Deep learning for predictive rendering of 3D printed objects

Akmaral Amanturdieva[1,*,†], Davit Gigilashvili[1,†] and Jiří Filip[2,†]

[1]Norwegian University of Science and Technology, Gjøvik, Norway
[2]The Czech Academy of Sciences, Institute of Information Theory and Automation, Prague, Czech Republic

Abstract

This study explores the development of a deep learning-based predictive rendering system for 3D printed objects, addressing the challenge of accurately predicting surface appearance from input parameters like surface normals, light angles, view positions, and tangent vectors. By utilizing the Deep Shading architecture, we present and explore a method that synthesizes rendered appearances. The dataset, sourced from controlled multi-view and illumination imaging conditions, serves as the foundation for training and evaluating the model. We tested various loss functions and training data demonstrating a promising performance in 3D printed appearance reproduction. Our findings contribute to the broader effort of improving predictive rendering systems for 3D printed objects, with potential applications in manufacturing, design, and material science.

Keywords

Deep Learning, Computer Graphics, Rendering,

1. Introduction

The introduction of 3D printing technology has revolutionized manufacturing and prototyping industries, enabling rapid and cost-effective production of complex geometries with diverse materials [1]. Despite these advancements, accurately predicting and visualizing the final appearance of printed objects remains a significant challenge [2]. Material properties, printing processes, and surface finishing techniques substantially influence the visual characteristics of the final product, creating a gap between digital design and physical reality. As a result, designers often resort to costly, time-consuming cycles of trial-and-error—printing, inspecting, and re-printing to achieve the desired look. Robust, physically based predictive rendering tools could close this loop, reducing wasted material and energy and making the entire workflow more sustainable and environmentally friendly.

The 3D printing workflow typically involves three primary stages: designing a digital model, preparing it for printing, and the actual fabrication process [3]. At the core of this workflow lies computer graphics technology, which facilitates the visualization of the object before physical production.

Classical computer graphics methodologies originate from physical modeling principles, focusing on geometry, surface properties, and camera settings [4]. The rendering process, transforming a scene definition into a simulated camera image, typically follows one of two approaches: rasterization or ray tracing. Rasterization maps geometry to the image domain in a feed-forward process, while ray tracing simulates light paths by casting rays from image pixels into the virtual scene, recursively modeling reflections and refractions [4]. The quality of rendered images depends heavily on the accuracy of these physical models and the sampling techniques employed to address the discrete nature of computer simulations.

While these classical rendering approaches have proven effective in many applications, they present significant limitations for 3D printing visualization. First, physically accurate models require extensive computational resources, making real-time visualization challenging. Second, traditional approaches struggle to capture the complex material properties and printing artifacts that emerge during the

MANER Conference Mainz/Darmstadt 2025 (MANER2025), August 29, 2025, Darmstadt, Germany
*Corresponding author.
†These authors contributed equally.
akmarala@ntnu.no (A. Amanturdieva); davit.gigilashvili@ntnu.no (D. Gigilashvili); filipj@utia.cas.cz (J. Filip)
© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2025-12-18

CEUR-WS.org/Vol-4135/paper6.pdf

fabrication process. Finally, creating realistic models demands considerable time and costly manual effort from skilled artists, rendering traditional computer graphics approaches time-consuming, expensive, and error-prone when applied to 3D printing visualization.

Recent advances in deep learning offer promising solutions to these challenges by learning to generate realistic visualizations based on patterns identified in training datasets. Neural rendering, a subset of these techniques, enables a statistical perspective on image generation [4], potentially bridging the gap between theoretical designs and the actual visual output of 3D printed objects. These approaches have demonstrated remarkable success in various applications, including view synthesis, material editing [5], and relighting [6] capabilities directly applicable to predicting the appearance of 3D printed objects.

This study aims to leverage deep learning techniques to develop a predictive rendering system specifically tailored for 3D printed objects. We focus on training a model capable of synthesizing the appearance of 3D printed objects based on a set of intrinsic and extrinsic parameters, including surface normals, light positions, viewing directions and tangent vectors. By incorporating these parameters, our approach seeks to provide manufacturers, designers, and researchers with accurate visual predictions before physical production, potentially reducing material waste and improving design iterations.

The paper is organized as follows: Section 2 reviews existing approaches to predictive rendering, focusing on image-based and intrinsic parameter-based methods. Section 3 describes our dataset and its key attributes. Section 4 outlines the methodology, including data preprocessing, input encoding, and architectural design of our models. Section 5 presents experimental results comparing different model configurations under varied conditions. Finally, Section 6 discusses implications, limitations, and future research directions.

2. Literature Review

Approaches for predicting rendered 3D object appearance can be broadly categorized into two types: image-based rendering methods and intrinsic image parameter-based techniques. Each category offers distinct advantages and limitations for visualizing 3D printed objects.

2.1. Image-Based Rendering

Image-based rendering (IBR) methods generate new images by manipulating existing image sets, typically through processes like warping and compositing to combine visual elements into a cohesive result [4]. The quality of generated images depends on factors such as geometry precision, quantity and spatial distribution of input views, and material properties, as certain materials exhibit significant appearance variations from different viewpoints [4].

Recent advances in deep learning have substantially improved IBR techniques. Hedman et al. [7] introduced Deep Blending, a novel approach that employs convolutional neural networks (CNNs) to predict optimal blending weights for each pixel, enabling seamless integration of input images without relying on handcrafted heuristics. IBRNet, presented by Wang et al. [8], offers a more generalized solution for multiview image-based rendering by synthesizing novel views of complex scenes through interpolation of sparse nearby views.

Building on these advances, Sun et al. [9] developed SIBRNet (Sparse Image-Based Rendering Network), which addresses the challenge of image-based rendering in sparse scene geometry through a two-stage approach combining geometry recovery and light blending.

These image-based approaches require multiple reference images of the actual printed object, limiting their predictive capabilities during the design phase.

2.2. Intrinsic Image Parameters-Based Approaches

Intrinsic image parameters-based approaches focus on decomposing and manipulating fundamental visual attributes that contribute to object appearance. These methods operate by explicitly modeling and controlling properties such as surface geometry, material characteristics, illumination conditions,

and viewpoint parameters. Unlike image-based approaches, these methods attempt to understand and model the underlying physical properties that generate the observed appearance. In this context, neural rendering establishes a mapping (I = M(c)), where ($c \in \mathbb{R}^{d_{\text{in}}}$) represents control parameters, and ($I \in \mathbb{R}^{H \times W \times 3}$) denotes the corresponding output image with height (H) and width (W)—a process that can be understood as complex sparse data interpolation [10].

Santo et al. [11] approached the inverse of this problem by introducing a Deep Photometric Stereo Network (DPSN) that maps reflectance observations to surface normals and reflectance properties. Unlike traditional methods, DPSN employs a data-driven approach using the MERL BRDF dataset [12], which includes measured BRDFs of diverse materials. The network predicts surface normals and reflectance coefficients per pixel from images under predefined light directions, enabling scene relighting under arbitrary lighting conditions.

Chen et al. [13] proposed PS-FCN, a fully convolutional neural network that predicts an object's normal map from an arbitrary number of input images taken under diverse lighting directions. Unlike DPSN [11], which requires a fixed set of predefined light directions, PS-FCN introduces an order-agnostic mechanism through max-pooling operations that enable robust feature aggregation regardless of the input images' order or number. Nalbach et al. [14] introduced a paradigm shift by utilizing deep learning to directly map deferred shading buffers to RGB outputs.

A recent breakthrough by Zeltner et al. [15] presents Real-time Neural Appearance Models (NAMs), which incorporates spatial material properties, per-pixel surface normals, BRDF parameters, lighting properties, and viewpoint information to enable fast, real-time appearance prediction.

These intrinsic parameter-based approaches show particular promise for predicting the appearance of 3D printed objects. However, they typically require extensive training data and struggle to capture the full range of printing artifacts and material behaviors that emerge during the physical fabrication process.

3. Dataset

The dataset for this study was provided by Institute of Information Theory and Automation [16]. It represents a comprehensive dataset designed to capture the complex interactions between viewing angles, illumination conditions, and surface properties. The dataset comprises multiple components including multi-view images, geometric information, and angular data, enabling detailed analysis of appearance characteristics under varying conditions.

3.1. Image Acquisition

The primary dataset consists of multi-view, multi-illumination images captured under systematically controlled conditions. Examples of the images from the dataset can be found in Figure 1. The acquisition protocol followed a structured sampling approach:

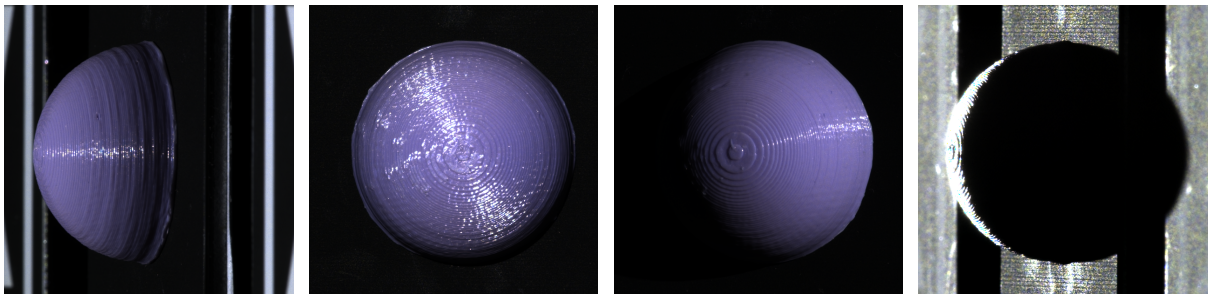
- Viewing Elevations: 6 distinct positions
- Illumination Positions: 81 distinct positions
- Total Images: $6 \times 81 = 486$ combinations
- Format: 8-bit SDR RGB PNG

Each image follows a standardized naming convention:

```
image_tlXXX_plXXX_tvXXX_pvXXX.png
```

where:

- `tl`/`pl`: Light source elevation/azimuth (in degrees)
- `tv`/`pv`: Viewing position elevation/azimuth (in degrees)



(a) Object captured at 75° viewing position elevation

(b) Object captured at 15° light source elevation and 120° light source azimuth

(c) Object captured at 60° light source elevation, 18° light source azimuth and 15° viewing position elevation

(d) Object captured at 75° light source elevation, 180° light source azimuth and 75° viewing position elevation

Figure 1: Example images from the dataset representing the same object with different geometric properties.

3.2. Auxiliary Data

3.2.1. Binary Masks

For each captured image the dataset contains corresponding binary masks (`mask_*`) that contain valid pixel regions. These masks are encoded as binary images where white pixels (255) indicate valid regions, facilitating precise spatial analysis and region-of-interest processing.

3.2.2. Normal and Tangent Maps

Surface geometry information is encoded through ideal-fitted normal maps (`normal_map_*`) for each viewpoint. In addition to surface normals, the dataset provides a per-pixel tangent map (`tangent_map_*`) for every viewpoint. Tangent maps encode the direction of increasing u texture coordinate on the surface and are useful for anisotropic appearance models or tangent-space shading. The normal and tangent vectors are color-coded using the following transformation:

$$\text{RGB} = 255 \times \frac{X + 1}{2} \tag{1}$$

where X represents the normalized surface normal/tangent vector at each pixel location.

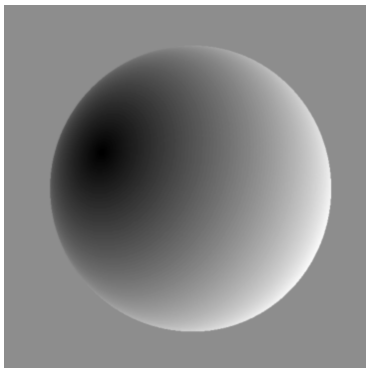
3.2.3. Angular Information Maps

The dataset includes per-pixel angular information encoded using a three-channel representation. in separate maps:

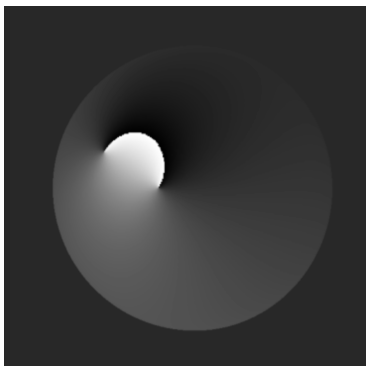
1. Light angle maps:
   - Elevation: `light_elev_*`
   - Azimuth: `light_azim_*`
2. View angle maps:
   - Elevation: `view_elev_*`
   - Azimuth: `view_azim_*`

An example image (Figure 3) with all the attributes can be observed in Figure 2.

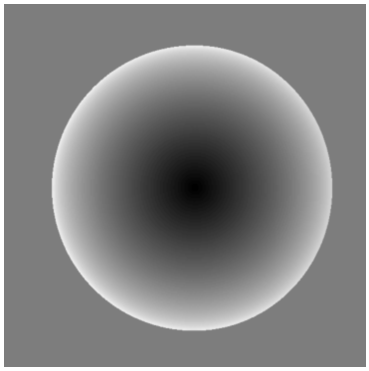
(a) Normal map visualization encoded in RGB colors



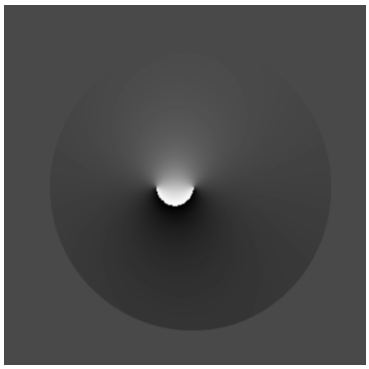
(b) Light elevation map displaying angular distribution at 30°



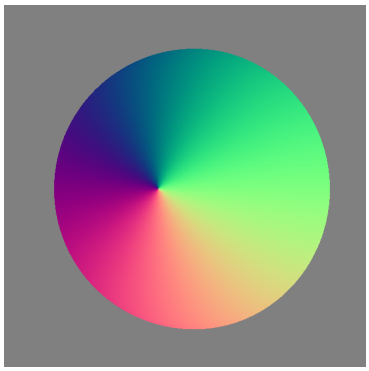
(c) Light azimuth map showing directional information at 150°



(d) View elevation map indicating observation angles at 15°



(e) View azimuth map showing viewing directions at 15°



(f) Tangent map visualization encoded in RGB colors

Figure 2: Visualization of color coded per-pixel attributes used in the appearance synthesis pipeline: (a) surface normal orientation, (b-c) light position parameters, (d-e) viewing direction parameters, and (f) tangent vector. Each map encodes specific geometric properties essential for accurate appearance prediction.

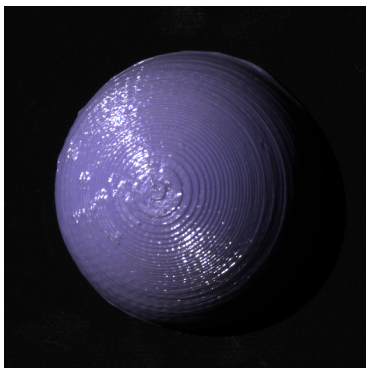


Figure 3: A photograph of the printed object probe shown in Figure 2

4. Methodology

4.1. Data Preparation

Data preparation methodology follows multiple processing stages designed to transform raw input data into a format optimized for deep learning model training.

Normal and Tangent maps to angles. For each pixel with 8-bit colour triplet $\mathbf{C} = (R, G, B) \in [0, 255]^3$ the unit normal/tangent is recovered as

$$\mathbf{n} = 2\frac{\mathbf{C}}{255} - 1 \in [-1, 1]^3. \tag{2}$$

θ and ϕ are then

$$\theta = \arccos(n_z), \qquad \phi = \operatorname{atan2}(n_y, n_x), \tag{3}$$

with $\phi \in (-\pi, \pi]$ and $\theta \in [0, \pi]$. In degrees we use $\phi^\circ = (\phi + \pi)\frac{180}{\pi} \bmod 360$ and $\theta^\circ = \theta\frac{180}{\pi}$.

Decoding light/view angles. The elevation and azimuth angles for both light direction and viewing position are initially stored in color-coded three-channel images. These are subsequently decoded into single-channel representations containing per-pixel angular values using the equation:

$$\alpha^\circ = \frac{256\ \textit{upper} + \textit{lower}}{65535}\ 360, \text{where upper is first and lower is a second channel.} \tag{4}$$

Cartesian direction encoding. All directional quantities: surface normal, tangent, light and view are finally expressed as unit vectors $\mathbf{v} = (x, y, z)$ and linearly mapped to $[0, 1]$ through $(\mathbf{v} + 1)/2$. For spherical angles (ϕ, θ) (used by normals and tangents)

$$\mathbf{v} = \big(\sin\theta\cos\phi,\ \sin\theta\sin\phi,\ \cos\phi\big), \tag{5}$$

while for azimuth a and elevation e (used by light and view)

$$\mathbf{v} = \big(\cos e\cos a,\ \cos e\sin a,\ \sin e\big). \tag{6}$$

This representation approach offers several significant advantages. The method ensures consistent input standardization across different parameter types while reducing dimensionality through efficient encoding. Furthermore, it facilitates the learning of rotation-independent features and provides a more intuitive representation of directional information within well-defined numerical bounds (0-1).

4.2. Training Data Generation

To enhance the training process and manage computational resources effectively, we implement a sophisticated patch-based preprocessing strategy. The process begins by loading binary masks that indicate valid pixel regions for each image. We then apply a sliding window approach using a 32 × 32 pixel window with a step size of a random number between 5 and 32 pixels, ensuring overlap between adjacent patches. Only patches containing exclusively valid pixels, as determined by the mask, are retained. For each valid patch, we record its coordinates, including the image index and spatial position (y, x). These coordinates are then used to generate corresponding patches from both the output images (rendered appearance) and input data (normal map, light, and view parameters).

The implementation of randomly overlapping patches in our preprocessing strategy serves multiple purposes. This approach significantly reduces block boundary artifacts in predictions while enhancing the capture of contextual information. The overlap between adjacent patches also improves model robustness by providing multiple perspectives of boundary regions. However, we must acknowledge certain challenges inherent in this approach, particularly the potential loss of global context and sensitivity to patch size and stride parameters.

Our patch-based preprocessing significantly expands the dataset, transforming 486 source images into approximately 80 000–100 000 patches. This expansion provides several key benefits for the learning process. The increased quantity of training data enables more effective learning of local features and patterns while maintaining invariance to global position. The approach also facilitates efficient stochastic sampling during training and optimizes computational resource utilization through controlled patch sizes.

The substantial increase in data volume necessitates careful consideration of computational resources. To address this challenge, we implement efficient batch processing techniques and optimize storage strategies for the extracted patches. This ensures that the benefits of our comprehensive patch-based approach are realized without overwhelming computational resources. Furthermore, the overlapping nature of our patches, while increasing memory requirements, provides essential redundancy that contributes to the robustness and accuracy of the model's predictions.

As a first step, we focus on the achromatic components of appearance; therefore, all images were processed in grayscale. Full-color rendering is beyond the scope of this work and will be addressed in future studies.

4.3. Model Architecture

Due to the simplicity and alignment with our research questions, in this work we train and test the model Deep Shading from [14].

Deep Shading model is a Convolutional Neural Network (CNN) with a U-shaped architecture to perform screen-space shading tasks. The architecture consists of an encoder-decoder structure with skip connections to preserve spatial details, inspired by U-Net designs. The encoder progressively downsamples spatial resolution while capturing high-level semantic features, while the decoder upsamples the features back to the original resolution, enabling precise pixel-wise predictions. The model takes 12-channel input containing multiple per-pixel attributes. These attributes are concatenated and processed through a series of convolutional layers with Leaky ReLU activations, batch normalization, and pooling operations. The latent representation from the encoder is passed to the decoder, where upsampling layers and transposed convolutions restore the original spatial dimensions. Skip connections between corresponding encoder and decoder layers ensure the retention of fine-grained details critical for pixel-wise tasks. The final output is a grayscale image representing the rendered object. Training is conducted using an L1 loss function, also known as mean squared error (MSE).

5. Experimental Results

In this section, we present the experimental setup, evaluation metrics, and key results obtained from the proposed methodology with the model introduced in [14]. The goal of these experiments is to assess the performance of the model in terms of accuracy, generalization, and robustness. We compare the method with baseline approach. Both qualitative and quantitative evaluations are presented. All of the experiments were conducted using PyTorch framework on NVIDIA RTX 4090 GPU.

5.1. Loss Function Design

L1 (photometric) loss The baseline uses the L1 photometric loss, which is the mean absolute error between predicted pixel values defined as:

$$\mathcal{L}_{L1} = \frac{1}{HW}\sum_{x,y}\left|I_{\text{pred}}(x,y) - I_{\text{gt}}(x,y)\right|, \tag{7}$$

where H and W represent the height and width of the image in pixels, (x, y) are the pixel coordinates, I_{pred} is the predicted image output from our model, and I_{gt} is the ground truth image. L1 is preferred over L2 (MSE) in image synthesis because it tends to preserve sharper details and avoid excessive blurring [17].

Gradient difference loss To enhance high-frequency detail, a gradient difference loss is added on top of L1. This term penalizes differences in image gradients between prediction and target, effectively aligning edges.

$$\mathcal{L}_{\text{grad}} = \frac{1}{HW}\sum_{x,y}\left(\left|\nabla_x I_{\text{pred}} - \nabla_x I_{\text{gt}}\right| + \left|\nabla_y I_{\text{pred}} - \nabla_y I_{\text{gt}}\right|\right), \tag{8}$$

where ∇_x and ∇_y represent the gradients in the horizontal and vertical directions, respectively, similar to the Gradient Difference Loss of [18].

Intensity-weighted loss The intensity loss term reweighs the pixel error according to brightness, so that highly illuminated regions (e.g. highlights) contribute more. Using a weighting function $w_{\text{light}}(x, y) \in [0, 1]$,

$$\mathcal{L}_{\text{int}} = \frac{1}{HW} \sum_{x,y} w_{\text{light}}(x, y) \left| I_{\text{pred}}(x, y) - I_{\text{gt}}(x, y) \right|, \tag{9}$$

where $w_{\text{light}}(x, y)$ is a weighting factor proportional to the pixel intensity in the ground truth image. The motivation comes from both human perception (errors in brightly lit areas are more noticeable) and physical reasoning (bright regions often correspond to direct reflections or light sources).

Specular-highlight loss This term targets view-dependent specular reflections. With view direction $\mathbf{V}$, light direction $\mathbf{L}$, surface normal $\mathbf{N}$ and halfway vector $\mathbf{H} = \frac{\mathbf{V}+\mathbf{L}}{\|\mathbf{V}+\mathbf{L}\|}$ one can identify pixels likely on specular highlights [19]

$$w_{\text{spec}}(x, y) = (\mathbf{N} \cdot \mathbf{H})^{16}, \quad \mathcal{L}_{\text{spec}} = \frac{1}{HW} \sum_{x,y} w_{\text{spec}}(x, y) \left| I_{\text{pred}}(x, y) - I_{\text{gt}}(x, y) \right| \tag{10}$$

where $w_{\text{spec}}(x, y)$ is a weighting factor that highlights areas with likely specular reflections based on the dot product of the normal vector and halfway vector raised to the power of 16. This power value approximates a typical specular exponent in the Blinn-Phong reflectance model [20].

Tangent-space directional loss Finally, the tangent gradient loss encourages the predicted image to vary along the surface in the same way as the ground truth when moving across the object's surface. Given surface tangent $\mathbf{T}$ and bitangent $\mathbf{B}$ at each pixel, directional derivatives of an image I are $D_{\mathbf{T}} I = \nabla_x I \, T_x + \nabla_y I \, T_y$ and $D_{\mathbf{B}} I = \nabla_x I \, B_x + \nabla_y I \, B_y$.

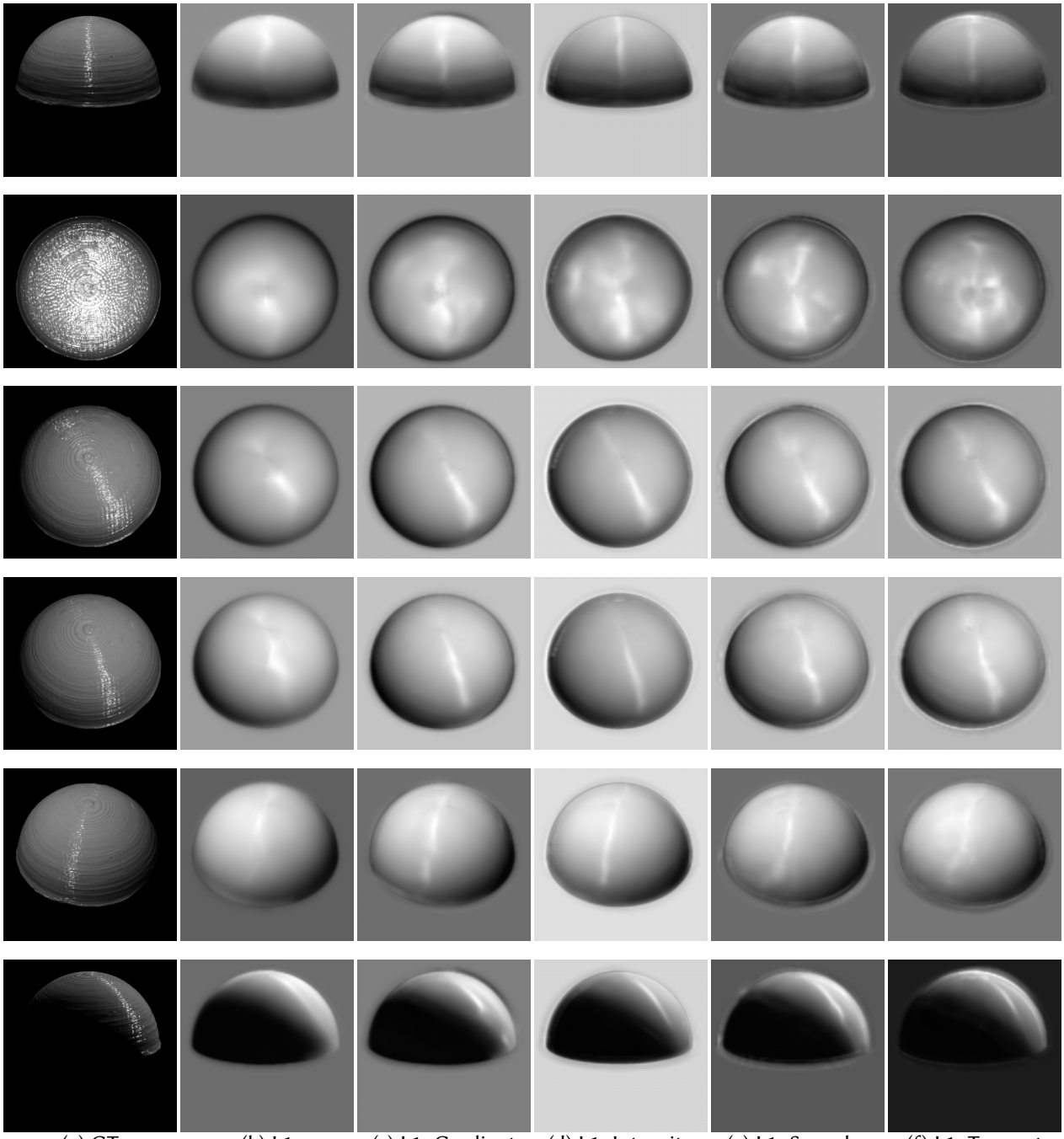
$$\mathcal{L}_{\text{tang}} = \frac{1}{HW} \sum_{x,y} \Big(\left| D_{\mathbf{T}} I_{\text{pred}} - D_{\mathbf{T}} I_{\text{gt}} \right| + \left| D_{\mathbf{B}} I_{\text{pred}} - D_{\mathbf{B}} I_{\text{gt}} \right| \Big), \tag{11}$$

where T_x, T_y, B_x, and B_y are the components of the tangent and bitangent vectors in the image plane. These directional derivatives $D_{\mathbf{T}}$ and $D_{\mathbf{B}}$ measure how the image intensity changes when moving along the tangent and bitangent directions on the surface. Intuitively, this term checks whether the network's output image has the correct directional lighting gradients as the viewer's eye moves across the surface. Recent work on polarized inverse rendering imposes tangent space consistency across views to better constrain shape and material estimation [21].

5.2. Training and Validation

During training and validation, the network was optimized with the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$). The learning rate was initialized at 1×10^{-3} and followed a step–decay schedule that halves the rate at predefined milestones. Each update processed mini-batches of 256 overlapping 32×32 pixel patches (random overlap) to enhance spatial coverage and reproducibility. To curb overfitting, early stopping terminated training when the validation loss failed to improve for 150 iterations 5 consecutive times.

We trained separate models using each loss combination (L1 alone, L1 + Gradient, L1 + Intensity, L1 + Specular, L1 + Tangent) on our predictive rendering task.



(a) GT (b) L1 (c) L1+Gradient (d) L1+Intensity (e) L1+Specular (f) L1+Tangent

Figure 4: Qualitative results on unseen data for different tested loss functions.

5.3. Qualitative Comparison

We present a side-by-side visualization of representative test images under each trained model to qualitatively assess the impact of the different loss terms (see Figure 4). For clarity, we compare: (a) the ground truth rendered image, (b) output with L1 loss only, (c) output with L1 + Gradient loss, (d) output with L1 + Intensity loss, (e) output with L1 + Specular loss, and (f) output with L1 + Tangent loss. Figure 4 provides a visual comparison highlighting key areas (edges, highlights, and subtle shading details).

Looking at the L1 baseline result (Fig. 4b), the overall structure and general appearance are correctly captured, but fine details show noticeable blurring. For instance, specular highlights on the objects'

glossy surfaces appear duller and more diffused compared to the ground truth. This demonstrates a well-known limitation of pixel-wise losses, they tend to minimize average error by blurring high-intensity features.

The L1 + Gradient model (Fig. 4c) shows modest improvement in edge preservation, particularly visible at object boundaries and contour transitions. However, the L1 + Intensity model (Fig. 4d) yields more substantial improvements in reproducing bright features. The specular highlights, for example, are much closer in intensity to the ground truth, they appear brighter and more prominent, whereas in the baseline they were noticeably muted.

The L1 + Specular model (Fig. 4e) demonstrates the most significant improvement in the placement, shape, and intensity of specular reflections. Because this model specifically penalizes errors when $\mathbf{N} \cdot \mathbf{H}$ is high (i.e., at mirror-reflection angles), it learns to accurately reproduce highlights with correct size, position, and intensity. The results show that the specular highlights are not only bright but also correctly localized.

Similarly, the L1 + Tangent model (Fig. 4f) maintains coherent shading patterns that follow the underlying geometry, resulting in more physically plausible renderings. The directional consistency of lighting gradients across the surface is particularly visible on curved regions of the objects.

Visually, the best performing models are those incorporating specular and tangent-space losses (Fig. 4e,f). These models produce results that most closely resemble the ground truth in terms of highlight reproduction and surface shading coherence. Interestingly, while the visual improvements are clear to human observers, traditional computational metrics like PSNR and SSIM (Table 1) fail to fully capture these perceptual enhancements. This highlights a known limitation in current image quality metrics.

In summary, our qualitative comparisons confirm that each added loss term effectively addresses its intended visual aspect, with the specular and tangent-space losses providing the most visually compelling results for realistic material rendering.

5.4. Quantitative Summary

Table 1
Performance on the test set. Best value per column is **bold**.

Loss	MAE	PSNR↑	SSIM↑
L1 (baseline)	**1.81**	**12.09**	**0.095**
L1 + Gradient	2.27	6.97	0.038
L1 + Intensity	2.83	10.12	0.008
L1 + Specular	1.88	9.22	0.056
L1 + Tangent	2.06	8.07	0.067

We evaluated each loss function configuration using standard image quality metrics: Mean Absolute Error (MAE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index (SSIM) as shown in Table 1. While the baseline L1 loss achieved the best numerical scores across these metrics, the perceptual quality of the results does not align with these measurements, which is a known limitation of traditional metrics when evaluating specular and high-frequency details. Among the other approaches, L1+Specular and L1+Tangent models produced the next best quantitative results, which better correspond with our qualitative assessment of visual quality.

This discrepancy between computational metrics and perceived quality highlights a fundamental challenge in material appearance evaluation. Standard image metrics tend to favor results that minimize average error across all pixels, regardless of perceptual importance, whereas human observers are particularly sensitive to specular highlights and coherent shading patterns that maintain physical plausibility. These findings suggest that for reflectance modeling applications, specialized perceptual metrics that better capture these visual aspects may be more appropriate than generic image quality measurements.

6. Discussion and Conclusion

The development of a predictive rendering system for 3D printed objects marks a significant step toward more accurate appearance prediction in manufacturing workflows. By using the Deep Shading architecture proposed in [14], we addressed the challenge of synthesizing rendered images from per-pixel attributes, which include surface normals, tangents, light positions, and view directions. Our approach aimed to provide a tool to visualize and analyze surface appearance before fabrication, thereby reducing the need for costly iterative prototyping.

Overall, the approach demonstrates promising results. However, a noticeable gap remains between the ground-truth photographs and even the best-performing neural predictions, indicating limitations that merit further investigation. In particular, the models still struggle to reproduce fine-scale layered textures characteristic of the printing, to maintain consistency across specular highlights, and to generalize to unseen lighting configurations.

Our investigation into various loss functions reveals important insights for neural rendering systems. While standard metrics favor the baseline L1 loss, visual assessment demonstrates that the L1+Specular and L1+Tangent models (Fig. 4e,f) produce perceptually superior results. Notably, these models show a promising tendency to reproduce hints of the layered structure inherent in 3D printed materials, capturing subtle surface characteristics that traditional metrics fail to quantify.

The divergence between computational metrics and visual quality underscores the need for specialized evaluation approaches in appearance modeling. Our findings suggest that loss functions targeting specific visual phenomena, such as specular highlights and directionally consistent shading, can significantly enhance the realism of rendered outputs despite showing modest improvements in conventional image quality metrics.

6.1. Limitations and Future Work

Despite these promising results, our current implementation represents a pilot experiment with several inherent limitations that present opportunities for future work. The study was conducted with a relatively small dataset and modest computational resources, which constrained both the model complexity and training scope. Our grayscale implementation, while sufficient for proof-of-concept, limits practical applicability in modern 3D printing workflows that increasingly rely on multi-material and color printing technologies. The training dataset focused on a limited range of images of the printed object, potentially limiting the generalization of fine surface details. Current parameter optimization uses fixed weighting schemes for loss function combinations, whereas adaptive or learned weighting strategies could yield superior results. The lighting model is restricted to directional illumination, excluding more complex scenarios involving area lights, environment lighting, and multiple sources with varying color temperatures that are common in real-world applications. Additionally, while our system demonstrates good performance on controlled test images, comprehensive validation against actual photographs of 3D printed objects under varied conditions remains essential to evaluate practical utility and identify potential discrepancies between predicted and manufactured appearances. Data augmentation techniques, including controlled noise introduction to account for manufacturing variability, represent unexplored avenues for improving model robustness. Moving forward, scaling to larger datasets, more sophisticated architectures, and enhanced computational resources could significantly advance the capability and reliability of predictive rendering systems for 3D printing applications, ultimately enabling more precise control over final object appearance and reducing the design-to-manufacturing iteration cycle.

We believe that continued advancement in appearance prediction systems will play a crucial role in bridging the gap between digital design and physical manufacturing, ultimately enabling more precise control over the final appearance of 3D printed objects. The development of specialized loss functions that target perceptually important aspects of material appearance represents a promising direction for future research in this area.

Acknowledgements

This research was partially supported by the Czech Science Foundation grant GA22-17529S.

Declaration on Generative AI

The author(s) have not employed any generative AI tools.

Code and Data Availability

The corresponding dataset is available upon request. The code is available via https://github.com/Dolphin000/deep-shading-project.

References

[1] N. Shahrubudin, T. C. Lee, R. Ramlan, An Overview on 3D Printing Technology: Technological, Materials, and Applications, Procedia Manufacturing 35 (2019) 1286–1296. URL: https://www.sciencedirect.com/science/article/pii/S2351978919308169. doi:10.1016/j.promfg.2019.06.089.

[2] W. Oropallo, L. A. Piegl, Ten challenges in 3D printing, Engineering with Computers 32 (2016) 135–148. URL: https://doi.org/10.1007/s00366-015-0407-0. doi:10.1007/s00366-015-0407-0.

[3] J. Horvath, Mastering 3D Printing, Apress, Berkeley, CA, 2014. URL: http://link.springer.com/10.1007/978-1-4842-0025-4. doi:10.1007/978-1-4842-0025-4.

[4] A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner, R. Pandey, S. Fanello, G. Wetzstein, J.-Y. Zhu, C. Theobalt, M. Agrawala, E. Shechtman, D. B. Goldman, M. Zollhöfer, State of the Art on Neural Rendering, Computer Graphics Forum 39 (2020) 701–727. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14022. doi:10.1111/cgf.14022.

[5] C. Liao, M. Sawayama, B. Xiao, Unsupervised learning reveals interpretable latent representations for translucency perception, PLOS Computational Biology 19 (2023) e1010878. doi:10.1371/journal.pcbi.1010878.

[6] L. Righetto, M. Khademizadeh, A. Giachetti, F. Ponchio, D. Gigilashvili, F. Bettio, E. Gobbetti, Efficient and user-friendly visualization of neural relightable images for cultural heritage applications, ACM Journal on Computing and Cultural Heritage 17 (2024) 54:1–54:24. doi:10.1145/3690390.

[7] P. Hedman, J. Philip, T. Price, J.-M. Frahm, G. Drettakis, G. Brostow, Deep blending for free-viewpoint image-based rendering, ACM Trans. Graph. 37 (2018) 257:1–257:15. URL: https://dl.acm.org/doi/10.1145/3272127.3275084. doi:10.1145/3272127.3275084.

[8] Q. Wang, Z. Wang, K. Genova, P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, T. Funkhouser, IBRNet: Learning Multi-View Image-Based Rendering, 2021. URL: http://arxiv.org/abs/2102.13090. doi:10.48550/arXiv.2102.13090, arXiv:2102.13090.

[9] Y. Sun, S. Zhou, R. Cheng, W. Tan, B. Yan, L. Fu, Learning Robust Image-Based Rendering on Sparse Scene Geometry via Depth Completion, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, New Orleans, LA, USA, 2022, pp. 7803–7813. URL: https://ieeexplore.ieee.org/document/9878554/. doi:10.1109/CVPR52688.2022.00766.

[10] A. Tewari, J. Thies, B. Mildenhall, P. Srinivasan, E. Tretschk, Y. Wang, C. Lassner, V. Sitzmann, R. Martin-Brualla, S. Lombardi, T. Simon, C. Theobalt, M. Niessner, J. T. Barron, G. Wetzstein, M. Zollhoefer, V. Golyanik, Advances in Neural Rendering, 2022. URL: http://arxiv.org/abs/2111.05849. doi:10.48550/arXiv.2111.05849, arXiv:2111.05849 [cs].

[11] H. Santo, M. Samejima, Y. Sugano, B. Shi, Y. Matsushita, Deep Photometric Stereo Network, in: 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), IEEE, Venice, 2017, pp. 501–509. URL: http://ieeexplore.ieee.org/document/8265276/. doi:10.1109/ICCVW.2017.66.

[12] W. Matusik, H. Pfister, M. Brand, L. McMillan, A data-driven reflectance model, ACM Transactions on Graphics (TOG) 22 (2003) 759–769. URL: https://www.merl.com/publications/TR2003-83. doi:10.1145/882262.882343.

[13] G. Chen, K. Han, K.-Y. K. Wong, PS-FCN: A Flexible Learning Framework for Photometric Stereo, 2018. URL: http://arxiv.org/abs/1807.08696. doi:10.48550/arXiv.1807.08696, arXiv:1807.08696 [cs].

[14] O. Nalbach, E. Arabadzhiyska, D. Mehta, H.-P. Seidel, T. Ritschel, Deep Shading: Convolutional Neural Networks for Screen Space Shading, Computer Graphics Forum 36 (2017) 65–78. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13225. doi:10.1111/cgf.13225.

[15] T. Zeltner, F. Rousselle, A. Weidlich, P. Clarberg, J. Novák, B. Bitterli, A. Evans, T. Davidovič, S. Kallweit, A. Lefohn, Real-time Neural Appearance Models, ACM Transactions on Graphics 43 (2024) 1–17. URL: https://dl.acm.org/doi/10.1145/3659577. doi:10.1145/3659577.

[16] Institute of Information Theory and Automation, 2024. URL: http://www.utia.cas.cz/.

[17] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1125–1134.

[18] M. Mathieu, C. Couprie, Y. LeCun, Deep multi-scale video prediction beyond mean square error, in: International Conference on Learning Representations (ICLR), 2016.

[19] S. Wu, S. Basu, T. Brödermann, L. Van Gool, C. Sakaridis, PBR–NeRF: Inverse rendering with physics-based neural fields, arXiv preprint arXiv:2412.09680 (2024). CVPR 2025, to appear.

[20] J. F. Blinn, Models of light reflection for computer synthesized pictures, in: Proceedings of the 4th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '77, Association for Computing Machinery, New York, NY, USA, 1977, pp. 192–198. doi:10.1145/563858.563893.

[21] C. Li, T. Ono, T. Uemori, S. Nitta, H. Mihara, A. Gatto, H. Nagahara, Y. Moriuchi, Neisf++: Neural incident stokes field for polarized inverse rendering of conductors and dielectrics (2024). URL: http://arxiv.org/abs/2411.10189. doi:10.48550/arXiv.2411.10189, arXiv:2411.10189.