# Fast Gradient Descent for Surface Capture Via Differentiable Rendering Supplemental

# Contents

1	Supplemental video	<b>2</b>
<b>2</b>	Qualitative results on DTU and NerF datasets	<b>2</b>
3	Detailed definitions used in the paper3.1Robust loss3.2Parameterizations3.3Discrete integrals	<b>4</b> 4 4
4	Losses	<b>5</b>
5	Analytical gradient derivations5.1Gradients of the photometric loss5.2Gradients of the ray color consistency loss5.3Gradients of the spatial regularization loss5.4Gradients of the spherical harmonics parsimony loss5.5Gradients of the ballooning loss5.6Gradients of the exposure regularization	<b>5</b> 6 6 7 7
6	Optimization	8

# 1 Supplemental video

Additional content can be downloaded from https://hal.inria.fr/hal-03748662.

### 2 Qualitative results on DTU and NerF datasets

We show qualitatives results for tables 1 and 2 of the main paper. We used a black background during the optimization, which is why the volumetric rendering columns also have a black background. Our method works best for highly textured, non-specular materials (scans 65, 106, 114, 122) and yields bloated / caved-in surfaces otherwise (scans 24, 37, 110). Note that the windows of the house (scan 24) are well reconstructed since they provide a strong texture gradient even at low resolutions.



Table 1: Qualitative results on the DTU dataset.



Table 2: Qualitative results on the nerf synthetic dataset.

# 3 Detailed definitions used in the paper

#### 3.1 Robust loss

The robust Huber norm and its derivative are:

$$||x||_{\mathcal{H}} = \begin{cases} \frac{x^2}{2} & \text{if } x < \epsilon\\ \epsilon(|x| - \frac{\epsilon}{2}) & \text{if } x \ge \epsilon \end{cases}$$
(1)

$$||x||'_{\mathcal{H}} = \begin{cases} x & \text{if } x < \epsilon \\ \epsilon \operatorname{sign}(x) & \text{if } x \ge \epsilon \end{cases}$$
(2)

If  $\mathbf{x}$  is a vector, we take the sum of the norms applied component-wise:

$$||\mathbf{x}||_{\mathcal{H}} = \sum_{\boldsymbol{\Gamma}} ||x_i||_{\mathcal{H}} \tag{3}$$

$$\left\| \mathbf{x} \right\|_{\mathcal{H}}^{\prime} = \begin{bmatrix} \|x_1\|_{\mathcal{H}} \\ \vdots \\ \|x_n\|_{\mathcal{H}}^{\prime} \end{bmatrix}$$
(4)

### 3.2 Parameterizations

$$\begin{aligned} \alpha_i \in [0,1] & (\text{transparency of a sample}) \\ \mathbf{c}_i \in [0,1]^3 & (\text{color of a sample}) \\ \alpha &= \text{lerp}(\bar{\alpha}) & (\text{transparency interpolation in the voxel grid}) \\ \mathbf{c} &= \text{lerp}(\bar{\mathbf{c}}) & (\text{color interpolation in the voxel grid}) \\ \bar{\mathbf{c}}(\mathbf{v}) &= \sum_{l=0}^{B} \sum_{m=-l}^{l} Y_{l,m}(\mathbf{v}) \boldsymbol{\beta}_{l,m} & (\text{spherical harmonic basis}) \\ g, b \in \mathbb{R}^+ & (\text{Per camera gain and bias}) \\ \mathbf{C}_{\text{gt}} &= g \mathbf{C}_{\text{uncorrected gt}} + b & (\text{Ground truth color}) \\ \mathbf{C}_{\text{bg}} &= g \mathbf{C}_{\text{uncorrected bg}} + b & (\text{Background color}) \end{aligned}$$

## 3.3 Discrete integrals

$$T_{k} = \prod_{j < k} \alpha_{j} \qquad (\text{partial transmittance})$$
$$\mathbf{C}_{k} = \sum_{j \leq k} (1 - \alpha_{j}) \mathbf{c}_{j} T_{j} \qquad (\text{partial color})$$
$$\mathbf{C} = \sum_{k < \infty} (1 - \alpha_{k}) \mathbf{c}_{k} T_{k} + T_{\infty} \mathbf{C}_{\text{bg}} \qquad (\text{total integral})$$

# 4 Losses

$$\mathcal{L}_{\text{photo}} = \sum_{r \in \text{rays}} ||\mathbf{C}(r) - \mathbf{C}_{\text{gt}}(r)||_{\mathcal{H}} \quad (\text{Robust photometric loss})$$

$$\mathcal{L}_{\text{sample}} = \sum_{\text{samples}} ||\mathbf{c}_{k} - \mathbf{C}||_{\mathcal{H}} T_{k}(1 - \alpha_{k}) \quad (\text{Ray color consistency})$$

$$\mathcal{L}_{\text{smooth } \alpha} = \sum_{\text{voxels}} ||\nabla \bar{\alpha}||_{2}^{2} \quad (\text{Spatial Regularization})$$

$$\mathcal{L}_{\text{SH-parsimony}} = \sum_{\text{voxels}} ||\boldsymbol{\beta}_{l,m}||_{\mathcal{H}} \quad (\text{Spherical harmonics parsimony})$$

$$\mathcal{L}_{\text{ballooning}} = \sum_{\text{voxels}} \bar{\alpha}^{2} \quad (\text{Ballooning occupancy bias})$$

# 5 Analytical gradient derivations

## 5.1 Gradients of the photometric loss

$$\mathcal{L}_{\text{photo}} = \sum_{r \in \text{rays}} ||\mathbf{C}(r) - \mathbf{C}_{\text{gt}}(r)||_{\mathcal{H}}$$
(5)

By applying the chain rule:

$$\frac{\partial \mathcal{L}_{\text{photo}}}{\partial \mathbf{c}_{\mathbf{k}}} = \frac{\partial \mathcal{L}_{\text{photo}}}{\partial \mathbf{C}} \frac{\partial \mathbf{C}}{\partial \mathbf{c}_{k}} \tag{6}$$

$$\frac{\partial \mathcal{L}_{\text{photo}}}{\partial \alpha_k} = \frac{\partial \mathcal{L}_{\text{photo}}}{\partial \mathbf{C}} \cdot \frac{\partial \mathbf{C}}{\partial \alpha_k} \tag{7}$$

Where:

$$\frac{\partial \mathcal{L}_{\text{photo}}}{\partial \mathbf{C}} = ||\mathbf{C} - \mathbf{C}_{\text{gt}}||_{\mathcal{H}}^{'}$$
(8)

$$\frac{\partial \mathbf{C}}{\partial \mathbf{c}_k} = (1 - \alpha_k) T_k \tag{9}$$

$$\frac{\partial \mathbf{C}}{\partial \alpha_k} = -\mathbf{c}_k T_k + \sum_{i>k} (1 - \alpha_i) \mathbf{c}_i \frac{T_i}{\alpha_k} + \frac{T_\infty}{\alpha_k} \mathbf{C}_{\mathrm{bg}}$$
(10)

$$= -\mathbf{c}_k T_k + \frac{1}{\alpha_k} \left( \sum_{i>k} (1 - \alpha_i) \mathbf{c}_i T_i + T_\infty \mathbf{C}_{\mathrm{bg}} \right)$$
(11)

$$= -\mathbf{c}_k T_k + \frac{1}{\alpha_k} \left( \mathbf{C} - \sum_{i \le k} (1 - \alpha_i) \mathbf{c}_i T_i \right)$$
(12)

$$= -\mathbf{c}_k T_k + \frac{\mathbf{C} - \mathbf{C}_k}{\alpha_k} \tag{13}$$

#### 5.2 Gradients of the ray color consistency loss

$$\mathcal{L}_{\text{sample}} = \sum_{\text{samples}} ||\mathbf{c}_k - \mathbf{C}||_{\mathcal{H}} T_k (1 - \alpha_k)$$
(14)

$$\frac{\partial \mathcal{L}_{\text{sample}}}{\partial \mathbf{c}_{k}} = ||\mathbf{c}_{k} - \mathbf{C}||_{\mathcal{H}}^{'} T_{k} (1 - \alpha_{k}) (1 - \frac{\partial \mathbf{C}}{\partial \mathbf{c}_{k}})$$
(15)

$$\frac{\partial \mathcal{L}_{\text{sample}}}{\partial \alpha_k} = ||\mathbf{c}_k - \mathbf{C}||'_{\mathcal{H}} \cdot \frac{\partial \mathbf{C}}{\partial \alpha_k} - ||\mathbf{c}_k - \mathbf{C}||_{\mathcal{H}} T_k$$
(16)

#### 5.3 Gradients of the spatial regularization loss

We convert the discrete sum to an integral and do a Taylor series expansion. We then apply the multi-dimensional integration by parts formula.

$$\mathcal{L}_{\text{smooth }\alpha} = \int_{\Omega} \nabla \bar{\alpha} \cdot \nabla \bar{\alpha} d\Omega \tag{17}$$

$$\mathcal{L}_{\text{smooth }\alpha}(\bar{\alpha}+h) = \int_{\Omega} \nabla \bar{\alpha} \cdot \nabla \bar{\alpha} + 2\nabla h \cdot \nabla \bar{\alpha} + \nabla h \cdot \nabla h d\Omega$$
(18)

$$= \mathcal{L}_{\text{smooth }\alpha} + 2 \int_{\Omega} \nabla h \cdot \nabla \bar{\alpha} d\Omega + o(h^2)$$
(19)

$$= \mathcal{L}_{\text{smooth }\alpha} + 2\left[-\int_{\Omega} h\Delta\bar{\alpha}d\Omega + \int_{\Gamma} h\nabla\bar{\alpha} \cdot nd\Gamma\right] + o(h^2)$$
(20)

$$= \mathcal{L}_{\text{smooth }\alpha} - 2 \int_{\Omega} h \Delta \bar{\alpha} d\Omega + o(h^2)$$
(21)

Where  $\Omega$  is a volume encompassing the voxel grid and  $\Gamma = \partial \Omega$  is its border. We make the assumption that  $\bar{\alpha}$  is constant outside the voxel grid so that  $\nabla \bar{\alpha} = 0$  on  $\Gamma$ . Thus  $\frac{\partial \mathcal{L}_{\text{smooth }\alpha}}{\partial \bar{\alpha}} = -2\Delta \bar{\alpha}$ . Note that we obtain the same result by staying in the discrete world and computing partial derivatives as usual, the notation is just more involved.

# 5.4 Gradients of the spherical harmonics parsimony loss

$$\mathcal{L}_{\text{SH-parsimony}} = \sum_{\text{voxels}} ||\beta_{l,m}||_{\mathcal{H}}$$
(22)

$$\frac{\partial \mathcal{L}_{\text{SH-parsimony}}}{\partial \boldsymbol{\beta}_{l,m}} = ||\boldsymbol{\beta}_{l,m}||_{\mathcal{H}}^{\prime}$$
(23)

#### Gradients of the ballooning loss 5.5

$$\mathcal{L}_{\text{ballooning}} = \sum_{\text{voxels}} \bar{\alpha}^2 \tag{24}$$

$$\frac{\partial \mathcal{L}_{\text{ballooning}}}{\partial \bar{\alpha}} = 2\bar{\alpha} \tag{25}$$

#### 5.6Gradients of the exposure regularization

$$\frac{\partial \mathcal{L}_{\text{photo}}}{\partial g_i} = ||\mathbf{C} - \mathbf{C}_{\text{gt}}||'_{\mathcal{H}} \cdot (T_{\infty}\mathbf{C}_{\text{uncorrected bg}} - \mathbf{C}_{\text{uncorrected gt}}) \qquad (26)$$

$$\frac{\partial \mathcal{L}_{\text{photo}}}{\partial b_i} = ||\mathbf{C} - \mathbf{C}_{\text{gt}}||_{\mathcal{H}}' \cdot \text{vec3}(T_{\infty} - 1)$$
(27)

$$\frac{\partial \mathcal{L}_{\text{sample}}}{\partial g_i} = ||\mathbf{c}_k - \mathbf{C}||'_{\mathcal{H}} \cdot \mathbf{C}_{\text{uncorrected bg}} T_{\infty} T_k (1 - \alpha_k)$$
(28)

$$\frac{\partial \mathcal{L}_{\text{sample}}}{\partial b_i} = ||\mathbf{c}_k - \mathbf{C}||'_{\mathcal{H}} \cdot \text{vec3}(1) T_k (1 - \alpha_k) T_{\infty}$$
(29)

We penalize exposure and bias coefficients so that the overall luminosity of the cameras stays roughly unchanged:

$$\mathcal{L}_{\text{expo}} = \sum_{i < n_{\text{cams}}} \log(g_i)^2 + b_i^2$$
(30)

$$\frac{\partial \mathcal{L}_{\text{expo}}}{\partial a_i} = 2 \frac{\log(g_i)}{a_i} \tag{31}$$

$$\frac{\partial \mathcal{L}_{expo}}{\partial g_i} = 2 \frac{\log(g_i)}{g_i}$$
(31)  
$$\frac{\partial \mathcal{L}_{expo}}{\partial b_i} = 2b_i$$
(32)

We also normalize the exposures towards one at each iteration:

$$g_i \longleftarrow g_i \left(\prod_{j < n_{\text{cams}}} g_j\right)^{\frac{-1}{n_{\text{cams}}}}$$
 (33)

#### Optimization 6

We use the ADAM optimizer with  $\beta_1 = 0.960$ ,  $\beta_2 = 0.953$ , with the following update rule:

$$\delta x = \frac{\partial \mathcal{L}}{\partial x} \tag{34}$$

$$g_1 = g_1\beta_1 + \delta x(1-\beta_1) \tag{35}$$

$$g_2 = \sqrt{(g_2)^2 \beta_2 + (\delta x)^2 (1 - \beta_2)}$$
(36)

$$x = x - \frac{g_1 \lambda_{\text{learning rate}}}{\sqrt{(g_2)^2 + 10^{-8}}}$$
(37)

We weight the losses by the following parameters:

$$\lambda_{\text{learning rate}} = 0.05781 \tag{38}$$

$$\lambda_{\rm photo} = 1 \tag{39}$$

$$\lambda_{\text{smooth }\alpha} = 0.00121 \tag{40}$$

$$\lambda_{\text{smooth }\beta_{0,0}} = 0.00216 \tag{41}$$

$$\lambda_{\text{smooth } \beta_{l,m}} = 0.00486$$
(42)
 $\lambda_{\text{sample}} = 0.04063$ 
(43)

$$\lambda_{\text{sample}} = 0.04063 \tag{43}$$

$$\lambda_{\text{smooth }\beta_{0,0}} = 0.00210 \tag{41}$$

$$\lambda_{\text{smooth }\beta_{l,m}} = 0.00486 \tag{42}$$

$$\lambda_{\text{sample}} = 0.04063 \tag{43}$$

$$\lambda_{\text{SH-parsimony}} = 0.00522 \tag{44}$$

$$\lambda_{\text{ballooning}} = 0.00002 \tag{45}$$

$$\lambda_{\text{ballooning}} = 0.00002 \tag{45}$$

$$\lambda_{\text{mask}} = 0.05264 \tag{46}$$

We also weight the per-sample losses  $(\lambda_{\text{photo}} \text{ and } \lambda_{\text{sample}})$  by  $\frac{1}{n_{\text{activeCameras}}+1}$ where  $n_{\text{activeCameras}}$  is the number of cameras for which the transmittance to the voxel of interest is above zero.