Multi-Fidelity Optimization for Inverse Material Rendering

Victor Stenvers
Utrecht University
Utrecht, Netherlands
v.stenvers@uu.nl

Peter Vangorp
Utrecht University
Utrecht, Netherlands
p.vangorp@uu.nl

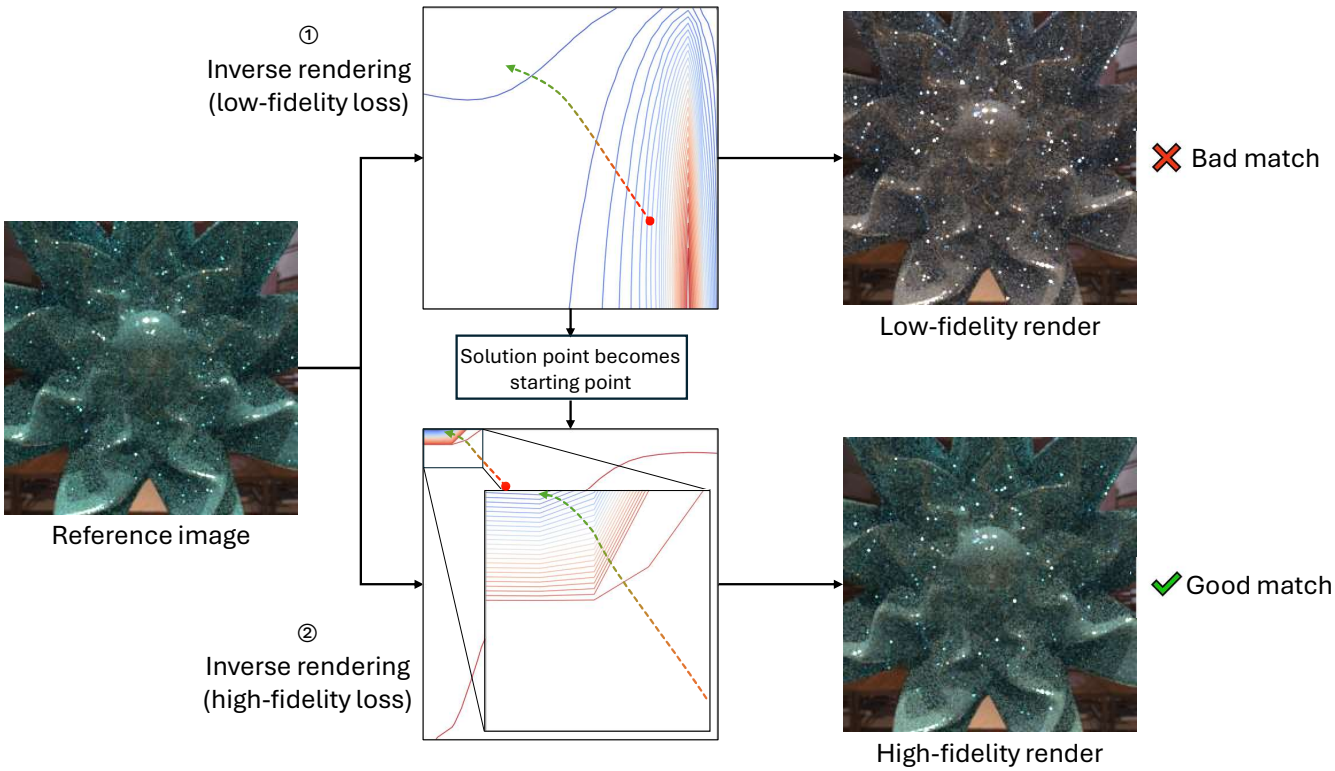

Figure 1: An overview of the multi-fidelity optimization (MFO) method for inverse material rendering. Given a reference image and initial scene information, the method optimizes for a matching material using a two-step approach. First, we optimize using a low-fidelity (LF) metric, generating a fast approximation of the reference. Then, using the scene information from the LF optimization as a starting point for a second optimization process, we optimize using a high-fidelity (HF) metric to refine the result. To illustrate the inner workings of the MFO method, we show representative loss surfaces with a plotted optimizer path, indicating its starting point with a red circle.

Abstract

Image quality metrics are typically used as loss functions for inverse material rendering. However, all these metrics have drawbacks. When optimizing with low-fidelity (LF) metrics, shorter runtimes can be achieved, but this also produces lower quality results. On the other hand, optimizing using high-fidelity (HF) metrics produces higher quality results, but causes significantly longer runtimes. We present an alternative method of applying image quality metrics to inverse material rendering by using multi-fidelity optimization (MFO). This optimization method is a two-stage process, where an LF optimization is utilized to quickly generate the starting point for an HF optimization. By evaluating the MFO approach for multiple material optimization scenarios, we show that this method produces high quality results, while having shorter runtimes than optimizations applying HF metrics. Given the efficacy of applying multi-fidelity optimization to inverse material rendering, we expect this technique to be useful for any optimization applying image quality metrics, encouraging a broader evaluation of MFO in inverse rendering.

CVMP '25, London, United Kingdom

ACM ISBN 979-8-4007-2117-5/25/12
https://doi.org/10.1145/3756863.3769712

CCS Concepts

• **Computing methodologies** → **Rendering**; • **Mathematics of computing** → *Continuous optimization.*

Keywords

Perception, Image Quality Metrics, Evaluation, Computer Graphics

ACM Reference Format:
Victor Stenvers and Peter Vangorp. 2025. Multi-Fidelity Optimization for Inverse Material Rendering. In *European Conference on Visual Media Production (CVMP '25), December 03–04, 2025, London, United Kingdom.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3756863.3769712

1 Introduction

Differentiable rendering, Inverse rendering [Patow and Pueyo 2003; Marschner 1998], a method for retrieving a scene description from a given image by inverting the rendering process, is a prevalent technique in the field of material optimization [Meyer et al. 2025; Ramamoorthi and Hanrahan 2023; Yu et al. 1999].

However, the current image quality metrics [Kavoosighafi et al. 2025] all have drawbacks. Low-fidelity metrics such as the mean square error (MSE) applied over pixel differences are fast, but these metrics do not always produce a result close to the reference render, as shown in figure 2b. While this issue can be resolved by using a high-fidelity metric such as HDR-ꟻLIP [Andersson et al. 2021], as shown in figure 2c, these metrics strongly impact the performance of the inverse rendering process.

In this work, we introduce a multi-fidelity optimization method [Gano et al. 2005; Zhou et al. 2023; Li and Li 2024] for material optimization using inverse rendering: by using the faster, but lower quality result from the LF optimization as the starting point for the slower HF optimization process, we improve the performance of the overall optimization process while also increasing the quality of the results.

We evaluated this optimization method for multiple material optimization contexts, investigating the runtimes and quality of the results for multiple image quality metrics.

This paper makes the following contributions:

- An evaluation of a multi-fidelity optimization method in various scenarios in the material optimization context, comparing the runtime and quality of the various results with optimizations applying other image quality metrics.
- We demonstrate a multi-fidelity optimization method for material optimization using inverse rendering. Showing improved runtime performance when compared with high-fidelity metrics, as well as an increase in quality over low-fidelity metrics and non-converging high-fidelity metrics.

2 Related Work

2.1 Inverse Rendering

Inverse rendering aims to perform the inverse function of rendering. While the goal of rendering is to transform a given scene description into an image, inverse rendering instead aims to transform a given image back into a scene description [Patow and Pueyo 2003; Marschner 1998].

(a) Reference image **(b) Low-fidelity inverse render** **(c) High-fidelity inverse render**

Figure 2: Example of a situation where the low-fidelity metric fails to find a material matching the reference (b), whereas the high-fidelity metric does (c), albeit with a longer runtime.

Inverse rendering is often an ill-posed problem because the rendering function is not uniquely invertible. Many different combinations of geometry, material, lighting, and camera parameters could all yield the same image. Inverse rendering techniques often try to recover only one of those aspects, assuming the others are known: only lighting [Nishino and Nayar 2004; D'Orazio et al. 2024], only material properties [Nguyen et al. 2013; Zsolnai-Fehér et al. 2020], or only the camera position [Smelyansky et al. 2002]. Additionally, inverse rendering rarely has direct solutions, causing most inverse rendering techniques to employ iterative optimization. This paper uses inverse material rendering as an example application to demonstrate the more widely applicable multi-fidelity optimization technique.

Differentiable rendering implementations perform conventional rendering and simultaneously return the gradient of the rendering function, i.e., the partial derivatives of the image pixel values with respect to the scene parameters [Kato et al. 2020; Gao and Qi 2024]. The derivatives could be used for backpropagation, so differentiable rendering functions could be integrated into neural networks. In this paper, the derivatives are used as the Jacobian in gradient-based optimization for inverse rendering.

Notable differentiable renderer implementations include the Mitsuba 3 ray tracer [Jakob et al. 2022] and the DIB-R rasterizer [Chen et al. 2019].

2.2 Image Quality Metrics

Image quality assessment (IQA) metrics quantify the difference between two images as a single numerical value or in some cases as a per-pixel heat map. In this paper, the numerical difference between the generated image and a reference image is used as the objective function in an optimization process. IQA metrics are commonly applied in that way for inverse rendering [Nguyen et al. 2013] or image-based generative neural networks [Goodfellow et al. 2014].

The most commonly used IQA metrics range from simple aggregates of pixel differences, such as (root) mean square error ((R)MSE) or peak signal-to-noise ratio (PSNR), to complex metrics that are trained or designed to emulate human visual perception, such as learned perceptual image patch similarity (LPIPS) [Zhang et al.

2018], ꟻLIP [Andersson et al. 2020, 2021], and the HDR visual difference predictor (HDR-VDP-3) [Mantiuk et al. 2023]. A more comprehensive table of state-of-the-art IQA metrics is presented by Kavoosighafi et al. [2025].

Simple and fast metrics such as MSE are often used as objective functions in inverse rendering optimizations or as loss functions in image-based generative neural networks. However, they are known to be overly sensitive to some types of differences and insufficiently sensitive to others [Wang et al. 2004] because of the way they aggregate pixel differences over a large number of pixels. As a consequence, they do not correlate well with human visual perception. Therefore, more elaborate metrics are intentionally designed to correlate with human visual perception and have been shown to be appropriately sensitive to certain salient differences in natural image features such as gloss [Cheeseman et al. 2025].

Differentiable implementations of some perception-inspired IQA metrics such as LPIPS or ꟻLIP exist, but these complex metrics incur a considerable computational cost compared to MSE, making them too slow in practice for many optimization or neural network applications. This paper presents a multi-fidelity optimization solution that mitigates the computational cost and enables the use of complex perception-inspired IQA metrics in iterative applications.

2.3 Multi-Fidelity Optimization

Warm-start optimization is a strategy that solves a first optimization problem to initialize a second, related optimization problem [Yıldırım and Wright 2002]. Typically the optimum found by the first problem is used as the starting point for the second, but it is conceivable that additional hyperparameters, e.g., the learning rate or momentum vector, could be carried over. The two optimization problems must be related in the sense that they have similar optima for warm-start to be beneficial: the second optimization will start close to its optimum and is therefore expected to converge in fewer iterations. The first optimization problem could be a simplification of the second, in the sense that it could use a simpler, approximate objective function. Alternatively, it could be a subset of the training dataset, or a previous run of the same optimization problem with slightly different parameters. Transfer learning, fine-tuning a pretrained machine learning model on a different dataset or for a different task [Pan and Yang 2010; Zhuang et al. 2020], can be considered a form of warm-start optimization.

Multi-fidelity optimization (MFO) is a form of warm-start optimization where the first optimization uses a low-fidelity (LF) surrogate objective function, and the second optimization uses the high-fidelity (HF) objective function that we actually want to optimize. The LF surrogate is a fast approximation that quickly gets sufficiently close to the optimum, where the computationally expensive HF optimization takes over and refines the final result. Both optimizations express the same objective, only at different levels of fidelity [Gano et al. 2005; Zhou et al. 2023; Li and Li 2024].

The main benefit of MFO is the speedup. Running only the HF optimization from the start would be computationally too expensive. The LF optimization is faster but it may converge to an approximate optimum instead of the true optimum of the HF objective function. The LF optimization does not even need to run until it has fully converged and may be cut off early when progress slows, indicating the optimum is near enough for the HF optimization to take over. MFO leverages both the speed of the LF optimization and the accuracy of the HF optimization.

An additional benefit of MFO is that it relies only on the LF objective function far from the optimum, and only on the HF objective function near the optimum. It elegantly handles objective functions with regions of unreliable gradients, e.g., that are flat or discontinuous, if the other objective function is being used in those regions. We have encountered LF objective functions with a wide, flat valley near the optimum, and HF objective functions with a flat plateau far from the optimum. MFO handles both cases much more reliably than either LF or HF alone.

MFO is fundamentally different from multi-criteria or multi-objective optimization (MOO), where a number of complementary objective functions are optimized simultaneously. Typically each objective function represents a distinct desirable aspect of the solution, and all objective functions are combined into a single weighted sum objective function [Ehrgott 2005]. This approach requires all objective functions to be evaluated in every iteration and therefore does not yield the speedup of MFO. MOO does solve the problem of flat regions where an individual objective function would get stuck due to a lack of gradient, as the other objective functions would provide sufficient gradients to continue improving.

There are few examples of MFO being used in computer graphics or computer vision. One notable application of MFO in inverse rendering is an inverse rendering method for producing interpretable basis bidirectional reflectance distribution functions (BRDFs) [Chung et al. 2024]. This method uses a first optimization with a subset of its MOO objectives, followed by the final optimization with the full set of objectives, including a computationally expensive sparsity objective that involves rendering an image.

3 Multi-Fidelity Optimization

Our MFO technique is a variant of the warm start optimization strategy. Starting with a given reference render and initial scene information, we first run our inverse rendering process using an LF metric, producing a lower quality solution to the optimization problem. The scene information produced by our LF optimization is then used as the new starting point for our HF inverse rendering process. From this starting point, we optimize for a small number of iterations with the HF metric, further improving the quality of the resulting material. We show an overview of our method in figure 1.

Switching Loss Function. There are multiple methods for changing from the LF metric to the HF metric, namely changing the loss function to the HF metric but keeping the internal state of the optimizer, such as learning rate or momentum; or starting a new optimization process with the HF metric and a reset internal state.

We chose to switch the metric by starting a new optimization process with the HF metric, as only reusing the LF solution is a more common approach while also being the least involved method. This method of switching loss functions is further evaluated in section 5.5.

Additionally, it is important to determine when to switch to the HF metric. We performed a small-scale experiment, evaluating and comparing the runtime performance when using LF and HF metrics, showing that the runtime for fully optimizing using an LF metric

is on par with optimizing for a small number of iterations using an HF metric. Therefore, in order to produce the best starting point for the second stage of the optimization, we fully optimize using the LF metric before switching to the HF optimization.

Stopping conditions. For all discussed optimization methods, we apply the same stopping conditions for the inverse rendering process. The optimization process is stopped when, for ten iterations: the loss does not change; or the best loss value does not improve. Additionally, the process is stopped if it reaches a user-specified maximum number of iterations.

4 Evaluation

In order to evaluate our novel approach and compare its performance to current optimization methods and their related metrics, we created a simple testing pipeline. This pipeline produces a number of scenes by selecting an object from a set of 15 objects from the Stanford 3D Scanning Repository [Stanford 2013] and the McGuire Computer Graphics Archive [McGuire 2017], as well as the blob by Vangorp et al. [2007] and surface 1 and 2 by Havran et al. [2016]. The object's material is generated with random parameter values for the diffuse color, specular, roughness and metallic parameters in the Disney principled BRDF [Burley 2012].

For each scene, a corresponding initial scene is produced by replacing the original material with another, random material. After all scene information has been determined, the scenes are rendered using Mitsuba [Jakob et al. 2022] and tonemapped using a global gamma exposure curve if a low dynamic range (LDR) metric is used.

Given both the reference and initial renders, as well as the initial scene information, we optimize over the material parameters using MSE, LPIPS [Zhang et al. 2018], LDR-ꟻLIP [Andersson et al. 2020], HDR-ꟻLIP [Andersson et al. 2021] or an MFO combination of MSE followed by ꟻLIP. We also extend LPIPS to support high dynamic range (HDR) using perceptual units [Mantiuk and Azimi 2021].

Noise in Rendering. The renders generated using our base scenario are noiseless. However, some amount of noise is typically present in most conventional renders. This noise could influence the performance of the optimization process by influencing the gradients of the loss function, thereby possibly steering the optimizer away from correct values.

Therefore, we also evaluated all optimization methods for noisy renders by sampling environment maps. Scenes are illuminated with seven environment maps: six from the USC High-Resolution Light Probe Image Gallery [Vision & Graphics Laboratory 2019] and one by Vogel [2010] and rendered using 4 samples per pixel (spp).

Available Parameters. Another consideration is the set of parameters for which to optimize. Not all material models use the same set of material parameters and in some cases certain parameters could be less interesting to the user. Additionally, more available parameters could result in more non-zero gradients per step, benefitting LF metrics by possibly preventing the optimization to find a plateau. This could therefore possibly reduce the relative efficacy of the multi-fidelity method.

Given that we optimize over all material parameters in the baseline evaluation, we evaluate this particular scenario by optimizing using a reduced set of material parameters. This reduced set consists of the diffuse color, roughness, specular and metallic material parameters, limiting the optimizer to the basic parameters of the Disney principled BRDF model.

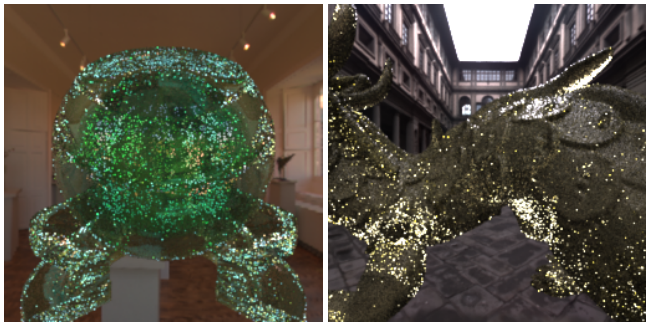

Figure 3: Examples of problematic renders produced by using a low spp in combination with transparent materials.

Transparency. The previous scenarios only utilized the BRDF component of our chosen material model, excluding transmissive or transparent materials. To cover this subset of materials, we generated transparent materials and evaluated all optimization methods on scenes using transparent materials. The transparent materials were generated by selecting a value in the range [0.75,1.0] for the specular transmission material parameter, a value in the range [0,0.6] for the roughness parameter and setting the metallic parameter value to zero. These values were chosen to maximize the number of transparent materials, as most parameter combinations outside these ranges would result in opaque materials.

These scenes are illuminated using the environment maps mentioned above in order to capture the full effect of using transparent materials. However, we use a higher spp of 32, as we noted that rendering these materials at lower values resulted in unrealistic images with localized peak values. An example of such problematic renders is shown in figure 3.

5 Results

The MFO method is evaluated for all scenarios described in section 4 by optimizing for at least 100 scenes per metric. All results are shown in table 1, grouped by LDR and HDR metrics, showing the mean runtimes and the relative speed increases compared to the corresponding ꟻLIP metric. The compared methods are:

- **MSE:** The low-fidelity-only optimization that is typically used in many optimization and neural network applications because it was the only computationally feasible approach.
- **LPIPS:** A high-fidelity-only optimization that seems to vary rather widely in runtime and quality.
- **ꟻLIP:** A high-fidelity-only optimization that we would want to use in applications because it produces consistently good quality results, but it is computationally infeasible.
- **MFO:** The multi-fidelity optimization (MSE followed by ꟻLIP) that produces consistently equal or even better quality than ꟻLIP alone and with a consistent speedup compared to ꟻLIP alone.

In addition to evaluating the runtimes for the various optimization methods, the quality of their results is also evaluated quantitatively using an independent IQA metric that was not used in the optimization: HDR-VDP-3 [Mantiuk et al. 2023]. This metric has been shown to correspond well to human perception of image distortions and natural image features such as gloss [Cheeseman et al. 2025]. Using HDR-VDP-3, the output renders of the various optimization processes are compared to the corresponding reference renders. This produces an objective quality value Q_{JOD} on the Just-Objectionable-Difference scale [Perez-Ortiz et al. 2020] where 10 represents a perfect match, 9 means that 75% of observers would notice the difference in a pairwise comparison, and lower quality values represent even more objectionable differences. The results of these comparisons are shown in table 2.

The quality value is also used to determine whether the optimization process was able to converge to the correct optimum for a given scene. In case the quality value was lower than a threshold value, the runtime for that scene is excluded from the average. The chosen threshold value is 9, as we found that this value ensures a visually good match but is also sufficiently lenient, accepting some small, less relevant differences. The inclusion percentages are shown in table 3.

All timing experiments were run on an NVIDIA RTX 3070.

5.1 Baseline

The *Base* column in table 1 shows the runtime statistics for the different optimization processes and metrics.

Of note is the runtime difference between the MFO method and the optimization only applying HDR-ꟻLIP: using the MFO method more than doubles the optimization speed. This, combined with the increased output quality shows that applying this strategy is worthwhile in the base case.

In contrast, the LDR version of the MFO method has a slightly longer runtime in comparison with the other LDR metrics. This is most likely due to the low runtimes for these metrics, meaning a small increase in the runtime could impact the relative runtime significantly. Given that we chose to run the LF metric (MSE) until its optimization process has fully converged, and running a number of HF optimization iterations afterwards, an increase in runtime was to be expected.

This implies MFO is more effective in cases where there is a larger runtime difference between the chosen LF and HF metrics, as can be observed in the runtime of the HDR version of the MFO method. HDR-ꟻLIP has a particularly slow runtime because it essentially tonemaps the HDR image at a number of exposures and then runs LDR-ꟻLIP at each exposure.

The MFO method produces results of a higher quality than the non-MFO methods, for both the LDR and HDR metrics. Unlike the non-MFO methods, it is able to find correct materials for optimizations that stall on plateaus, as the HF stage will most likely find gradients due to its higher quality. This is illustrated in the example of the outputs for the various stages of the MFO method in figure 5, row 1. Additionally, as can be noted from table 2, the minimum quality of the results is also improved, indicating a more stable optimization.

5.2 Noise

The noisy scenario examines the effect of stochastic rendering noise on the optimization results. When examining the runtime statistics for the noisy scenario in the *Noise* column in table 1, it can be noted that the performance for all metrics is roughly in line with their performance in the baseline experiment, meaning the MFO method yields the best quality and, for the HDR variant, significantly improves in runtime over HDR-ꟻLIP.

One notable exception is the runtime when using LPIPS, which has drastically improved. Contrasting this with the lower quality of the LPIPS results (tables 2 and 3), however, shows that the majority of noisy LPIPS optimizations did not reach an adequate optimum. That makes the runtime improvement less promising. From inspecting the results, we noticed that for a large portion of the excluded scenes the optimization process only produced slight deviations from the starting material parameters, causing the process to get stuck after a small number of iterations.

5.3 Parameter Sets

The *Parameters* column in tables 1 and 2 shows that the HDR MFO method performs consistently with the previous scenarios, producing the best results while significantly improving on the runtime of HDR-ꟻLIP.

The LDR variant shows similar performance, producing the highest quality results. Additionally, unlike in the previous scenarios, it outperforms its LDR-ꟻLIP counterpart on runtime as well, yielding a 40% speedup. This is most likely the result of leveraging the significant runtime improvement for the MSE metric.

Finally, for this scenario, LPIPS seems to have the same issue as in the *Noise* scenario.

5.4 Transparency

The results in the *Transparency* column in table 1 show that the runtimes for all metrics have increased significantly due to the higher samples per pixel (spp) as discussed in section 4.

Additionally, the overall quality has been reduced. This is also noticable in table 3, showing a drop of around 50% in the inclusion percentages, meaning a large number of result images still include noticeable differences. Despite this drop in quality, the MFO method still produces the best results for both variants, with the HDR variant being on par with HDR-ꟻLIP. However, when considering the minimum quality values, the HDR MFO method will be slightly more consistent.

Interestingly, both MFO variants have the highest runtimes. From evaluating the results, we observed that the HF metric (LDR- or HDR-ꟻLIP) is mostly responsible for the higher quality outputs in this scenario. This implies that the starting point provided by the LF optimization (MSE) provides little benefit to the performance of the HF optimization. However, optimizing for the LF metric still adds to the runtime, meaning that the overall runtime for the MFO method will be longer.

5.5 Ablations

In addition to evaluating the performance and quality of the MFO method in various scenarios, we also investigated a set of edgecases by ablation.

Table 1: Mean runtime statistics over all sample scenes for the different metrics when used for material optimization in various scenarios as described in section 4. The values shown are the mean runtime (mm:ss) and the increase in speed relative to the corresponding LDR- or HDR-ꟻLIP metric.

Image metric	Base	Noise	Parameters	Transparency
MSE (LDR)	01:02 (1.5x)	01:13 (1.4x)	00:39 (2.3x)	04:36 (1.3x)
LPIPS	01:58 (0.8x)	00:23 (4.4x)	00:22 (4.1x)	02:30 (2.3x)
LDR-ꟻLIP	01:35 (1x)	01:40 (1x)	01:31 (1x)	05:44 (1x)
MFO (LDR)	01:38 (0.9x)	01:51 (0.9x)	01:03 (1.4x)	07:55 (0.72x)
MSE (HDR)	00:54 (8.1x)	01:13 (4.2x)	00:33 (10.6x)	05:10 (2x)
LPIPS (PU)	02:30 (2.9x)	00:26 (11.7x)	00:37 (9.5x)	03:14 (3.2x)
HDR-ꟻLIP	07:17 (1x)	05:03 (1x)	05:50 (1x)	10:25 (1x)
MFO (HDR)	03:08 (2.3x)	03:15 (1.6x)	02:52 (2x)	14:06 (0.7x)

Table 2: The HDR-VDP-3 quality values over all sample scenes for the different metrics when used for material optimization in various scenarios as described in section 4, shown as: mean (min/max).

Image metric	Base	Noise	Parameters	Transparency
MSE (LDR)	9.84 (8.05/9.99)	9.45 (6.88/9.99)	9.81 (7.76/9.99)	8.30 (4.01/9.99)
LPIPS	9.72 (1.49/9.99)	7.87 (5.14/9.93)	7.82 (5.23/9.99)	7.94 (3.59/9.96)
LDR-ꟻLIP	9.69 (6.45/9.99)	9.54 (7.03/9.99)	9.54 (5.96/9.99)	8.35 (5.88/9.99)
MFO (LDR)	**9.89 (8.24/9.99)**	**9.66 (5.74/9.99)**	**9.84 (6.13/9.99)**	**8.43 (5.21/9.99)**
MSE (HDR)	9.19 (4.24/9.99)	9.39 (7.45/9.99)	9.35 (4.98/9.99)	7.61 (4.35/9.98)
LPIPS (PU)	9.38 (4.58/9.99)	7.43 (4.08/9.89)	7.44 (1.77/9.98)	7.28 (4.28/9.82)
HDR-ꟻLIP	9.58 (5.54/9.99)	9.34 (6.26/9.99)	9.66 (6.38/9.99)	**8.17 (4.59/9.99)**
MFO (HDR)	**9.83 (7.58/9.99)**	**9.52 (7.16/9.99)**	**9.91 (7.60/9.99)**	**8.17 (4.89/9.99)**

Table 3: The percentage of scenes passing the threshold, thereby included to calculate the mean runtimes as shown in table 1.

Image metric	Base	Noise	Parameters	Transparency
MSE LDR	95%	84%	96%	34%
LPIPS	90%	13%	14%	24%
LDR-ꟻLIP	88%	86%	84%	36%
MFO LDR	97%	88%	95%	36%
MSE (HDR)	74%	80%	80%	15%
LPIPS (PU)	80%	10%	11%	9%
HDR-ꟻLIP	83%	76%	86%	35%
MFO (HDR)	92%	82%	97%	33%

Switching Loss Function. In order to evaluate the method of switching from the LF metric to the HF metric, we performed a qualitative comparison. This evaluated both methods, namely changing the loss function to the HF metric but keeping the internal state of the optimizer, such as learning rate or momentum; or starting a new optimization process with the HF metric and a reset internal state.

From the resulting images, we found that creating a new optimization process produces results with better output more consistenly, suggesting that the naive implementation is the preferred option. A set of resulting images is shown in figure 4, showing the constrasting consistency of the two methods.

Exploring the optimization process shows that the inconsistency when keeping the internal state is mostly likely caused by transferring the learning rate, causing the HF optimization process to be less able to further correct the previously found parameter values.

Lower spp for Transparency. While rendering with a lower spp for the transparency scenario resulted in unrealistic images, these images can still be used to evaluate the robustness of the MFO method. Therefore, we evaluated the performance and quality of the optimization process for the various metrics with renders using transparent material and 4 spp, showing the results in table 4.

We found that for these scenes, the runtime performance of the MFO method is comparable with the runtime performance for the noisy scenario. This was to be expected, given the performance of the MFO method for noisy situations, as described in section 5.2.

In terms of quality, however, the MFO method is outperformed by all other metrics except LPIPS. From inspecting the various results, this drop in relative performance is difficult to explain. In most cases, the HF optimization does not improve the results of the LF optimization, as was the case for all other scenarios.

5.6 Discussion

As noted, the MFO method does not always improve the runtime performance of the optimization process, most notably when applied to LDR optimizations. This raises the question of when to

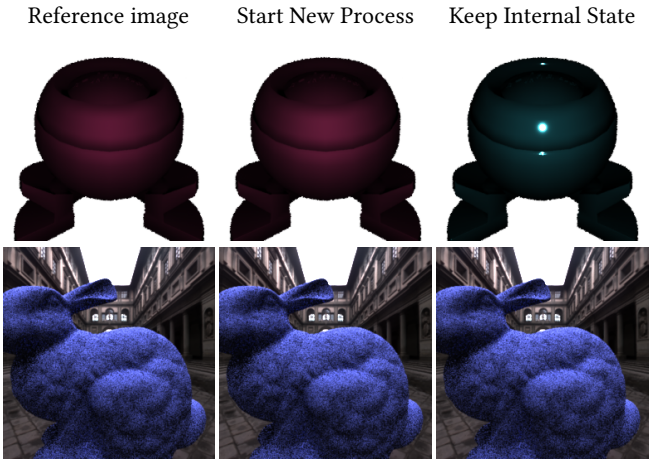

Figure 4: Results from the ablation evaluating the methods for switching from the low-fidelity (LF) metric to the high-fidelity (HF) metric, showing the reference image and the results from optimizing with each method.

Table 4: The mean runtime statistics and HDR-VDP-3 quality values over all sample scenes for the different metrics when used for material optimization in the *Transparency* scenario while using 4 samples per pixel (spp). The values shown are the mean runtime (mm:ss), the increase in speed relative to the corresponding LDR- or HDR-ꟻLIP metric, and the quality shown as: mean (min/max).

Image metric	Time	Quality
MSE (LDR)	01:07 (1.8x)	**8.97 (5.89/9.99)**
LPIPS	00:25 (4.9x)	7.46 (5.12/9.36)
LDR-ꟻLIP	02:03 (1x)	8.03 (5.82/9.99)
MFO (LDR)	02:08 (1x)	7.99 (5.21/9.99)
MSE (HDR)	01:00 (4.7x)	8.79 (6.22/9.99)
LPIPS (PU)	00:42 (6.6x)	7.01 (4.06/9.22)
HDR-ꟻLIP	04:39 (1x)	**8.85 (4.49/9.99)**
MFO (HDR)	04:27 (1x)	8.67 (5.24/9.99)

apply multi-fidelity optimization, and if it should always be the preferred option. If quality is the primary concern and if a significantly faster LF objective function is available, applying MFO will generally be beneficial. Many image-based optimization or neural network applications could benefit from the improved quality of complex IQA metrics, and MFO will be applicable in a wide range of computer graphics, computer vision, and machine learning applications by enabling the use of more complex and costly objective functions than was previously feasible.

However, if one is purely concerned with shorter runtimes, applying an LF metric by itself could be sufficient, with MFO still being a viable secondary option. Alternatively, while we did not evaluate this, an adjusted version of MFO could be applied where

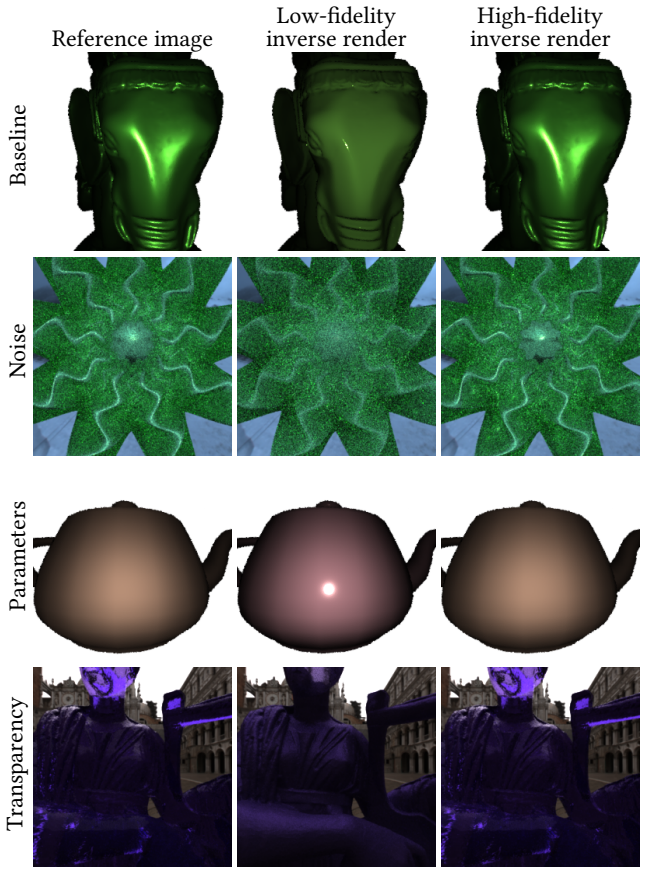

Figure 5: Examples of multi-fidelity optimization (MFO) per scenario illustrating a situation where the low-fidelity (LF) optimization fails to find a material matching the reference. The high-fidelity (HF) optimization, using the result of the LF optimization as a starting point, does manage to find a correct material.

the number of iterations per optimization process is limited, possibly yielding quality improvements while compromising less on runtime.

6 Conclusion and Future Work

By evaluating multiple scenarios, isolating various parts of the inverse material optimization process, we have shown that multi-fidelity optimization is beneficial for material optimization using inverse rendering. While this technique does not always improve the efficiency of the optimization process, it does generally improve the quality of the results and should therefore be considered a viable option for all material optimization settings.

This paper has explored the applicability of MFO in the context of material optimization using inverse rendering. However, given the advantages of applying this method, it could be a valuable addition to any optimization applying image difference metrics. Future work could therefore explore the usability of MFO in other

fields, such as environment map optimization, three-point lighting and shape optimization.

While we covered diverse scenarios in material optimization, additional scenarios could be introduced for a more complete evaluation. In all scenarios, we used a relatively simple scene consisting of a single object illuminated by either a point light or an environment map. This, however, is a simplified version of real-world scenes, which typically consist of large numbers of objects and light sources. A further evaluation of the MFO method using scenes commonly used in rendering, such as the Amazon Lumberyard Bistro scene [Lumberyard 2017], could further explore the applicability of multi-fidelity optimization in real-world applications. Another scenario to consider is limiting the number of material parameters available to the optimizer by evaluating the MFO method using other material models instead of simulating these material models by reducing the (material) parameters available to the optimizer.

References

Pontus Andersson, Jim Nilsson, Tomas Akenine-Möller, Magnus Oskarsson, Kalle Åström, and Mark D. Fairchild. 2020. ꟻLIP: A Difference Evaluator for Alternating Images. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 3, 2 (2020), 15:1–15:23. doi:10.1145/3406183

Pontus Andersson, Jim Nilsson, Peter Shirley, and Tomas Akenine-Möller. 2021. Visualizing Errors in Rendered High Dynamic Range Images. In *Eurographics Short Papers*. doi:10.2312/egs.20211015

Brent Burley. 2012. Physically Based Shading at Disney. SIGGRAPH 2012 Course: Practical Physically Based Shading in Film and Game Production.

Jacob R. Cheeseman, James A. Ferwerda, Takuma Morimoto, and Roland W. Fleming. 2025. Gloss discrimination: Toward an image-based perceptual model. *Journal of Vision* 25, 10 (2025), 6. doi:10.1167/jov.25.10.6

Wenzheng Chen, Jun Gao, Huan Ling, Edward Smith, Jaakko Lehtinen, Alec Jacobson, and Sanja Fidler. 2019. Learning to predict 3D objects with an interpolation-based differentiable renderer. In *Advances In Neural Information Processing Systems (NeurIPS)*. arXiv:1908.01210 [cs.CV]

Hoon-Gyu Chung, Seokjun Choi, and Seung-Hwan Baek. 2024. Differentiable Inverse Rendering with Interpretable Basis BRDFs. arXiv:2411.17994 [cs.CV]

Antonio D'Orazio, Davide Sforza, Fabio Pellacini, and Iacopo Masi. 2024. Environment Maps Editing using Inverse Rendering and Adversarial Implicit Functions. In *Smart Tools and Applications in Graphics*. arXiv:2410.18622 [cs.CV]

Matthias Ehrgott. 2005. *Multicriteria optimization* (2nd ed.). Springer. doi:10.1007/3-540-27659-9

Shawn E. Gano, John E. Renaud, and Brian Sanders. 2005. Hybrid Variable Fidelity Optimization by Using a Kriging-Based Scaling Function. *AIAA Journal* 43, 11 (2005), 2422–2430. doi:10.2514/1.12466

Ruicheng Gao and Yue Qi. 2024. A Brief Review on Differentiable Rendering: Recent Advances and Challenges. *Electronics* 13, 17 (2024). doi:10.3390/electronics13173546

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (Eds.), Vol. 27. arXiv:1406.2661 [stat.ML]

Vlastimil Havran, Jiri Filip, and Karol Myszkowski. 2016. Perceptually Motivated BRDF Comparison using Single Image. *Computer Graphics Forum* 35, 4 (2016), 1–12. doi:10.1111/cgf.12944

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

Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. 2020. Differentiable Rendering: A Survey. arXiv:2006.12057 [cs.CV]

Behnaz Kavoosighafi, Rafał K. Mantiuk, Saghi Hajisharif, Ehsan Miandji, and Jonas Unger. 2025. Perceived quality of BRDF models. *Computer Graphics Forum* 44, 4 (2025). doi:10.1111/cgf.70162

Ke Li and Fan Li. 2024. Multi-Fidelity Methods for Optimization: A Survey. arXiv:2402.09638 [cs.LG]

Amazon Lumberyard. 2017. Amazon Lumberyard Bistro, Open Research Content Archive (ORCA). http://developer.nvidia.com/orca/amazon-lumberyard-bistro

Rafał K. Mantiuk and Maryam Azimi. 2021. PU21: A novel perceptually uniform encoding for adapting existing quality metrics for HDR. In *Picture Coding Symposium*.

Rafał K. Mantiuk, Dounia Hammou, and Param Hanji. 2023. HDR-VDP-3: A multi-metric for predicting image differences, quality and contrast distortions in high dynamic range and regular content. arXiv:2304.13625 [eess.IV]

Stephen Robert Marschner. 1998. *Inverse rendering for computer graphics*. Cornell University.

Morgan McGuire. 2017. *Computer Graphics Archive*. https://casual-effects.com/data

Johannes Meyer, Lukas Dippon, and Christian Kludt. 2025. Material characterization by inverse rendering. In *OCM 2025-7th International Conference on Optical Characterization of Materials, March 26th–27th, 2025, Karlsruhe, Germany: Conference Proceedings*. KIT Scientific Publishing, 65.

Chuong H. Nguyen, Daniel Scherzer, Tobias Ritschel, and Hans-Peter Seidel. 2013. Material Editing in Complex Scenes by Surface Light Field Manipulation and Reflectance Optimization. *Computer Graphics Forum* 32, 2pt2 (2013), 185–194. doi:10.1111/cgf.12038

Ko Nishino and Shree K. Nayar. 2004. Eyes for relighting. *ACM Transactions on Graphics* 23, 3 (2004), 704–711. doi:10.1145/1015706.1015783

Sinno Jialin Pan and Qiang Yang. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering* 22, 10 (2010), 1345–1359. doi:10.1109/TKDE.2009.191

Gustavo Patow and Xavier Pueyo. 2003. A Survey of Inverse Rendering Problems. *Computer Graphics Forum* 22, 4 (2003), 663–687. doi:10.1111/j.1467-8659.2003.00716.x

Maria Perez-Ortiz, Aliaksei Mikhailiuk, Emin Zerman, Vedad Hulusic, Giuseppe Valenzise, and Rafał K. Mantiuk. 2020. From Pairwise Comparisons and Rating to a Unified Quality Scale. *IEEE Transactions on Image Processing* 29 (2020), 1139–1151. doi:10.1109/TIP.2019.2936103

Ravi Ramamoorthi and Pat Hanrahan. 2023. *A Signal-Processing Framework for Inverse Rendering* (1 ed.). Association for Computing Machinery. https://doi.org/10.1145/3596711.3596748

Vadim N. Smelyansky, Robin D. Morris, F. O. Kuehnel, David A. Maluf, and Peter Cheeseman. 2002. Dramatic Improvements to Feature Based Stereo. In *Proceedings of the 7th European Conference on Computer Vision-Part II (ECCV '02)*. Springer-Verlag, 247–261.

Stanford. 2013. *Stanford 3D Scanning Repository*. https://graphics.stanford.edu/data/3Dscanrep/

Peter Vangorp, Jurgen Laurijssen, and Philip Dutré. 2007. The influence of shape on the perception of material reflectance. *ACM Transactions on Graphics* 26, 3 (2007), 77:1–9. doi:10.1145/1276377.1276473

Vision & Graphics Laboratory. 2019. *High-Resolution Light Probe Image Gallery*. https://vgl.ict.usc.edu/Data/HighResProbes/

Bernhard Vogel. 2010. *Light Probes*. http://dativ.at/lightprobes/

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. doi:10.1109/TIP.2003.819861

E. Alper Yıldırım and Stephen J. Wright. 2002. Warm-start strategies in interior-point methods for linear programming. *SIAM Journal on Optimization* 12, 3 (2002), 782–810. doi:10.1137/S1052623400369235

Yizhou Yu, Paul Debevec, Jitendra Malik, and Tim Hawkins. 1999. Inverse global illumination: recovering reflectance models of real scenes from photographs. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*. ACM Press/Addison-Wesley Publishing Co., 215–224. doi:10.1145/311535.311559

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. arXiv:1801.03924 [cs.CV]

Qi Zhou, Min Zhao, Jiexiang Hu, and Mengying Ma. 2023. *Multi-fidelity Surrogates: Modeling, Optimization and Applications*. Springer. doi:10.1007/978-981-19-7210-2

Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. 2020. A Comprehensive Survey on Transfer Learning. *Proc. IEEE* 109, 1 (2020), 43–76. doi:10.1109/JPROC.2020.3004555

Károly Zsolnai-Fehér, Peter Wonka, and Michael Wimmer. 2020. Photorealistic Material Editing Through Direct Image Manipulation. *Computer Graphics Forum* 39, 4 (2020), 107–120. doi:10.1111/cgf.14057