EUROGRAPHICS 2026 / B. Masia and J. Thies
(Guest Editors)

Volume 45 (2026), Number 2

Supplementary Material for Strain-Field Based Segmentation for Fabric Formwork

Abhinit Sati, Tiffany Bao, Jeff Tedi, Edward Chien, Emily Whiting

Boston University, USA

1. Cloth Material Model

We use a modified version of the cloth model proposed by [WOR11] in conjunction with Arcsim [NSO12] for cloth simulation. [WOR11] proposed a modified version of Hooke's law, describing the non-linear constitutive relationship between the strain tensor ε and the stress tensor $\boldsymbol{\sigma}$ as follows:

$$\boldsymbol{\sigma} = \begin{bmatrix} \sigma_{uu} \\ \sigma_{vv} \\ \sigma_{uv} \end{bmatrix} = \mathbf{K}(\varepsilon_{uu}, \varepsilon_{vv}, \varepsilon_{uv}) \begin{bmatrix} \varepsilon_{uu} \\ \varepsilon_{vv} \\ \varepsilon_{uv} \end{bmatrix} = \mathbf{K}(\varepsilon)\varepsilon \tag{1}$$

where both the stress and strain tensor are expressed in their vectorized representations. The first 2 entries measure stretch/stress along the canonical warp (u) and weft (v) basis, and the third entry measures the angle change between them to describe shearing. The stiffness tensor $\mathbf{K}$ maps the strain tensor to the stress tensor. The strain tensor is parameterized using its eigendecomposition in terms of λ_{max} and ϕ, where λ_{max} captures the constitutive nonlinearity, and ϕ encodes the material anisotropy. $\mathbf{K}$ is expressed as a function of the two as follows

$$\mathbf{K}(\lambda_{max}, \phi) = \begin{bmatrix} k_{uu}(\lambda_{max}, \phi) & k_{uv}(\lambda_{max}, \phi) & 0 \\ k_{uv}(\lambda_{max}, \phi) & k_{vv}(\lambda_{max}, \phi) & 0 \\ 0 & 0 & k_{ss(\lambda_{max}, \phi)} \end{bmatrix} \tag{2}$$

where k_{uu} is the stiffness in the longitudinal direction, k_{vv} is the stiffness in the transverse direction, k_{uv} is the Poisson ratio and k_{ss} is the shear stiffness [Str08]. It is important to note that $\mathbf{K}$ in its current form describes orthotropic materials (a subset of anisotropic materials) ideally suited for woven fabrics as demonstrated by [WOR11] and [CTT17].

2. Non-linear Isotropic Material Model

Inspired by the work of [FZZ*20], we propose using a non-linear isotropic material model for simulation, as it eliminates the need to make initial assumptions about stiffness in any particular direction and ensures that all directions are treated uniformly. We set the stiffness matrix to be a function of the *max stretch* λ_{max}, to ensure the strain-limiting behavior exhibited by most fabrics is still intact. If the stiffness parameters are unaffected by the strain angle, we can characterize the material as being isotropic. We can then reformulate the stiffness tensor defined in equation 2, independent of the strain angle as follows

$$\mathbf{K}(\lambda_{max}) = \begin{bmatrix} k_{uu}(\lambda_{max}) & k_{uv}(\lambda_{max}) & 0 \\ k_{uv}(\lambda_{max}) & k_{vv}(\lambda_{max}) & 0 \\ 0 & 0 & k_{ss(\lambda_{max})} \end{bmatrix} \tag{3}$$

We set the stiffness values along the longitudinal and transverse axes to be the same, i.e $k_{uu} = k_{vv}$. We then employ the following expression for isotropic materials, which establishes a relationship between longitudinal and transverse stiffness, Poisson's ratio, and shear stiffness, derived from Lamé's second parameter equation

$$k_{ss} = \frac{k_{uu}}{2(1 + k_{uv})} \tag{4}$$

We elaborate more on the values we choose for these material parameters in section 11. The two sets of parameters, *rest* and *max stretch*, aid in modeling the initial soft elastic phase followed by the hard deformation limit typical of biphasic materials such as woven cloth [TCT22].

3. Force Jacobians for Implicit Integration

We compute the Force Jacobians for use in the backward Euler solver [BW23] for all forces described in the main paper in the following subsections.

3.1. Stretch/Membrane Forces and their Jacobian

We model membrane forces as the gradient of the non-linear St-VK stretch energy defined for every triangle t, in terms of the Green strain tensor $\boldsymbol{\epsilon}$ expressed in Voigt form

$$E_{\text{stretch},t} = \frac{1}{2} \boldsymbol{\epsilon}^T \mathbf{K}(\boldsymbol{\epsilon}) \boldsymbol{\epsilon} \tag{5}$$

We can describe the internal force on a vertex on the deformed mesh as follows

© 2026 The Author(s)
Computer Graphics Forum © 2026 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

$$\mathbf{f_m}(\bar{\mathbf{x}}_\mathbf{i}) = -\sum_{t\in\mathcal{N}(i)} \frac{A_t}{3}\frac{\partial E_{\text{stretch},t}}{\partial \bar{\mathbf{x}}_\mathbf{i}} \tag{6}$$

where A_t is the area of the rest triangle t in 2D and $\mathcal{N}(i)$ is the one-ring neighborhood of vertex i. Given that our system is dominated by in-plane stretching due to the inflation caused by the infill pressure of the liquid plaster, we disregard bending forces as their contribution is insignificant compared to the stretching forces.

For the Jacobian of the stretching forces, we use the implementation provided in the codebase of *ArcSim* [NSO12].

3.2. Fluid Forces and their Jacobian

Like [ZFS*19], we model the hydrostatic fluid pressure force on a vertex $\bar{\mathbf{x}}_\mathbf{i}$ given a suspension orientation direction $\mathbf{d}$ via the following equation:

$$p(\bar{\mathbf{x}}_\mathbf{i}, \mathbf{d}) = \rho g(\bar{\mathbf{x}}_\mathbf{d} - \bar{\mathbf{x}}_\mathbf{i})\cdot \mathbf{d} \tag{7}$$

ρ is the density of the liquid plaster, g is the acceleration due to gravity and $\bar{\mathbf{x}}_\mathbf{d}$ is the zero-pressure reference point. The force on vertex $\bar{\mathbf{x}}_\mathbf{i}$ can be expressed in terms of the pressure as follows

$$\mathbf{f_p}(\bar{\mathbf{x}}_\mathbf{i}) = A_\mathbf{i}\, p(\bar{\mathbf{x}}_\mathbf{i}, \mathbf{d})\mathbf{n_i} \tag{8}$$

where $A_\mathbf{i}$ is the area of a small region around vertex $\mathbf{i}$ ($\frac{1}{3}$ of the total area of the triangles in its one-ring neighborhood) and $\mathbf{n_i}$ is the corresponding surface normal.

The force Jacobian is computed as follows,

$$\frac{\partial \mathbf{f_p}}{\partial \bar{\mathbf{x}}_\mathbf{i}} = -A_\mathbf{i}\rho g(\mathbf{n_i}\otimes\mathbf{d}) \tag{9}$$

where $\otimes$ denotes the outer product of 2 vectors. We assume the area around the vertex, $A_\mathbf{i}$ and normal, $\mathbf{n_i}$ won't change too much during the course of the simulation, and compute the derivatives by keeping them constant to simplify calculations. Since the fluid force is computed on a per-vertex basis, the Jacobian is a 3 × 3 block inserted into the global system matrix.

3.3. Jacobian for Seam Forces

These are only incorporated in the full non-linear orthotropic simulation, after we obtain the segmentation.

We model seams in our framework using springs with high stiffness with the familiar expression for Hooke's law. Let $\mathbf{i}$ and $\mathbf{j}$ be the vertex indices that form a seam edge $(\mathbf{i},\mathbf{j})$. The seam force on vertex $\mathbf{i}$ is then given by

$$\mathbf{f_s}(\bar{\mathbf{x}}_\mathbf{i}) = -\kappa(\|\bar{\mathbf{x}}_\mathbf{i} - \bar{\mathbf{x}}_\mathbf{j}\| - L)\frac{\bar{\mathbf{x}}_\mathbf{i} - \bar{\mathbf{x}}_\mathbf{j}}{\|\bar{\mathbf{x}}_\mathbf{i} - \bar{\mathbf{x}}_\mathbf{j}\|} \tag{10}$$

κ describes the spring constant and $L = \|\hat{\mathbf{x}}_\mathbf{i} - \hat{\mathbf{x}}_\mathbf{j}\|$ is the rest length of the seam edge $(\mathbf{i},\mathbf{j})$. The force on vertex $\mathbf{j}$, $\mathbf{f_s}(\bar{\mathbf{x}}_\mathbf{j})$ is simply $-\mathbf{f_s}(\bar{\mathbf{x}}_\mathbf{i})$. Given the average length of seams in our segmentation and the Young's modulus of Nylon (used for stitching), estimated at 1 GPa, we set $\kappa = 5\times 10^5$ N/m.

Similar to [Stu22], we compute

$$\frac{\partial \mathbf{f_s}}{\partial \bar{\mathbf{x}}_\mathbf{i}} = \mathbf{H_s} = -\kappa\left(\left(1 - \frac{L}{\|\bar{\mathbf{x}}_\mathbf{ij}\|}\right)(\mathbf{I} - \mathbf{p}\otimes\mathbf{p}) + \mathbf{p}\otimes\mathbf{p}\right) \tag{11}$$

Since the spring force is computed as a 6-dimensional vector (for both points in the spring $\mathbf{i}$ and $\mathbf{j}$), the Jacobian is a 6 × 6 block inserted into the global system matrix

$$\begin{bmatrix} \mathbf{H_s} & -\mathbf{H_s} \\ -\mathbf{H_s} & \mathbf{H_s} \end{bmatrix}$$

$\bar{\mathbf{x}}_\mathbf{ij} = \bar{\mathbf{x}}_\mathbf{i} - \bar{\mathbf{x}}_\mathbf{j}$ and $\mathbf{p} = \frac{\bar{\mathbf{x}}_\mathbf{i} - \bar{\mathbf{x}}_\mathbf{j}}{\|\bar{\mathbf{x}}_\mathbf{i} - \bar{\mathbf{x}}_\mathbf{j}\|}$ are used to simplify the expression.

3.4. Putting it all together

The solver for updating the state of the simulation is set up in the standard fashion:

$$\mathbf{A}\Delta\mathbf{v} = \mathbf{b}$$

where $\mathbf{A} = \mathbf{M} - h^2\left(\frac{\partial \mathbf{f_p}}{\partial \bar{\mathbf{x}}} - \frac{\partial \mathbf{f_m}}{\partial \bar{\mathbf{x}}} - \frac{\partial \mathbf{f_s}}{\partial \bar{\mathbf{x}}}\right)$ and $\mathbf{b} = h\mathbf{f_n} + h^2\left(\frac{\partial \mathbf{f_m}}{\partial \bar{\mathbf{x}}} + \frac{\partial \mathbf{f_s}}{\partial \bar{\mathbf{x}}}\right)\mathbf{v_n}$. $\mathbf{f_n}$ and $\mathbf{v_n}$ are the total forces and velocities at the previous timestep respectively. $\Delta\mathbf{v}$ is used to update the velocities, which in turn is used to update vertex positions. $\mathbf{M}$ is the mass matrix.

4. Field Smoothing for Denoising

For the first smoothing procedure that reduces overall noise, we want to maintain a balance between smoothness and the alignment to the original field for all regions on the mesh - both *isotropic* and *anisotropic*. To accomplish this, we modify the original smoothness energy described by [BZK09] and integrate soft constraints for following the original vector field

$$E_{smooth}(\theta) = \underbrace{\sum_{e_{ij}\in E}(\theta_i - \theta_j + \kappa_{ij} + \frac{\pi}{2}p_{ij})^2}_{\text{smoothness term}} + \alpha\underbrace{\sum_{f_k\in F}(\theta_k - \theta_k^t)^2}_{\text{soft constraint}} \tag{12}$$

where θ_i, θ_j and θ_k specify the angle used to parameterize the 4 vector field for faces i, j and k respectively, w.r.t a reference edge of each face. p_{ij} describes the period jump that helps map a vector field vector from a face to its neighboring face across edge ij, shared between face i and face j. θ_k^t specifies the angle used to describe the original vector field. α is a user parameter for adjusting the balance between the desired smoothness of the field and its adherence to the original field.

© 2026 The Author(s)
Computer Graphics Forum © 2026 The Eurographics Association and John Wiley & Sons Ltd.

5. Fiber Orientation that Eliminates Shear

Let $\mathbf{F}$ be the deformation gradient, $\mathbf{C} = \mathbf{F}^\top \mathbf{F}$ the right Cauchy–Green tensor, and

$$\varepsilon = \tfrac{1}{2}(\mathbf{C} - \mathbf{I})$$

the Green–Lagrange strain. Let $\{\mathbf{m}_1, \mathbf{m}_2\}$ denote orthonormal *material axes* (warp/weft) in the reference configuration. We use the structural invariants [Ogd97]

$$I_4 = \mathbf{m}_1^\top \mathbf{C}\mathbf{m}_1, \quad I_6 = \mathbf{m}_2^\top \mathbf{C}\mathbf{m}_2, \quad I_8 = \mathbf{m}_1^\top \mathbf{C}\mathbf{m}_2.$$

Note that the Green–Lagrange shear component in the material basis is

$$\varepsilon_{uv} = \tfrac{1}{2} I_8 = \tfrac{1}{2}\mathbf{m}_1^\top \mathbf{C}\mathbf{m}_2.$$

Proposition 1 (Perfect alignment implies zero material–frame shear) Let $\mathbf{C}$ be symmetric positive definite with orthonormal eigenvectors $\{\mathbf{s}_1, \mathbf{s}_2\}$ and eigenvalues $\{\lambda_1, \lambda_2\}$, i.e. $\mathbf{C}\mathbf{s}_i = \lambda_i \mathbf{s}_i$. If the material axes are

$$\mathbf{m}_1 = \mathbf{s}_1, \qquad \mathbf{m}_2 = \mathbf{s}_2,$$

then the mixed invariant vanishes:

$$I_8 = \mathbf{m}_1^\top \mathbf{C}\mathbf{m}_2 = 0,$$

and hence $\varepsilon_{uv} = 0$.

Proof With $\mathbf{m}_1 = \mathbf{s}_1$ and $\mathbf{m}_2 = \mathbf{s}_2$,

$$\mathbf{m}_1^\top \mathbf{C}\mathbf{m}_2 = \mathbf{s}_1^\top (\mathbf{C}\mathbf{s}_2) = \mathbf{s}_1^\top (\lambda_2 \mathbf{s}_2) = \lambda_2 \mathbf{s}_1^\top \mathbf{s}_2 = 0,$$

since the eigenvectors of $\mathbf{C}$ are orthonormal. Therefore $I_8 = 0$ and $\varepsilon_{uv} = \frac{1}{2} I_8 = 0$. □

Corollary 1 (Right–angle preservation) Let θ be the angle between the deformed images of $\mathbf{m}_1$ and $\mathbf{m}_2$. Then

$$\cos\theta = \frac{(\mathbf{F}\mathbf{m}_1)\cdot(\mathbf{F}\mathbf{m}_2)}{||\mathbf{F}\mathbf{m}_1||\,||\mathbf{F}\mathbf{m}_2||} = \frac{\mathbf{m}_1^\top \mathbf{C}\mathbf{m}_2}{\sqrt{I_4}\sqrt{I_6}}.$$

Under the alignment of the Proposition, $\cos\theta = 0$, so the warp–weft orthogonality is preserved ($\theta = 90°$).

6. Drift Penalty for Separatrices

We modify the cost function from [PPM*16] to penalize a path drifting away from the vector field

$$cost(\mathbf{v}, \mathbf{w}) = \frac{1}{\lambda_{\mathbf{u}}} ||\mathbf{w} - \mathbf{v}|| \left(1 + \alpha \left(\frac{angle(\mathbf{u}, \mathbf{w} - \mathbf{v})}{\theta_{max}}\right)^2\right) \tag{13}$$

where $\mathbf{v}$ is the current vertex, $\mathbf{w}$ is a candidate vertex and $\mathbf{u}$ is the vector field direction that contains $\mathbf{w}$ in its sector to ensure field coherency [PPM*16]. θ_{max} and α are user defined parameters describing the maximum angle deviation allowed from the vector field and drift penalty respectively. $\lambda_{\mathbf{u}}$ is the magnitude of the stretch along direction $\mathbf{u}$.

7. Singularities of 4-vector fields

The table above describes the different singularities that show up in 4-vector fields.

Index	Valence
$+\frac{1}{4}$	3
$+\frac{1}{2}$	2
$-\frac{1}{4}$	5
$-\frac{1}{2}$	6

Table 1: *Index of rotations for singularities and their corresponding valences*

8. Developability

We measure developability of mesh segments using the Covariance energy from [SGC18]. This energy is leveraged by a spectral clustering approach that splits segments to ensure quasi-developability.

8.1. Covariance energy

Intuitively, this energy measures how *hinge* like a vertex is by examining its one ring neighborhood (star of the vertex). A vertex can be qualified as a hinge, if and only if all the triangles in its star have normals that lie on a common plane, captured by the following expression

$$\eta_i = \min_{|\mathbf{u}|=1} \sum_{ijk \in O} \theta_i^{jk} \langle \mathbf{u}, \mathbf{n}_{ijk} \rangle^2 \tag{14}$$

η_i measures the area of the polygon created by projecting the normals of the faces in the star of vertex i on the Gauss sphere. The closer it is to 0, the more *developable* our vertex is. $\mathbf{u}$ represents the orientation of the best fit plane, ijk are the vertex indices that make up a face in the star of i and θ_i^{jk} is the interior angle of the face at vertex i. $\mathbf{n}_{ijk}$ is a normal to face ijk and O is the star of vertex i. We can obtain η_i by solving an eigenvalue problem, details of which can be found in the Appendix section of [SGC18]. We can define the developability of a segment as follows

$$\varepsilon^\eta = \sum_{i \in V} \eta_i \tag{15}$$

We decide if a segment needs to be split if this total is more than a user defined threshold.

8.2. Spectral Clustering with Covariance energies

1. Construct an affinity matrix $\mathbf{A} \in \mathbb{R}^{m \times m}$ from the dual graph of the mesh, where m is the number of faces in the mesh. Cell (i, j) of this matrix is a non-zero entry if face i and j are adjacent to each other. We set $\mathbf{A}[i, j] = e^{-p \cdot w}$, where p is a user defined parameter and w is computed as the difference of the covariance energy values at the 2 vertices forming the edge (i, j). This ensures the probability of making a cut along a high covariance energy vertex is maximized.

© 2026 The Author(s)
Computer Graphics Forum © 2026 The Eurographics Association and John Wiley & Sons Ltd.

2. Compute the unnormalized Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D} \in \mathbb{R}^{m \times m}$ is a diagonal matrix whose entry (i,i) is the sum of the elements in the i^{th} row of $\mathbf{A}$.
3. Compute the first k eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ of $\mathbf{L}$, where k is the number of clusters/segments specified as input.
4. Define $\mathbf{U} \in \mathbb{R}^{m \times k}$ such that it contains vectors $\mathbf{u}_1, \ldots, \mathbf{u}_k$ as columns.
5. Let $\mathbf{y}_i \in \mathbb{R}^k$ be the vector corresponding to the ith row of matrix $\mathbf{U}$, for all $i \in 1, \ldots, m$.
6. Cluster all $\mathbf{y}_i$ with the k-means clustering algorithm to generate clusters $C_1, \ldots, C_k$.

9. Instron machine tensile tests details

The machine used for tensile testing is the Instron 2712-04x, equipped with pneumatic grippers pressurized at 60 psi for all tests. All of our samples have a length of 40 mm and width of 20 mm. The maximum load capacity for the pneumatic gripper attachment is 500 N, and the highest values reached during our experiments were around 350 N. The rate of displacement is 2 mm/min. We conduct tensile tests on all the samples until fracture occurs. The resulting curves represent only the elastic response of the fabric. These curves are truncated at the onset of the plastic region, which occurs after the material fractures.

10. Detailed steps for setting up the simulation for estimating material parameters

- Clamp the top edges of the sample and apply downward in-plane forces to the bottom edges to simulate the fabric being pulled during a tensile test.
- In our tensile tests, the displacement rate (2 mm/min) is constant and serves as the input to the system, with the corresponding force required to maintain that rate as the output. In our simulation program, the forces are the input, and the displacement is computed in response to these forces. We ensure that the forces applied in the simulation are synchronized with the time elapsed when the same force is applied during the tensile test. This is accurate and works well since the simulation is quasi-static (the magnitudes of the internal and external forces are almost equal at all time-steps).
- The simulation is run until equilibrium is reached, with force and displacement values computed and stored at each time step. Displacement is measured as the difference in length between the center points of the top and bottom edges on the cloth mesh. This data is then stitched together to create the full force-displacement curve for the simulation.

11. Material Parameters for Isotropic Simulation

We modify the piecewise orthotropic material model described by [WOR11] to a piecewise-linear isotropic model. In the table below, the *Rest* row represents the stiffness values in the relaxed state, while the *Max stretch* row corresponds to the stiffness values when the material is stretched to its limit, allowing us to capture the biphasic stretching stiffness behavior of woven fabrics.

	k_{uu}	k_{uv}	k_{vv}	k_{ss}
Rest	300 N/m	2	300 N/m	37.5 N/m
Max stretch	600 N/m	2	600 N/m	75 N/m

Table 2: *Stiffness values used in the non-linear isotropic simulation*

12. Material parameters for the woven fabric

The table below gives the material parameters we obtained for the woven fabric used for fabrication.

Parameters	Origin	0°	22.5°	45°	67.5°	90°
k_{uu}	36496.94	453939.78	321081.07	147439.0	439895.04	418775.97
k_{uv}	17.77	24.67	37.87	11.50	52.62	82.44
k_{vv}	56329.3	499773.53	643526.19	261801.58	383126.43	488043.68
k_{ss}	3171.91	10794.15	46434.4	13953.55	12605.24	44474.28

Table 3: *Stiffness values for various orientations of the woven fabric used for making the fabric formwork.*

References

[BW23] BARAFF D., WITKIN A.: Large steps in cloth simulation. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 2023, pp. 767–778. 1

[BZK09] BOMMES D., ZIMMER H., KOBBELT L.: Mixed-integer quadrangulation. *ACM Transactions On Graphics (TOG) 28*, 3 (2009), 1–10. 2

[CTT17] CLYDE D., TERAN J., TAMSTORF R.: Modeling and data-driven parameter estimation for woven fabrics. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2017), pp. 1–11. 1

[FZZ*20] FANG G., ZHANG T., ZHONG S., CHEN X., ZHONG Z., WANG C. C.: Reinforced fdm: Multi-axis filament alignment with controlled anisotropic strength. *ACM Transactions on Graphics (TOG) 39*, 6 (2020), 1–15. 1

[NSO12] NARAIN R., SAMII A., O'BRIEN J. F.: Adaptive anisotropic remeshing for cloth simulation. *ACM transactions on graphics (TOG) 31*, 6 (2012), 1–10. 1, 2

[Ogd97] OGDEN R. W.: *Non-Linear Elastic Deformations*. Dover, 1997. 3

[PPM*16] PIETRONI N., PUPPO E., MARCIAS G., SCOPIGNO R., CIGNONI P.: Tracing field-coherent quad layouts. In *Computer graphics forum* (2016), vol. 35, Wiley Online Library, pp. 485–496. 3

[SGC18] STEIN O., GRINSPUN E., CRANE K.: Developability of triangle meshes. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 1–14. 3

[Str08] STRONG A. B.: *Fundamentals of composites manufacturing: materials, methods and applications*. Society of manufacturing engineers, 2008. 1

[Stu22] STUYCK T.: *Cloth simulation for computer graphics*. Springer Nature, 2022. 2

[TCT22] TANG P., COROS S., THOMASZEWSKI B.: A second order cone programming approach for simulating biphasic materials. In *Computer Graphics Forum* (2022), vol. 41, Wiley Online Library, pp. 87–93. 1

[WOR11] WANG H., O'BRIEN J. F., RAMAMOORTHI R.: Data-driven elastic models for cloth: modeling and measurement. *ACM transactions on graphics (TOG) 30*, 4 (2011), 1–12. 1, 4

[ZFS*19] ZHANG X., FANG G., SKOURAS M., GIESELER G., WANG C. C., WHITING E.: Computational design of fabric formwork. *ACM Transactions on Graphics 38*, 4 (2019), 1–13. 2

© 2026 The Author(s)
Computer Graphics Forum © 2026 The Eurographics Association and John Wiley & Sons Ltd.