Eurographics Symposium on Rendering 2021
A. Bousseau and M. McGuire
(Guest Editors)

Volume 40 (2021), Number 4

Supplemental Material: Point-Based Neural Rendering with Per-View Optimization

1. Derivation of probability-based depth

Following from the discussion in Sec. 3.3 in the main paper, we assume D_n follows distribution $\mathcal{DF}$ based on a mixture model with density $\mathcal{DF}_{D_n}(d)$. We first assume that there exists at least one such point (thus the symbol $\exists$):

$$\mathcal{DF}^{\exists}_{D_n}(d) = \sum_{i\in\mathcal{N}_n} f(d,d_{n,i})\left(\alpha_{n,i}\prod_{j=1}^{i-1}(1-\alpha_{n,j})\right)$$

$$\mathcal{DF}^{\exists}_{D_n}(d) = \sum_{i\in\mathcal{N}_n} f(d,d_{n,i})\beta_{n,i} \tag{1}$$

With:

$$\beta_{n,i} = \alpha_{n,i}\prod_{j=1}^{i-1}(1-\alpha_{n,j})$$

where $f(d,d_{n,i},\sigma_{ni,})$ is the pdf of a distribution that describes the likelihood of point $\rho_{n,i}$ being at depth d. This can be an arbitrary single mode distribution such as a uniform or normal distribution. The distribution of Eq. 1 does not integrate to one, because there is also a probability that no points exist at that pixel for that view:

$$p^{\emptyset}_{D_n} = \prod_{i\in\mathcal{N}_n}(1-\alpha_{n,i}) = \beta_{n,\infty}$$

We thus complete $\mathcal{DF}^{\exists}_{D_n}(d)$ with a point at infinity to make it a probability density function. In order to simplify the remaining derivations we keep the notation $\mathcal{N}_n$ for the set of splats from input view *n completed* with that point at infinity, and use the corresponding $\beta_{n,\infty}$ for this point, leading to:

$$\mathcal{DF}_{D_n}(d) = \sum_{i\in\mathcal{N}_n} f_i(d,d_{n,i})\beta_{n,i}$$

Note that implementation-wise adding the point at infinity is important to avoid corner cases. We can then compute the probability that the (projected) depth of a view n is smaller than all of the other views:

$$P(D_n < \min_{m\neq n}(D_m)) = \int_{-\infty}^{+\infty} P(t < \min_{m\neq n}(D_m)|D_n = t)\ \mathcal{DF}_{D_n}(t)\ dt$$

We have:

$$P(D_n < \min_{m\neq n}(D_m)) = \int_{-\infty}^{+\infty} P(t < \min_{m\neq n}(D_m)|D_n = t)\ \mathcal{DF}_{D_n}(t)\ dt$$

$$P(D_n < \min_{m\neq n}(D_m)) = \int_{-\infty}^{+\infty} \prod_{m\neq n} P(t < D_m)\ \mathcal{DF}_{D_n}(t)\ dt$$

$$P(D_n < \min_{m\neq n}(D_m)) = \sum_{i\in\mathcal{N}_n}\beta_{n,i}\int_{-\infty}^{+\infty}\prod_{m\neq n} P(t < D_m)\ f_i(t,d_{n,i})\ dt$$

$$P(D_n < \min_{m\neq n}(D_m)) = \sum_{i\in\mathcal{N}_n}\beta_{n,i}\int_{-\infty}^{+\infty}\prod_{m\neq n}\left(\int_t^{\infty}\sum_{j\in\mathcal{N}_m} f_j(s,d_{m,j})\beta_{m,j}ds\right)\ f_i(t,d_{n,i})\ dt$$

$$P(D_n < \min_{m\neq n}(D_m)) = \sum_{i\in\mathcal{N}_n}\beta_{n,i}\int_{-\infty}^{+\infty}\prod_{m\neq n}\left(\sum_{j\in\mathcal{N}_m}\beta_{mj}\int_t^{\infty} f_j(s,d_{mj})ds\right)\ f_i(t,d_{n,i})\ dt$$

Thus:

$$P(D_n < \min_{m\neq n}(D_m)) = \sum_{i\in\mathcal{N}_n}\beta_{n,i}\int_{-\infty}^{+\infty}\prod_{m\neq n}\left(\sum_{j\in\mathcal{N}_m}\beta_{m,j}\int_t^{\infty} f_j(s,d_{m,j})ds\right)\ f_i(t,d_{n,i})\ dt$$

A natural candidate for f would be a normal distribution. Unfortunately this expression would be very costly to compute and does not have a closed form solution. Instead we use a simple symmetric triangle distribution of support 2σ, which is easy to evaluate. Example plots of these distributions are presented in Fig. 1. This gives f_i:

$$f_i(s,d_{n,i})ds = \begin{cases}\frac{\sigma-|t-x|}{\sigma^2} & \text{if } d_{n,i}-\sigma < t \leq d_{n,i}+\sigma,\\ 0 & \text{otherwise}\end{cases}$$

And:

$$\int_t^{\infty} f_i(s,d_{n,i})ds = T(t,d_{n,i},\sigma) = \begin{cases} 1 & \text{if } t < d_{n,i}-\sigma, \\ 0 & \text{if } t > d_{n,i}+\sigma, \\ 1-\frac{(x-t+\sigma)^2}{2\sigma^2} & \text{if } d_{n,i}-\sigma < t \leq d_{n,i}, \\ \frac{(t+\sigma-x)^2}{2\sigma^2} & \text{if } d_{n,i} < t \leq d_{n,i}+\sigma, \end{cases}$$

$$P(D_n < \min_{m\neq n}(D_m)) =$$

$$\sum_{i\in\mathcal{N}_n} \beta_{n,i} \int_{d_{n,i}-\sigma}^{d_{n,i}+\sigma} \prod_{m\neq n} \left(\sum_{j\in\mathcal{N}_m} \beta_{m,j} T(t,d_{m,j},\sigma) \right) f_i(t,d_{n,i})\, dt$$

This functional still has no simple closed form formula, so we approximate it with numerical integration:

$$P(D_n < \min_{m\neq n}(D_m)) \approx$$

$$\frac{2\sigma}{S} \sum_{i\in\mathcal{N}_n} \beta_{n,i} \sum_{t=1}^{S} \prod_{m\neq n} \left(\sum_{j\in\mathcal{N}_m} \beta_{m,j} T(s(t),d_{m,j},\sigma) \right) f_i(s(t),d_{n,i}) \tag{2}$$

With:

$$s(t) = d_{ni} - \sigma + \frac{t}{S+1}, \text{ and } S \text{ the number of samples}$$

In our experiments, we found that setting $S = 1$ provided satisfactory results in all our tests. This computation is done in parallel per pixel using CUDA.

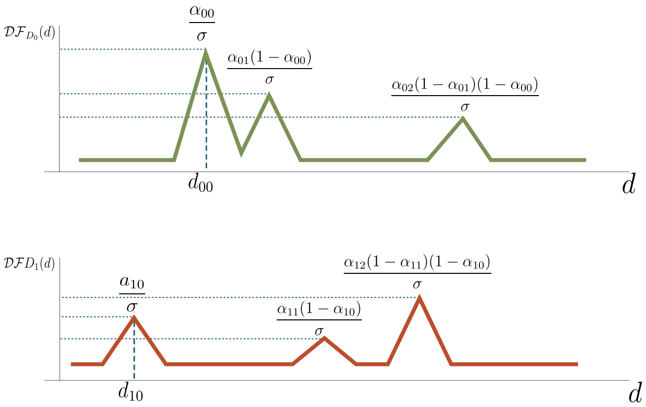
$\mathcal{DF}_{D_0}(d)$
$\frac{\alpha_{00}}{\sigma}$
$\frac{\alpha_{01}(1-\alpha_{00})}{\sigma}$
$\frac{\alpha_{02}(1-\alpha_{01})(1-\alpha_{00})}{\sigma}$
d_{00}
d
$\mathcal{DF}D_1(d)$
$\frac{\alpha_{10}}{\sigma}$
$\frac{\alpha_{11}(1-\alpha_{10})}{\sigma}$
$\frac{\alpha_{12}(1-\alpha_{11})(1-\alpha_{10})}{\sigma}$
d_{10}
d

Figure 1: *Simplified example of two* pdfs *of a pixel for view I_0 and I_1. Our soft depth test is the computation of the probability of the random variable D_i drawn by its corresponding PDF, $\mathcal{DF}_{D_i}$ shown above, to have a smaller depth than all other D_j.*

2. Multi-view Harmonization details

In Fig. 2, we illustrate the evolution of the optimization of μ_i.

3. Additional Results and Comparisons

In Fig. 3 we show comparisons of our method with two baseline methods, ULR and Textured Mesh.

In Fig. 4 we see that lowering the size of the model often does not affect the results substantially and higher number of cameras used for rendering in the same time/memory budget compensates for the smaller neural renderer.

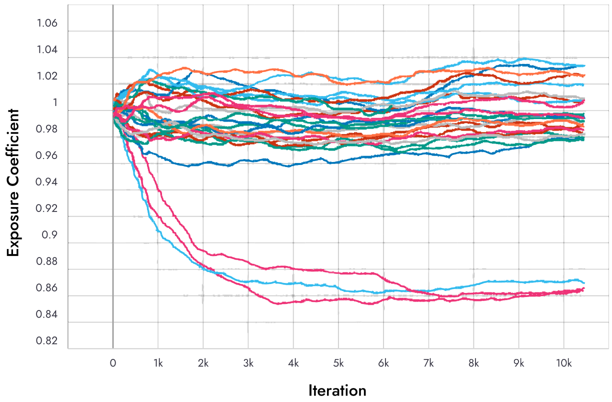

Figure 2: *The optimization of the μ_i for each view for 10k iterations during training of "Ponche" scene. Each line with a different color corresponds to a different input view; we clearly see a group of three images that are brighter (see Fig. 7 in the main paper).*

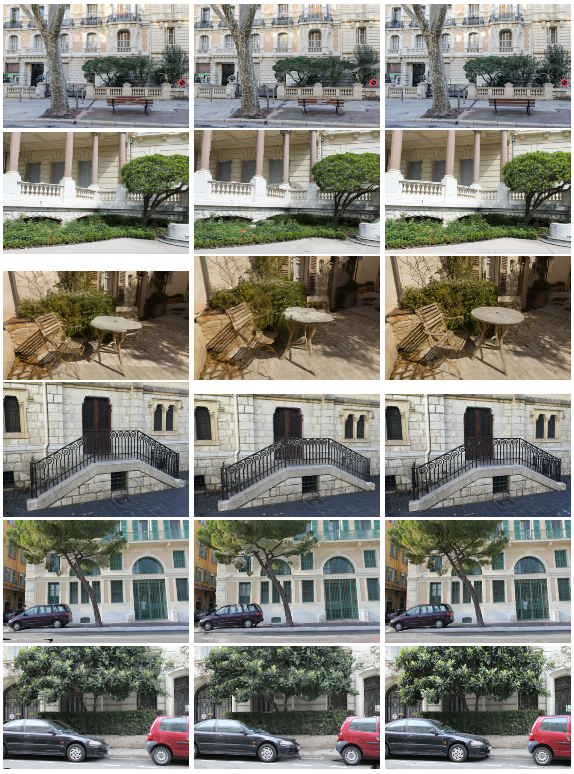

Figure 3: *Left to right: Unstructured Lumigraph Rendering, Textured Mesh, and our method.*

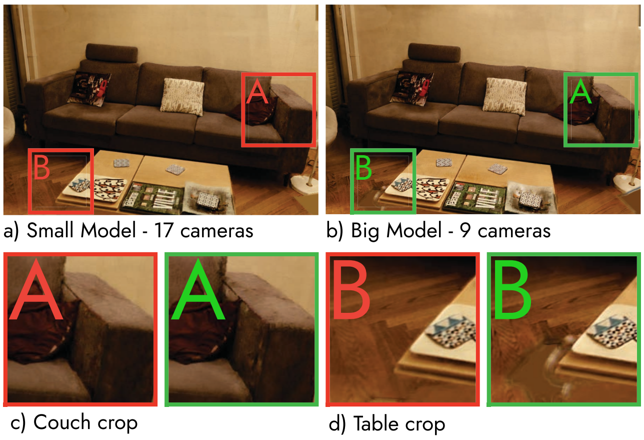

Figure 4: *In this challenging indoor scene, we use the small neural network to use a larger number of cameras.*

Table 1: ***Quantitative analysis for the full rendering pipeline against the interactive approximation in the paper. For this experiment both methods use a smaller model to allow the latter for interactive frame-rates and isolate the impact of the point splatting approximation.***

	MSE ↓
Museum	0.007
Ponche	0.004
Stairs	0.006
Street	0.008