Supplementary Material:
3D shape reconstruction of semi-transparent worms

Thomas P. Ilett* Omer Yuval* Thomas Ranner* Netta Cohen*† David C. Hogg*†
University of Leeds, Leeds, United Kingdom

In this supplementary material we provide additional detail for some of the steps in our method and include the ranges and values for the various parameters and hyperparameters. In Appendix G we provide a more detailed comparison between our method and the two other methods available. We report the results of an ablation study in Appendix H and in Appendix I we describe the three supporting videos (available here: `https://doi.org/10.6084/m9.figshare.22310650`) that demonstrate the method and showcase some of the results.

A. Geometric model: The Bishop frame

3D curves are typically expressed in the Frenet frame TNB where T refers to the normalised tangent of the curve, N is the 'normal' vector defined as the normalised arc-length derivative of T and B is the 'binormal' vector obtained through the cross product $B = T \times N$. This frame is defined along the curve using the Frenet-Serret formulas:

$$\dot{T} = \kappa N, \tag{S1}$$

$$\dot{N} = -\kappa T + \tau B, \tag{S2}$$

$$\dot{B} = -\tau N, \tag{S3}$$

where dot denotes the arc-length derivative d/ds and κ and τ are scalar fields generally called curvature and torsion respectively. For simplicity we leave the arc length parameter s implicit in all equations.

A difficulty with the Frenet formulation is that the torsion, τ, is strictly undefined for straight curves, or locally wherever $\kappa = 0$. Zero (or near-zero) curvature is expected in an animal that propagates sinusoidal waves along its body and at these points we cannot guarantee a unique and consistent parametrisation. To overcome this ambiguity we use the Bishop frame [S1], given by TM^1M^2 where T again refers to the normalised tangent of the curve and M^1, M^2 form an orthogonal basis. The Bishop equations define how the frame changes along the curve:

$$\dot{T} = m^1M^1 + m^2M^2, \tag{S4}$$

$$\dot{M}^1 = -m^1T, \tag{S5}$$

$$\dot{M}^2 = -m^2T, \tag{S6}$$

where m^1, m^2 are scalar fields analogous to κ, τ that express the curvature in the M^1 and M^2 directions respectively.

While the Bishop frame improves the zero-curvature problem, it does leave a degree of freedom in the choice of the initial value of M^1 (M^1_{init}) that can point in any direction perpendicular to the initial tangent T_{init}. Any rotation of M^1_{init} around T_{init} will result in a different (m^1, m^2) representation of the curvature, but this rotation angle can easily be recovered and different representations subsequently aligned.

As Bishop describes in [S1] (and expanded here for completeness) the two frames are related through their scalar field components. κ can be recovered from m^1, m^2 using Eqs. (S1) and (S4) as:

$$\kappa = \left|\frac{dT}{ds}\right| = \left|m^1M^1 + m^2M^2\right| = \sqrt{(m^1)^2 + (m^2)^2}. \tag{S7}$$

To recover the torsion τ that describes the rotation of the Frenet frame around N let θ be the angle between N and M^1, then

$$N = M^1\cos\theta + M^2\sin\theta, \tag{S8}$$

$$B = -M^1\sin\theta + M^2\cos\theta, \tag{S9}$$

$$m^1 = \kappa\cos\theta \text{ and} \tag{S10}$$

$$m^2 = \kappa\sin\theta. \tag{S11}$$

Differentiating Eq. (S8) with respect to arc length and substituting from Eqs. (S5) and (S6) we have:

$$\dot{N} = \dot{\theta}\left(-M^1\sin\theta + M^2\cos\theta\right) \tag{S12}$$

$$+ T\left(m^1\cos\theta - m^2\sin\theta\right) \tag{S13}$$

$$= \dot{\theta}B - \kappa T(\cos^2\theta + \sin^2\theta) \tag{S14}$$

$$\implies \tau = \dot{\theta}. \tag{S15}$$

*{T.Ilett, O.Yuval, T.Ranner, N.Cohen, D.C.Hogg}@leeds.ac.uk
† Equal contribution.

Thus, in the words of Bishop, "κ and an indefinite integral $\int \tau ds$ are polar coordinates for the curve (m^1, m^2)".

B. Project: Pinhole camera model

The imaging setup is modelled using a triplet of pinhole camera models with tangential and radial distortion [S2]. A single pinhole camera model is used to project 3D points into an image plane using a perspective transformation. The 15 parameters required for each camera model are summarised in Table S3. These are divided into intrinsic, extrinsic and distortion parameters. The intrinsic parameters are (f_x, f_y, c_x, c_y), where f_x and f_y are the focal lengths and (c_x, c_y) is a principal point usually set to the image centre. The extrinsic parameters – angles (ϕ_0, ϕ_1, ϕ_2) and a translation vector t – define the extrinsic transformation $M = [R|t]$, where

$$R = R_z(\phi_0)R_y(\phi_1)R_x(\phi_2), \tag{S16}$$

$$t = \begin{pmatrix} t_0 \\ t_1 \\ t_2 \end{pmatrix} \tag{S17}$$

and R is a rotation matrix composed of three axial rotations:

$$R_z(\phi_0) = \begin{pmatrix} \cos\phi_0 & -\sin\phi_0 & 0 \\ \sin\phi_0 & \cos\phi_0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{S18}$$

$$R_y(\phi_1) = \begin{pmatrix} \cos\phi_1 & 0 & \sin\phi_1 \\ 0 & 1 & 0 \\ -\sin\phi_1 & 0 & \cos\phi_0 \end{pmatrix} \text{ and} \tag{S19}$$

$$R_x(\phi_2) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_2 & -\sin\phi_2 \\ 0 & \sin\phi_2 & \cos\phi_2 \end{pmatrix}. \tag{S20}$$

The radial and tangential distortion coefficients, (k_1, k_2, k_3) and (p_1, p_2) respectively, complete the parametrisation.

Due to imperfections in the camera model (and possibly environmental vibrations of the setup), using fixed pinhole camera model parameters yields results with errors that vary over time and space (*i.e.* the accuracy depends on the worm position). Allowing the camera parameters to change freely between frames resolves the reconstruction errors, but the problem becomes under-determined and introduces drift into both the camera and curve parameters resulting in incorrect tracking. To compensate for these errors without allowing full freedom of movement we use fixed pinhole camera parameters and introduce frame-dependent variables that emulate relative movement between the cameras, hence limiting drift and providing stable reconstructions.

To this end, the standard pinhole camera model is extended to include $\eta^s = (dx, dy, dz)$, relative shifts along the local coordinate axes (see Sec. 4.1 in the main text). These parameters approximate changes in the relative positions and rotations of the cameras by applying pixel translations after the perspective transformation. Without loss of generality η^s can be limited to one direction per camera, thus capturing only relative shifts. The shifts used in camera index c are given by $(s_x, s_y)_c$ where:

$$(s_x, s_y)_0 = (dx, 0), \tag{S21}$$

$$(s_x, s_y)_1 = (0, -dy), \text{ and} \tag{S22}$$

$$(s_x, s_y)_2 = (0, dz). \tag{S23}$$

For 3D object point (X, Y, Z), the corresponding projected image point (u, v) is generated using the following procedure (when $z \neq 0$):

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + t, \tag{S24}$$

$$x' = \frac{x}{z} + \frac{s_x}{f_x}, \tag{S25}$$

$$y' = \frac{y}{z} + \frac{s_y}{f_y}, \tag{S26}$$

$$r^2 = x'^2 + y'^2, \tag{S27}$$

$$k = 1 + k_1 r^2 + k_2 r^4 + k_3 r^6, \tag{S28}$$

$$x'' = kx' + 2p_1 x'y' + p_2(r^2 + 2x'^2), \tag{S29}$$

$$y'' = ky' + p_1(r^2 + 2y'^2) + 2p_2 x'y', \tag{S30}$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f_x x'' + c_x \\ f_y y'' + c_y \end{pmatrix}. \tag{S31}$$

Note the inclusion of the shift parameters in Eqs. (S25) and (S26).

C. Rendering parameters: Tapering

The rendering stage generates super-Gaussian blobs at the projected image locations of each curve vertex (n). The shape of the blobs in camera c depends on the optimisable parameters: the scale σ_c, the intensity ι_c and the exponent used in the Gaussian ρ_c. To capture the worm shape we taper the values of σ_c and ι_c from their optimisable values along the middle 60% down to fixed minimum values σ_{min} and ι_{min} respectively at the ends. The tapered outputs $\bar{\sigma}_c, \bar{\iota}_c \in \mathbb{R}^N$ are calculated thus:

$$\bar{\sigma}_{c,n} = \begin{cases} \sigma_{\text{min}}(1 - \frac{5n}{N}) + \sigma_c \frac{5n}{N} & 0 \leq n < N/5 \\ \sigma_c & N/5 \leq n < 4N/5 \\ \sigma_c(1 - \frac{n-4N/5}{N-4N/5}) + \sigma_{\text{min}} \frac{n-4N/5}{N-4N/5} & 4N/5 \leq n < N, \end{cases} \tag{S32}$$

and

$$\bar{\iota}_{c,n} = \begin{cases} \iota_{\text{min}}(1 - \frac{5n}{N}) + \iota_c \frac{5n}{N} & 0 \leq n < N/5 \\ \iota_c & N/5 \leq n < 4N/5 \\ \iota_c(1 - \frac{n-4N/5}{N-4N/5}) + \iota_{\text{min}} \frac{n-4N/5}{N-4N/5} & 4N/5 \leq n < N. \end{cases} \tag{S33}$$

These values are used in Eq. (2) in the main text.

D. Mask generation

The input images are masked to focus the pixel-errors to a single region, local to the predicted curve, that is consistent across all three views and excludes any interference that does not correspond to the same mass. The masks $M \in \mathbb{R}^{3 \times w \times w}$ are generated in a similar way to the renders R (see Sec. 4.2 in the main text), but with a few notable differences. First, the blobs B are normalised and weighted by the relative scores,

$$B'_{c,n} = \frac{B_{c,n}}{\sum_{i,j} B_{c,n}} \cdot \hat{S}_n, \tag{S34}$$

then combined by taking the maximum values as before,

$$M'_{c,i,j} = \max\{B'_{c,n,i,j}\}_{n=0,\dots,N-1}, \tag{S35}$$

and finally passed through a threshold:

$$M_{c,i,j} = \begin{cases} 1 & M'_{c,i,j} \geq \Theta \\ 0.2 & M'_{c,i,j} < \Theta. \end{cases} \tag{S36}$$

For $\Theta = 0$ we have $M = 1$ everywhere and no masking occurs. For $\Theta \sim 1$ the mask shrinks around the blobs that correspond to the highest scoring vertices, making $M = 0.2$ almost everywhere. In all our experiments we fix $\Theta = 0.1$ as this appears to produce a good balance. Note that we do not completely exclude the remaining points, but just reduce their intensity. This allows some gradient to flow from outside the detection region which is especially important in the early stages when none of the curve may be intersecting the correct pixel mass. It is also important to detach the masks from the gradient computation at this stage otherwise the curve will simply shrink and fade away from the high-intensity pixels thus minimising pixel errors simply by detecting fewer pixels.

E. Centre-shifting

The curve is periodically shifted along its length to centre it over the pixel mass in all three views. An unbalanced alignment can be seen from the score profile when the centre-of-mass index of $\hat{S}$ ($\bar{n}$) is not in the middle of the curve (*i.e.* $\bar{n} \neq N/2$). We can then shift the curve along its length using $\bar{n}$ as the new midpoint, removing vertices from the low-scoring end and adding new vertices to the high-scoring end. The low-scoring end will consequently improve, and since there is no expectation that the new vertices will match the images this typically means the high-scoring end worsens; rectifying the imbalance.

To perform a centre-shift we calculate the centre of mass of the score profile and the degree of imbalance as:

$$\bar{n} = \frac{\sum_n n\hat{S}_n}{\sum_n \hat{S}_n}, \text{ and} \tag{S37}$$

$$n_s = \bar{n} - N/2. \tag{S38}$$

Then we update the curvature by shifting the values and decreasing linearly to zero at the ends. I.e. for $n_s > 0$,

$$K_n \leftarrow \begin{cases} K_{n+n_s} & 0 \leq n < N - n_s, \\ K_{N-n_s-1}(1 - \frac{N-n_s-n+1}{n_s}) & N - n_s \leq n < N, \end{cases} \tag{S39}$$

and similarly for $n_s < 0$. Finally, new position and orientation parameters are calculated from the adjusted midpoint and updated curvatures using Eq. (1) from the main text:

$$(P, T, M^1) \leftarrow F\left(P_{\bar{n}}, T_{\bar{n}}, M^1_{\bar{n}}, K, l, \bar{n}\right). \tag{S40}$$

This process is illustrated in Fig. 6 in the main text.

Centre-shifting the curve occurs between gradient descent optimisation steps. In practise, shifting after every step quickly leads to instabilities as the new points are not afforded the time required to align them with the images. Furthermore, it is unrealistic to expect a perfect balance can be sustained and an unconstrained n_s means large shifts may be applied, possibly due to a change in the camera parameters or some transient interference, that would destroy extensive sections of the curve. To mitigate these problems we only apply centre-shifting every α steps when $|n_s| > \beta N$ and then restrict the shift size to γ (*i.e.* $n_s \leftarrow \min\{n_s, \gamma\}$). In our experiments we find values of $\alpha \in [3, 6]$, $\beta \in (0.05, 0.1)$ and $\gamma \in [1, 2]$ provide the necessary stabilisations (Table S6).

F. Optimisation

The non-optimisable parameter values and ranges that are used in our experiments are outlined in Table S6. The biggest factors affecting the choice of parameters are the magnification and individual worm size – both of which vary between experiments. These determine the required image size, w, and inform the estimates for the length bounds, l_{min} and l_{max}. The super-Gaussian blobs are generated in corresponding $w \times w$ images, so the minimum scales and intensities at the tips, σ_{min} and ι_{min}, must also change accordingly with the image and worm size.

Table S7 lists the weighting coefficients used in the combined loss calculation (Eq. (15) in the main text). Values of

ω_{sm} and ω_{t} may vary between experiments to capture the different dynamics observed in the different environmental conditions (specifically, concentration of the gelatin). For example, when the worm is deforming quickly (in low-viscosity experiments) there are large postural changes between frames and therefore the temporal loss $\mathcal{L}_{\mathrm{t}}$ is relatively big. In this case a smaller value of ω_{t} is used to prevent the reconstruction lagging behind the worm. Similarly, when the worm is deforming slowly (in high-viscosity experiments) it frequently forms tightly coiled postures in which case the smoothness loss $\mathcal{L}_{\mathrm{t}}$ is large and a smaller value for ω_{sm} is more suitable. As discussed in the main text, when reconstructing full sequences the initial curve and parameters are used for the initial guesses to the subsequent frame. This preserves head-tail orientation and consistency through complex manoeuvres.

The learning rates are shown in Table S8 – these are fixed for all experiments. Optimisable camera, curve and render parameters are summarised in Tables S3 to S5 respectively.

G. Comparisons with previous methods

In Fig. 8 in the main text we compare results against two existing methods [S3,S4]. These methods only provide projected midline coordinates, so in order to use our pipeline to generate renders and calculate $\mathcal{L}_{\mathrm{px}}$ we need to provide renderer (blob) parameters. We can use values found for our midlines, but this introduces a bias towards our method as these parameters are only optimal for our midlines. To mitigate against this we initialise with our values but re-optimise for each frame until convergence (keeping the curve and camera parameters fixed) to ensure optimal rendering parameters are found for each midline. In Fig. S1 we show the effect of the re-optimisation across the same clip. As expected, re-optimisation reduces the loss for both methods, but the improvement is fairly marginal. The improved losses are used for comparison in the main text.

In Table S1 we evaluate midline quality of all three methods against ground-truth manual annotations. This unbiased evaluation (as only projected midline points are used) shows our method to be more accurate. However, the sample size is limited, more so as the other methods only provide results for roughly half of the available annotations.

H. Ablation study

We demonstrate some of the effects of masking (Fig. 5), centre-shifting (Fig. 6) and varying parameters (Movie 1) throughout the main text. In Table S2 we present the results from a more thorough ablation study conducted over a typical $\sim 5\,\mathrm{min}$ clip to clarify the importance of the different components of our method.

Optimisable camera (a), rendering parameters (b) and centre-shifting (c) yield considerable benefit; setting $\omega_{\mathrm{sc}} > 0$ (d) and regularisation (f) incur a marginal cost ($\tilde{\mathcal{L}} > 0.99$) but recover poses with high transparency (d) and ensure realistic (smooth) poses (f); finally, masking (e) incurs a small cost but provides robustness to sequences with interference (*e.g.* Movie 1).

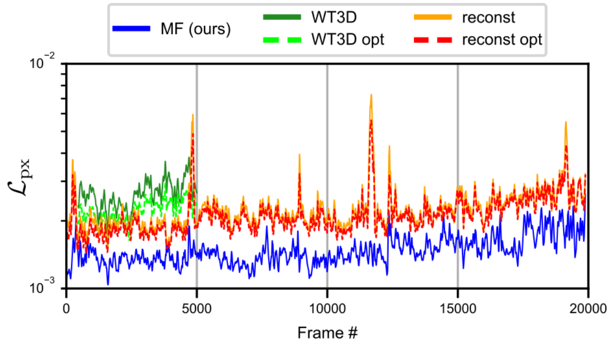

Figure S1. A comparison between the losses obtained when using our renderer parameters for Yuval's Worm-Tracker 3D (WT3D) [S4] and Salfelder *et al.*'s 'reconst' [S3] midlines *vs.* the losses obtained after re-optimising the renderer parameters to better suit their midlines. Our method (MF) is shown for reference. We show moving averages over 25 frames (1 s).

Method	# poses	Score	Total
MF (ours)	487	**1.53** (0.51)	4 h 37 min
reconst [S3]	226	1.54 (0.69)	1 h 32 min
MF (ours)	226	**1.34** (0.53)	
WT3D [S4]	237	2.64 (0.92)	45 min
MF (ours)	237	**1.46** (0.59)	

Table S1. Mean (and standard deviation) pixel distances between predicted points and hand-annotated points (see Fig. 7). Total refers to the overall reconstructed duration using each method.

I. Supplementary results

Three supporting videos are available here: `https://doi.org/10.6084/m9.figshare.22310650`.

In the accompanying Movie 1, the effects of adjusting some of the parameters listed in Tables S6 and S7 on the solution are demonstrated. In this video we show the optimisation process using paired examples. The same frame and randomised initial guess are used for each pair and the optimisation is run for a fixed 2000 steps. The frames are selected to demonstrate a range of challenging conditions – especially for achieving convergence from a random guess. The first of each pair shows successful optimisation using parameter values in the ranges specified in Tables S6 and S7. The second of each pair shows the effect that chang-

Variant	$< \mathcal{L}_{\text{px}} >$	$\tilde{\mathcal{L}} = < \mathcal{L}_{\text{px}} > /$ Ref
Ref	3.99e−3 (1.13e−3)	1.000
a	7.21e−3 (1.97e−3)	1.807
b	4.64e−3 (1.42e−3)	1.163
c	4.29e−3 (1.71e−3)	1.075
d	3.99e−3 (1.12e−3)	1.000
e	3.69e−3 (9.00e−4)	0.925
f	3.96e−3 (1.08e−3)	0.992

Table S2. Ablation results (mean (and standard deviation) pixel errors and normalised mean) across a typical 7200-frame clip. Variants: a) no camera parameter optimisation, b) rendering parameters (σ, ι, ρ) fixed to averages from the reference results, c) no centre-shifting, d) no scores-loss ($\omega_{\text{sc}} = 0$), e) no input masking, f) no regularisation losses ($\omega_{\text{sm}} = \omega_{\text{t}} = 0$).

ing one of the parameters has on the converged solution.

We include two further videos demonstrating examples of both successful and less-successful sequence reconstructions. In Movie 2 we showcase successful examples. First, when the worm is well resolved in all three views. Next, when there is interference from dirt or bubbles and/or poor focus in one or more views. Lastly, through complex coiling manoeuvres that include significant self-occlusion.

The limitations of our method are illustrated in Movie 3. These examples, taken from otherwise successful reconstructions, demonstrate that when significant loss of focus is combined with coiling, heavily occluded, postures the reconstruction can fail. The exact degree of failure is difficult to ascertain for the exact same reasons and only by watching the full sequences can we be convinced that the reconstruction is incorrect. This suggests that incorporating more temporal information may help to resolve these fail-cases, but we leave this for future investigation.

References

[S1] Richard L. Bishop. There is more than one way to frame a curve. *Amer. Math. Monthly*, 82(3):246–251, Mar. 1975. 1

[S2] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Mar. 2004. 2

[S3] Felix Salfelder, Omer Yuval, Thomas P Ilett, David C Hogg, Thomas Ranner, and Netta Cohen. Markerless 3D spatio-temporal reconstruction of microscopic swimmers from video. In *Visual observation and analysis of Vertebrate And Insect Behavior 2020*, 2021. 4

[S4] Omer Yuval. *The neuromechanical control of Caenorhabditis elegans head motor behaviour in 3D environments*. PhD thesis, University of Leeds, 2022. 4

Parameter	**Purpose**	**Domain**
f_x,f_y	Focal lengths	$\mathbb{R}^+$
(c_x, c_y)	Principal image point	$\mathbb{R}^+$
ϕ_0, ϕ_1, ϕ_2	Rotation angles	$[0, 2\pi)$
t	Translation (position) vector	$\mathbb{R}^3$
(k_1, k_2, k_3)	Radial distortion coefficients	$\mathbb{R}^+$
(p_1, p_2)	Tangential distortion coefficients	$\mathbb{R}^+$
η^s	Relative shifts	$\mathbb{R}^3$

Table S3. Camera model parameters. With the exception of η^s these are defined for each camera. η^s is shared between the models as per Eqs. (S21) to (S23).

Parameter	**Purpose**	**Domain**
P	3D curve vertex coordinates	$\mathbb{R}^{N\times 3}$
T	Normalised curve tangent vectors at each vertex location	$\mathbb{R}^{N\times 3}$
M^1	Normalised curvature orientation vectors at each vertex location	$\mathbb{R}^{N\times 3}$
K	Vector curvature	$\mathbb{R}^{N\times 2}$
l	Curve length	$(l_{\text{min}}, l_{\text{max}})$

Table S4. Curve and Bishop frame parameters.

Parameter	**Purpose**	**Domain**
σ_c	Standard deviation of the super-Gaussian blobs along the untapered middle 60% of the worm in camera c	$[\sigma_{\text{min}}, \infty)$
ι_c	Intensity scaling factor for the super-Gaussian blobs along the untapered middle 60% of the worm in camera c	$[\iota_{\text{min}}, \infty)$
ρ_c	Exponent used in the super-Gaussian blobs in camera c	$(0, \infty)$

Table S5. Rendering parameters.

Parameter	Purpose	Value/Range
w	(Square) image size	$200–350$ px
N	Number of discrete curve vertices	128
l_{min}	Minimum curve length	$0.5–1$ mm
l_{max}	Maximum curve length	$1–2$ mm
k_{max}	Maximum curvature constraint	3 osculating circles
σ_{min}	Standard deviation of the super-Gaussian functions at the tips	$2–4$ px
ι_{min}	Intensity scaling factor of the super-Gaussian functions at the tips	$0.15–0.3$
Θ	Mask threshold	0.1
α	Frequency of centre-shift adjustments (number of gradient descent steps)	3–6 steps
β	Centre-shift adjustment sensitivity	$0.05–0.1$
γ	Maximum centre-shift adjustment	1–2 vertices

Table S6. Non-optimisable parameter values and ranges used in our experiments. Listed in the order they appear in the text.

Parameter	Purpose	Value/Range
ω_{px}	Weighting of the pixel loss $\mathcal{L}_{\text{px}}$	0.1
ω_{sc}	Weighting of the scores loss $\mathcal{L}_{\text{sc}}$	0.01
ω_{sm}	Weighting of the smoothness loss $\mathcal{L}_{\text{sm}}$	$10–100$
ω_{t}	Weighting of the temporal loss $\mathcal{L}_{\text{t}}$	$10–100$
ω_{i}	Weighting of the intersections loss $\mathcal{L}_{\text{i}}$	$0.1–1$

Table S7. Weighting coefficients for the different loss terms.

Parameter	Purpose	Value
λ_p	Learning rate for the curve parameters $\{P, T, M^1, K, l\}$	$1\text{e}{-3}$
λ_r	Learning rate for the rendering parameters $\{\sigma, \iota, \rho\}$	$1\text{e}{-4}$
λ_η	Learning rate for the camera parameters η	$1\text{e}{-5}$
λ_{min}	Minimum learning rate for all parameters	$1\text{e}{-6}$

Table S8. Learning rates for the different parameter groups.