Advanced Optimization for Machine Learning in Computer Graphics

Michael Fischer

University College London, United Kingdom

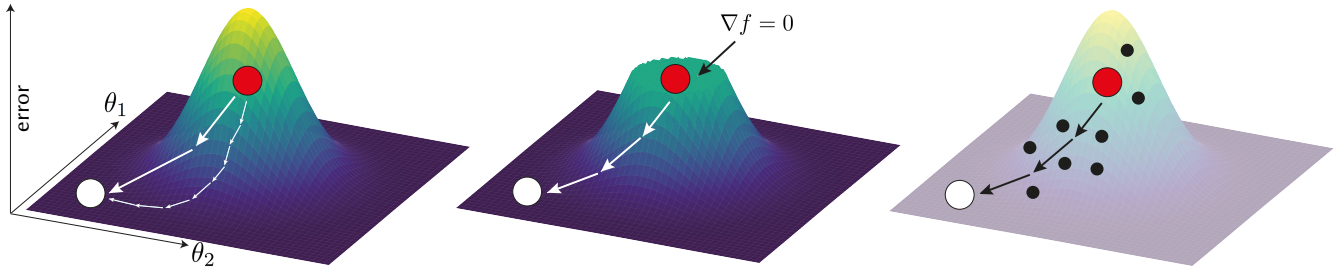

Figure 1: *We research methods to achieve faster convergence (left: Metappearance, Sec. 3.1), remove plateaus from the cost landscape (middle: PRDPT, Sec. 3.2), and enable gradient-based optimization on forward models with unknown cost landscapes (right: ZeroGrads, Sec. 3.3).*

Abstract

Gradient-based optimization has become the de-facto standard for optimization in the graphics community. However, it is rarely questioned whether we can do "better" than simple gradient descent (GD) - in fact, it is easy to construct examples where GD is easily outperformed by other methods. My research to date explores such scenarios from three distinct angles: Firstly, we show that we can use meta-learning to optimize over GD itself, accelerating convergence times by several orders of magnitude at similar visual quality. Secondly, we replace traditional (and sometimes uninformative) gradients by a variational formulation, which enables differentiable rendering on problems where previous approaches did not converge due to plateaus in the cost landscape. Finally, we explore the use of a neural network as a surrogate loss, which we can differentiate to yield low-variance gradients on otherwise non-differentiable problems, enabling successful optimization of high-dimensional inverse problems.

1. Introduction

In the course of daily activities, a modern graphics researcher will encounter numerous optimization challenges - be it the "learning" of a neural network or solving an inverse problem like estimating reflectance maps from a single flash image. Both tasks, for a successful optimization outcome, require an efficient traversal of the cost landscape and are typically addressed using (stochastic) gradient descent, often in combination with optimizers like Adam. Yet, a straightforward application of gradient descent can be inefficient, for example due to regions of shallow slopes causing prolonged convergence times, or plateaus – regions of zero gradient – halting the optimization altogether. Moreover, gradient descent via automatic differentiation (AD) is only applicable in scenarios where the forward model is differentiable, which excludes a large part of modern graphics tools, such as Blender or Photoshop.

My PhD research is dedicated to exploring these issues and identifying effective solutions. I commenced my doctorate studies at University College London (UCL) in November 2020, under the guidance of Tobias Ritschel and Niloy Mitra, and am now in the third year of a four-year program. My experience includes a summer internship at Meta Reality Labs in Redmond, Washington, and I will be joining Adobe Research London as a research scientist intern this summer, anticipating graduation from my doctorate studies in early 2025.

In this manuscript, I will first present an overview of relevant literature in Sec. 2, proceed to detail the projects undertaken during my PhD in Sec. 3, and conclude with a discussion on the forthcoming work and anticipated contributions in Sec. 4.

2. Related Work

Gradient Descent is a well-studied technique that has applications in many real-world graphics and vision tasks. However, it is evident that, on a plateau, gradient *descent*, by definition, will stall since there is no gradient, effectively hindering or preventing successful optimization altogether. It is trivial to construct such a scenario, and plateaus are surprisingly common in real-life rendering applications such as rasterization (triangle- and depth-tests). Our work Plateau-Reduced Differentiable Path Tracing (PRDPT) takes inspiration from differentiable rasterization [LLCL19, PGBD22], where

plateaus are smoothed by employing smooth approximations of the step function, and derives a variational formulation [SB12] for gradient descent in inverse path tracing, which leads to improved convergence on inverse path tracing problems.

Derivative-Free Optimization (DFO), can instead be used when we do not have information about the loss landscape and thus cannot compute its gradient analytically. However, we have the ability to "sample" this loss landscape by running the forward model with a specific set of parameters that we would like to sample the loss for, and then either try to estimate the gradient from these samples, or employ search-based algorithms. Search-based methods (direct search, genetic algorithms, CMA-ES, particle-swarm optimization, simulated annealing, ...) often employ heuristic search criteria (e.g., the fitness and procreation rates in genetic algorithms) and thus do not scale well to higher dimensions, and need a high number of function evaluations for convergence. Gradient estimation methods, on the other hand, suffer from high per-iteration cost (e.g., finite differences, SPSA, PRDPT) and noisy gradient estimates, especially in higher-dimensional settings.

Meta-learning finally transcends "regular" learning by optimizing over the optimization itself. In our case, we use the model-agnostic meta learning (MAML) framework [FAL17], and optimize for an ideal model initialization and per-parameter learning rate [LZCL17]. This can be interpreted as learning from previous optimization runs, i.e., building up intuition about the shape of the loss landscape, and how it is best traversed. Moreover, the intricacies of the algorithm allow for insights into data-sampling and real-time training of models with sparse observations.

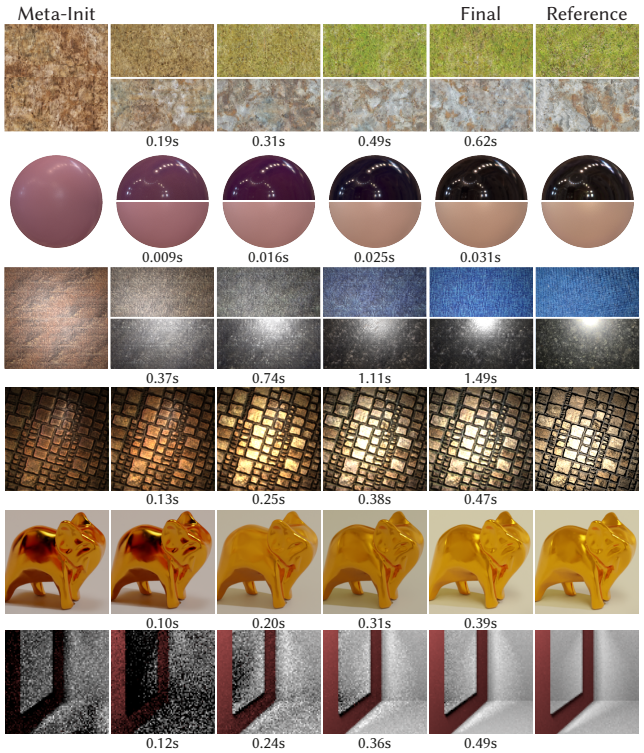

Figure 2: *Convergence at 25, 50 and 75% of the inner-loop gradient steps (columns). Metappearance faithfully encodes the reference at interactive runtimes.*

3. Current Work

3.1. Metappearance

The training paradigms for neural networks for visual appearance reproduction can be grouped into three categories: I) general models, where a network is trained on a large corpus of data and then can run inference on unseen data points, II) over-fitting, where each network is overfitted onto a single data point (e.g., a NeRF overfitted to a scene), and III) fine-tuning, a hybrid of both, where a general model's prediction is quickly overfitted onto a specific datapoint. General models are fast as they perform feed-forward inference, but often lack accuracy, as they must compress the entire dataset into a limited set of weights, whereas overfitted models usually are very accurate, but slow to train and, once trained, not adaptable anymore.

In Metappearance [FR22], we employ *meta-learning* to combine the advantages of both methods: the fast forward-times of general networks and their applicability to new, unseen datapoints with the accuracy and high quality of overfitted networks. Metappearance employs a nested optimization loop: in the inner-loop, we randomly sample an optimization task (e.g., overfitting an MLP to a BRDF) and perform a small number (≤ 10) of gradient descent steps towards the reference solution for this task. In the subsequent outer loop, we then compute the gradient of the resulting network w.r.t. its initialization – the "meta-gradient" – and change the initialization accordingly, such that the next inner-loop iteration yields improved performance. By iterating this process over many random-sampled optimization tasks (e.g., overfitting all MERL BRDFs), our meta-learned network can learn to encode strong data-specific priors into its initialization (mimicking the generality principle from general networks). Moreover, the meta-learned network is forced to make do with only a small number of gradient descent steps, leading to fast convergence, and is explicitly encouraged to provide high-fidelity output through the meta gradient descent. An often-overlooked aspect of this algorithm is that it also encourages high-quality reproductions with scarce data: if a GD step in the inner-loop uses batchsize b, and we perform k such steps, the total number of samples seen by the learner is kb, which – for the case of MERL BRDFs – is up to 99.5% less than overfitting the entire BRDF, and yet still achieves similar visual quality.

We show results on a wide variety of visual appearance tasks (textures, BRDFs, svBRDFs, illumination, and even the full light transport in a scene, cf. Fig. 2) and extensively evaluate how Metappearance compares to the traditional training paradigms in terms of final visual fidelity and training time (Fig. 3). Metappearance also serves as the foundation of a subsequent publication where we learn the ideal sampling-pattern

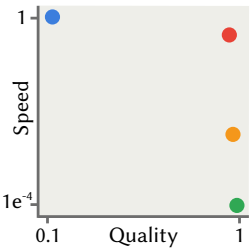

Figure 3: *Quality-Speed plot for the training of general, overfitted, fine-tuned and meta-learned models.*

for BRDF reconstruction from sparse samples [LFR23].

3.2. Plateau-Reduced Differentiable Path Tracing

In addition to optimizing over the loss landscapes, we also need to ensure that they are continuous and well-behaved. As detailed in Sec. 2, this is often not the case in inverse rendering scenarios, as we compare in image space only. A a small motivational example is shown in Fig. 4: the task here is to move the blue sphere to its reference position. Gradient-based optimization will converge only in case a), whereas in b) and c) we are on a plateau, i.e., the image-space loss does not change when moving the sphere, and a change in parameter thus does not lead to a change in observed loss, leading to a zero gradient and to the optimizer not being able to recover the correct position.

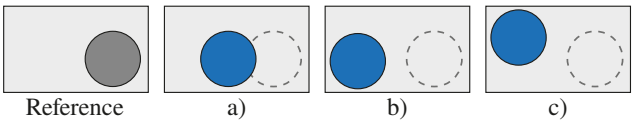

Figure 4: *An example of a plateau in the cost landscape: a) will converge, b) and c) will not due to a plateau in the cost landscape.*

To alleviate this problem, we take inspiration from differentiable rendering and the concept of "differentiating through blurring", and propose to convolve the rendering equation with an additional blur kernel. The rendering equation describes each pixel P as the integral over the renderer $\mathcal{R}$ executed for all light paths x in Ω, i.e.,

$$P(\theta) = \int_{\Omega} \mathcal{R}(x,\theta)\,dx, \tag{1}$$

We now introduce an additional blur kernel over the parameter space Θ, leading to the *smooth* rendering equation:

$$\begin{aligned} Q(\theta) = \kappa \star P(\theta) &= \int_{\Theta} \kappa(\tau) \int_{\Omega} \mathcal{R}(x,\theta-\tau)\,dx d\tau \\ &= \int_{\Theta\times\Omega} \kappa(\tau)\mathcal{R}(x,\theta-\tau)\,dx d\tau, \end{aligned} \tag{2}$$

which has the gradient estimator

$$\widehat{\frac{\partial Q}{\partial \theta}} \approx \frac{1}{N}\sum_{i=1}^{N} \underbrace{\frac{\partial \kappa}{\partial \theta}(\tau_i)}_{\text{Diff. Kernel}} \underbrace{P(\theta-\tau_i)}_{\text{Renderer}}, \tag{3}$$

In our work PRDPT [FR23a], we derive an efficient estimator for the smooth rendering equation and its gradient and show two possible implementations thereof. We employ a Gaussian smoothing kernel, as it a) has infinite support, i.e., removes plateaus everywhere, b) has a C^1-continuous derivative, and c) has a closed-form probability density function (PDF), allowing importance-sampling the derivative kernel via the inverse CDF method.

As our formulation captures the full behaviour of light transport, we can solve inverse problems including caustics, shadows, global illumination and other sophisticated light transport effects which differentiable rasterizers can neither model nor optimize. Moreover, our smoother formulation allows us to descent along a smoothed version of the original objective function, leading to improved convergence in inverse rendering tasks, as shown in Fig. 5.

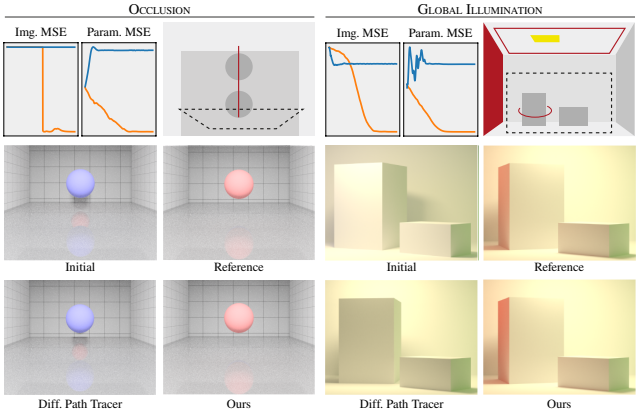

Figure 5: *Inverse rendering problems with our formulation and Mitsuba [NDVZJ19]. The tasks and variable parameters are illustrated in the top right subfigure of each experiment. Mitsuba often fails to achieve the desired outcome as it lands on a plateau and cannot recover.*

3.3. Zero Grads: Learning Local Surrogate Losses

While we have previously shown that we can optimize gradient descent itself and smooth the loss landscape's plateaus, all these techniques are of little use when we do not have information about the actual loss landscape. As discussed in Sec. 2, employing traditional gradient-free optimizers is infeasible here, as graphics problems often reach high dimensionality, and densely sampling the loss landscape (as is often done for off-line surrogate learning [GSST19]) usually is expensive, as it requires a full rendering or simulator run.

In this work, we thus combine ideas from response surface modelling [KM10] with our previous variational formulation and propose a method to learn the loss landscape on-the-fly (i.e., during parameter optimization) by sparsely sampling around the current parameter position and then fitting a neural interpolant to these samples. The intuition behind the locality-principle is that the exact shape of the cost landscape in remote regions is unimportant for the current gradient step. The neural network, our *surrogate* h_ϕ, thus learns the mapping $h_\phi(\theta) \to \mathcal{L}(\theta)$, i.e., it maps parameter values θ to loss values $\mathcal{L}(\theta)$. As the surrogate is an analytic forward model which can be readily differentiated by modern deep-learning frameworks, querying the parameter gradient for gradient descent now reduces to a simple gradient-query of the network: $\partial_\theta = \partial h_\phi(\theta)/\partial\theta$.

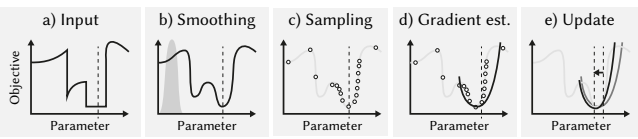

Figure 6: *Conceptual illustration of the steps in ZeroGrads.*

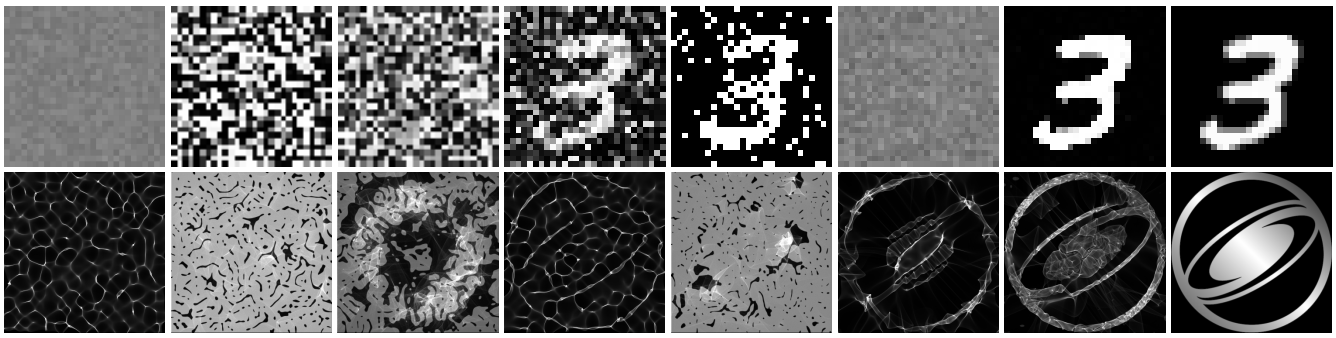

Figure 7: *Comparison of our method, ZeroGrads (Sec. 3.3) against traditional derivative-free optimizers (FR22 implements PRDPT (Sec. 3.2)). In the upper row, we learn an MLP that replicates digits from the MNIST dataset (approx. 35,000 optimization parameters). In the lower row, we optimize the height-field of a glass slab such that it refracts incoming light to resemble a reference caustic (1,024 optimization parameters).*

We further can derive an unbiased gradient estimator for the surrogate gradient $\partial\phi$ on the smooth cost landscape, but omit this part for brevity and refer to [FR23b] for details.

We show results on two high-dimensional optimization tasks in Fig. 7 and hypothesize that the superiority of our method can be explained by the centerpiece of our approach, the *neural proxy*: in contrast to PRDPT and SPSA, ZeroGrads uses a neural network as proxy function, whose state acts as hysteresis and endows our method with inertia, limiting the estimated losslandscape's spatiotemporal change by the network's adaptability. PRDPT and SPSA, in contrast, re-build a (linear) gradient estimate during every iteration of the optimization, effectively ignoring information about the loss-landscape from previous iterations. As this gradient estimate is a stochastic approximation, it will exhibit noise and variance, which highlights the main difference between the approaches: while SPSA and PRDPT estimate the *parameter gradient* ∂_θ (subject to variance), ZeroGrads estimates the surrogate gradient ∂_ϕ, but *analytically* computes the parameter gradient ∂_θ. This allows us to move the higher-variance estimate into the neural network's parameter update, where the estimate's noise is smoothed by the aforementioned hysteresis.

To analyze this behaviour, we plot the variance of the gradient-magnitude over the course of the optimization in Fig. 8 (here specifically for the caustic optimiziation, cf. Fig. 7) and find that the vastly different scales on the y-axes of each subplot confirm our hypothesis: the variance in the gradient-magnitude of our method is orders of magnitude lower than that of the other approaches. While this does not allow reasoning about the *correctness* of the derived gradients, it explains why our approach outperforms the competitors in the provided examples and scales favourably to very high (tested up to 250k) dimensions.

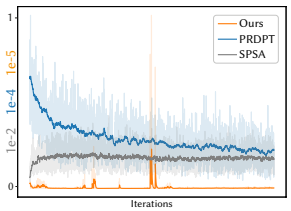

Figure 8: *Gradient variance over the course of optimization. Note the different scales on the y-axis.*

4. Future Work and Expected Contributions

For my future PhD work, I plan to apply the insights gained from my previous work to two new domains. Firstly, it would be interesting to investigate whether we can smooth the sampling trajectory in stochastic diffusion models (e.g., DDPMs) by applying our variational formulation to the update rule. This could enable smoother trajectories and hence faster sampling due to fewer sampling steps while retaining high-quality outputs.

Secondly, an interesting area of research would be the design of sampling algorithms that lead to improved *gradients* during inverse MC path tracing - similar to what variance reduction techniques such as importance sampling achieve for the primal (forward) rendering. Gradient descent most likely does not need a fully converged image (as long as there is enough signal to discern the influence of the optimization parameters), but can allot the ray sampling budget such that it produces more informative gradients which will lead to less variance and hence faster optimization times.

5. Acknowledgements of Funding

My tuition fees at UCL and my living stipend are funded by Meta Reality Labs, Grant Nr. 5034015. I am further supported by a GPU donation from Meta and a recipient of the 2024 Rabin Ezra scholarship.

References

[FAL17] FINN C., ABBEEL P., LEVINE S.: Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning* (2017), PMLR, pp. 1126–1135. 2

[FR22] FISCHER M., RITSCHEL T.: Metappearance: Meta-learning for visual appearance reproduction. *ACM Transactions on Graphics (TOG) 41*, 6 (2022), 1–13. 2

[FR23a] FISCHER M., RITSCHEL T.: Plateau-reduced differentiable path tracing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 4285–4294. 3

[FR23b] FISCHER M., RITSCHEL T.: Zero grads ever given: Learning local surrogate losses for non-differentiable graphics. *arXiv preprint arXiv:2308.05739* (2023). 4

[GSST19] GRABOCKA J., SCHOLZ R., SCHMIDT-THIEME L.: Learning surrogate losses. *arXiv preprint arXiv:1905.10108* (2019). 3

[KM10] KHURI A. I., MUKHOPADHYAY S.: Response surface methodology. *Wiley Interdisciplinary Reviews: Computational Statistics 2*, 2 (2010), 128–149. 3

[LFR23] LIU C., FISCHER M., RITSCHEL T.: Learning to learn and sample brdfs. In *Computer Graphics Forum* (2023), vol. 42, Wiley Online Library, pp. 201–211. 3

[LLCL19] LIU S., LI T., CHEN W., LI H.: Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2019), pp. 7708–7717. 1

[LZCL17] LI Z., ZHOU F., CHEN F., LI H.: Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835* (2017). 2

[NDVZJ19] NIMIER-DAVID M., VICINI D., ZELTNER T., JAKOB W.: Mitsuba 2: A retargetable forward and inverse renderer. *ACM Transactions on Graphics (TOG) 38*, 6 (2019), 1–17. 3

[PGBD22] PETERSEN F., GOLDLUECKE B., BORGELT C., DEUSSEN O.: Gendr: A generalized differentiable renderer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 4002–4011. 1

[SB12] STAINES J., BARBER D.: Variational optimization. *arXiv preprint arXiv:1212.4507* (2012). 2