

# Hardware-Rasterized Ray-Based Gaussian Splatting

## Supplementary Material

Samuel Rota Bulò, Nemanja Bartolovic, Lorenzo Porzi, Peter Kortschieder  
 Meta Reality Labs, Zürich

### Abstract

*This document complements our main paper with the following additional contributions:*

- derivation of results,
- additional quantitative experiments,
- qualitative comparisons.

### A. Derivations

**Proposition A.1.**  $\tau(\mathbf{x})\mathbf{x}$  is the point of maximum density along ray  $\mathbf{x} \in \mathbb{R}^3 \setminus \{\mathbf{0}\}$  for a Gaussian distribution with parameters  $(\boldsymbol{\mu}, \Sigma)$ .

*Proof.* Let  $\alpha\mathbf{x}$  parametrize all points along ray  $\mathbf{x}$ . The point of maximum density for the given 3D Gaussian is obtained by solving

$$\min_{\alpha} f(\alpha) := (\alpha\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1}(\alpha\mathbf{x} - \boldsymbol{\mu}).$$

Since the problem is convex, we find the solution by simply setting to zero the derivative of  $f$ , yielding

$$\alpha^* := \tau(\mathbf{x}),$$

where  $\tau$  is defined as per Sec. 3.2. It follows that  $\alpha^*\mathbf{x} = \tau(\mathbf{x})\mathbf{x}$  is the point of maximum density along the ray as required.  $\square$

**Proposition A.2.** If  $c^2 \leq \kappa$  holds for a given Gaussian primitive, then its support spans the whole image.

*Proof.* By plugging Eq. (5) into the support condition Eq. (3), unfolding the definition of  $\tau(\mathbf{x})$ , and after simple algebraic manipulations, we end up with the following relation:

$$\frac{(\boldsymbol{\mu}^\top \Sigma^{-1} \mathbf{x})^2}{\mathbf{x}^\top \Sigma^{-1} \mathbf{x}} \geq 0 \geq c^2 - \kappa, \quad (1)$$

which holds for any  $\mathbf{x} \in \mathbb{R}^3 \setminus \{\mathbf{0}\}$ , or in other terms all possible camera rays.  $\square$

**Proposition A.3.** If  $c^2 > \kappa$  holds for a given Gaussian primitive, then  $\tau(\mathbf{x}) \neq 0$  for all  $\mathbf{x} \in \mathbb{R}^3 \setminus \{\mathbf{0}\}$ .

*Proof.* Following a derivation similar to Prop. A.2, we arrive at the following relation

$$\tau(\mathbf{x})(\boldsymbol{\mu}^\top \Sigma^{-1} \mathbf{x}) \geq c^2 - \kappa > 0.$$

Since the condition  $c^2 > \kappa$  implies that  $\boldsymbol{\mu} \neq \mathbf{0}$ , and also  $\mathbf{x} \neq \mathbf{0}$ , it follows that  $\boldsymbol{\mu}^\top \Sigma^{-1} \mathbf{x} \neq 0$ . Hence, necessarily  $\tau(\mathbf{x}) \neq 0$  for all  $\mathbf{x} \in \mathbb{R}^3 \setminus \{\mathbf{0}\}$ .  $\square$

**Proposition A.4.**  $\tau(\mathbf{e}) = 1$  for all  $\mathbf{e} \in \mathcal{E}$  assuming  $c^2 > \kappa$ .

*Proof.* Given  $\mathbf{e} \in \mathcal{E}$  we have by definition of  $\mathcal{E}$  that  $\mathbf{e} = \tau(\mathbf{x})\mathbf{x}$  holds for some  $\mathbf{x} \in \mathbb{R}^3 \setminus \{\mathbf{0}\}$ . By Prop. A.3,  $\tau(\mathbf{x}) \neq 0$  assuming  $c^2 > \kappa$ , which implies that  $\mathbf{e} \neq \mathbf{0}$ . Then, we can write  $\mathbf{x} = \alpha\mathbf{e}$  by setting  $\alpha = \tau(\mathbf{x})^{-1}$  and, therefore,  $\mathbf{e} = \tau(\mathbf{x})\mathbf{x} = \tau(\alpha\mathbf{e})\alpha\mathbf{e} = \tau(\mathbf{e})\mathbf{e}$ , the last equality following by unfolding the definition of  $\tau$ . Accordingly,  $\tau(\mathbf{e}) = 1$  necessarily holds.  $\square$

**Proposition A.5.** For any  $\mathbf{e} \in \mathcal{E}$ , the following holds:

$$\mathbf{e}^\top \Sigma^{-1} \mathbf{e} = \mathbf{e}^\top \Sigma^{-1} \boldsymbol{\mu} = c^2 - \kappa$$

provided that  $c^2 > \kappa$ .

*Proof.* By Prop. A.4, we have that  $\tau(\mathbf{e}) = 1$  holds for any  $\mathbf{e} \in \mathcal{E}$ . By definition of  $\tau$  it follows that  $\mathbf{e}^\top \Sigma^{-1} \mathbf{e} = \mathbf{e}^\top \Sigma^{-1} \boldsymbol{\mu}$ . This together with the constraint  $\mathcal{D}_{\text{ray}}(\mathbf{e}; \boldsymbol{\mu}, \Sigma) = (\mathbf{e} - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{e} - \boldsymbol{\mu}) = \kappa$  from  $\mathcal{E}$  yields the required relation by simple algebraic manipulations.  $\square$

### A.1. Derivation of $R_{\hat{\boldsymbol{\mu}} \leftarrow \mathbf{v}}$

We define the rotation matrix aligning  $\mathbf{v}$  to  $\hat{\boldsymbol{\mu}}$  as follows

$$R_{\hat{\boldsymbol{\mu}} \leftarrow \mathbf{v}} := \begin{cases} R_{\hat{\boldsymbol{\mu}}, \mathbf{v}} & \text{if } \hat{\boldsymbol{\mu}}^\top \mathbf{v} \geq 0 \\ R_{\hat{\boldsymbol{\mu}}, -\mathbf{v}} P & \text{else} \end{cases}$$

where  $P := \begin{bmatrix} -1 & & \\ & 1 & \\ & & -1 \end{bmatrix}$ . Here, we use the formula

$R_{\mathbf{x}, \mathbf{y}} := 2 \frac{(\mathbf{x} + \mathbf{y})(\mathbf{x} + \mathbf{y})^\top}{(\mathbf{x} + \mathbf{y})^\top (\mathbf{x} + \mathbf{y})} - I$ , which yields a rotation matrix aligning 3D vector  $\mathbf{y}$  to  $\mathbf{x}$ . This is a special case of the Rodriguez rotation formula that we obtain considering

a  $180^\circ$  rotation around the axis  $\mathbf{x} + \mathbf{y}$ . Since our goal is to align  $\mathbf{v}$  to  $\hat{\mu}$ ,  $R_{\hat{\mu}, \mathbf{v}}$  could already serve that purpose. Unfortunately,  $R_{\hat{\mu}, \mathbf{v}}$  is not defined when  $\hat{\mu} = -\mathbf{v}$ . For this reason, we distinguish two cases: If  $\hat{\mu}^\top \mathbf{v} \geq 0$ , then Rodriguez rotation formula yields a valid solution, so we return  $R_{\hat{\mu}, \mathbf{v}}$ . Otherwise, we first rotate the space  $180^\circ$  around  $[0 \ 1 \ 0]$  with matrix  $P$ , so that  $\mathbf{v}$  points in the opposite direction, and then apply  $R_{\hat{\mu}, -\mathbf{v}}$  to align  $-\mathbf{v}$  to  $\hat{\mu}$ . Indeed,  $R_{\hat{\mu}, -\mathbf{v}} P \mathbf{v} = R_{\hat{\mu}, -\mathbf{v}}(-\mathbf{v}) = \hat{\mu}$ .

## A.2. Derivation of Eq. (14)

We start simplifying the objective in Eq. (13):

$$\begin{aligned} & \|\Phi^{-1}(\mathbf{u}) - \Phi^{-1}(\mathbf{0})\|^2 \\ & \stackrel{(a)}{\propto} \|\mathbf{Q}[\mathcal{H}(a\mathbf{u}) - \mathcal{H}(a\mathbf{0})]\|^2 \\ & \stackrel{(b)}{\propto} \|\mathbf{Q}_{:,0:2}\mathbf{u}\|^2 \\ & \stackrel{(c)}{=} \mathbf{u}^\top \mathbf{Q}_{:,0:2}^\top \mathbf{Q}_{:,0:2} \mathbf{u} = \mathbf{u}^\top \mathbf{B} \mathbf{u}. \end{aligned}$$

Here, in (a) we unfold the definition of  $\Phi^{-1}$  as per main paper, neglect multiplicative factors not depending on  $\mathbf{u}$  and refactor terms; (b) follows from the fact that  $\mathcal{H}(\mathbf{z}) - \mathcal{H}(\mathbf{z}')$  yields a 3D vector with a null  $z$ -coordinate for any  $\{\mathbf{z}, \mathbf{z}'\} \subset \mathbb{R}^2$ , so  $\mathbf{Q}(\mathcal{H}(\mathbf{z}) - \mathcal{H}(\mathbf{z}')) = \mathbf{Q}_{:,0:2}(\mathbf{z} - \mathbf{z}')$ . In addition, we neglected again multiplicative factors not depending on  $\mathbf{u}$ . Finally, (c) follows from unfolding the norm and noting that the resulting matrix of quadratic coefficients coincides with  $\mathbf{B}$  as per main paper.

## A.3. Derivation of Eq. (15)

According to the description preceding Eq. (15), we have that  $V_{\text{ray}} := \frac{c^2}{c^2 - \kappa} \Phi^{-1}(U_{\text{ray}} \mathbf{0})$ , where we assume that  $\Phi^{-1}$  is applied column-wise to the input matrix. This can then be rewritten as follow:

$$\begin{aligned} V_{\text{ray}} & \stackrel{(a)}{=} c \mathbf{Q} \mathcal{H}(a U_{\text{ray}} \mathbf{0}) \\ & \stackrel{(b)}{=} c \mathbf{Q} \mathcal{H}\left(\frac{1}{c} U_{\text{ray}} Z_{\text{ray}}\right) \\ & \stackrel{(c)}{=} c \left[ \frac{1}{c} \mathbf{Q}_{0:2} U_{\text{ray}} Z_{\text{ray}} + R_p S R_{\hat{\mu} \leftarrow \mathbf{v}} \mathbf{v} \mathbf{1}^\top \right] \\ & \stackrel{(d)}{=} c \left[ \frac{1}{c} \mathbf{Q}_{0:2} U_{\text{ray}} Z_{\text{ray}} + R_p S \hat{\mu} \mathbf{1}^\top \right] \\ & \stackrel{(e)}{=} \mathbf{Q}_{0:2} U_{\text{ray}} Z_{\text{ray}} + \hat{\mu} \mathbf{1}^\top \\ & \stackrel{(f)}{=} T_{\text{ray}} \mathcal{H}(Z_{\text{ray}}). \end{aligned}$$

Here, (a) follows by unfolding the definition of  $\Phi^{-1}$  and simplifying the scalar factors; (b) is obtained by using the relation  $a\mathbf{0} = \frac{1}{c}Z_{\text{ray}}$  with  $Z_{\text{ray}}$  defined as per main paper; (c) follows from the fact that we can write  $A\mathcal{H}(X) = A[X^\top \ 0]^\top + A\mathbf{v}\mathbf{1}^\top = A_{0:2}X + A\mathbf{v}\mathbf{1}^\top$ , where  $\mathbf{1}$  is a vector of

ones, and  $Q := R_p S R_{\hat{\mu} \leftarrow \mathbf{v}}$  as per definition in Eq. (12); (d) applies the relation  $R_{\hat{\mu} \leftarrow \mathbf{v}} \mathbf{v} = \hat{\mu}$ ; (e) results from unfolding the definition of  $\hat{\mu}$  as per main paper and simplifying matrix/scalar multiplications; (f) follows from the relation  $AX + \mathbf{y}\mathbf{1}^\top = [A \ \mathbf{y}] \mathcal{H}(X)$  and from the definition of  $T_{\text{ray}}$  as per main paper.

## A.4. Derivation of RayGS's fragment shader

We show how we derived the formula used in the fragment shader starting from Eq. (5), where  $\mathbf{x} := V_{\text{ray}} \boldsymbol{\alpha}$  for a given interpolating coefficient vector  $\boldsymbol{\alpha}$  (*i.e.* nonnegative and summing up to 1):

$$\begin{aligned} \mathcal{D}_{\text{ray}}(\mathbf{x}; \boldsymbol{\mu}, \Sigma) &= (\tau(\mathbf{x})\mathbf{x} - \boldsymbol{\mu})^\top \Sigma^{-1} (\tau(\mathbf{x})\mathbf{x} - \boldsymbol{\mu}) \\ &\stackrel{(a)}{=} c^2 + \tau(\mathbf{x})^2 \mathbf{x}^\top \Sigma^{-1} \mathbf{x} - 2\tau(\mathbf{x}) \mathbf{x}^\top \Sigma^{-1} \boldsymbol{\mu} \\ &\stackrel{(b)}{=} c^2 - \frac{(\mathbf{x}^\top \Sigma^{-1} \boldsymbol{\mu})^2}{\mathbf{x}^\top \Sigma^{-1} \mathbf{x}} \\ &\stackrel{(c)}{=} c^2 - \frac{(\mathbf{x}^\top R_p S^{-2} R_p^\top \boldsymbol{\mu})^2}{\|S^{-1} R_p^\top \mathbf{x}\|^2} \\ &\stackrel{(d)}{=} c^2 \left( 1 - \frac{(\mathbf{x}^\top R_p S^{-1} \hat{\boldsymbol{\mu}})^2}{\|S^{-1} R_p^\top \mathbf{x}\|^2} \right) \\ &\stackrel{(e)}{=} c^2 \left[ 1 - \frac{(\boldsymbol{\alpha}^\top V_{\text{ray}}^\top R_p S^{-1} \hat{\boldsymbol{\mu}})^2}{\|S^{-1} R_p^\top V_{\text{ray}} \boldsymbol{\alpha}\|^2} \right] \\ &\stackrel{(f)}{=} c^2 \left[ 1 - \frac{\left( \mathcal{H}\left(\frac{1}{c} U_{\text{ray}} Z_{\text{ray}} \boldsymbol{\alpha}\right)^\top Q^\top R_p S^{-1} \hat{\boldsymbol{\mu}} \right)^2}{\|S^{-1} R_p^\top Q \mathcal{H}\left(\frac{1}{c} U_{\text{ray}} Z_{\text{ray}} \boldsymbol{\alpha}\right)\|^2} \right] \\ &\stackrel{(g)}{=} c^2 \left[ 1 - \frac{\left( \mathcal{H}\left(\frac{1}{c} U_{\text{ray}} Z_{\text{ray}} \boldsymbol{\alpha}\right)^\top R_{\hat{\boldsymbol{\mu}} \rightarrow \mathbf{v}} \hat{\boldsymbol{\mu}} \right)^2}{\|R_{\hat{\boldsymbol{\mu}} \rightarrow \mathbf{v}} \mathcal{H}\left(\frac{1}{c} U_{\text{ray}} Z_{\text{ray}} \boldsymbol{\alpha}\right)\|^2} \right] \\ &\stackrel{(h)}{=} c^2 \left[ 1 - \frac{\left( \mathcal{H}\left(\frac{1}{c} U_{\text{ray}} Z_{\text{ray}} \boldsymbol{\alpha}\right)^\top \mathbf{v} \right)^2}{\|\mathcal{H}\left(\frac{1}{c} U_{\text{ray}} Z_{\text{ray}} \boldsymbol{\alpha}\right)\|^2} \right] \\ &\stackrel{(i)}{=} c^2 \left[ 1 - \frac{1}{1 + c^{-2} \|Z_{\text{ray}} \boldsymbol{\alpha}\|^2} \right] \\ &\stackrel{(j)}{=} [c^{-2} + \|Z_{\text{ray}} \boldsymbol{\alpha}\|^{-2}]^{-1}. \end{aligned}$$

Here, (a) follows by simple algebraic manipulations and by considering  $c := \sqrt{\boldsymbol{\mu}^\top \Sigma^{-1} \boldsymbol{\mu}}$  as per main paper; (b) follows by unfolding the definition of  $\tau$  as per main paper and by simple algebraic manipulations; (c) follows by unfolding  $\Sigma^{-1} := R_p S^{-2} R_p^\top$  and rewriting the denominator into a squared norm; (d) follows by substituting  $S^{-1} R_p^\top \boldsymbol{\mu} = c \hat{\boldsymbol{\mu}}$ , where  $\hat{\boldsymbol{\mu}}$  is as per main paper, and factorizing; (e) follows by unfolding the definition of  $\boldsymbol{\alpha}$  provided above; (f) follows by unfolding  $V_{\text{ray}}$  using the relation (b) in Appendix A.3, using the identity  $\mathcal{H}(Z_{\text{ray}}) \boldsymbol{\alpha} = \mathcal{H}(Z_{\text{ray}} \boldsymbol{\alpha})$ , and simplifying scalar factors; (g) follows by unfolding the definition of  $Q$  as per

main paper and simplifying matrix multiplications; (h) follows by noting that the norm of a rotated vector yields the norm of the vector and that  $R_{\mu \rightarrow v} \hat{\mu} = v$ ; (i) follows by from the identities  $\mathcal{H}(z)^\top v = 1$  and  $\|\mathcal{H}(xz)\|^2 = 1 + x^2 \|z\|^2$ , and that the norm is invariant to rotations of the argument. Finally, (j) follows by rearranging and simplifying terms.

### A.5. Derivation of $P_{\text{MIP}}(\mathbf{x})$ in Sec. 5

We start by approximating the 3D-2D projection operator  $\pi(\mathbf{x})$  to the first-order around  $\mathbf{x}_0$ , yielding

$$\hat{\pi}(\mathbf{x}) := \pi(\mathbf{x}_0) + J_\pi(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0).$$

Assuming  $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ , we have that  $\mathbf{u} := \hat{\pi}(\mathbf{x}) \sim \mathcal{N}(\hat{\pi}(\boldsymbol{\mu}), J\Sigma J^\top)$ , where  $J := J_\pi(\mathbf{x}_0)$ . For each (unit) camera ray  $\mathbf{x}$ , we take  $\mathbf{x}_0 := \tau(\mathbf{x})\mathbf{x}$ , where  $\tau$  is as per main paper, and define a 2D Gaussian distribution  $\mathcal{N}(\pi(\mathbf{x}_0), \sigma_x^2)$ . The density of rays  $\mathbf{x}$  is then determined by the expectation of  $\mathcal{N}(\mathbf{u}; \pi(\boldsymbol{\mu}), J\Sigma J^\top)$  with  $\mathbf{u} \sim \mathcal{N}(\pi(\mathbf{x}_0), \sigma_x^2 \mathbf{I})$ , which yields

$$\begin{aligned} P_{\text{MIP}}(\mathbf{x}) &:= \int_{\mathbb{R}^2} \mathcal{N}(\mathbf{u}; \pi(\boldsymbol{\mu}), J\Sigma J^\top) \mathcal{N}(\mathbf{u}; \pi(\mathbf{x}_0), \sigma_x^2 \mathbf{I}) d\mathbf{u} \\ &= \mathcal{N}(\pi(\mathbf{x}_0); \pi(\boldsymbol{\mu}), J\Sigma J^\top + \sigma_x^2 \mathbf{I}). \end{aligned}$$

We consider a projection operator that varies with the viewing ray  $\mathbf{x}$ . Specifically, we project 3D points to the spherical tangent space  $\mathcal{T}_x \subset \mathbb{R}^3$  of  $\mathbf{x}$  by leveraging the logarithmic map. The 3D tangent vector can be mapped to 2D points on the tangent plane by leveraging *any* local coordinate frame expressed as unitary, orthogonal columns of  $\mathbf{F}_x \in \mathbb{R}^{3 \times 2}$ , which satisfies  $\mathbf{F}_x^\top \mathbf{F}_x = \mathbf{I}$  and  $\mathbf{F}_x^\top \mathbf{x} = \mathbf{0}$ . This yields the following projection operator:

$$\pi_x(\mathbf{z}) := \mathbf{F}_x^\top \text{Log}_{\mathbf{x}} \left( \frac{\mathbf{z}}{\|\mathbf{z}\|} \right).$$

Then,  $\pi_x(\mathbf{x}_0) = \mathbf{0}$  and the Jacobian  $J_x := J_{\pi_x}(\mathbf{x}_0)$  of this projection operator evaluated at  $\mathbf{x}_0$  is given by  $J_x = \frac{\mathbf{F}_x^\top}{\tau(\mathbf{x})}$ . It follows that

$$\hat{\pi}_x(\mathbf{z}) = \frac{1}{\tau(\mathbf{x})} \mathbf{F}_x^\top \mathbf{z},$$

which yields the following projected and smoothed 2D Gaussian distribution

$$\begin{aligned} P_{\text{MIP}}(\mathbf{x}) &= \mathcal{N}(\mathbf{0}; \mathbf{F}_x^\top \boldsymbol{\mu}; \mathbf{F}_x^\top \Sigma \mathbf{F}_x + (\tau(\mathbf{x})\sigma_x)^2 \mathbf{I}) \tau(\mathbf{x})^2 \\ &= \mathcal{N}(\mathbf{0}; \mathbf{F}_x^\top \boldsymbol{\mu}; \mathbf{F}_x^\top [\Sigma + (\tau(\mathbf{x})\sigma_x)^2 \mathbf{I}] \mathbf{F}_x) \tau(\mathbf{x})^2 \\ &:= \mathcal{N}(\mathbf{0}; \mathbf{F}_x^\top \boldsymbol{\mu}; \mathbf{F}_x^\top \hat{\Sigma}_x \mathbf{F}_x) \tau(\mathbf{x})^2, \end{aligned}$$

where we set  $\hat{\Sigma}_x := \Sigma + (\tau(\mathbf{x})\sigma_x)^2 \mathbf{I}$ . To evaluate the Gaussian distribution on a given view and ray  $\mathbf{x}$  we need to compute  $\boldsymbol{\mu}^\top \mathbf{F}_x (\mathbf{F}_x^\top \hat{\Sigma}_x \mathbf{F}_x)^{-1} \mathbf{F}_x^\top \boldsymbol{\mu}$  and  $\det(\mathbf{F}_x^\top \hat{\Sigma}_x \mathbf{F}_x)$ . In order to

get rid of the dependency on  $\mathbf{F}_x$  we can rewrite those expressions as follows:

$$\begin{aligned} \boldsymbol{\mu}^\top \mathbf{F}_x (\mathbf{F}_x^\top \hat{\Sigma}_x \mathbf{F}_x)^{-1} \mathbf{F}_x^\top \boldsymbol{\mu} \\ \stackrel{(a)}{=} \boldsymbol{\mu}^\top \mathbf{F}_x \mathbf{F}_x^\top \left[ \hat{\Sigma}_x^{-1} - \frac{\hat{\Sigma}_x^{-1} \mathbf{x} \mathbf{x}^\top \hat{\Sigma}_x^{-1}}{\mathbf{x}^\top \hat{\Sigma}_x^{-1} \mathbf{x}} \right] \mathbf{F}_x \mathbf{F}_x^\top \boldsymbol{\mu} \\ \stackrel{(b)}{=} \boldsymbol{\mu}^\top (\mathbf{I} - \mathbf{x} \mathbf{x}^\top) \left[ \hat{\Sigma}_x^{-1} - \frac{\hat{\Sigma}_x^{-1} \mathbf{x} \mathbf{x}^\top \hat{\Sigma}_x^{-1}}{\mathbf{x}^\top \hat{\Sigma}_x^{-1} \mathbf{x}} \right] (\mathbf{I} - \mathbf{x} \mathbf{x}^\top) \boldsymbol{\mu} \\ \stackrel{(c)}{=} \boldsymbol{\mu}^\top \left[ \hat{\Sigma}_x^{-1} - \frac{\hat{\Sigma}_x^{-1} \mathbf{x} \mathbf{x}^\top \hat{\Sigma}_x^{-1}}{\mathbf{x}^\top \hat{\Sigma}_x^{-1} \mathbf{x}} \right] \boldsymbol{\mu} \\ \stackrel{(d)}{=} (\boldsymbol{\mu} - \hat{\tau}(\mathbf{x}) \mathbf{x})^\top \hat{\Sigma}_x^{-1} (\boldsymbol{\mu} - \hat{\tau}(\mathbf{x}) \mathbf{x}) = \mathcal{D}_{\text{ray}}(\mathbf{x}; \boldsymbol{\mu}, \hat{\Sigma}_x), \end{aligned}$$

where  $\hat{\tau}(\mathbf{x}) := \frac{\boldsymbol{\mu}^\top \hat{\Sigma}_x^{-1} \mathbf{x}}{\mathbf{x}^\top \hat{\Sigma}_x^{-1} \mathbf{x}}$ , and

$$\det(\mathbf{F}_x^\top \hat{\Sigma}_x \mathbf{F}_x) \stackrel{(a)}{=} \det(\hat{\Sigma}_x) \mathbf{x}^\top \hat{\Sigma}_x^{-1} \mathbf{x}.$$

Equalities (a) follow from Prop. A.6 by taking  $X := W^\top \hat{\Sigma} W$ , where  $W = [\mathbf{F}_x, \mathbf{x}]$ , and considering  $A := \mathbf{F}_x^\top \hat{\Sigma}_x \mathbf{F}_x$ . Since  $W$  is unitary, we have that  $X^{-1} = W^\top \hat{\Sigma}_x^{-1} W$  and  $\det(X) = \det(\hat{\Sigma}_x)$ . It follows that  $D = \mathbf{F}_x^\top \hat{\Sigma}_x^{-1} \mathbf{F}_x$ ,  $e = \mathbf{F}_x^\top \hat{\Sigma}_x^{-1} \mathbf{x}$  and  $f = \mathbf{x}^\top \hat{\Sigma}_x^{-1} \mathbf{x}$ . Moreover we have that  $b = \mathbf{F}_x^\top \hat{\Sigma}_x \mathbf{x}$  and  $c = \mathbf{x}^\top \hat{\Sigma}_x \mathbf{x}$ . Equality (b) follows by noting that columns of  $W$  are eigenvectors of  $\mathbf{I} - \mathbf{x} \mathbf{x}^\top$  with eigenvalues given by  $[1, 1, 0]$  and therefore  $\mathbf{I} - \mathbf{x} \mathbf{x}^\top = \mathbf{F}_x \mathbf{F}_x^\top$ . Finally, equalities (c) and (d) follow by simple algebraic manipulations.

The final form of  $P_{\text{MIP}}(\mathbf{x})$ , up to constant multiplicative factors, is given by

$$P_{\text{MIP}}(\mathbf{x}) \propto \frac{\tau(\mathbf{x})^2}{\sqrt{\det(\hat{\Sigma}_x) \mathbf{x}^\top \hat{\Sigma}_x^{-1} \mathbf{x}}} \exp\left(-\frac{1}{2} \mathcal{D}_{\text{ray}}(\mathbf{x}; \boldsymbol{\mu}, \hat{\Sigma}_x)\right).$$

### A.6. Useful results

**Proposition A.6.** Assume

$$X := \begin{bmatrix} A & b \\ b^\top & c \end{bmatrix} \quad \text{and} \quad X^{-1} := \begin{bmatrix} D & e \\ e^\top & f \end{bmatrix}.$$

Then  $A^{-1} = D - \frac{ee^\top}{f}$  and  $\det(A) = \frac{\det(X)}{c - b^\top A^{-1} b}$ .

*Proof.* Since  $XX^{-1} = \mathbf{I}$  we have that  $AD + be^\top = \mathbf{I}$  and therefore  $D = A^{-1} - A^{-1}be^\top$ . We have also that  $Ae + bf = \mathbf{0}$  and therefore  $e/f = -A^{-1}b$ . Hence by substituting we have  $D = A^{-1} + ee^\top/f$ , from which the result about the inverse of  $A$  follows.

By properties of the determinant we have that  $\det(X) = \det(A)(c - b^\top A^{-1} b)$ , from which the result about the determinant of  $A$  trivially follows.  $\square$

## B. Additional quantitative experiments

### B.1. GS vs VKGS

In Tab. B.1, we report the results obtained by GS versus the Vulkan counterpart VKGS. We have split the table into

two sections to distinguish scenes from MipNerf360 (top) and Tanks&Temples (bottom). For each scene, we report the size of the model in terms of number of primitives and report left-to-right speed comparisons in terms of FPS and quality metrics in terms of PSNR, SSIM and LPIPS, averaged over all the test views. We disregard by now the last two columns in the table. Staring from the rendering speed, it is crystal clear that the Vulkan implementation outclasses the CUDA-based from GS with  $2\times$  average speedup, despite GS including the most recent optimizations in the renderer. Except for a few cases, the quality metrics are not significantly different. However, the fact that there are differences indicates potential misalignment between the implementations and the results favor GS because the model has been trained with the same renderer. We noticed that the bigger discrepancies happen on models that suffer from a lot of big semi-transparent floater.

on RTX2080 Scene	<i>N</i>	FPS $\uparrow$ GS VKGS	PSNR $\uparrow$ GS VKGS	SSIM $\uparrow$ GS VKGS	LPIPS $\downarrow$ GS VKGS
bicycle	6.13M	52 <b>144</b>	25.24 24.99	0.768 0.752	0.229 0.243
bonsai	1.24M	105 <b>249</b>	31.98 31.47	0.938 0.926	0.253 0.221
counter	1.22M	82 <b>193</b>	28.69 28.57	0.905 0.894	0.262 0.233
flowers	3.64M	95 <b>171</b>	21.52 21.45	0.600 0.594	0.366 0.367
garden	5.83M	58 <b>126</b>	27.41 26.98	0.867 0.852	0.119 0.127
kitchen	1.85M	65 <b>151</b>	30.32 30.10	0.921 0.908	0.158 0.156
room	1.59M	83 <b>223</b>	30.63 30.70	0.913 0.900	0.289 0.260
stump	4.96M	75 <b>168</b>	26.55 26.25	0.772 0.761	0.244 0.253
treehill	3.78M	81 <b>169</b>	22.49 22.41	0.634 0.621	0.367 0.37
barn	0.89M	129 <b>216</b>	29.06 26.02	0.880 0.860	0.201 0.238
caterpillar	1.07M	129 <b>212</b>	24.36 22.16	0.824 0.796	0.234 0.264
ignatius	2.25M	108 <b>186</b>	22.21 21.10	0.823 0.797	0.187 0.210
meetingroom	1.05M	122 <b>258</b>	26.23 23.15	0.893 0.854	0.209 0.239
truck	2.54M	105 <b>170</b>	25.19 24.21	0.876 0.852	0.178 0.168

Table B.1. Speed and quality comparison between GS and VKGS on scenes from MipNeRF360 and Tanks&Temples. The last two columns provide speed comparison of VKGS versus VKRayGS on MipNeRF360 using the same GS scene model.

## C. Qualitative examples

In Figs. C.1 to C.3 we provide the rendering of the first test image of all scenes we evaluated on. We report results obtained with GOF and our VKRayGS. The goal is to validate that visually there is barely any visible difference between the two renderings, while our method being order of magnitude faster. Clearly, given that the quality scores do not perfectly match, there are some differences that are mainly due to discrepancies between the Vulkan-based rendering pipeline and the one from GOF. *E.g.* we might have different ways of culling primitives that introduce differences mainly at the borders, or different ways of addressing out-of-bound colors, which could make saturated areas darker. Aligning differences between the Vulkan-based renderer and GOF are beyond our contributions, and thus out of our paper’s scope.

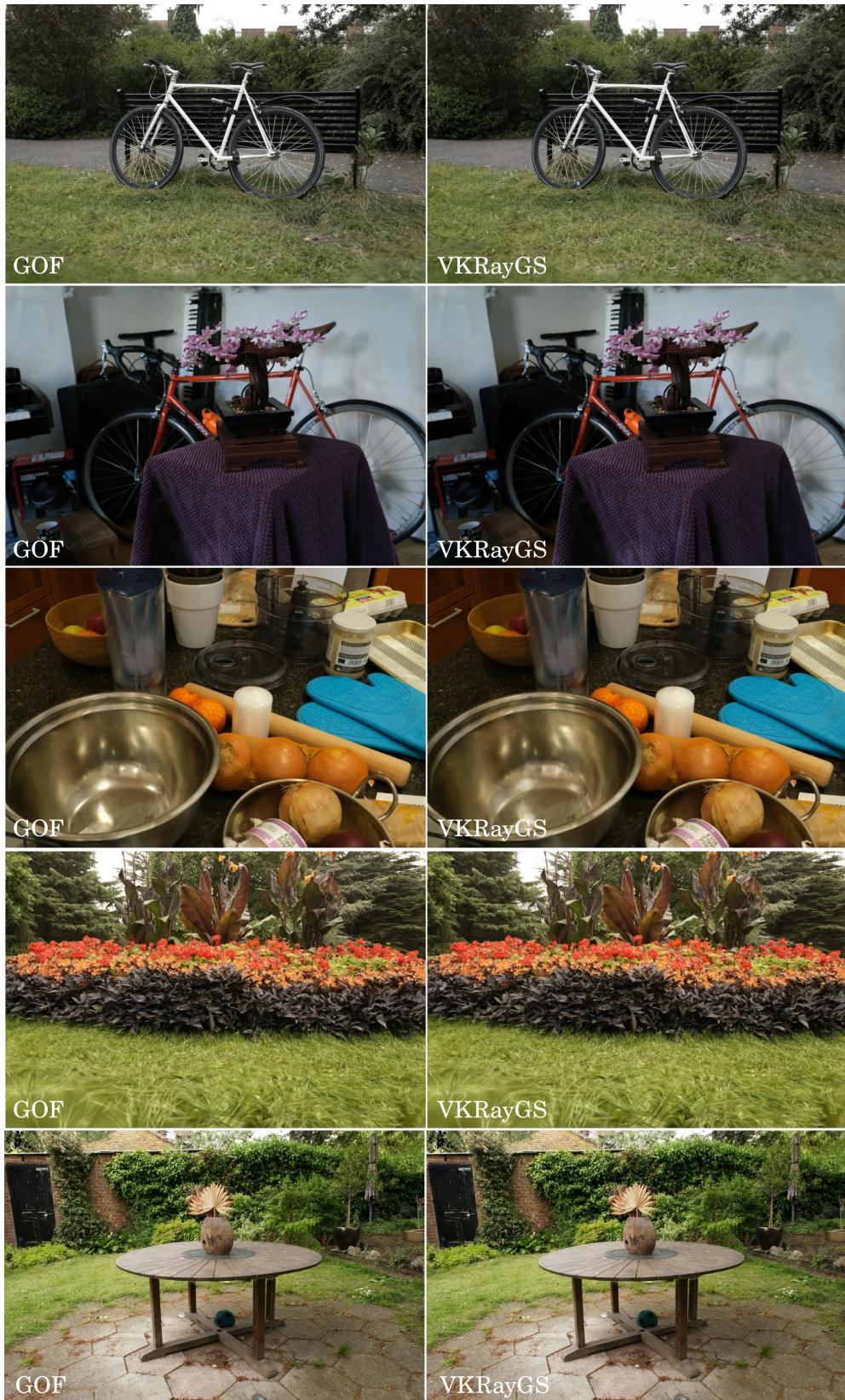


Figure C.1. First test images of the MipNerf360 scenes *bicycle*, *bonsai*, *counter*, *flowers* and *garden*, rendered by GOF and our method VKRayGS.



Figure C.2. First test images of the MipNerf360 scenes *kitchen*, *room*, *stump* and *treehill*, rendered by GOF and our method VKRayGS.

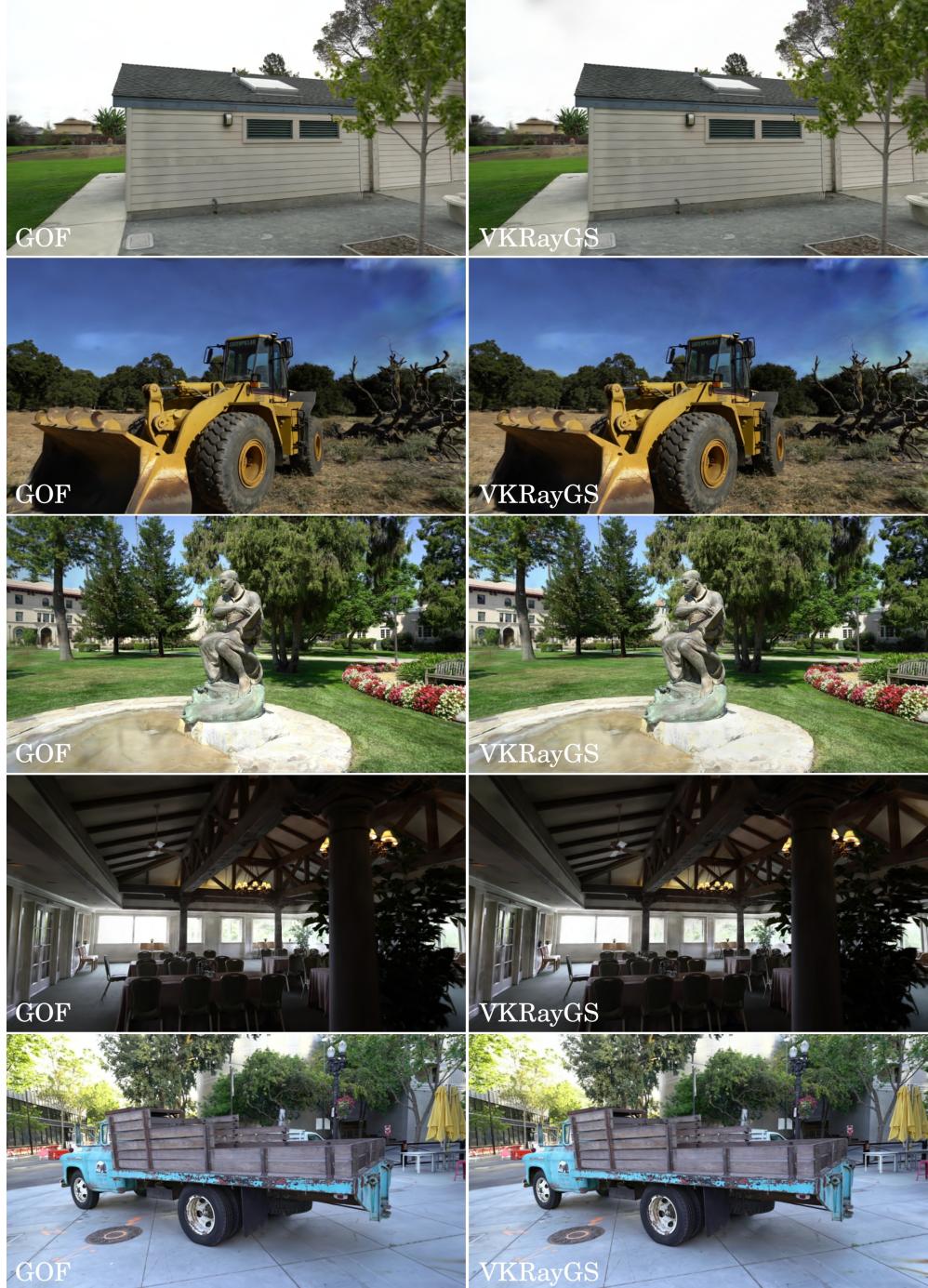


Figure C.3. First test images of the Tanks&Temples scenes *barn*, *caterpillar*, *ignatius*, *meetingroom* and *truck*, rendered by GOF and our method VKRayGS.