and Julia (nightstand)

Objects from SketchFab users
Virtual Museums of Małopolska (bust) and
Thomas Flynn (lion)

**Figure 1: We develop an object insertion pipeline and interface that enables iterative editing of illumination-aware composite images. Our pipeline leverages off-the-shelf computer vision methods and differentiable rendering to reconstruct a 3D representation of a given scene. Users can add 3D objects and render them with physically accurate lighting effects.**

**ACM Reference Format:**
Weikun Peng, Sota Taira, Chris Careaga, and Yağız Aksoy. 2025. Interactive Object Insertion with Differentiable Rendering. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Posters (SIGGRAPH Posters '25), August 10-14, 2025*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3721250.3743019

## 1 INTRODUCTION

Compositing virtual objects into real-world imagery, referred to as object insertion, has a number of applications across film visual effects, augmented reality, and even interior design. Creating physically realistic composites can be a tedious process requiring an artist to perform manual editing of illumination effects for a given object and background scene. This manual process lacks interactive feedback, making it difficult to finetune aspects of the composite, such as object location, size, and orientation. In this work, we propose a modern framework and accompanying user interface to bring recent advancements in computational photography to artists and designers in an accessible and extensible manner. Specifically, we follow the method of Careaga and Aksoy [2025] and leverage state-of-the-art mid-level vision estimations to build a virtual 3D scene from a single image. We then use differentiable rendering and optional user constraints to determine the lighting conditions in the scene. Finally, we allow the user to place 3D objects into the scene and render them using the estimated illumination, resulting in a final realistic composite. Our method brings together ideas from the past decade of inverse rendering research to create an open-source tool for artists and designers. Our implementation is publicly available at: https://github.com/willipwk/image-composing

## 2 METHOD

Our pipeline consists of multiple steps that can be carried out in an iterative fashion via user interaction. Our interface represents a modern take on prior automatic object insertion methods from the literature [Karsch et al. 2011]. Our scene reconstruction process follows closely the methodology of [Careaga and Aksoy 2025]. In this section, we detail each step of our pipeline. The entire process is summarized in Figure 2.

*Preprocessing*. The input to our interface is a single image, $I$, that will be used as the background scene for subsequent object insertions. We begin our pipeline by first preprocessing the input image to create a 3D representation of the scene. Specifically, we

*Denotes equal contribution.

*SIGGRAPH Posters '25, August 10-14, 2025, Vancouver, BC, Canada*
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1549-5/2025/08.
https://doi.org/10.1145/3721250.3743019

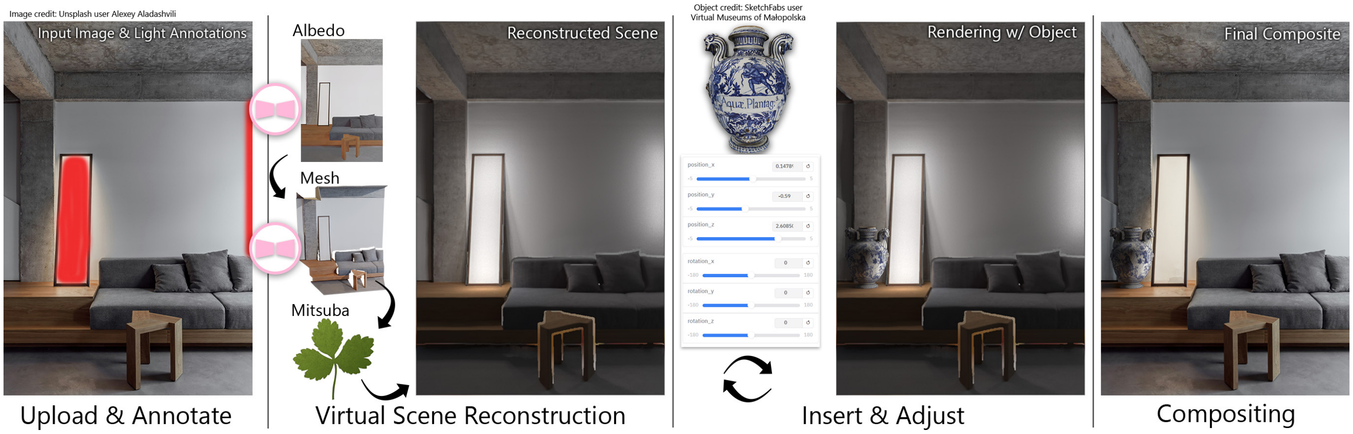


Image credit: Unsplash user Alexey Aladashvili

Object credit: SketchFabs user Virtual Museums of Małopolska

position_x 0.1478( -5 5
position_y -0.59 -5 5
position_z 2.6085( -5 5
rotation_x 0 -180 180
rotation_y 0 -180 180
rotation_z 0 -180 180

**Figure 2: Our pipeline starts with virtual reconstruction of an input image by estimating albedo and geometry using off-the-shelf methods. After generating the mesh, we use light source annotations and Mitsuba 3 [Jakob et al. 2022] to optimize illumination via differentiable rendering. Users can then insert 3D objects into the rendered version of the scene, and adjust location, orientation, and size. Finally, we render a high-quality version of the objects and composite them into the original image.**

use the method of [Careaga and Aksoy 2024] to extract an albedo image from the scene. This provides a representation of the image without any lighting effects present. Then, using MoGe [Wang et al. 2025], we extract a 3D point map, which we triangulate to create a mesh. For each pixel in the image, we texture the corresponding vertex in the mesh using the estimated albedo color. With this mesh and camera parameters estimated from MoGe, we have a 3D scene representation that can be imported into any rendering engine.

*Lighting Optimization.* In order for the user to insert objects with accurate lighting, we need to model the illumination present in the scene. We leverage recent advances in differentiable rendering to accomplish this task. Specifically, we load the scene into Mitsuba [Jakob et al. 2022], a differentiable path-tracing engine, and add an environment map and multiple point lights. Optionally, the user can provide annotations of the light sources visible in the scene to better initialize the location of the point lights. If no annotations are provided, point lights are initialized in a grid over the scene mesh. We utilize gradient-based optimization to determine the values of the environment map and point lights that best reconstruct the input image. This process results in an approximation of the scene, complete with accurate illumination.

*Object Insertion and Compositing.* After reconstructing the input scene, users can begin adding virtual objects into the scene. Our interface allows users to add multiple objects and iteratively fine-tune placement, orientation, and size. After each edit, the scene is re-rendered to update the appearance of the inserted object. Given the approximate appearance of the reconstructed scene, our pipeline concludes by performing differential compositing of the rendered objects into the original scene. Concretely, we create two renderings of the reconstructed scene, one with the inserted objects, denoted $O$, and one without, denoted $N$. Then, using a mask indicating which pixels belong to the inserted objects, $M$, and the original input image $I$, we compute the final composite image, $C$, using the differential compositing equation from [Debevec 1998]:

$$C = M \cdot O + (1 - M) \cdot (B + \alpha(O - N)) \tag{1}$$

where $\alpha$ is a scalar to adjust the weight of the changed value in the image background. Our interface allows the users to alter the value of $\alpha$. By compositing the rendered object using this equation, we can transfer any illumination changes caused by the rendered objects (e.g., shadows) to the input image, maintaining the realistic appearance of the original scene.

## 3 RESULTS

Some example results can be seen in Figures 1, 2 and **??**. Our scene reconstruction works well in various scenarios and can handle local lighting effects such as a desk lamp. Inserted objects exhibit physically accurate illumination effects and can even cast shadows onto the environment. Our user interface allows users to fine-tune various aspects of the object insertion. If the lighting optimization does not result in an accurate reconstruction, users can alter lighting annotations and re-run the optimization process. Users can also add multiple objects and alter their characteristics. The reconstructed scene can be rendered at a low sample rate and resolution, meaning that the user can get near instant feedback after each edit. Once the user is happy with their composition, our application renders the scene at full resolution with a high sample count in order to do the final differential compositing process. A demo of our application can be found in the supplementary video.

## 4 CONCLUSION

In this work, we propose a pipeline and interface that enables users to composite 3D objects into photographs with accurate lighting effects. While our method can handle various lighting conditions, improvements could be made to the scene reconstruction algorithm to better model outdoor illumination, and out-of-frame occluders. We hope that our interface serves as a starting point for future development and helps bring years of research to the hands of capable artists.

## REFERENCES

Chris Careaga and Yağız Aksoy. 2024. Colorful Diffuse Intrinsic Image Decomposition in the Wild. *ACM Trans. Graph.* 43, 6, Article 178 (2024), 12 pages.

Chris Careaga and Yağız Aksoy. 2025. Physically Controllable Relighting of Photographs. In *Proc. SIGGRAPH*.

Paul Debevec. 1998. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proc. SIGGRAPH*.

Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang. 2022. *Mitsuba 3 renderer*. https://mitsuba-renderer.org.

Kevin Karsch, Varsha Hedau, David Forsyth, and Derek Hoiem. 2011. Rendering synthetic objects into legacy photographs. *ACM Trans. Graph.* (2011).

Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. 2025. MoGe: Unlocking Accurate Monocular Geometry Estimation for Open-Domain Images with Optimal Training Supervision.