Differentiable Shadow Mapping for Efficient Inverse Graphics Supplementary Material

Markus Worchel Marc Alexa

TU Berlin

1. Implementation and Renderer Details

We implemented our method in Python, on top of the automatic differentiation framework PyTorch [13] and use the differentiable rasterization primitives by Laine et al. [10] as foundation for our renderer. Our rendering pipeline follows a deferred shading architecture [4], so we first rasterize the scene geometry and then perform a shading pass that computes the light-material interaction. This architecture allows us to consider an arbitrary number of lights. Currently, all surfaces use a BRDF (bidirectional reflectance distribution function) with only the Lambert diffuse term but an extension to more physically-based material models, like the Disney "principled" BRDF [3] or variants of it [7, 9], is trivial.

For rendering the shadow maps, we use the same rasterization primitives as for the primary camera. The camera built for each light depends on the light type: (1) a directional light uses a camera with orthographic project that is placed far outside of the scene, facing the origin from the specified direction; (2) a spot light uses a camera that is placed at its position, facing the desired direction, and using a perspective projection with a configurable field of view.

2. Experiment Details

Our scenes are roughly centered at the origin and after loading 3D models, we scale them to the $[-1, 1]^3$ cube. This allows us to use consistent near and far planes across experiments, both for the frusta of the actual cameras and the frusta of the light cameras.

For all experiments – except those explicitly stating otherwise – we use a shadow map resolution of 256×256. We implement two filters for pre-filtering the shadow map: a box filter and a Gaussian filter. We use both filters in the experiments, usually with kernel size $k = 3$ or $k = 5$. The Gaussian filter produces more pleasing visual results whereas using the box filter is a little faster. Generally, we observed robustness to the exact choice of filter.

Convergence and Jacobian Experiments. We use a "minimal plane" setup for the convergence and Jacobian experiments, consisting of a single planar shadow receiver and a single planar occluder (both parallel), illuminated by a directional light orthogonal to them. The camera that produces the shadow images used for optimization is located between the receiver and the occluder, focusing the receiver. The highly tessellated and slightly rotated occluder that we use for computing the Jacobian has 130,000 faces. For the simple experiment that investigates the effect of shadow map filtering on the convergence, we displace the occluder horizontally and the goal of the optimization is to recover the original position, only by using the shadow images.

Light Direction Estimation. The experiment of light direction estimation has the following general setup: an object is placed on a rectangular floor plane and illuminated by n directional lights. We render a reference image with a fixed camera that observes the object. The goal of the optimization is to recover the target light directions from some initial state, only by comparing the rendered image to the target image.

We run the experiment for 6 objects in total. For each parameter setting (e.g. filter size) we perform 150 runs of the experiment, with randomized target and initial conditions. We ensure that these configurations are deterministic, so – as an example – the optimizations for filter size $k = 3$ and $k = 15$ use the same set of 150 configurations. The average accuracy is computed as an average over the 6 objects and 150 runs; our error bounds are based on the standard deviation, respectively.

The accuracy (alignment) is computed as follows: Given the n target light directions $\{\mathbf{l}_i\}$ and the estimated light directions $\{\mathbf{l}_j^{\text{opt}}\}$, we first perform a greedy matching between both sets, trying to maximize the dot product $\mathbf{l}_i \cdot \mathbf{l}_{k_i}^{\text{opt}}$ between a direction i and its match k_i. Note that this mapping is bijective, so there is a one-to-one correspondence between directions. We then compute the alignment as the average dot product over all directions:

$$\frac{1}{n} \sum_i \mathbf{l}_i \cdot \mathbf{l}_{k_i}^{\text{opt}} \tag{1}$$

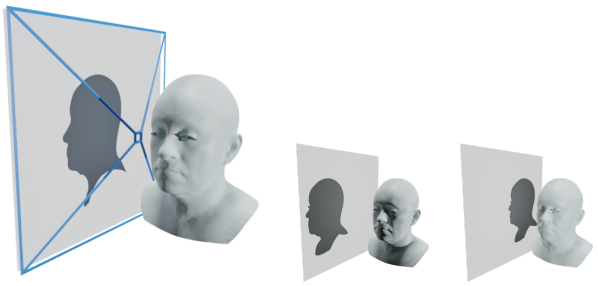

Figure 1. General setup of the face reconstruction experiment. The face model is placed in front of a planar shadow receiver and the camera (blue) is placed between them, focusing the receiver (left). For the experiments with three shadows, we add two additional lights with small horizontal offsets (middle, right).

Monocular Pose Estimation. The general setup for the pose estimation application is as follows: the scene consists of a single planar receiver that is illuminated by a directional light orthogonal to it. The object whose pose we wish to recover is placed in front of the receiver, such that it casts a shadow. For each optimization run, we first randomly offset the object in the plane parallel to the receiver and randomly rotate it around its up-axis (the random offsets and rotations are deterministic). Then, we recover the original pose in an optimization, which compares rendered images to a reference image that shows the object without offset or rotation. The camera is located at a position where it sees both the object and the shadow receiver plane.

We use Mitsuba 3 [6] for creating the reference images and as a baseline in the optimization. Since Mitsuba 3 does not support differentiation for directional lighting, we use a far-away, rectangular area light as a proxy in the optimization. We calibrate this proxy to match the directional light. In our experiments, timings for Mitsuba 3 showed no noticeable dependence on the choice of emitter.

We use ADAM [8] for the optimization, with a step size of 0.01 and $\beta_1 = 0.9$, $\beta_2 = 0.999$ and run 150 iterations.

Face Reconstruction from Profile Shadows. The basic setup for face reconstruction is similar to that of pose estimation: a single planar shadow receiver is illuminated by a directional light orthogonal to it. The head model is placed in front of the plane so that is casts a shadow. However, here the camera is located between the head and the receiver, focusing the receiver (see Figure 1). For the setup with three shadows, we add two additional directional lights.

We use artificial geometry to evaluate the quality of our fit quantitatively. When computing the geometric distance between the reference geometry and our prediction, we only compare the meshes from the neck upwards, because the shadow of the upper torso is not included in the optimization.

In the experiment that uses silhouette images of real public figures, we first scale the images and align them roughly with the shadow observed by the camera in our scene. We also optimize the translation of the head model to reduce the effect of small inaccuracies in this alignment.

We use ADAM [8] with a step size of 0.01 for the translation and a step size of 0.2 for the identity weights used by the morphable face model, both with $\beta_1 = 0.9$, $\beta_2 = 0.999$. We run the optimization for 400 iterations.

Shadow Art. The basic shadow art setup is again similar to the setup for face reconstruction and pose estimation: a single shadow receiver illuminated by an orthogonal directional light with the geometry that is optimized in front of the plane. The experiments with two views use a second shadow receiver that is placed orthogonal to the first one, illuminated by a second directional light orthogonal to it (see Figure 2). For each receiver plane, we place an orthographic camera between the object and the plane: these are the cameras producing the shadow images used in the optimization.

In selected experiments, we use a perspective camera that observes the receiver plane from the side, demonstrating that light rays and view rays do not need to coincide with our framework. We also replace the planar shadow receiver with a curved receiver to show that the receiver does not necessarily need to be flat. In our demo code, we include a shadow art sample, which reproduces these settings for a single view.

The shadow art experiments that deform a mesh start with a sphere of 12,800 triangles and finish in roughly 15 seconds for each object. We run 400 iterations with ADAM [8], using a step size of 0.2, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\lambda = 20$ for gradient preconditioning [12], and a weight of 0.2 for the normal consistency regularization.

The shadow art experiments that optimize the translations, scales, and rotations of an object collection initially start with one instance of the object and duplicate the number of objects every 150 iterations (each object spawns two objects with small offsets to avoid interpenetration). We run 1100 iterations with ADAM [8], using a step size of 0.005, $\beta_1 = 0.9$, $\beta_2 = 0.999$.

Interactive Modeling from Shadows. The interactive modeling setup is equivalent to the one-view shadow art setup. In contrast to shadow art, the intend here is to modify an *existing* model based on user interaction with the shadow image. We first load an input mesh, generate a shadow image from the unmodified mesh, and write the image to the disk. The optimization loop then reads back this shadow image in each iteration and uses it as a target image to drive the model deformation.

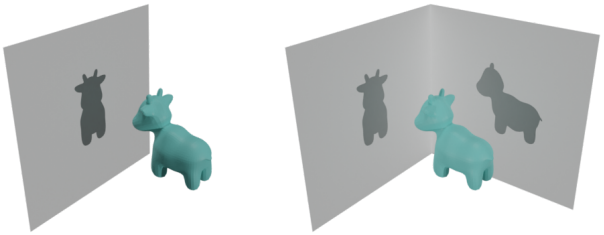

Figure 2. Shadow art setup for one view (left) and two views (right), with a reference object.

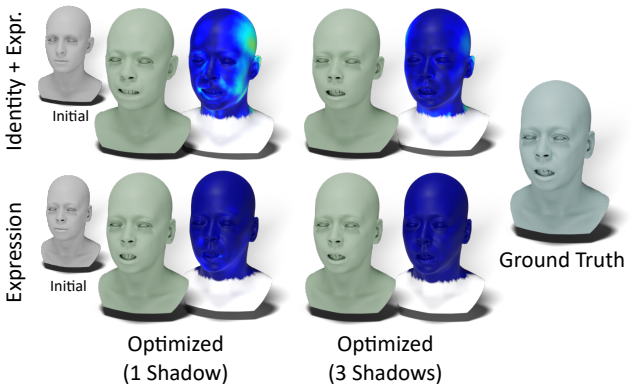

Figure 3. We use a morphable face model that allows controlling the face identity and the expression. We optimize both (top row) and only the expression (bottom row) based on the shadows cast by the model.

3. Additional Results

Monocular Pose Estimation. Table 1 contains the quantitative results for all resolutions (128^2, 512^2, 1024^2) as well as time measurements per iteration. We can perform approximately 70 (128^2), 40 (512^2), and 20 (1024^2) optimization steps per second, which allows following the optimization in real-time.

Face Reconstruction from Profile Shadows. Table 2 contains the quantitative results for four artificial faces models. In the main paper, we only optimize the identity weights of the morphable face model, which effectively determine the fundamental *shape* of the face. We also experimented with optimizing the identity and expression as well as only the expression, while keeping the identity fixed (see Figure 3). The latter might be useful if the shape of the head is known (e.g. because it was scanned previously) and the task is to recover facial expressions using shadows, for example in the context of face motion capture.

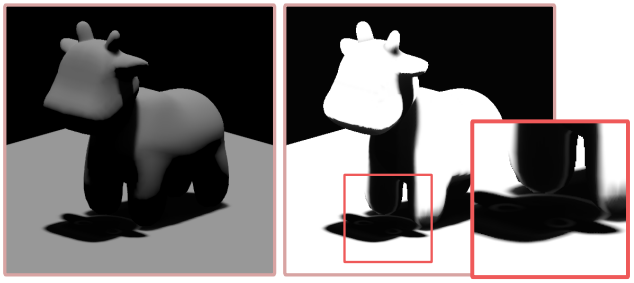

Figure 4. Shaded image (left) and visibility (right) with visible light bleeding artifact.

4. Limitation: Light Bleeding

It is well known that Variance Shadow Maps [5] – our particular choice of pre-filtered shadow maps – suffer from light bleeding in regions with high depth variance (see Figure 4). While we noticed no immediate impact in our experiments, it might be desirable to reduce light bleeding by using heuristics [11] or implementing other pre-filtered shadow maps [1, 2, 14] in our framework.

References

[1] Thomas Annen, Tom Mertens, Philippe Bekaert, Hans-Peter Seidel, and Jan Kautz. Convolution shadow maps. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, page 51–60, Goslar, DEU, 2007. Eurographics Association. 3

[2] Thomas Annen, Tom Mertens, Hans-Peter Seidel, Eddy Flerackers, and Jan Kautz. Exponential shadow maps. In *Proceedings of Graphics Interface 2008*, GI '08, page 155–161, CAN, 2008. Canadian Information Processing Society. 3

[3] Brent Burley. Physically Based Shading at Disney. In *SIGGRAPH Courses: Practical Physically Based Shading in Film and Game Production*, 2012. 1

[4] Michael Deering, Stephanie Winner, Bic Schediwy, Chris Duffy, and Neil Hunt. The triangle processor and normal vector shader: A vlsi system for high performance graphics. In *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '88, page 21–30, New York, NY, USA, 1988. Association for Computing Machinery. 1

[5] William Donnelly and Andrew Lauritzen. Variance shadow maps. In *Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games*, I3D '06, page 161–165, New York, NY, USA, 2006. Association for Computing Machinery. 3

[6] Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang. Mitsuba 3 renderer, 2022. https://mitsuba-renderer.org. 2, 4

[7] Brian Karis. Real shading in unreal engine 4. *SIGGRAPH 2013 Course: Physically Based Shading in Theory and Practice*, 2013. 1

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun,

Table 1. Quantitative results for the pose estimation experiments for different scenes and resolutions. We measure the rotation error $\Delta\varphi$, the translation error $\Delta\mathbf{t}$, the total runtime, and runtime per iteration, averaged over ten runs. We include results for a GPU with ray-tracing cores ("RT") and for a setting where we use our renderer as reference ("Our Ref."). Best scores in **bold**, second best scores underlined.

	Mitsuba 3 [6]			Rasterizer + Shadows (Ours)			Rasterizer		
	$\downarrow \Delta\varphi[^\circ]$	$\downarrow \Delta\mathbf{t}$	$\downarrow$ time [s] (/it)	$\downarrow \Delta\varphi[^\circ]$	$\downarrow \Delta\mathbf{t}$	$\downarrow$ time [s] (/it)	$\downarrow \Delta\varphi[^\circ]$	$\downarrow \Delta\mathbf{t}$	$\downarrow$ time [s] (/it)
128×128									
Bunny	**0.31**	**0.05**	54.13 (0.361)	0.69	0.27	2.27 (0.015)	3.04	5.56	**1.06** (**0.007**)
Dragon	**2.78**	4.01	52.46 (0.349)	3.32	**3.94**	2.09 (0.014)	8.01	14.31	**1.03** (**0.007**)
Hand	**1.44**	5.24	53.17 (0.354)	3.66	**4.02**	2.08 (0.014)	6.80	12.42	**1.03** (**0.007**)
Spot	**0.02**	**0.11**	58.19 (0.388)	7.40	12.83	2.11 (0.014)	12.47	44.70	**1.05** (**0.007**)
512×512									
Bunny	**0.23**	**0.10**	325.29 (2.168)	0.33	0.22	3.76 (0.024)	0.31	0.26	**1.58** (**0.009**)
Dragon	**2.86**	**3.91**	306.46 (2.043)	3.32	4.53	3.64 (0.023)	8.02	11.63	**1.54** (**0.009**)
Hand	**1.41**	**5.18**	317.40 (2.116)	1.68	5.36	3.66 (0.023)	1.85	5.66	**1.51** (**0.009**)
Spot	**0.03**	**0.02**	381.80 (2.545)	0.05	0.05	3.76 (0.023)	5.80	23.15	**1.77** (**0.010**)
Spot (RT)	**0.03**	**0.02**	53.26 (0.354)	0.05	0.05	1.93 (0.012)	6.92	21.52	**1.05** (**0.006**)
Spot (Our Ref.)	0.05	0.12	384.24 (2.561)	**0.02**	**0.06**	3.60 (0.022)	5.83	23.12	**1.71** (**0.010**)
1024×1024									
Bunny	**0.22**	**0.12**	1245.60 (8.304)	0.31	0.21	7.72 (0.047)	0.33	0.25	**3.71** (**0.021**)
Dragon	**2.86**	**3.87**	1151.74 (7.678)	3.22	4.48	7.53 (0.046)	7.85	11.48	**3.18** (**0.018**)
Hand	**1.42**	**5.38**	1200.74 (8.005)	1.61	5.43	7.72 (0.048)	1.76	5.66	**3.13** (**0.018**)
Spot	**0.03**	**0.03**	1441.56 (9.610)	**0.03**	0.04	7.64 (0.046)	2.58	3.93	**3.19** (**0.017**)

Table 2. Quantitative results for face reconstruction from shadows, using artificial head geometry generated with a morphable face model. Best scores in **bold**.

	1 Shadow			3 Shadows		
Face ID	$\downarrow D_{\text{mean}}$	$\downarrow D_{\text{Hausdorff}}$	$\downarrow$ time [s] (/it)	$\downarrow D_{\text{mean}}$	$\downarrow D_{\text{Hausdorff}}$	$\downarrow$ time [s] (/it)
1	0.011	0.114	**18.86** (**0.047**)	**0.006**	**0.066**	27.11 (0.068)
6	0.012	0.108	**19.00** (**0.048**)	**0.010**	**0.106**	27.32 (0.068)
8	0.014	0.129	**19.36** (**0.048**)	**0.007**	**0.054**	27.37 (0.068)
131	0.014	0.069	**18.09** (**0.045**)	**0.008**	**0.046**	27.46 (0.069)
mean	0.013	0.105	**18.83** (**0.047**)	**0.008**	**0.068**	27.32 (0.068)

editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 2

[9] Sebastien Lagarde and Charles de Rousiers. Moving frostbite to physically based rendering 3.0. *SIGGRAPH 2014 Course: Physically Based Shading in Theory and Practice*, 2014. 1

[10] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Trans. Graph.*, 39(6), nov 2020. 1

[11] Hubert Nguyen. *Gpu Gems 3*. Addison-Wesley Professional, first edition, 2007. 3

[12] Baptiste Nicolet, Alec Jacobson, and Wenzel Jakob. Large steps in inverse rendering of geometry. *ACM Trans. Graph.*, 40(6), dec 2021. 2

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc. 1

[14] Christoph Peters and Reinhard Klein. Moment shadow mapping. In *Proceedings of the 19th Symposium on Interactive 3D Graphics and Games*, i3D '15, page 7–14, New York, NY, USA, 2015. Association for Computing Machinery. 3