Supplement to Complex Wrinkle Field Evolution

ZHEN CHEN, The University of Texas at Austin
DANNY M. KAUFMAN, Adobe
MÉLINA SKOURAS, Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK
ETIENNE VOUGA, The University of Texas at Austin

CCS Concepts: • **Computing methodologies** → **Mesh geometry models**;

Additional Key Words and Phrases: discrete differential geometry, complex representation, wrinkle fields, singularities

ACM Reference format:
Zhen Chen, Danny M. Kaufman, Mélina Skouras, and Etienne Vouga. 2023. Supplement to Complex Wrinkle Field Evolution. *ACM Trans. Graph.* 1, 1, Article 1 (January 2023), 19 pages.
https://doi.org/10.1145/3592397

1 GEODESIC DISCUSSION

In this section, we derive the geodesic formula given in the main text.

1.1 Distance Energy Derivation

We start with the full derivation of the distance-measure energy term. To study the wrinkle evolution controlled by *CWF*, we assume that the underlying base surface and the corresponding rest shape do not change over time. Therefore

$$\dot{\epsilon} = -\frac{\mathrm{d}(\Re(z))}{\mathrm{d}t}F - \Re(z)\frac{\mathrm{d}F}{\mathrm{d}t}, \tag{1}$$

$$F := \boldsymbol{I}^{-1}(\mathrm{d}\arg \boldsymbol{z})^T(\mathrm{d}\arg \boldsymbol{z}). \tag{2}$$

Letting $\omega = \mathrm{d}\arg \boldsymbol{z}$ and $\boldsymbol{z} = a\tilde{\boldsymbol{z}} = a(\cos\theta + i\sin\theta)$, we have

$$\begin{aligned}\|\dot{\epsilon}\|_{\mathrm{SV}}^2 = &\left((\dot{a})^2\cos^2\theta + (a\dot{\theta})^2\sin^2\theta - a\dot{a}\dot{\theta}\sin 2\theta\right)\|F\|_{\mathrm{SV}}^2 \\ &+ a^2\cos^2\theta\|\dot{F}\|_{\mathrm{SV}}^2 + 2a\cos\theta(\dot{a}\cos\theta - a\dot{\theta}\sin\theta)(F:\dot{F}),\end{aligned} \tag{3}$$

where

$$\|M\|_{\mathrm{SV}}^2 = \frac{\alpha}{2}\mathrm{Tr}^2(M) + \beta\mathrm{Tr}(M^2), \tag{4}$$

with α and β the Lamé parameters, and

$$A : B := \frac{\alpha}{2}\mathrm{Tr}(A)\mathrm{Tr}(B) + \beta\mathrm{Tr}(AB) = B : A. \tag{5}$$

Following the coarse-graining idea of Aharoni et al. [2017], the trigonometric terms can be eliminated by averaging with the conjugate correspondence:

$$\begin{aligned}\|\dot{\bar{\epsilon}}\|_{\mathrm{SV}}^2 = &\left((\dot{a})^2\sin^2\theta + (a\dot{\theta})^2\cos^2\theta + a\dot{a}\dot{\theta}\sin 2\theta\right)\|F\|_{\mathrm{SV}}^2 \\ &+ a^2\sin^2\theta\|\dot{F}\|_{\mathrm{SV}}^2 + 2a\sin\theta(\dot{a}\sin\theta + a\dot{\theta}\cos\theta)(F:\dot{F}).\end{aligned} \tag{6}$$

Therefore,

$$\begin{aligned}\int_{\mathcal{M}}\|\dot{\epsilon}\|_{\mathrm{SV}}^2\,\mathrm{d}A &\approx \int_{\mathcal{M}}\frac{1}{2}\left(\|\dot{\epsilon}\|_{\mathrm{SV}}^2 + \|\dot{\bar{\epsilon}}\|_{\mathrm{SV}}^2\right)\,\mathrm{d}A \\ &= \int_{\mathcal{M}}\frac{(\dot{a})^2 + a^2\left|\dot{\tilde{\boldsymbol{z}}}\right|^2}{2}\|F\|_{\mathrm{SV}}^2 + \frac{a^2}{2}\|\dot{F}\|_{\mathrm{SV}}^2 + a\dot{a}(F:\dot{F})\,\mathrm{d}A \\ &= \frac{1}{2}\int_{\mathcal{M}}\left|\dot{\tilde{\boldsymbol{z}}}\right|^2\|F_{a,\omega}\|_{\mathrm{SV}}^2 + \|\dot{F}_{a,\omega}\|_{\mathrm{SV}}^2\,\mathrm{d}A,\end{aligned} \tag{7}$$

where we use

$$\left|\dot{\tilde{\boldsymbol{z}}}\right|^2 = \left|(\exp i\theta)\,\dot{\theta}\right| = |\dot{\theta}| \tag{8}$$

$$F_{a,\omega} = a\boldsymbol{I}^{-1}\omega^T\omega. \tag{9}$$

Thus the distance on paths $\gamma(t)$, which connects two endpoints $\boldsymbol{z}^0$ and $\boldsymbol{z}^1$ is given by:

$$d(\gamma) = \frac{1}{2}\int_0^1\int_{\mathcal{M}}\left|\dot{\tilde{\boldsymbol{z}}}\right|^2\|F_{a,\omega}\|_{\mathrm{SV}}^2 + \|\dot{F}_{a,\omega}\|_{\mathrm{SV}}^2\,\mathrm{d}A\,\mathrm{d}t, \tag{10}$$

with $\boldsymbol{z} = a\tilde{\boldsymbol{z}}$ and $\mathrm{d}\arg \boldsymbol{z} = \omega$, or equivalently $(\mathrm{d} - i\omega)\tilde{\boldsymbol{z}} = 0$..

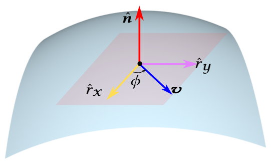
1.2 Geodesics Computation

The geodesic γ^* between two endpoints $\boldsymbol{z}^0$ and $\boldsymbol{z}^1$ can be found by solving the following variational problem:

$$\begin{aligned}\min_{\boldsymbol{z}}\ &\frac{1}{2}\int_0^1\int_{\mathcal{M}}\left|\dot{\tilde{\boldsymbol{z}}}\right|^2\|F_{a,\omega}\|_{\mathrm{SV}}^2 + \|\dot{F}_{a,\omega}\|_{\mathrm{SV}}^2\,\mathrm{d}A\,\mathrm{d}t \\ \text{s.t.}\quad &(\mathrm{d} - i\omega)\tilde{\boldsymbol{z}} = 0,\ \boldsymbol{z} = a\tilde{\boldsymbol{z}},\ |\tilde{\boldsymbol{z}}| = 1 \\ &\boldsymbol{z}(0) = \boldsymbol{z}^0,\ \boldsymbol{z}(1) = \boldsymbol{z}^1.\end{aligned} \tag{11}$$

To solve Equation (11), we write $u = \sqrt{a}\omega$, and $f = \|u\|_{\boldsymbol{I}^{-1}}^2$, so that we have

$$\begin{aligned}\|F_{a,\omega}\|_{\mathrm{SV}}^2 &= (\frac{\alpha}{2} + \beta)f^2 \\ \|\dot{F}_{a,\omega}\|_{\mathrm{SV}}^2 &= 2(\alpha + \beta)\left(u\boldsymbol{I}^{-1}\dot{u}^T\right)^2 + 2\beta f\|\dot{u}\|_{\boldsymbol{I}^{-1}}^2.\end{aligned} \tag{12}$$



Consider the tangent vector $\boldsymbol{v}$ corresponding to the one-form u, where $\boldsymbol{v} = \mathrm{d}r\boldsymbol{I}^{-1}u^T$.This can be expressed as the rotation of the **unit** basis tangent vector $\hat{r}_x$ w.r.t **unit** base surface normal $\hat{\boldsymbol{n}}$, with rotation angle $\phi(t)$, and rescaling by its norm:

$$\boldsymbol{v} = \|\boldsymbol{v}\|R(\hat{\boldsymbol{n}}, \phi(t))\hat{r}_x = \sqrt{f}R(\hat{\boldsymbol{n}}, \phi(t))\hat{r}_x, \tag{13}$$

0730-0301/2023/1-ART1
https://doi.org/10.1145/3592397

where $\|\boldsymbol{v}\|^2 = \boldsymbol{v}^T\boldsymbol{v} = u\boldsymbol{I}^{-1}u^T = \|u\|^2_{\boldsymbol{I}^{-1}} = f$ and $R(\hat{\boldsymbol{n}}, \phi(t))$ is the rotation matrix with axis $\hat{\boldsymbol{n}}$ and angle ϕ. This expression gives us

$$\begin{aligned} u\boldsymbol{I}^{-1}\dot{u}^T &= \frac{1}{2}\dot{f} \\ \|\dot{u}\|^2_{\boldsymbol{I}^{-1}} &= \frac{1}{4f}\dot{f}^2 + f\dot{\phi}^2\hat{r}_x^T\left(\frac{\partial R}{\partial \phi}\right)^T\frac{\partial R}{\partial \phi}\hat{r}_x. \end{aligned} \tag{14}$$

Given that $\|\hat{\boldsymbol{n}}\| = \|\hat{r}_x\| = 1$, $\hat{\boldsymbol{n}} \cdot \hat{r}_x = 0$, and $\hat{\boldsymbol{n}}$ is the rotation axis, it is easy to check that

$$\hat{r}_x^T\left(\frac{\partial R}{\partial \phi}\right)^T\frac{\partial R}{\partial \phi}\hat{r}_x = 1. \tag{15}$$

Therefore we have

$$\|\dot{F}_{a,\omega}\|^2_{\mathrm{SV}} = (\frac{\alpha}{2} + \beta)\dot{f}^2 + 2\beta f^2\dot{\phi}^2. \tag{16}$$

The optimization (11) becomes:

$$\begin{aligned} &\underset{f,\phi,\tilde{z},a}{\arg\min}\frac{1}{2}\int_0^1\int_{\mathcal{M}}(\frac{\alpha}{2} + \beta)(f^2\left|\dot{\tilde{z}}\right|^2 + \dot{f}^2) + 2\beta f^2\dot{\phi}^2\ \mathrm{d}A\mathrm{d}t \\ &\text{s.t.}\quad \omega(t) = \boldsymbol{u}(t)/\sqrt{a(t)} = \sqrt{f(t)/a(t)}\hat{r}_x^T\left[R(\hat{n}, \phi(t))\right]^T\mathrm{d}r \\ &\qquad\ \phi(0) = \phi^0,\ \phi(1) = \phi^1,\ \tilde{z}(0) = \tilde{z}^0,\ \tilde{z}(1) = \tilde{z}^1 \\ &\qquad\ f(0) = f^0,\ f(1) = f^1,\ a(0) = a^0,\ a(1) = a^1 \\ &\qquad\ [\mathrm{d} - i\omega(t)]\,\tilde{z}(t) = 0,\ |\tilde{z}(t)| = 1. \end{aligned} \tag{17}$$

Similarly to what we discussed in the main text, we replace the last two constraints by $\tilde{z} \in \mathrm{Opt}_{\omega}$, leading to the following well-defined optimization problem:

$$\begin{aligned} &\underset{f,\phi,\tilde{z},a}{\arg\min}\frac{1}{2}\int_0^1\int_{\mathcal{M}}(\frac{\alpha}{2} + \beta)(f^2\left|\dot{\tilde{z}}\right|^2 + \dot{f}^2) + 2\beta f^2\dot{\phi}^2\ \mathrm{d}A\ \mathrm{d}t \\ &\quad\text{s.t.}\quad \tilde{z} \in \mathrm{Opt}_{\omega}, \\ &\mathrm{Opt}_{\omega} := \underset{\tilde{z}}{\arg\min}\int_{\mathcal{M}}\left[\left\|(\mathrm{d} - i\omega(t))\,\tilde{z}\right\|^2 + \left(|\tilde{z}|^2 - 1\right)^2\right]\mathrm{d}A, \end{aligned} \tag{18}$$

with the same boundary conditions.

We solve it by a penalty method as:

$$\begin{aligned} &\underset{f,\phi,\tilde{z},a}{\arg\min}\ \frac{1}{2}\int_0^1\int_{\mathcal{M}}(\frac{\alpha}{2} + \beta)(f^2\left|\dot{\tilde{z}}\right|^2 + \dot{f}^2) + 2\beta f^2\dot{\phi}^2\ \mathrm{d}A\ \mathrm{d}t \\ &\qquad + c_1\int_0^1\int_{\mathcal{M}}\left\|(\mathrm{d} - i\omega)\,\tilde{z}\right\|^2 + \left(|\tilde{z}|^2 - 1\right)^2\ \mathrm{d}A\ \mathrm{d}t, \end{aligned} \tag{19}$$

with boundary conditions:

$$\phi(0) = \phi^0,\ \phi(1) = \phi^1,\ \theta(0) = \theta^0,\ \theta(1) = \theta^1 \tag{20}$$

$$f(0) = f^0,\ f(1) = f^1,\ a(0) = a^0,\ a(1) = a^1, \tag{21}$$

$$\tag{22}$$

and

$$\omega = \sqrt{f(t)/a(t)}\hat{r}_x^T\left[R(\hat{n}, \phi(t))\right]^T\mathrm{d}r. \tag{23}$$

We further approximate this problem by replacing f^2 by its average over time:

$$\begin{aligned} &\underset{f,\phi,\tilde{z},a}{\arg\min}\frac{1}{2}\int_{\mathcal{M}}\left(\frac{(f^0)^2 + (f^1)^2}{2}\int_0^1\left(\frac{\alpha}{2} + \beta\right)\left|\dot{\tilde{z}}\right|^2 + 2\beta\dot{\phi}^2\mathrm{d}t\right)\mathrm{d}A \\ &\qquad + \left(\frac{\alpha}{4} + \frac{\beta}{2}\right)\int_{\mathcal{M}}\int_0^1\dot{f}^2\ \mathrm{d}t\ \mathrm{d}A \\ &\qquad + c_1\int_{\mathcal{M}}\int_0^1\left\|(\mathrm{d} - i\omega)\,\tilde{z}\right\|^2 + \left(|\tilde{z}|^2 - 1\right)^2\ \mathrm{d}t\ \mathrm{d}A. \end{aligned} \tag{24}$$

This indicates that

$$f(t) = (1 - t)f^0 + tf^1.$$

The optimal ϕ should satisfy

$$\begin{aligned} \beta\ddot{\phi} &= -c_1\overline{(\mathrm{d} - i\omega)\tilde{z}}\left(i\tilde{z}\frac{\mathrm{d}\omega}{\mathrm{d}\phi}\right) \\ &= -c_1\overline{(\mathrm{d} - i\omega)\tilde{z}}\left(i\tilde{z}\omega^{\mathrm{perp}}\right), \end{aligned} \tag{25}$$

where

$$\omega^{\mathrm{perp}}(t) = \sqrt{f(t)/a(t)}\hat{r}_x^T\left[R\left(\hat{n}, \phi(t) + \frac{\pi}{2}\right)\right]^T\mathrm{d}r. \tag{26}$$

Away from the singularities, $\tilde{z}$ and ω are almost compatible, that is $(\mathrm{d} - i\omega)\tilde{z} \approx 0$. Therefore we have $\ddot{\phi} = 0$, which indicates

$$\phi(t) = (1 - t)\phi^0 + t\phi^1. \tag{27}$$

Near the singularities, $\tilde{z}$ is close to zero. In this case, the wrinkles displacement ($= a\Re(\tilde{z})\hat{\boldsymbol{n}} \approx 0$) is negligible. The value of ϕ does not matter. Therefore, we can extend Equation (27) for the whole surface. We further assume that amplitude varies slowly temporally, so that it can be approximated using the linear relation,

$$a(t) = (1 - t)a^0 + ta^1. \tag{28}$$

We can get the final remaining $\tilde{z}$ by solving the optimization problem:

$$\begin{aligned} &\underset{\tilde{z}}{\arg\min}\int_{\mathcal{M}}\left(g_{\mathrm{bd}}\int_0^1\left|\dot{\tilde{z}}\right|^2\mathrm{d}t\right)\mathrm{d}A \\ &\qquad + c\int_{\mathcal{M}}\int_0^1\left\|(\mathrm{d} - i\omega(t))\,\tilde{z}\right\|^2 + \left(|\tilde{z}|^2 - 1\right)^2\ \mathrm{d}t\mathrm{d}A, \end{aligned} \tag{29}$$

where

$$c = c_1/\left(\frac{\alpha}{4} + \frac{\beta}{2}\right),\ g_{\mathrm{bd}} = \frac{(f^0)^2 + (f^1)^2}{2} \tag{30}$$

To sum up, in the smooth case, the approximated geodesic can be computed as follows:

$$a(t) = (1 - t)a^0 + ta^1 \tag{31}$$

$$f(t) = (1 - t)f^0 + tf^1 = (1 - t)a^0\|\omega^0\|^2_{\boldsymbol{I}^{-1}} + ta^1\|\omega^1\|^2_{\boldsymbol{I}^{-1}} \tag{32}$$

$$\phi(t) = (1 - t)\phi^0 + t\phi^1 \tag{33}$$

$$\omega(t) = \sqrt{f(t)/a(t)}\hat{r}_x^T\left[R(\hat{n}, \phi(t))\right]^T\mathrm{d}r, \tag{34}$$

and $\tilde{z}$ is the optimal solution for Equation (29) with constant penalty coefficient c. For the regions with singularities, where $a^0|_{\boldsymbol{P}} = 0$, or $a^1|_{\boldsymbol{P}} = 0$, $\phi(t)$ is not well defined. In that case, we simply linearly interpolate $\omega(t)$ as $(1 - t)\omega^0 + t\omega^1$.

1.3 Discrete Geodesics

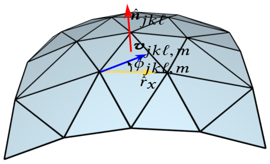
In the previous section, we provide the formula to compute the smooth geodesic between two complex fields z^0 and z^1. In this section, we provide the corresponding discretization w.r.t. two *CWFs*: $(\omega^0, a^0, \tilde{z}^0)$ and $(\omega^1, a^1, \tilde{z}^1)$, where $\omega^{0,1}$ are the edge-discretized one-forms, while $a^{0,1}$ and $\tilde{z}^{0,1}$ are defined per-vertex.

We start with some notation. Let N_j be the set of all triangle faces incident to vertex $\boldsymbol{V}_j$, and N_{jk} the faces incident to edge $\boldsymbol{E}_{jk}$. Notice that Equation (31) can be directly discretized for per-vertex amplitude as

$$a_j^t = (1-t)a_j^0 + ta_j^1. \tag{35}$$

What remains is to discuss the discretization of Equation (34) for ω, and the corresponding optimization (29) for $\tilde{z}$.

Discrete f. We first discuss the discretization of f. From Equation (13), we know that in the smooth setting, f represents the squared magnitude of the tangential vector $\boldsymbol{v} = \sqrt{a}\mathrm{d}\boldsymbol{r}\boldsymbol{I}^{-1}\omega$. We discretize it as a scalar per face-vertex; that is, for each face $\boldsymbol{F}_{jk\ell}$ and each vertex $\boldsymbol{V}_m, m \in \{j,k,\ell\}$, we assign an $f_{jk\ell,m}$. More specifically, for each vertex $\boldsymbol{V}_m, m \in \{j,k,\ell\}$, we discretize $\boldsymbol{v}$ as follows:



$$\boldsymbol{v}_{jk\ell,m} = \sqrt{a_m}\boldsymbol{u}_{jk\ell,m} \tag{36}$$

where $\boldsymbol{u}_{jk\ell,m}$ can be computed according to Equation (51) in § 3. Then we set

$$f_{jk\ell,m} = \|\boldsymbol{v}_{jk\ell,m}\|^2 = a_m \left\|\boldsymbol{u}_{jk\ell,m}\right\|^2. \tag{37}$$

Since a_m^0, a^1, $\boldsymbol{u}_{jk\ell,m}^0$, and $\boldsymbol{u}_{jk\ell,m}^1$ are given by the boundary conditions, we conclude that

$$f_{jk\ell,m}^t = (1-t)a_m^0 \left\|\boldsymbol{u}_{jk\ell,m}^0\right\|^2 + ta_m^1 \left\|\boldsymbol{u}_{jk\ell,m}^1\right\|^2, \tag{38}$$

and that for each vertex $\boldsymbol{V}_j$,

$$(g_{\mathrm{bd}})_j = \left(\frac{(f^0)^2 + (f^1)^2}{2}\right)_j \tag{39}$$

$$= \frac{1}{2|\mathrm{N}_j|} \sum_{F_{jk\ell} \in \mathrm{N}_j} \left(a_j^0 \left\|\boldsymbol{u}_{jk\ell,j}^0\right\|^2\right)^2 + \left(a_j^1 \left\|\boldsymbol{u}_{jk\ell,j}^1\right\|^2\right)^2. \tag{40}$$

Discrete ω formula. Our next step is to discretize Equation (34). Consider an edge $\boldsymbol{E}_{jk}$. Consider one of its incident faces $F_{jk\ell}$ and one of its endpoints $\boldsymbol{V}_m, m \in \{j,k\}$. We can get the corresponding a_m^t and $f_{jk\ell,m}^t$ according to Equations (35) and (38) respectively. We choose

$$r_x = e_{jk} := \boldsymbol{V}_k - \boldsymbol{V}_j \tag{41}$$

$$\hat{r}_x = r_x/\|r_x\|, \tag{42}$$

and $\phi_{jk\ell,m}$ is the corresponding angle between $\hat{r}_x$ and $\boldsymbol{v}_{jk\ell,m}$, we have

$$\phi_{jk\ell,m}^t = (1-t)\phi_{jk\ell,m}^0 + t\phi_{jk\ell,m}^1. \tag{43}$$

The one-form contribution from vertex $\boldsymbol{V}_m$ on face $F_{jk\ell}$, is then computed as

$$\omega_{jk\ell,m}^t = \sqrt{f_{jk\ell,m}^t/a_m^t}\,\hat{e}_{jk}^T \left[R(\hat{\boldsymbol{n}}_{jk\ell}, \phi_{jk\ell,m}^t)\right]^T e_{jk} \tag{44}$$

$$= \|e_{jk}\|\sqrt{f_{jk\ell,m}^t/a_m^t}\cos\left(\phi_{jk\ell,m}^t\right). \tag{45}$$

Therefore

$$\omega_{jk}^t = \frac{\|\boldsymbol{V}_k - \boldsymbol{V}_j\|}{2|\mathrm{N}_{jk}|} \sum_{F_{jk\ell} \in \mathrm{N}_{jk}} \sum_{m \in \{j,k\}} \sqrt{f_{jk\ell,m}^t/a_m^t}\cos\left(\phi_{jk\ell,m}^t\right). \tag{46}$$

Discrete $\tilde{z}$ formula. The final step is to discretize Equation (29). In the main text, we have already provided the discretization of $|(\mathrm{d} - i\omega)\tilde{z}|$ and $\left(|\tilde{z}|^2 - 1\right)^2$. Additionally, Equation (40) offers the discretization of g_{bd}. The only remaining part of Equation (29) is $\dot{\tilde{z}}$, which can be directly discretized using standard finite difference:

$$\dot{\tilde{z}}_j^t = \frac{\tilde{z}_j^{t+\delta t} - \tilde{z}_j^t}{\delta t}. \tag{47}$$

Optimizing Equation (29) with respect to $\left\{\tilde{z}_j^{n\cdot\delta t} = x_j^{n\cdot\delta t} + iy^{n\cdot\delta t}{}_j\right\}_j^n$ yields the final optimal $\tilde{z}$.

2 LOOP SUBDIVISION RULES FOR 1-FORMS

We modified the Loop scheme provided in [de Goes et al. 2016; Wang et al. 2006], such that the corresponding mask for the one-form is still consistent with the boundary fixed Loop scheme. We listed the changes to the boundary rules in Figure 1. For the interior rules, we refer the reader to the supplementary material provided by de Goes et al. [2016].

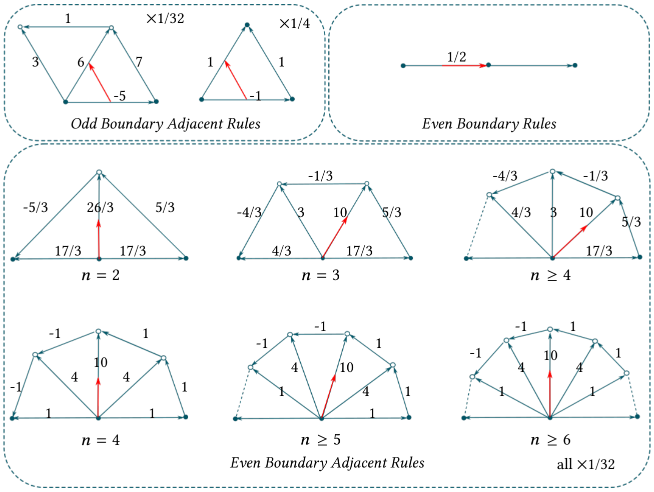


Fig. 1. Loop subdivision rules for boundary 1-forms

3 CONTANGENT VECTOR COMPUTATION

In this section, we provide the formula of acting the cotangent vector ω at a point $\boldsymbol{P}$ (in the tangent space) on a tangent direction d. Setting

$\boldsymbol{P} = \boldsymbol{P}_0$ and $d = \boldsymbol{P}_1^* - \boldsymbol{P}_0$ will give us the corresponding formual mentioned in the main text.

To begin with, consider a face $\boldsymbol{F}$ with vertices $\boldsymbol{V}_0, \boldsymbol{V}_1$ and $\boldsymbol{V}_2$, and edge 1-forms ω_{01}, ω_{12}, and ω_{20}. The barycentric coordinates of $\boldsymbol{P}$ is $(\alpha_0, \alpha_1, \alpha_2)$:

$$\boldsymbol{P} = \alpha_0 \boldsymbol{V}_0 + \alpha_1 \boldsymbol{V}_1 + \alpha_2 \boldsymbol{V}_2 \tag{48}$$

For any tangent direction d,

$$\omega(\boldsymbol{P})(d) = \boldsymbol{v}_{\tan}(\boldsymbol{P}) \cdot d \approx \sum_{i=0}^{2} \alpha_i (\boldsymbol{v}_i \cdot d) \tag{49}$$

where $\boldsymbol{v}_{\tan}$ is the corresponding tangent gradient of ω. Here we approximate $\boldsymbol{v}_{\tan}$ by linearly blending from the corresponding values at triangle corners (denoted as $\boldsymbol{v}_i$).

The tangential gradient $\boldsymbol{v}_0$ of vertex $\boldsymbol{V}_0$ should satisfy

$$\begin{aligned} \boldsymbol{v}_0 \cdot (\boldsymbol{V}_1 - \boldsymbol{V}_0) &= \omega_{01} \\ \boldsymbol{v}_0 \cdot (\boldsymbol{V}_2 - \boldsymbol{V}_0) &= -\omega_{20} \end{aligned} \tag{50}$$

Expressing $\boldsymbol{v}_0$ in the tangential plane basis gives us:

$$\boldsymbol{v}_0 = x(\boldsymbol{V}_1 - \boldsymbol{V}_0) + y(\boldsymbol{V}_2 - \boldsymbol{V}_0), \tag{51}$$

where

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \|\boldsymbol{V}_1 - \boldsymbol{V}_0\|^2 & (\boldsymbol{V}_1 - \boldsymbol{V}_0) \cdot (\boldsymbol{V}_2 - \boldsymbol{V}_0) \\ (\boldsymbol{V}_1 - \boldsymbol{V}_0) \cdot (\boldsymbol{V}_2 - \boldsymbol{V}_0) & \|\boldsymbol{V}_2 - \boldsymbol{V}_0\|^2 \end{bmatrix}^{-1} \begin{bmatrix} \omega_{01} \\ -\omega_{20} \end{bmatrix} \tag{52}$$

$\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ can be computed in a similar manner.

4 TIMING

We report the timing of our interpolation approach in Table 1. We ran our experiments on a desktop with a 8-core Intel Core i9-9900K CPU, clocked at 3.6 GHz and 64 GB of memory. We use 50 frames in total and linear interpolants inbetween to get a final set of 200 frames used for animation generation. To get optimized results, we let the Newton solver run until the gradient L2-norm is smaller than 10^{-6}.

5 ADDITIONAL EDITING EXAMPLES

We used our wrinkle editing tools to create several additional examples of smooth interpolation of wrinkles on complex surfaces. Please see the supplemental video for animations of these examples. Figure 2 shows the result of locally rotating a small user-specified patch of the circular wave on the cylinder by ninety degrees. Figure 3 shows interpolation between two wrinkle patterns on a hand, where the second keyframe was created from the first by rotating the left red patch 45 degrees and halving the wrinkle frequency and doubling the amplitude in the right red patch. In Figure 4, we generate the target frame by rotating the wrinkles on the cheek (the left red patch) by 90 degrees, increasing the wrinkle frequency by 1.5 times in the middle red patch beneath the mouth, and doubling the frequency on the bottom red patch. Figure 5 shows an interpolation of two user-designed waves on Spot, where different red patches have different rotation angles. Figure 6 shows the possibility to design and interpolate wrinkles on cloth. All of these examples show complicated wrinkle shapes with many singularities (the blue dots in the amplitude figures). We successfully get *local* smooth wrinkle evolution on all of these user-designed examples.

Editing Simulation Results. Our tools can be used to edit wrinkles originally computed via physical simulation. We show two examples: Chen et al. [2021]'s dress and pants. For the dress (Figure 7), we set the second keyframe by uniformly enlarging the wrinkle frequency and at the same time halving the wrinkle amplitude to mimic a change in cloth thickness. For the pants (Figure 8), we change the frequency on the two legs differently, by enlarging the frequency in the middle of the left leg and shrinking the frequency in the middle of the right leg. Again we changed the amplitude to keep $\|a \cdot \omega\|$ constant. We find that our approach mimic wrinkle motion reasonably well.

6 EXTRA COMPARISON EXAMPLES

Figures 9 and 10 show extra comparisons between our approach and alternative keyframe interpolation methods.

7 ABLATION STUDY

In this section we study different choices for the number of guide frames N' (with $N = 200$ fixed), and solver penalty c.

7.1 Number of Guide Frames

In this section, we show the results of different choices of number of guide frames N', where we fix the penalty coefficient to $10^3 g_{\text{ave}}$. In Figures 11, 12, 13, we show the results of wrinkles, amplitude and phase patterns respectively for different number of frames, where $N' = 1$ implies that we directly linearly interpolate keyframes. We find that as we increase the number of frames, the final interpolated results turn to converge. In particular, the differences between $N' = 25, 50, 100$ and 200 are subtle, especially for the last three. (You can see subtle differences between $N' = 25$ and the others at $t = 0.5$ and $t = 0.875$.)

7.2 Penalty Coefficient

As we discussed in the main text, when actually solving for the geodesic between two *CWF*s, we use a penalty $c = 10^3 g_{\text{ave}}$ to control the balance between temporal coherence and compatibility, where the later affects the semantic meaning of the final upsampled wrinkled surface. In Figures 14, 15, we show different choices of c with 50 frames. If c is too small, for example $c = 10 g_{\text{ave}}$, the kinetic term dominates the optimization, and the end result has bad compatibility. The corresponding final wrinkled surface has lots of unexpected singularities, with undesired aliased visual artifacts. On the other hand, if c is too large, for example $c = 10^5 g_{\text{ave}}$, the compatibility term and unit term dominate the optimization, and temporal incoherence will not be penalized. This leads to spatially appealing, but temporally incoherent, wrinkle sequences. For the values in between, the smaller the c, the more "local" patches of rotating regions we will have. At the same time, E_{compat} is a quadratic convex term, while E_{unit} is a quartic, non-convex term. Increasing c will increase the impact of these two terms. We found that it will slow down the optimization, then speed it up (see the first column in Figure 14). How to find the best choice for c for each example to achieve the best trade-off between animation quality (spatial and temporal) and solving speed needs further exploration, and we treat this as future work.

Table 1. The time and convergence information for all the examples shown in the main paper (marked as $*$) and this supplementary document. |V| and |F| are the number of vertices and faces of the base coarse mesh, and |V′| and |F′| are the corresponding values of the upsampled wrinkled mesh, where 4 or 5 levels of upsampling are used. The major part of time (~90%) is costed by the CHOLMOD linear solve within the Newton's method. Note that the efficiency can be improved by fine-tuning the number of frames we use. See § 7.1 for more discussion.

Models	Figures	|V|	|F|	|V′|	|F′|	#iterations	CHOLMOD time (s)	total time(s)
torus global rotation	10	400	800	400K	800K	243	123.30	127.13
cylinder local rotation	2	639	1222	246K	491K	129	30.61	31.92
bunny global rotation	10*	502	1000	512K	1.0M	381	254.53	263.09
bunny local rotation	7*	502	1000	512K	1.0M	127	89.36	92.31
fertility global enlargement	5*	494	1000	511K	1.0M	298	178.91	185.74
fertility local enlargement	6*	494	1000	511K	1.0M	94	63.52	65.69
phantasma global rotation	1*	850	1696	868K	1.7M	320	414.51	426.39
phantasma local enlargement	9	850	1696	868K	1.7M	109	172.01	176.05
spot composite editing	18*	502	1000	512K	1.0M	154	82.19	85.65
spot user designed editing	5	502	1000	512K	1.0M	212	123.21	127.92
user designed face	4	1026	1999	1.0M	2.0M	151	334.46	341.79
user designed hand	3	1012	2000	1.0M	2.0M	94	173.12	177.69
user designed dress	6	1333	2586	1.5M	3.0M	144	324.59	333.65
pants	8	1677	3304	1.6M	3.3M	262	710.14	728.63
dress	7	1525	2950	1.5M	3.0M	96	303.08	310.65
Average	-	820	1617	802K	1.6M	188	225.17	231.62

REFERENCES

Hillel Aharoni, Desislava Todorova, Octavio Albarran, Lucas Goehring, Randall Kamien, and Eleni Katifori. 2017. The smectic order of wrinkles. *Nature Communications* 8 (07 2017), 15809.

Zhen Chen, Hsiao-Yu Chen, Danny M. Kaufman, Mélina Skouras, and Etienne Vouga. 2021. Fine Wrinkling on Coarsely Meshed Thin Shells. *ACM Trans. Graph.* 40, 5, Article 190 (aug 2021), 32 pages.

Fernando de Goes, Mathieu Desbrun, Mark Meyer, and Tony DeRose. 2016. Subdivision Exterior Calculus for Geometry Processing. *ACM Trans. Graph.* 35, 4, Article 133 (jul 2016), 11 pages.

Felix Knöppel, Keenan Crane, Ulrich Pinkall, and Peter Schröder. 2015. Stripe Patterns on Surfaces. *ACM Trans. Graph.* 34, 4, Article 39 (jul 2015), 11 pages.

Ke Wang, Weiwei, Yiying Tong, Mathieu Desbrun, and Peter Schröder. 2006. Edge Subdivision Schemes and the Construction of Smooth Vector Fields. *ACM Trans. Graph.* 25, 3 (jul 2006), 1041–1048.

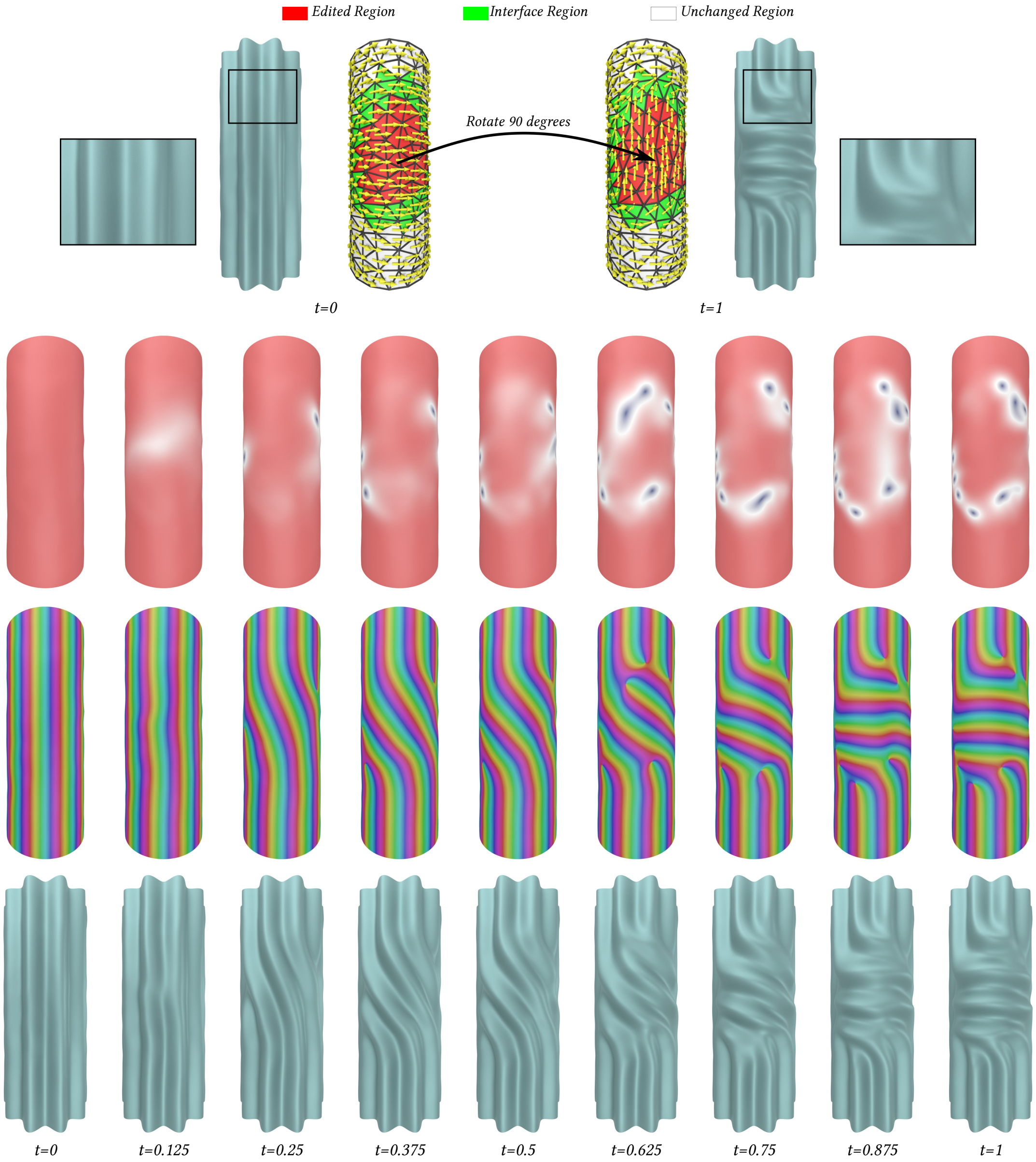


Fig. 2. Local rotation of an axial wrinkle pattern on a cylinder. The two keyframes are given in the left/right side; the wrinkle frequency has been rotated by ninety degrees in the red region of the second keyframe. The corresponding wrinkle animation is given in the **Interpolation Results** video in the supplementary materials near 00:20-00:37.

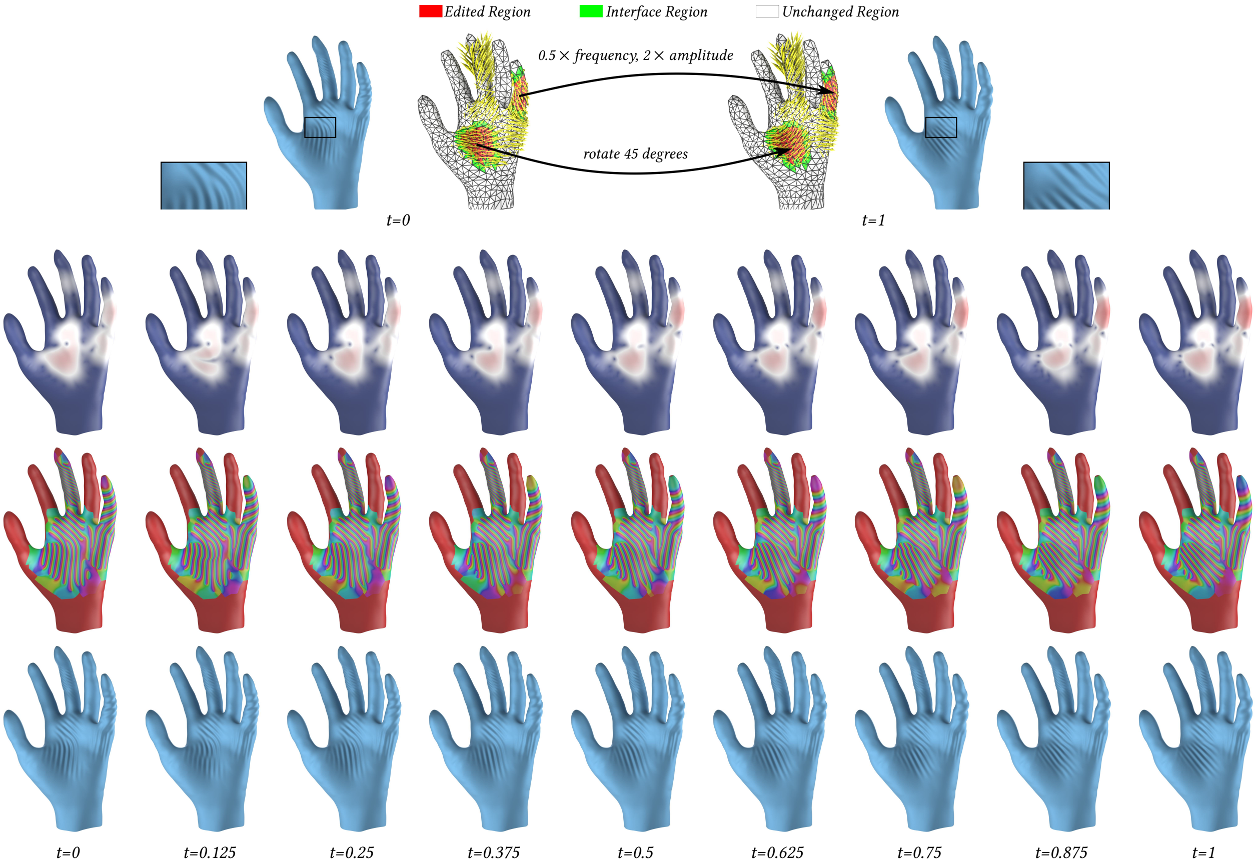


Fig. 3. *CWF* interpolation of two wrinkle patterns on the hand. Please check the **Interpolation Results** video in the supplementary materials for more details (timestamp 03:13-03:29).

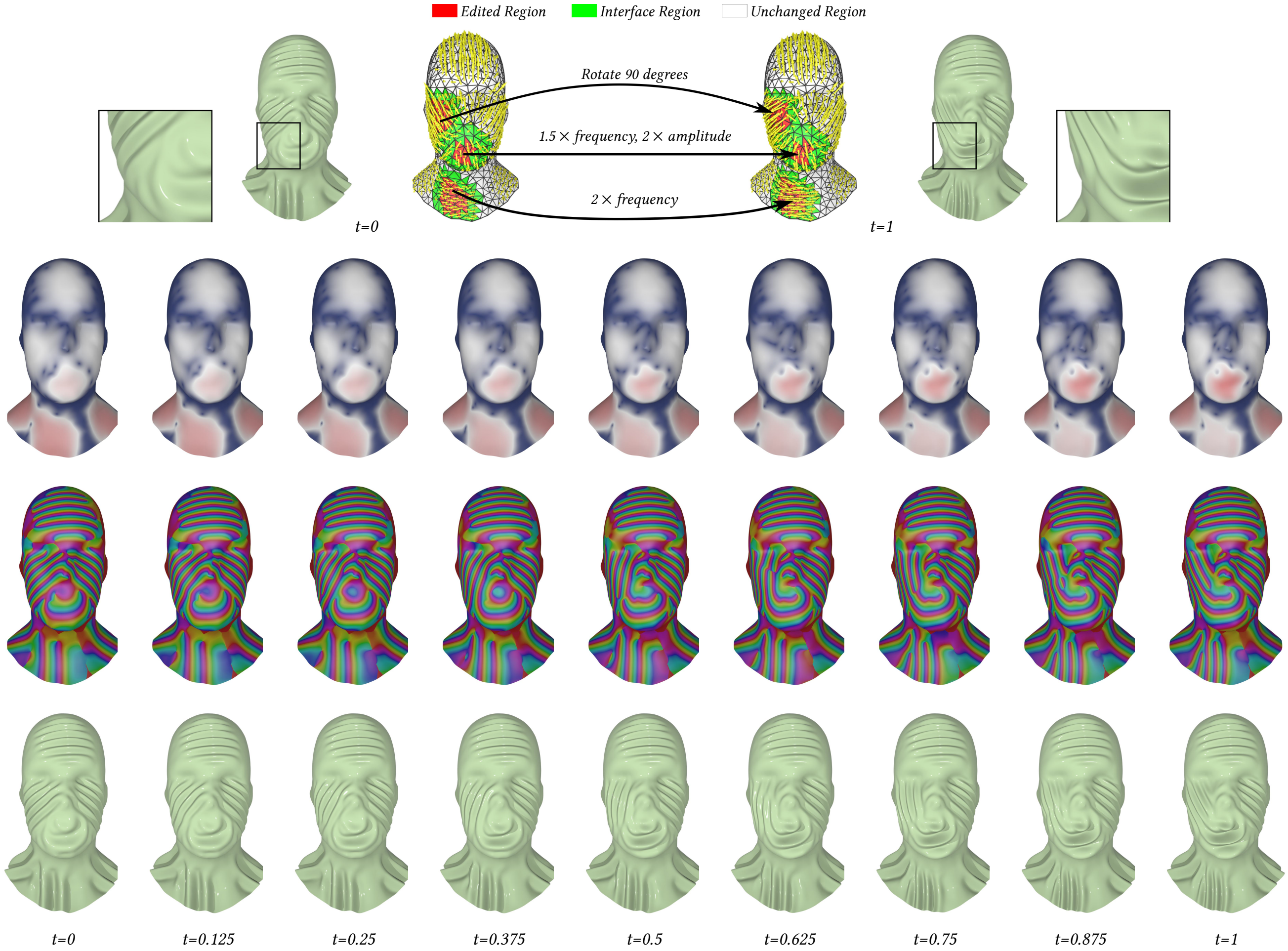


Fig. 4. *CWF* interpolation of two wrinkle patterns on the face. Please check the **Interpolation Results** video in the supplementary materials for more details (timestamp 02:56–03:12).

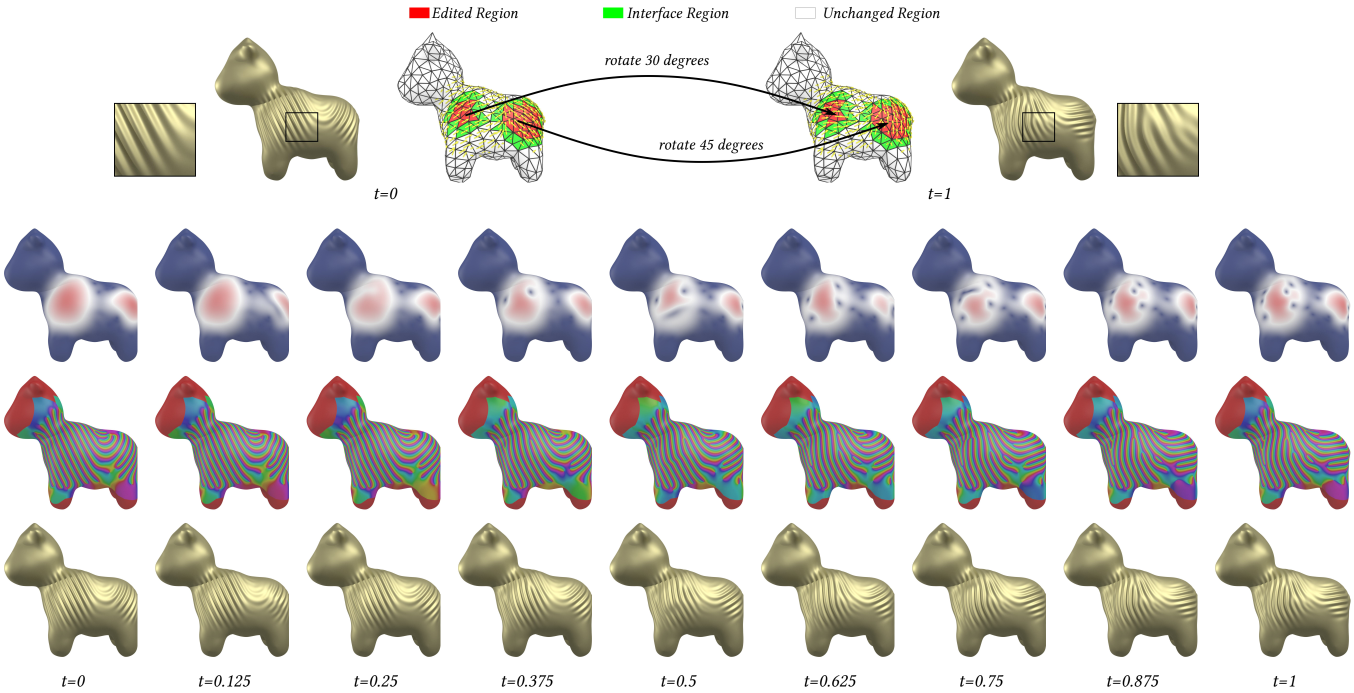


Fig. 5. *CWF* interpolation of two wrinkle patterns on Spot the cow. Please refer to the **Interpolation Results** video in the supplementary materials for the corresponding wrinkle animation (timestamp 02:38-02:55).

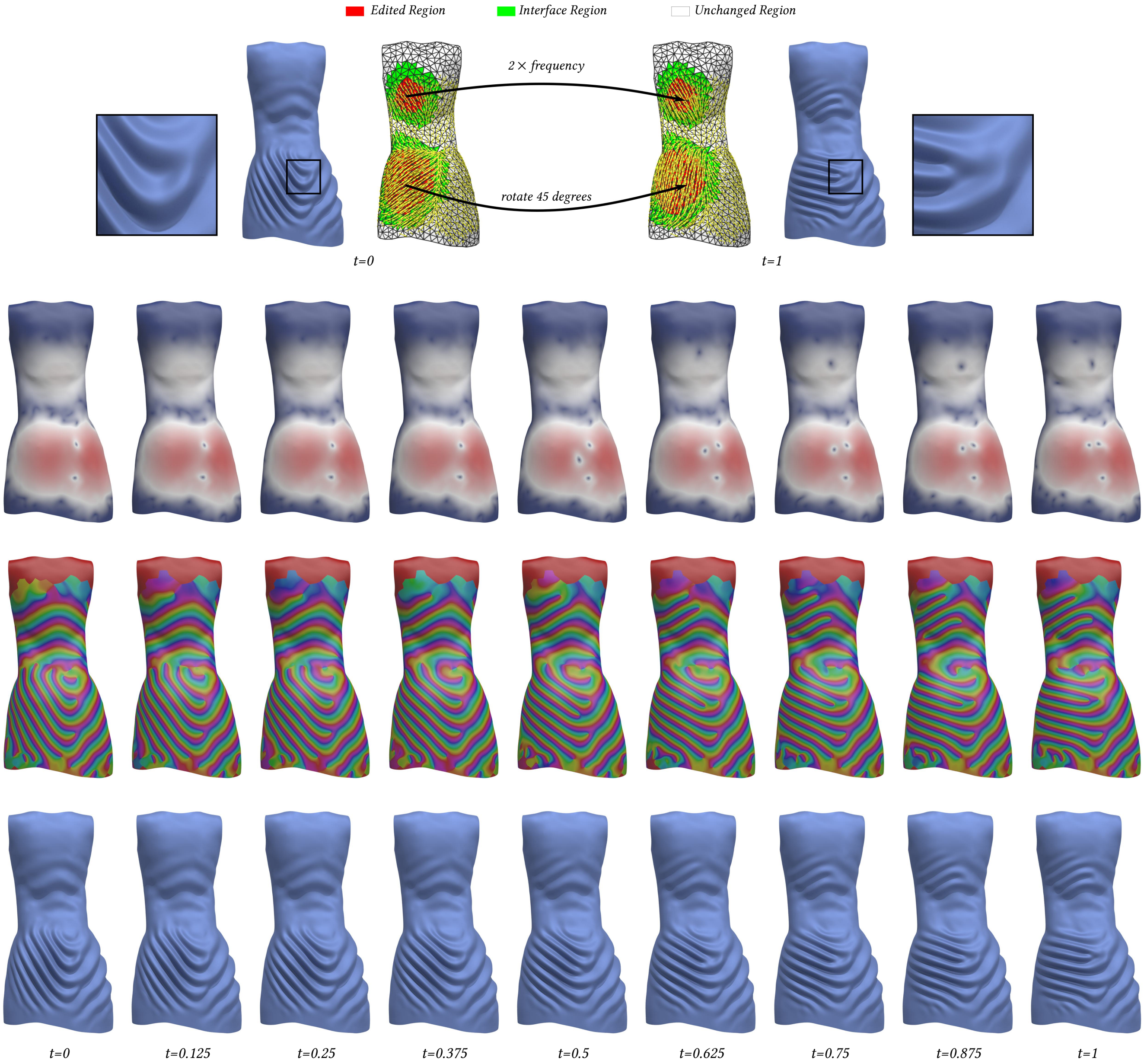


Fig. 6. *CWF* interpolation of two wrinkle patterns on the dress. One can easily see how the wrinkles and the singularities move in the **Interpolation Results** video of the supplementary materials (timestamp 03:30-03:46)

.

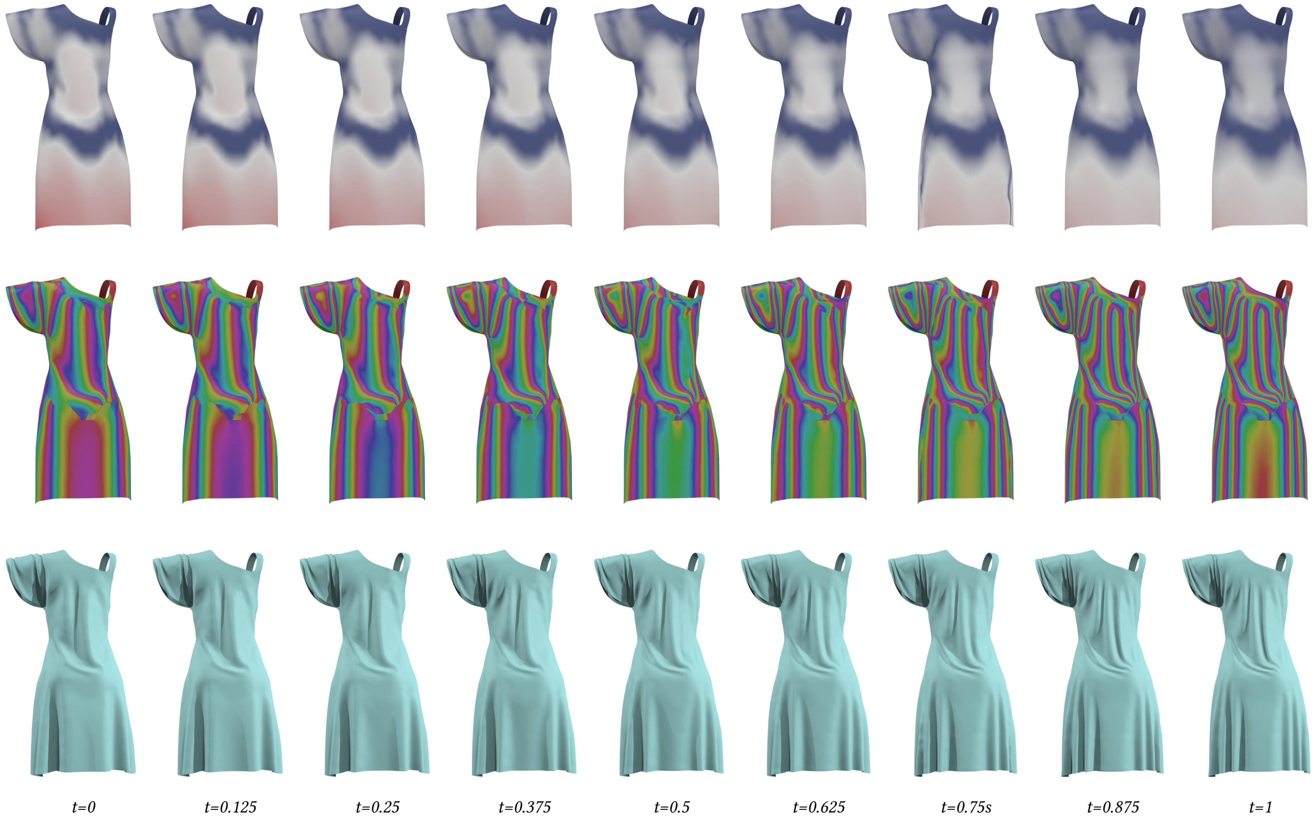


Fig. 7. We uniformly double the wrinkle frequency and halve the amplitude on the dress examples from Chen et al. [2021], emulating replacing the cloth with a thinner material. Please check the **Interpolation Results** video in the supplementary materials at 03:47-04:03 for the corresponding wrinkle animation.

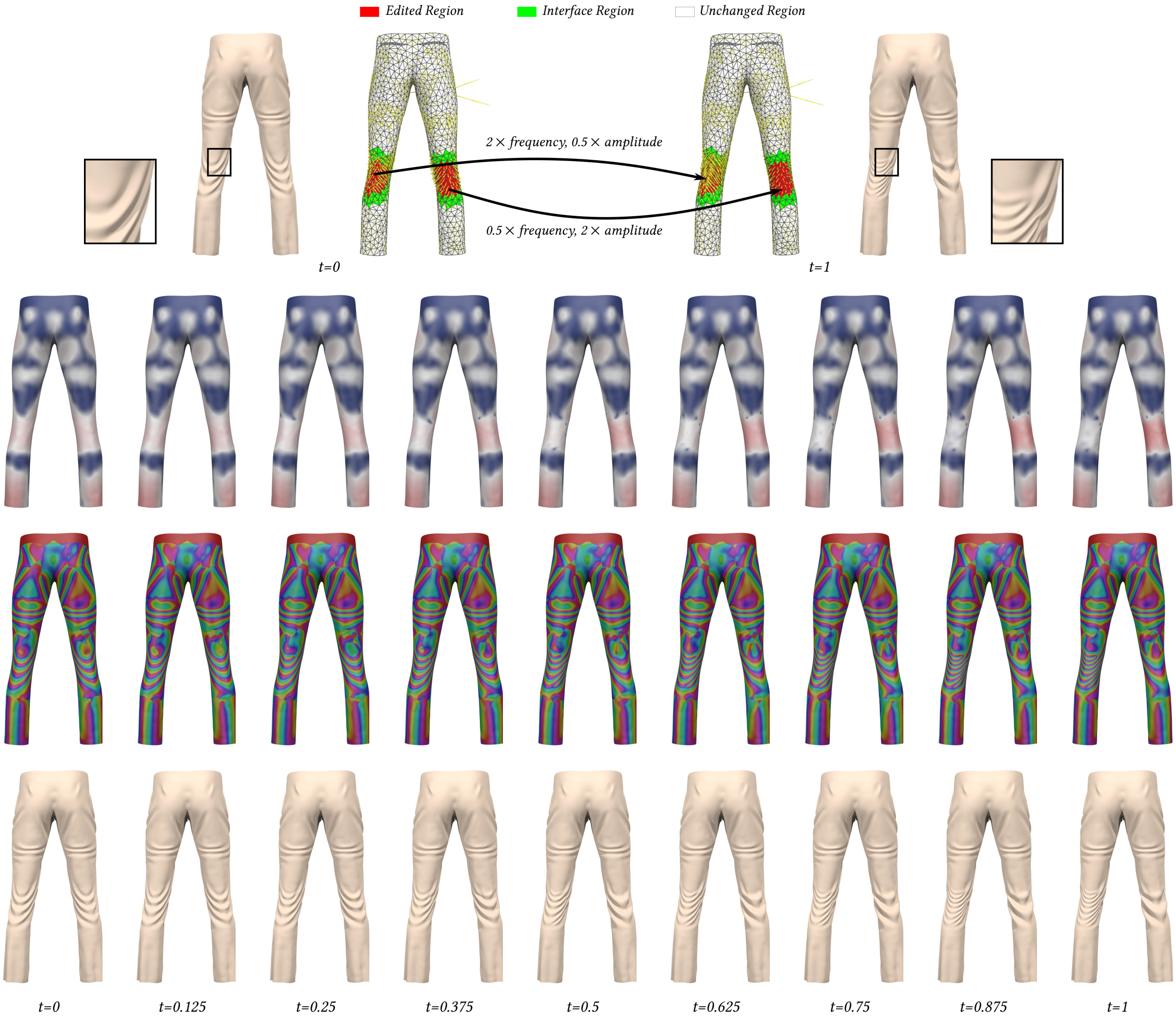


Fig. 8. *CWF* interpolation of two wrinkle patterns on the pants, where the initial frames are generated using the code provided by Chen et al. [2021]. Please check the **Interpolation Results** video in supplementary materials at 04:04-04:21 for the corresponding wrinkle animation.

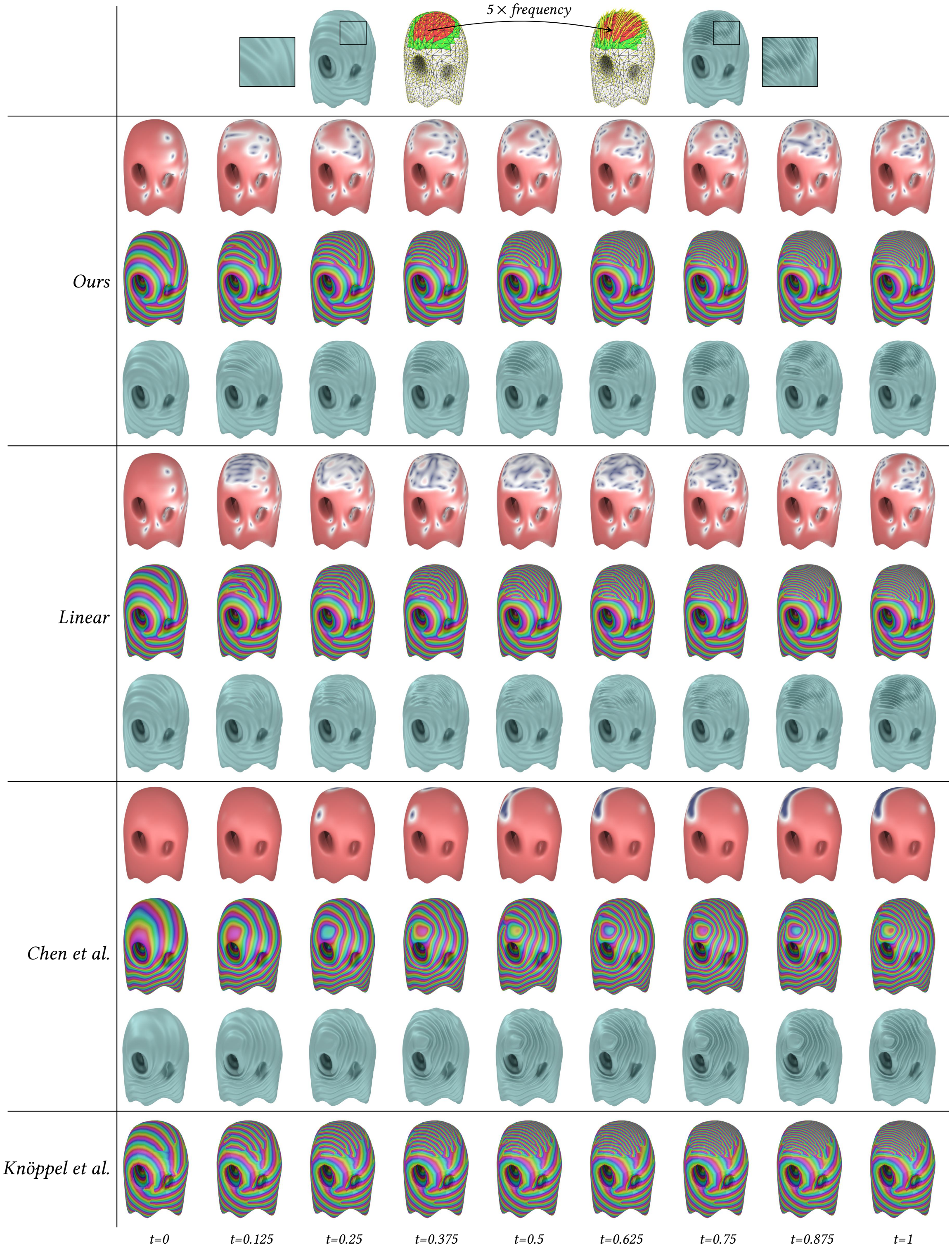


Fig. 9. The results of enlarging a local patch of the phantasma model (see the first row for details). You can see clear amplitude artifacts when using direct linear interpolation, and strange phase patterns when using the method proposed by Chen et al. [2021]. Although the method of Knöppel et al. [2015] produces beautiful static phase patterns, it suffers from temporal incoherence. Similar temporal incoherence can be seen in Chen et al. [2021]. These coherence issues can be better visualize in the **main** supplementary video (at 04:05-04:27) and in the **Comparisons** supplementary video (at 01:57-02:58).

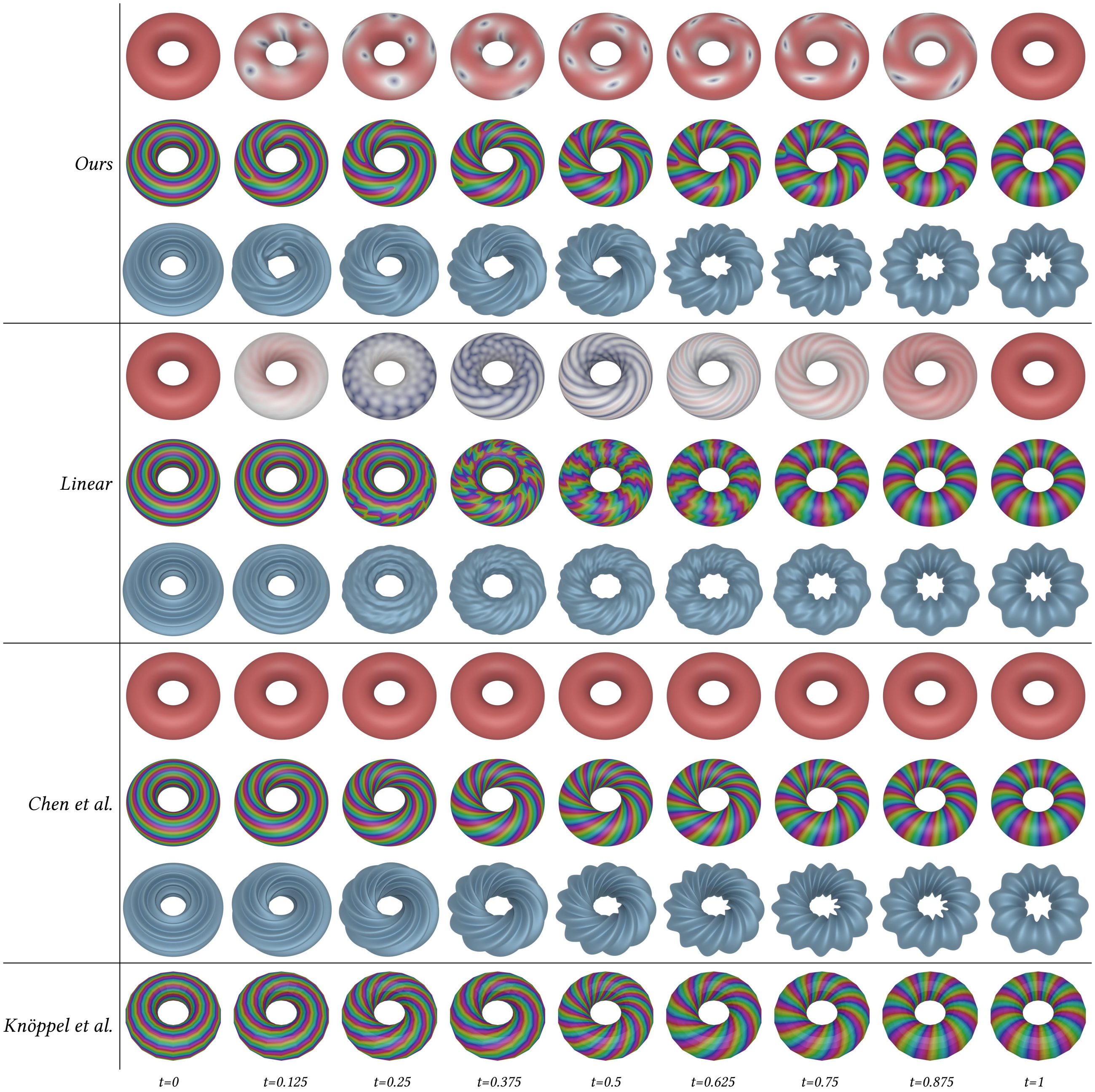


Fig. 10. The results of rotating wrinkles on a torus by ninety degrees. You can see clear amplitude artifacts when using direct linear interpolation. Although in this example Chen et al. [2021] and Knöppel et al. [2015] produce beautiful static phase patterns with no singularities, the corresponding results suffer from temporal incoherence. These coherence issues can be seen in the **main** supplementary video at 02:50-03:44 and in the **Comparisons** supplementary video at 00:05-00:59.

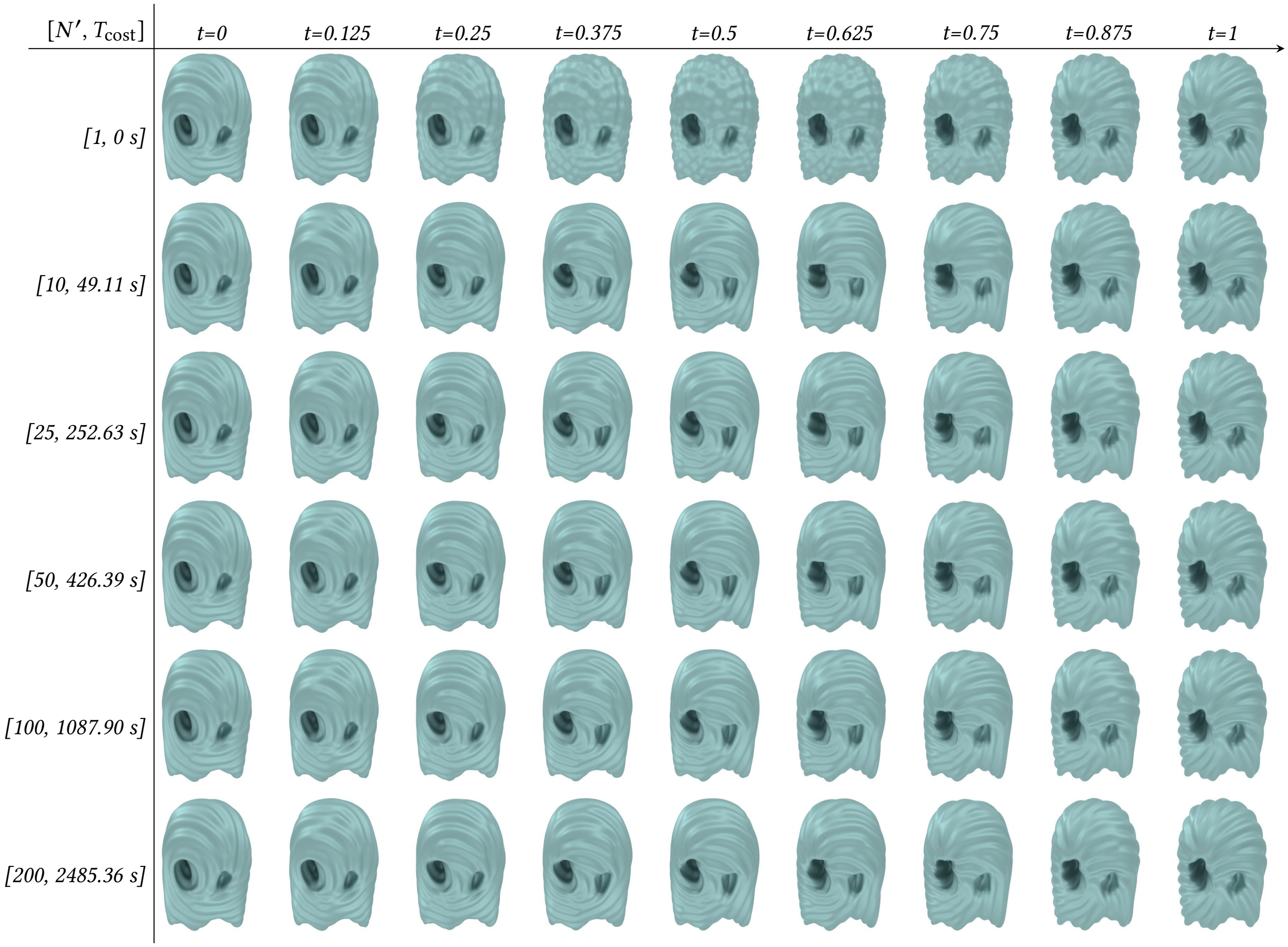


Fig. 11. Interpolation of rotating wrinkle patterns on the phantasma model by ninety degrees, where [•, •] are the number of guide frames, and the computation time (in seconds), respectively. Notice that the differences between $N' = 25, 50, 100$ and 200 are subtle. You can refer to the **Ablation Experiments** video in the supplementary materials (00:14-00:31) for details.

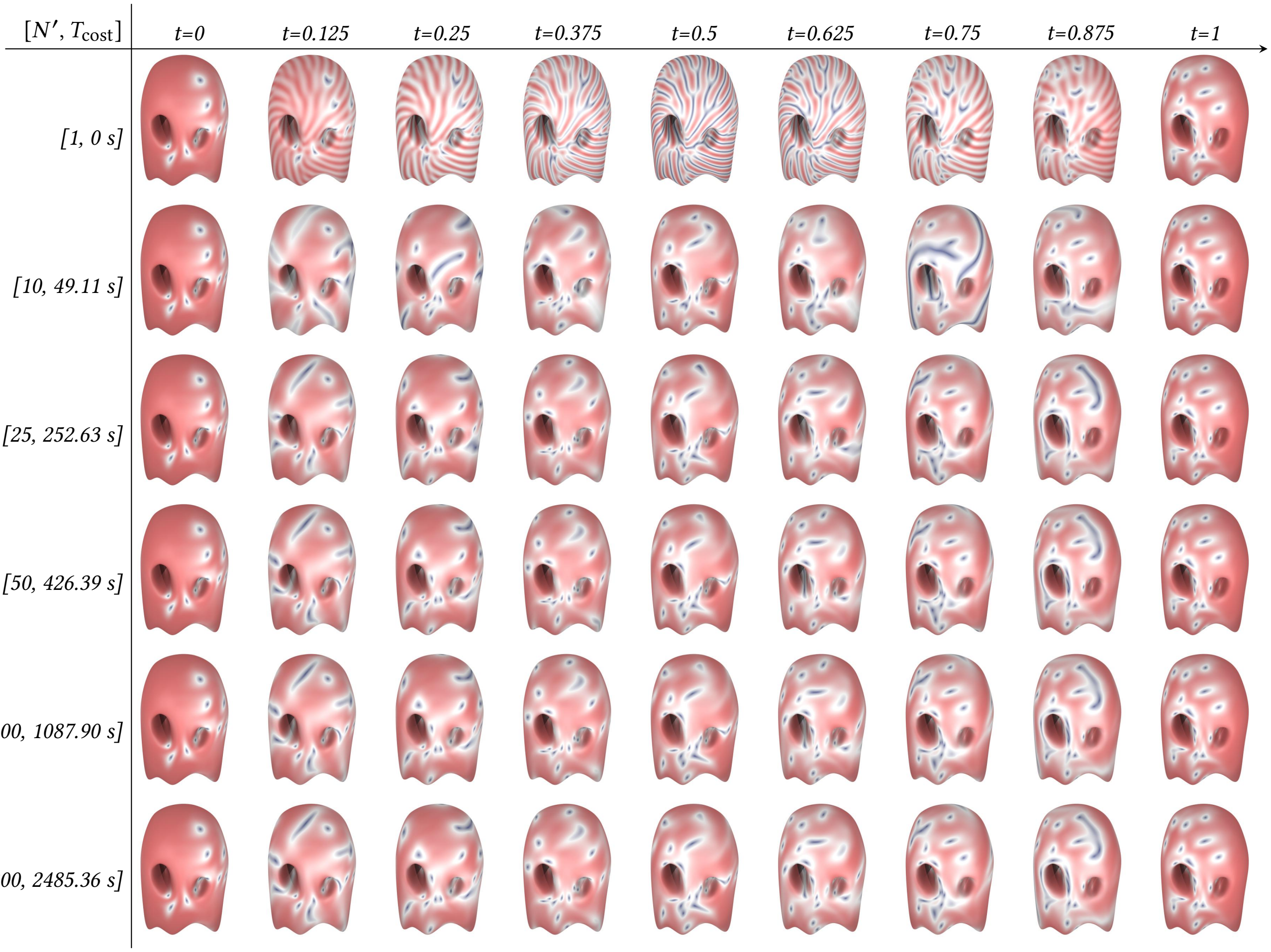


Fig. 12. The corresponding amplitude patterns of the example in Figure 11. Again, please refer to the **Ablation Experiments** video in the supplementary materials (00:14-00:31) for details.

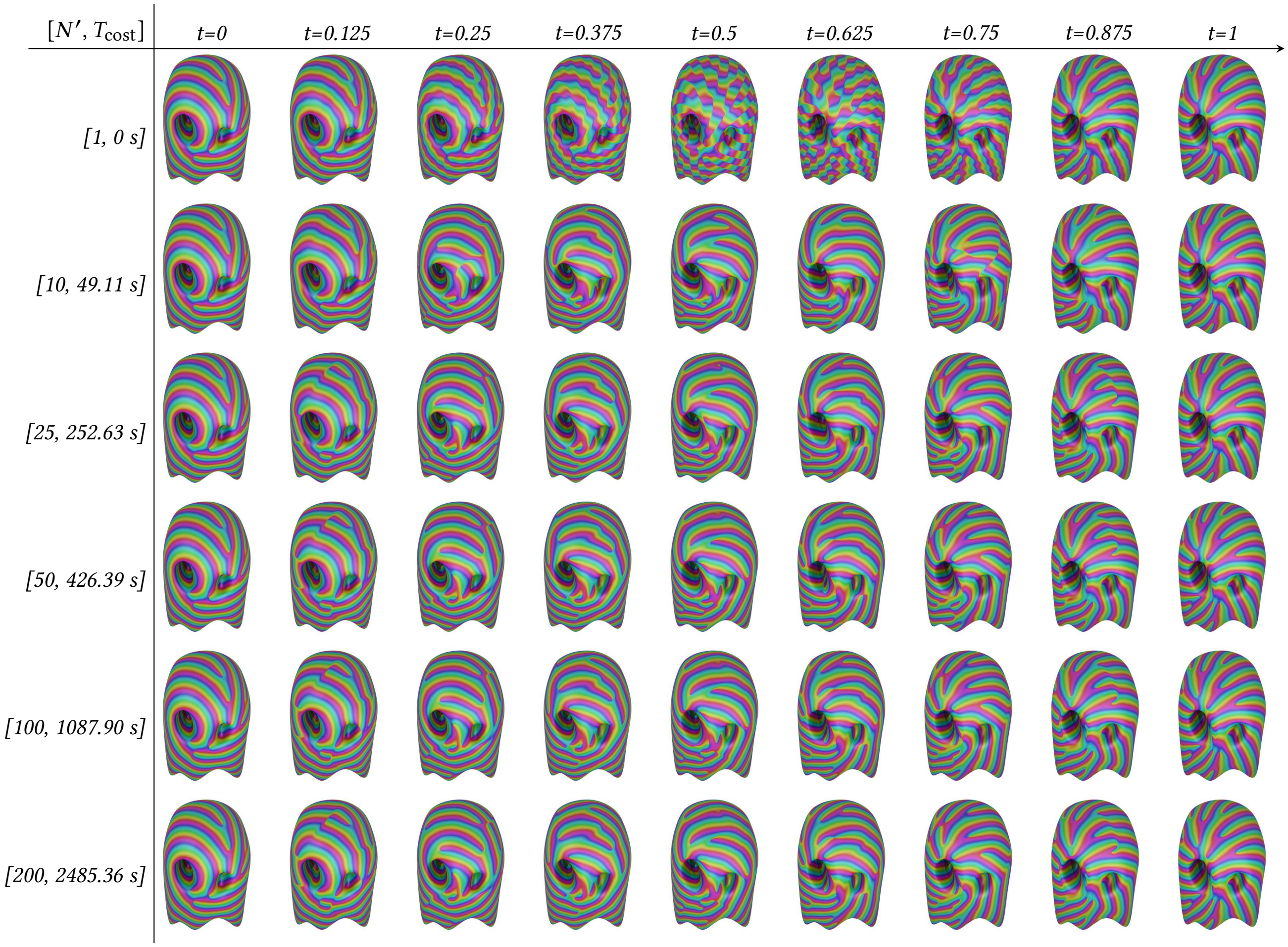


Fig. 13. The corresponding phase patterns of the example in Figure 11. Again, please refer to the **Ablation Experiments** video in the supplementary materials (00:14-00:31) for details.

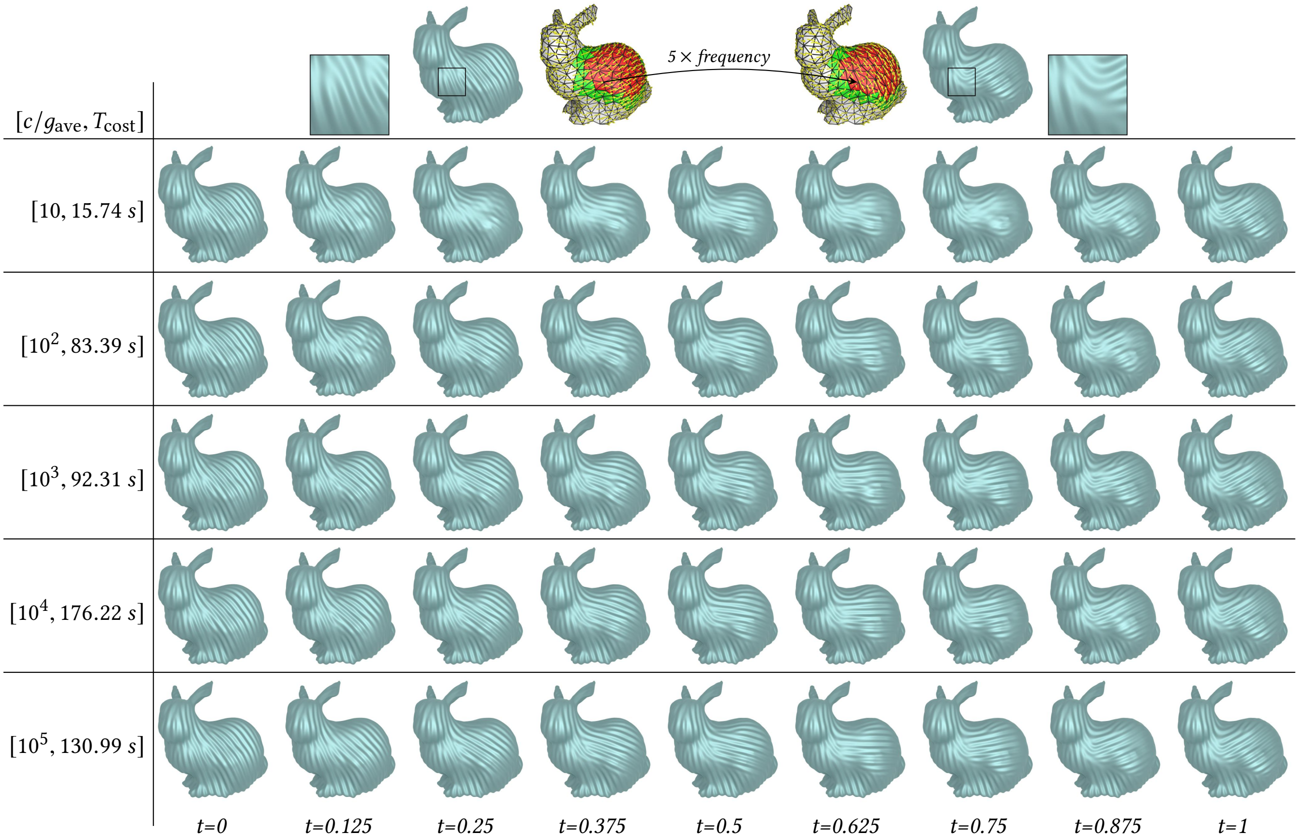


Fig. 14. *CWF* interpolation of two wrinkle patterns on the Stanford bunny model, where the target wrinkles are from locally rotating the corresponding patch of the first keyframe by ninety degrees. T_{cost} is the time cost of the interpolation solve. Please refer to the **Ablation Experiments** video in the supplementary materials at timestamp 00:44-01:07 for details.

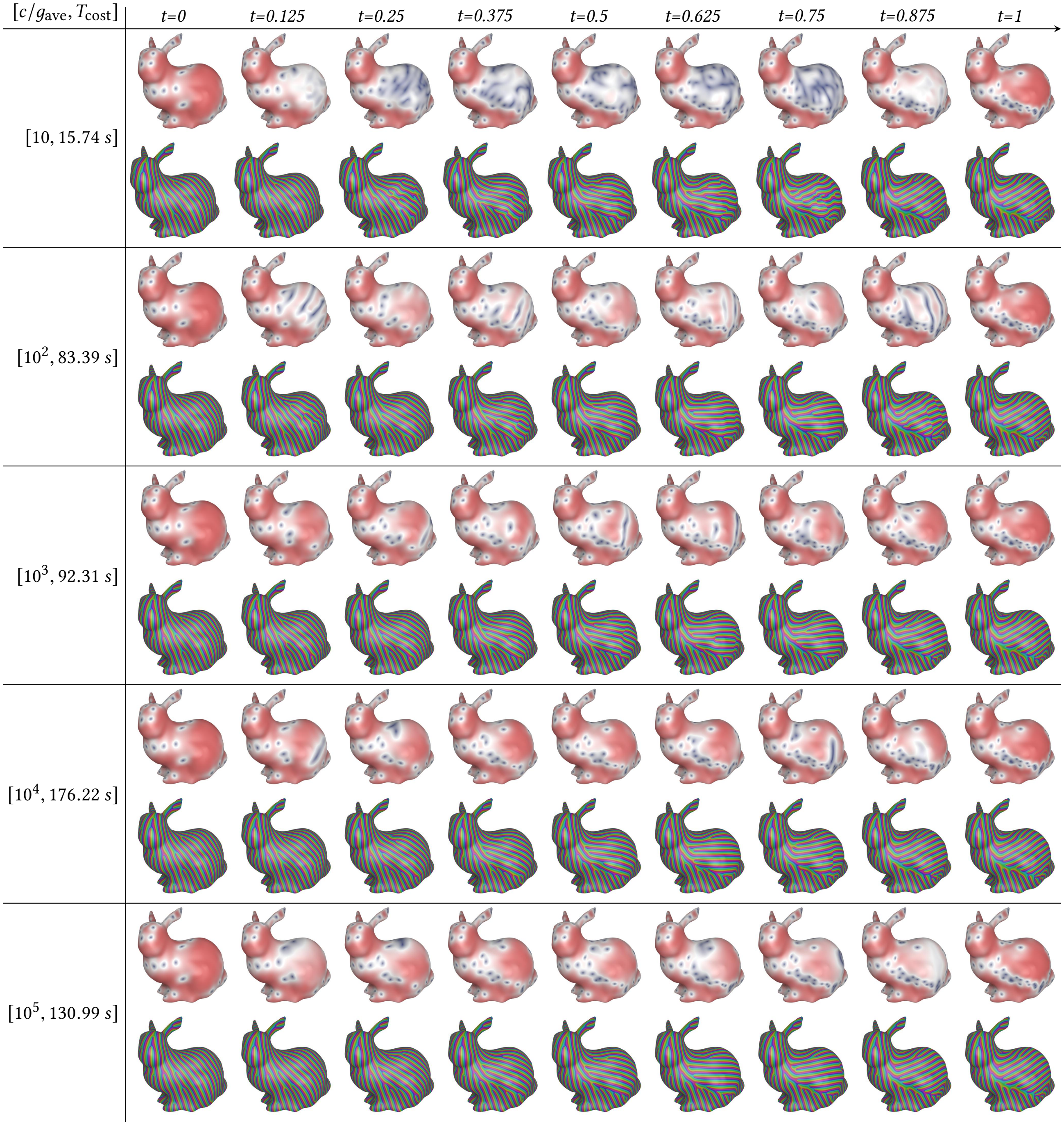


Fig. 15. The corresponding amplitude (odd rows) and phase (even rows) patterns of the example in Figure 14. Please refer to the **Ablation Experiments** video in the supplementary materials at timestamp 00:44-01:07 for details.