PIE-NeRF: Physics-based Interactive Elastodynamics with NeRF

Supplementary Material

A. Weight derivatives

The displacement field of with Q-GMLS interpolation is $\mathbf{u}(\mathbf{x}) = \mathbf{J}(\mathbf{x})\mathbf{q}$(i.e., Eq (9), where $\mathbf{J} = [N_1\mathbf{I}, N_1^1\mathbf{I}, N_1^2\mathbf{I}, ..., N_1^{11}\mathbf{I}, ...] \in \mathbb{R}^{3\times 30n}$. The Jacobian and Hessian matrices of $\mathbf{J}$, $\nabla\mathbf{J}_k^\top$ and $\nabla^2\mathbf{J}_k^\top$, are required in IP integration and IP ray warping.

The computation of derivatives boils down to the computation of the Jacobian and Hessian matrices of Q-GMLS weighting functions N_i, N_i^j and N_i^{jk}. Recall our weighting functions:

$$
\begin{aligned}
N_i(\mathbf{x}) &= \mathbf{p}^\top(\mathbf{x})\mathbf{G}^{-1}(\mathbf{x})\mathbf{p}(\mathbf{x}_i)w(\mathbf{x}-\mathbf{x}_i),\\
N_i^j(\mathbf{x}) &= \mathbf{p}^\top(\mathbf{x})\mathbf{G}^{-1}(\mathbf{x})\mathbf{p}_{,j}(\mathbf{x}_i)w(\mathbf{x}-\mathbf{x}_i),\\
N_i^{jk}(\mathbf{x}) &= \mathbf{p}^\top(\mathbf{x})\mathbf{G}^{-1}(\mathbf{x})\mathbf{p}_{,jk}(\mathbf{x}_i)w(\mathbf{x}-\mathbf{x}_i),
\end{aligned}
$$

where $\mathbf{p}(\mathbf{x}) = [1, x, y, z, x^2, xy, xz, y^2, yz, z^2]^\top$ is the Q-GMLS polynomial basis, and $\mathbf{p}_{,j}$, $\mathbf{p}_{,jk}$ are its first- and second-order derivative respectively, which are trivial to compute. The notations can be simplified as:

$$
\begin{aligned}
N_i(\mathbf{x}) &= \mathbf{S}(\mathbf{x})\mathbf{p}(\mathbf{x}_i),\\
N_i^j(\mathbf{x}) &= \mathbf{S}(\mathbf{x})\mathbf{p}_{,j}(\mathbf{x}_i),\\
N_i^{jk}(\mathbf{x}) &= \mathbf{S}(\mathbf{x})\mathbf{p}_{,jk}(\mathbf{x}_i),
\end{aligned}
\tag{1}
$$

for $\mathbf{S}(\mathbf{x}) = w(\mathbf{x}-\mathbf{x}_i)\mathbf{p}(\mathbf{x})\mathbf{G}^{-1}(\mathbf{x})$. Hence, once we get the Jacobian and Hessian of $\mathbf{S}(\mathbf{x})$, we can easily compute $\nabla\mathbf{J}$ and $\nabla^2\mathbf{J}$, since $\mathbf{x}_i$ does not depend on $\mathbf{x}$. In our implement, the kernel function is $w(\mathbf{x}-\mathbf{x}_i)$ is $w(\mathbf{x}-\mathbf{x}_i) = (1-\|\mathbf{d}\|^2)^3$, whose derivatives are:

$$
\begin{aligned}
\nabla w(\mathbf{x}) &= -6(1-\mathbf{x}^\top\mathbf{x})^2\mathbf{x},\\
\nabla^2 w(\mathbf{x}) &= 6(1-\mathbf{x}^\top\mathbf{x})(4\mathbf{x}\mathbf{x}^\top - (1-\mathbf{x}^\top\mathbf{x})\mathbf{I}).
\end{aligned}
\tag{2}
$$

To compute the derivatives of $\mathbf{G}(\mathbf{x})^{-1}$, we first derive the derivatives of $\mathbf{G}(\mathbf{x})$ as:

$$
\begin{aligned}
\nabla\mathbf{G}(\mathbf{x}) &= \sum_{i=1}^{n}[\mathbf{p}(\mathbf{x}_i)\mathbf{p}^\top(\mathbf{x}_i) + \sum_j \mathbf{p}_{,j}(\mathbf{x}_i)\mathbf{p}_{,j}^\top(\mathbf{x}_i)\\
&\quad + \sum_{j,k}\mathbf{p}_{,jk}(\mathbf{x}_i)\mathbf{p}_{,jk}^\top(\mathbf{x}_i)] \otimes \nabla w(\mathbf{x}-\mathbf{x}_i),\\
\nabla^2\mathbf{G}(\mathbf{x}) &= \sum_{i=1}^{n}[\mathbf{p}(\mathbf{x}_i)\mathbf{p}^\top(\mathbf{x}_i) + \sum_j \mathbf{p}_{,j}(\mathbf{x}_i)\mathbf{p}_{,j}^\top(\mathbf{x}_i)\\
&\quad + \sum_{j,k}\mathbf{p}_{,jk}(\mathbf{x}_i)\mathbf{p}_{,jk}^\top(\mathbf{x}_i)] \otimes \nabla^2 w(\mathbf{x}-\mathbf{x}_i).
\end{aligned}
\tag{3}
$$

$\nabla\mathbf{G}(\mathbf{x})^{-1}$ and $\nabla^2\mathbf{G}(\mathbf{x})^{-1}$ can then be computed as:

$$
\begin{aligned}
\mathbf{G}(\mathbf{x})_{,j}^{-1} &= -\mathbf{G}(\mathbf{x})^{-1}\mathbf{G}_{,j}(\mathbf{x})\mathbf{G}(\mathbf{x})^{-1},\\
\mathbf{G}(\mathbf{x})_{,jk}^{-1} &= \mathbf{G}(\mathbf{x})^{-1}\mathbf{G}_{,k}(\mathbf{x})\mathbf{G}(\mathbf{x})^{-1}\mathbf{G}_{,j}(\mathbf{x})\mathbf{G}(\mathbf{x})^{-1}\\
&\quad + \mathbf{G}(\mathbf{x})^{-1}\mathbf{G}_{,j}(\mathbf{x})\mathbf{G}(\mathbf{x})^{-1}\mathbf{G}_{,k}(\mathbf{x})\mathbf{G}(\mathbf{x})^{-1}\\
&\quad - \mathbf{G}(\mathbf{x})^{-1}\mathbf{G}_{,jk}(\mathbf{x})\mathbf{G}(\mathbf{x})^{-1}.
\end{aligned}
\tag{4}
$$

Putting together we have:

$$
\begin{aligned}
\mathbf{S}_{,j}(\mathbf{x}) &= w_{,j}(\mathbf{x}-\mathbf{x}_i)\mathbf{G}^{-1}(\mathbf{x})\mathbf{p}(\mathbf{x})\\
&\quad + w(\mathbf{x}-\mathbf{x}_i)\mathbf{G}^{-1}(\mathbf{x})\mathbf{p}_{,j}(\mathbf{x})\\
&\quad + w(\mathbf{x}-\mathbf{x}_i)\mathbf{G}_{,j}^{-1}(\mathbf{x})\mathbf{p}(\mathbf{x}),\\
\mathbf{S}_{,jk}(\mathbf{x}) &= w_{,jk}(\mathbf{x}-\mathbf{x}_i)\mathbf{G}^{-1}(\mathbf{x})\mathbf{p}(\mathbf{x})\\
&\quad + w_{,j}(\mathbf{x}-\mathbf{x}_i)\mathbf{G}_{,k}^{-1}(\mathbf{x})\mathbf{p}(\mathbf{x})\\
&\quad + w_{,j}(\mathbf{x}-\mathbf{x}_i)\mathbf{G}^{-1}(\mathbf{x})\mathbf{p}_{,k}(\mathbf{x})\\
&\quad + w_{,k}(\mathbf{x}-\mathbf{x}_i)\mathbf{G}^{-1}(\mathbf{x})\mathbf{p}_{,j}(\mathbf{x})\\
&\quad + w(\mathbf{x}-\mathbf{x}_i)\mathbf{G}_{,k}^{-1}(\mathbf{x})\mathbf{p}_{,j}(\mathbf{x})\\
&\quad + w(\mathbf{x}-\mathbf{x}_i)\mathbf{G}^{-1}(\mathbf{x})\mathbf{p}_{,jk}(\mathbf{x})\\
&\quad + w_{,k}(\mathbf{x}-\mathbf{x}_i)\mathbf{G}_{,j}^{-1}(\mathbf{x})\mathbf{p}(\mathbf{x})\\
&\quad + w(\mathbf{x}-\mathbf{x}_i)\mathbf{G}_{,jk}^{-1}(\mathbf{x})\mathbf{p}(\mathbf{x})\\
&\quad + w(\mathbf{x}-\mathbf{x}_i)\mathbf{G}_{,j}^{-1}(\mathbf{x})\mathbf{p}_{,k}(\mathbf{x}).
\end{aligned}
\tag{5}
$$

B. Energy integration

B.1. Potential energy

We assume that each IP is a small elastic cuboid Ω_k. The elastic potential energy is computed as:

$$
\begin{aligned}
U_k &= \int_{\Omega_k}\Psi(\mathbf{F}(\mathbf{h}))\mathrm{d}V\\
&= \int_{-\frac{h_1}{2}}^{\frac{h_1}{2}}\int_{-\frac{h_2}{2}}^{\frac{h_2}{2}}\int_{-\frac{h_3}{2}}^{\frac{h_3}{2}}\Psi\left(\mathbf{F}(\sum_{i=1}^{3}x_i\mathbf{c}_i)\right)\mathrm{d}x_1\mathrm{d}x_2\mathrm{d}x_3.
\end{aligned}
\tag{6}
$$

We give the derivation for two specific energies namely ARAP and Neo-Hookean. For simplicity, we omit the notations of $\mathrm{d}x_1\mathrm{d}x_2\mathrm{d}x_3$. Also note that:

$$
\int_{\Omega_k}x_ix_j\mathbf{c}_i\mathbf{c}_j^\top = 0, \forall i \neq j.
$$

ARAP elasticity. The ARAP energy density is

$$
\Psi(\mathbf{F}) = ||\mathbf{F}-\mathbf{R}||^2. \tag{7}
$$

With first-order approximate the deformation gradient inside the cuboid, the integrated potential is:

$$U_k = \int_{\Omega_k} ||\mathbf{F} + \nabla\mathbf{F}\sum_{i=1}^{3} x_i\mathbf{c}_i - \mathbf{R}||^2 =$$

$$\frac{V}{12}\mathrm{tr}\left((\mathbf{F}-\mathbf{R})(\mathbf{F}-\mathbf{R})^\top\right) + \frac{V}{12}\sum_{i=1}^{3} h_i^2\,||\nabla\mathbf{F}\cdot\mathbf{c}_i||^2 . \quad (8)$$

Here, $\nabla\mathbf{F}$ can be obtained as:

$$\nabla\mathbf{F} = \sum_{j=1}^{n} \mathbf{u}_j \otimes \nabla^2 N_j, \quad (9)$$

or using generalized coordinates:

$$\nabla\mathbf{F} = \mathbf{q}\cdot\nabla^2\mathbf{J}^\top . \quad (10)$$

Neo-Hookean elasticity. The Neo-Hookean energy density is:

$$\Psi(\mathbf{F}) = \frac{\mu}{2}\left(\mathrm{tr}\,(\mathbf{F}\mathbf{F})^\top - 3\right) - \mu\log J + \frac{\lambda}{2}\log^2 J, \quad (11)$$

where $J = \det(\mathbf{F})$. The first term $\mathrm{tr}(\mathbf{F}\mathbf{F}^\top)$ is integrated as:

$$\int_{\Omega_k} \mathrm{tr}\left((\mathbf{F}+\nabla\mathbf{F}\sum_{i=1}^{3} x_i\mathbf{c}_i)(\mathbf{F}+\nabla\mathbf{F}\sum_{i=1}^{3} x_i\mathbf{c}_i)^\top\right)$$
$$= V\mathrm{tr}(\mathbf{F}\mathbf{F}^\top) + \frac{V}{12}\sum_{i=1}^{3} h_i^2\,||\nabla\mathbf{F}\cdot\mathbf{c}_i||^2 . \quad (12)$$

For the two log terms, we integrate based on the first-order approximation $\log J(\mathbf{x}) = \log J + \frac{\nabla J}{J}\cdot\Delta\mathbf{x}$ at the center of the cuboid. The second term is:

$$\int_{\Omega_k} \log J + \frac{\nabla J}{J}\cdot\Delta\mathbf{x} = V\log J, \quad (13)$$

and the third term is:

$$\int_{\Omega_k} \log^2 J + 2\log J\frac{\nabla J}{J}\cdot\Delta\mathbf{x} + \frac{1}{J^2}\nabla J^\top\Delta\mathbf{x}\Delta\mathbf{x}^\top\nabla J$$
$$= V\log^2 J + \frac{V}{12J^2}\nabla J^\top\,\mathbf{C}\,\nabla J, \quad (14)$$

where $\mathbf{C}$ is the covariance matrix.

We also need to derive ∇J. Let $\mathbf{F} = [\mathbf{f}_1|\mathbf{f}_2|\mathbf{f}_3]$. The determinant of $\mathbf{F}$ can thus be calculated as $J = \det(\mathbf{F}) = \mathbf{f}_1\times\mathbf{f}_2\cdot\mathbf{f}_3$. Using the chain rule, we get

$$\nabla J = \frac{\partial J}{\partial\mathbf{F}} : \nabla\mathbf{F}, \quad (15)$$

where $\frac{\partial J}{\partial\mathbf{F}} = [\mathbf{f}_2\times\mathbf{f}_3\,|\mathbf{f}_3\times\mathbf{f}_1|\,\mathbf{f}_1\times\mathbf{f}_2]$. Finally, we can assemble the potential energy of Neo-Hookean using Eqs. (12), (13) and (14).

B.2. Derivatives of potential energy

To obtain the generalized energy gradient ($\mathbf{f}_{int}$) and Hessian ($\frac{\partial\mathbf{f}_{int}}{\partial\mathbf{q}}$), we focus on $\Psi_a = \frac{V}{12}\sum_{i=1}^{3} h_i^2\,||\nabla\mathbf{F}\cdot\mathbf{c}_i||^2$ and $\Psi_b = \frac{V}{12J^2}\nabla J^\top\,\mathbf{C}\,\nabla J$.

Re-write Ψ_a as:

$$\Psi_a = \frac{V}{12}\sum_{i_1}\sum_{i_2}\mathbf{u}_{i_1}^\top\mathbf{u}_{i_2}\mathrm{tr}\left(\nabla^2 N_{i_1}\nabla^2 N_{i_2}\mathbf{C}\right), \quad (16)$$

where $\mathbf{u}_i$ is a vector of $\mathbf{q}$'s $3i$-th row to $(3i+2)$-th row, $N_i\mathbf{I}$ is a matrix consist of the $3i$-th to $(3i+2)$-th columns of $\mathbf{J}$. As Ψ_a is quadratic w.r.t. $\mathbf{u}$, we can directly write the stiffness matrix block of IP e as:

$$\mathbf{K}^e_{a,i_1,i_2} = \frac{V}{6}\mathrm{tr}\left(\nabla^2 N_{i_1}\nabla^2 N_{i_2}\mathbf{C}\right)\mathbf{I}. \quad (17)$$

We assemble blocks of all the IPs to get the system stiffness matrix $\mathbf{K}_a$ of Ψ_a. $\mathbf{f}_{int,a}$ is then computed as $\mathbf{f}_{int,a} = \mathbf{K}_a\cdot\mathbf{q}$.

Ψ_b contains many non-linear terms, whose derivatives are more involved. We use the chain rule to calculate $f_{int,b}$ and Hessian $\mathbf{K}_b$. To assemble these terms, we need $\frac{\partial J}{\partial\mathbf{q}}$, $\frac{\partial^2 J}{\partial\mathbf{q}^2}$, $\frac{\partial\nabla J}{\partial\mathbf{q}}$ and $\frac{\partial^2\nabla J}{\partial\mathbf{q}^2}$, which are computed as:

$$\begin{aligned}
\frac{\partial J}{\partial\mathbf{q}} &= \frac{\partial J}{\partial\mathbf{F}}\frac{\partial\mathbf{F}}{\partial\mathbf{q}},\\
\frac{\partial^2 J}{\partial\mathbf{q}^2} &= \frac{\partial\mathbf{F}}{\partial\mathbf{q}}^\top : \frac{\partial^2 J}{\partial\mathbf{F}^2} : \frac{\partial\mathbf{F}}{\partial\mathbf{q}},\\
\frac{\partial\mathbf{F}}{\partial\mathbf{q}} &= \nabla\mathbf{J}.
\end{aligned} \quad (18)$$

Here, $\frac{\partial J}{\partial\mathbf{F}}$ and $\frac{\partial^2 J}{\partial\mathbf{F}^2}$ can be derived as in any J-based hyperelastic energy models. Note that $\mathbf{J}$ should not be confused with the determinant J. In the following, we denote $\mathbf{f}^j = \frac{\partial\mathbf{f}}{\partial q_j}$ as the partial differentiation of $\mathbf{f}$ w.r.t. j-th row of $\mathbf{q}$ to avoid high-order tensor notations:

$$\begin{aligned}
J^j_{,i} = {}&\mathbf{f}^j_{1,i}\times\mathbf{f}_2\cdot\mathbf{f}_3 + \mathbf{f}^j_1\times\mathbf{f}_{2,i}\cdot\mathbf{f}_3 + \mathbf{f}^j_1\times\mathbf{f}_2\cdot\mathbf{f}_{3,i}\\
&+ \mathbf{f}_{1,i}\times\mathbf{f}^j_2\cdot\mathbf{f}_3 + \mathbf{f}_1\times\mathbf{f}^j_{2,i}\cdot\mathbf{f}_3 + \mathbf{f}_1\times\mathbf{f}^j_2\cdot\mathbf{f}_{3,i}\\
&+ \mathbf{f}_{1,i}\times\mathbf{f}_2\cdot\mathbf{f}^j_3 + \mathbf{f}_1\times\mathbf{f}_{2,i}\cdot\mathbf{f}^j_3 + \mathbf{f}_1\times\mathbf{f}_2\cdot\mathbf{f}^j_{3,i}.
\end{aligned} \quad (19)$$

$$\begin{aligned}
J^{jk}_{,i} = {}&\mathbf{f}^j_{1,i}\times\mathbf{f}^k_2\cdot\mathbf{f}_3 + \mathbf{f}^j_1\times\mathbf{f}^k_{2,i}\cdot\mathbf{f}_3 + \mathbf{f}^j_1\times\mathbf{f}^k_2\cdot\mathbf{f}_{3,i}\\
&+ \mathbf{f}^j_{1,i}\times\mathbf{f}_2\cdot\mathbf{f}^k_3 + \mathbf{f}^j_1\times\mathbf{f}_{2,i}\cdot\mathbf{f}^k_3 + \mathbf{f}^j_1\times\mathbf{f}_2\cdot\mathbf{f}^k_{3,i}\\
&+ \mathbf{f}^k_{1,i}\times\mathbf{f}^j_2\cdot\mathbf{f}_3 + \mathbf{f}^k_1\times\mathbf{f}^j_{2,i}\cdot\mathbf{f}_3 + \mathbf{f}^k_1\times\mathbf{f}^j_2\cdot\mathbf{f}_{3,i}\\
&+ \mathbf{f}_{1,i}\times\mathbf{f}^j_2\cdot\mathbf{f}^k_3 + \mathbf{f}_1\times\mathbf{f}^j_{2,i}\cdot\mathbf{f}^k_3 + \mathbf{f}_1\times\mathbf{f}^j_2\cdot\mathbf{f}^k_{3,i}\\
&+ \mathbf{f}^k_{1,i}\times\mathbf{f}_2\cdot\mathbf{f}^j_3 + \mathbf{f}^k_1\times\mathbf{f}_{2,i}\cdot\mathbf{f}^j_3 + \mathbf{f}^k_1\times\mathbf{f}_2\cdot\mathbf{f}^j_{3,i}\\
&+ \mathbf{f}_{1,i}\times\mathbf{f}^k_2\cdot\mathbf{f}^j_3 + \mathbf{f}_1\times\mathbf{f}^k_{2,i}\cdot\mathbf{f}^j_3 + \mathbf{f}_1\times\mathbf{f}^k_2\cdot\mathbf{f}^j_{3,i}.
\end{aligned} \quad (20)$$

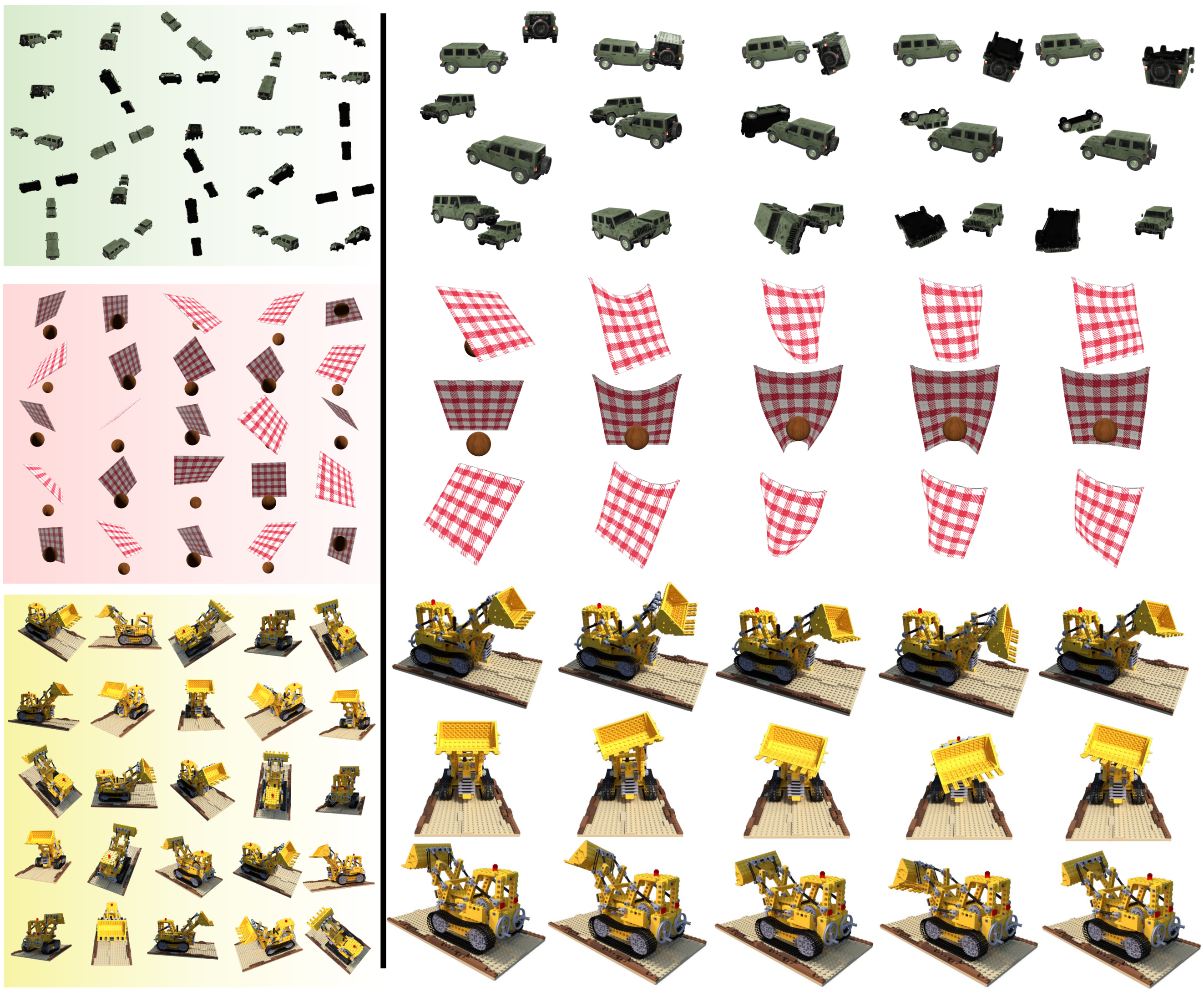

Figure 1. **Multiview dynamics:** We show more results using PIE-NerF. The left shadowed column gives the input NGP-NeRF training data i.e., static views from different angles. On the right, we show three dynamic scenes, rendered from three different camera poses.

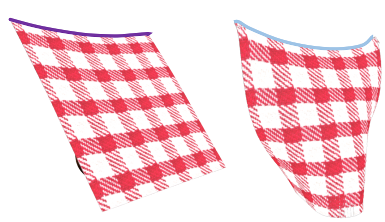

Figure 2. **Locking:** Linear MLS easily yields locking artifact (left) while Q-GMLS used in PIE-NeRF produces plausible results with the same number of DOFs (right).

With these components, the Jacobian and Hessian of Ψ_b can be assembled as:

$$\frac{\partial \Psi_b}{\partial \mathbf{q}} = -\frac{V}{6J^3}\nabla J^\top \mathbf{C} \nabla J \frac{\partial J}{\partial \mathbf{q}} + \frac{V}{6J^2}\nabla J^\top \mathbf{C} \frac{\partial \nabla J}{\partial \mathbf{q}}, \quad (21)$$

and

$$\frac{\partial^2 \Psi_b}{\partial \mathbf{q}^2} = \frac{V}{2J^4}\nabla J^\top \mathbf{C} \nabla J \frac{\partial J}{\partial \mathbf{q}} \otimes \frac{\partial J}{\partial \mathbf{q}} - \frac{2V}{3J^3}(\nabla J^\top \mathbf{C} \frac{\partial \nabla J}{\partial \mathbf{q}}) \otimes \frac{\partial J}{\partial \mathbf{q}} + \frac{V}{6J^2} \frac{\partial \nabla J}{\partial \mathbf{q}}^\top \mathbf{C} \frac{\partial \nabla J}{\partial \mathbf{q}}. \quad (22)$$

C. Linear locking

Linear locking refers to the situation where deformation elements exhibit smaller displacements and appear stiffer than

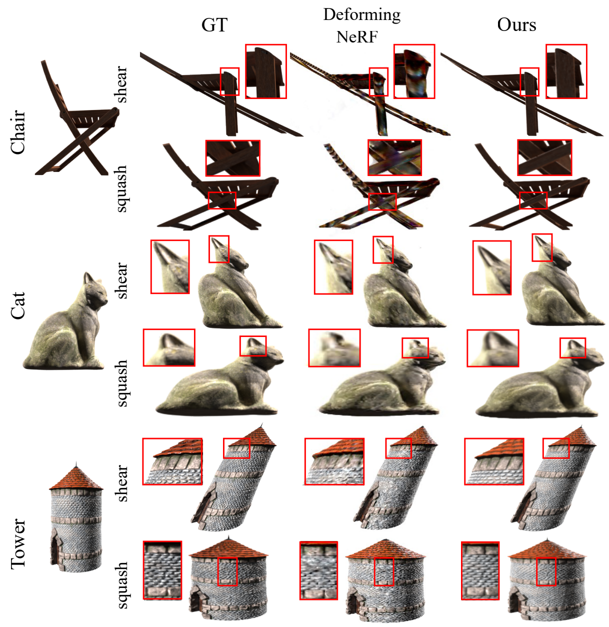

Figure 3. **Comparison with Deforming-NeRF:** We conducted shearing and squashing deformations on various models, with the ground truth generated by direct mesh vertex transformations.

Deformation		Shear		Squash	
Metrics		MSE↓	PSNR↑	MSE↓	PSNR↑
Chair	DN*	0.0096	21.9787	0.0076	23.1212
	Ours	**0.0022**	**28.8971**	**0.0012**	**31.8755**
Cat	DN	0.0041	26.0174	0.0025	28.3460
	Ours	**0.0019**	**29.5765**	**0.0012**	**31.7907**
Tower	DN	0.0066	23.7911	0.0120	20.9451
	Ours	**0.0027**	**27.9968**	**0.0037**	**26.4454**

Table 1. **Quantitative benchmark of Fig. 3**: Compared to deforming NeRF (*DN in the table), our method consistently demonstrates better performance – lower MSE values and higher PSNR scores across all cases.

they actually are. This is due to the limited capacity of linear deformation models to accurately represent bending and shearing. When simulating codimensional shapes like rods and shells, linear GMLS tends to yield locking artifacts. As shown in Fig. 2, linear GMLS cannot generate correct bending behavior when the cloth collides with the sphere. The cloth acts like a stiff plane.

D. More results

Fig. 1 reports three additional tests of PIE-NeRF including collision, thin-shell elasticity, and stiff materials. The resulting novel motions of the scene from different views are shown in the figure. In the first example, the cars move under gravity and an initial velocity. They then collide with each other. In the second case, we fix the top corners of a piece of NeRF cloth, which drops on a wooden ball under gravity. In this example, the quadratic interpolation used in Q-GMLS effectively captures the nonlinear cloth dynamics while locking will occur if one chooses to use linear MLS with the same number of simulation DOFs. In the last case, we apply forces to the shovel of the Lego excavator, which has a relatively stiffer material ($5\times$ stiffer than other examples), resulting in interesting and novel dynamics.

We benchmark our method against Deforming-NeRF[2] for shear and squash deformation. Given a known deformation gradient field $\mathbf{u}(\mathbf{x})$, the ground truth is generated by directly applying it to original mesh vertex positions and rendered via BlenderNeRF [1]. Deforming-NeRF uses $\mathbf{u}$ to modify its cage mesh to drive the object's deformation. In our method, we apply $\mathbf{u}$ on IPs and render the deformed scene using quadratic ray warping. The qualitative results are shown in Fig. 3. The quantitative results are listed in Table 1.

References

[1] Maxime Raafat. BlenderNeRF, May 2023. 4

[2] Tianhan Xu and Tatsuya Harada. Deforming radiance fields with cages. In *European Conference on Computer Vision*, pages 159–175. Springer, 2022. 4