CONTENTS

I. TUBULAR ENABLES MEASUREMENTS OF DEFORMING TISSUE SURFACES ACROSS SYSTEMS

Our approach cartographically projects dynamic surfaces onto a fixed material frame of reference. As shown in Fig. S1, this approach enables measurements which would be otherwise challenging. First, minimizing tissue motion in the material frame of reference enables cell tracking on dynamic surfaces (Fig. S1A). This is shown for a tracked quartet of cells in a *w;48Y-GAL4/UAS-CAAX:mCherry;klar/+* embryo, which has a fluorescent plasma membrane marker in the midgut. We processed this dataset as follows: after an iLastik pass to identify the interior of the midgut tissue, TubULAR's default level sets minimization captured the apical surface of the endoderm by minimizing a Chan-Vese functional on the output of the iLastik training. The result from each timepoint was fed into the level sets optimization of the subsequent timepoint to track the tissue surface across all timepoints. We pushed this surface 2.5μm outward to approximately intersect the endoderm midplane. We then mapped this dynamic surface to a 2D material parameterization and tracked cells using a semi-automated approach detailed in [1].

Second, we can measure coarse-grained fields such as the anisotropy of fluorescent signals. In Fig. S1B, we extract the orientation of the anisotropic tissue structure by applying a Radon transform [4, 5] to the projected surface of the circular muscle layer in the midgut of a *w;UAS-CAAX:mCherry/+;Mef2-GAL4,klar/+* embryo. The circular muscle layer consists of long, thin cells that ensheath the midgut endoderm, and we measure their local orientation in the material frame of reference to be tightly ordered at an angle of $\pi/2$, corresponding to the $\hat{\phi}$ direction in the material frame.

Third, by tracking both cell motion and tissue motion in dynamic tissue surfaces, we can distinguish the contributions of cell rearrangments and cell shape changes to tissue shape change, as depicted schematically in (Fig. S1C). We work through an example of inferring such 'tissue tectonics' in Section IX 'Example of inferring intercalation rates using TubULAR.' Lastly, as depicted schematically in Fig. S1D, our method also enables mode decomposition of a system's dynamics.

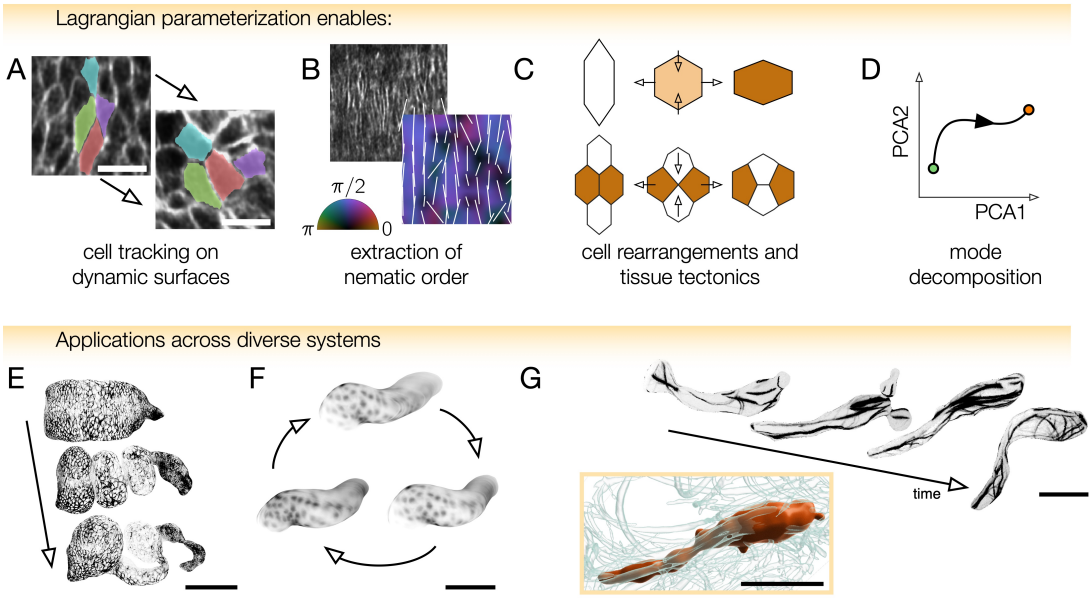

FIG. S1. **Across diverse systems, parameterization of tubes in a material (Lagrangian) frame enables cell tracking on dynamic, curved surfaces, measurements of tissue anisotropy, decomposing tissue shape change into cell rearrangements and cell shape changes, and mode decomposition.** *(A)* Computational microscopy enables dynamic tissue surface extraction and cell tracking. After selectively imaging the endodermal layer in a *w;48Y-GAL4/UAS-CAAX:mCherry;klar* embryo, we track cells, with four highlighted that exhibit intercalations highlighted here. Scale bars are 10 μm. *(B)* Performing a radon transform on a patch of circular visceral muscle cells – which form a 'palisade' structure ensheathing the midgut endoderm – returns a measure of tissue anisotropy aligned with the circumferential direction ($\phi = \pi/2$, violet). *(C)* Constructing material-frame pullback images facilitates measurements of tissue shape change, cell shape change, and cell intercalation rates. *(D)* Following the tissue's deformation enables mode decomposition of the dynamics, offering descriptions with reduced complexity, shown schematically as a path of the system through mode space after performing principal component analysis (PCA). *(E-G)* This method tracks deformations for organs and *in vitro* systems alike, including the Drosophila midgut [1], the beating zebrafish heart [2], and a phase-separated droplet deforming in a cytoskeletal gel [3]. Scale bars are 100 μm for all panels. *(G)* We follow a DNA-nanostar-based droplet (red) deforming in an active fluid [3]. Here, mechanical coupling between the interface of a liquid-liquid phase separated DNA droplet and a surrounding active microtubule fluid (cyan rods) generates continuous deformation towards droplet breakup. In the time series, microtubule fluorescence within 10 μm of the interface is projected onto the deforming droplet surface.

Our approach finds applications across diverse systems (Fig. S1E-G). In addition to tracking the dynamics of *Drosophila* midgut morphogenesis (Fig. S1E) and zebrafish heart beating (Fig. S1F) showcased in the main text, we also applied TubULAR's surface extraction and visualization to the surface of a deforming DNA droplet in a microtubule active fluid [3]. Fig. S1G shows a snapshot of a DNA droplet in an active microtubule gel. Briefly, DNA droplets are assembled from multi-armed DNA nanostructures with self-interacting complementary overhangs [6]. Kinesin motors are bound to a subpopulation of the DNA nanostars in the system. Active flows are generated by microtubule filaments depleted through non-adsorbing polymers such as polyethylene glycol (PEG) and powered by clusters of kinesin motors. Kinesin motors convert chemical energy from the environment and generate inter-filament sliding [7]. The DNA droplets are covalently bound to kinesin motors resulting in mechanical coupling between the microtubule bundles and the surface of the droplets. The microtubules (light blue fluorescent intensity isosurfaces in 3D rendering, black in grayscale `TexturePatch` time series) align with the elongating DNA-nanostar droplet, suggesting that droplet elongation is driven by microtubule alignment.

II. TUBULAR PIPELINE COMPONENTS AND SCOPE

While many existing tools aid tracking of cells in 3D, our toolkit allows the user to measure tissue motion with respect to fully parameterized, dynamic surfaces with complex geometry (Fig. S2). TubULAR thereby extends efforts that analyze tissue motion on static geometries [8], quasi-2D surfaces [9, 10], and otherwise simple geometries [11].

In addition to detailed documentation and example pipelines available on GitHub, we summarize the steps in our approach in Fig. S3. This gives a typical sequence of method calls to extract surfaces, create an initial sequence of parameterizations constrained for minimal tis-

sue motion in the pullback plane, and compute covariant measures of tissue dynamics. Subsequent steps further refine the material coordinate definition to remove residual motion in pullback images and read out measures of velocity and strain in this Lagrangian frame (along 'material pathlines'). The last set of steps visualize the material motions, decompose them into divergence, curl, rate of area change, and measures of anisotropic deformation like the Beltrami coefficient. Finally, methods compute principal components of the tissue dynamics and decompose movement into eigenfunctions of the Laplace-Beltrami operator. Such eigenfunctions are akin to spherical harmonics for a sphere or Bessel functions for a cylinder, but defined on an arbitrary surface. Fig. S4, meanwhile, gives an overview of the class structures included in the toolkit.

The TubULAR workflow assumes individual TIFF files for each timepoint matching a common naming convention. For example, `Timepoint_000000.tif`, `Timepoint_000001.tif`, etc would be found in the same directory. If the dataset has multiple channels, all channels to be used should be found within the same file for a given timepoint, and the metadata instructs TubULAR how to interpret the axis order.

An output directory for the data will be generated upon instantiation of the TubULAR class. The directory structure information for inputs and outputs of the class are stored in the `dir` property of the TubULAR class instance. While multiple surfaces per timepoint can be analyzed by separate class instances of TubULAR, the default workflow is designed for a single surface per timepoint. For example, if the user would like to analyze the relative motion between two concentric tissue layers, the user could obtain a surface that lies approximately in between the two layers. In this scheme, the output of one TubULAR class instance rendering tissue texture at some positive distance from the mesh surfaces could be compared against the output of another TubULAR class instance rendering tissue texture at some negative distance from the mesh surfaces. In such a scenario, the two instances could share the same mesh directory (and other properties) to ensure no differences in measured motion arose from differences in the constrained mapping procedure.

III. INTEGRATION OF TUBULAR WITH IMSANE

ImSAnE [12] is a tissue cartography package in MATLAB used in the developmental biology community. Since the aims of ImSAnE overlap with those of TubULAR, we have implemented integration of the two toolkits in two ways: an ImSAnE Experiment class instance can be passed to TubULAR, or all of TubULAR's functionality can be accessed entirely within ImSAnE.

We found existing methods within ImSAnE (and in other software) to be unable to automatically follow

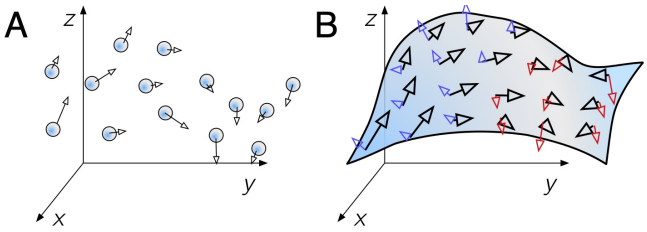

FIG. S2. **TubULAR advances analysis from bare 3D tracks to flows across and along dynamic surfaces.** *(A)* Many existing tools such as Imaris aid tracking isolated cells or other objects in 3D. Such methods do not immediately provide measurements of tissue deformations due to the lack of reference surface. *(B)* TubULAR builds surfaces on which motion is decomposed into in-plane and out-of-plane motion. In this schematic, motion along the surface normal is blue for outward and red for inward, while black arrows illustrate tangential motion.

tissue-scale motions of the full surfaces demonstrated in the main text. In Fig. S5, we highlight a comparison of TubULAR against ImSAnE's `meshWrapper` fitter. Here, the surface is divided by five Voronoi patches (blue, red, yellow, violet, and green colors in Fig. S5B). The patches were chosen by identifying mesh vertices that are far apart at $t = t_0 = 0$ min, then point-matching each vertex onto the subsequent timepoint. This point-matching repeated throughout the timecourse of morphogenesis. Individual patches of the surface can be fit in ImSAnE, but the motion of these patches in pullback space can be erratic (Fig. S5C). In contrast, TubULAR parameterizes the entire surface in a single coordinate chart, modulo two endcaps at the anterior and posterior poles (Fig. S5D). Here, we chose a 100×100 vertex paramerization, with mesh vertices equally spaced in a rectangular grid in $s\phi$ pullback coordinates, so the discretization is visible in the surface normals. Tissue motion in the material coordinates is minimized in this method, so that coloring surfaces across different timepoints and overlaying their pullback images results in a largely grayscale image (Fig. S5E).

Fig 1F of the main text showed that pullback images made via ImSAnE's `cylinderMeshWrapper` likewise do not have stabilized motion in the pullback plane. The overall image also changed size from timepoint to timepoint. As shown in Fig. S6, we also found that existing methods within ImSAnE that attempt to fit the entire surface were not able to handle complex geometries such as the folded midgut.

A. Passing ImSAnE to TubULAR

An ImSAnE `Experiment` class instance can be passed to TubULAR as a first argument instead of a metadata struct. If no meshes have been found using ImSAnE already, then this step can be done with the newly created TubULAR instance, along with subsequent analysis. Al-

```
Instantiation
Set experiment metadata and any custom options.....xp=struct(...); opts=struct(...);
Instantiate TubULAR.....tubi = TubULAR(xp, opts)

Surface extraction
Create downsampled volumes to use for iLastik & level sets.....tubi.prepareIlastik()
Extract the surfaces using level sets on output of iLastik or raw data..tubi.getMeshes()

Parameterization
Define global reference frame (APDV).....tubi.computeAPDVCoords()
Define endcap points and one point for a virtual seam.....tubi.computeAPDpoints()
Align meshes into global frame.....tubi.alignMeshesAPDV()
Optional: render data on dynamic surfaces (can be slow).....tubi.plotSeriesOnSurfaceTexturePatch()
Compute initial centerlines.....tubi.extractCenterlineSeries()
Temporally average centerlines and fix any inconsistencies.....tubi.generateCleanCntrlines()
Remove the endcaps.....tubi.sliceMeshEndcaps()
Clean up the mesh endcaps.....tubi.cleanCylMeshes()
Cut the meshes along virtual seams.....tubi.generateCurrentCutMesh()
Constrained parameterization of the mesh into (u,v) and (s,φ) frames....tubi.generateCurrentSPCutMesh()
Generate pullback images.....tubi.generateCurrentPullbacks()

Dynamics
Initial pass of smoothing embedding over time.....tubi.smoothDynamicSPhiMeshes()
Create pullbacks of smoothed (s,φ) frames.....tubi.generateCurrentPullbacks()
Tile the images in φ (as 'double-cover').....tubi.doubleCoverPullbackImages()
Measure any residual material motion in the parameterization.....tubi.measurePIV2d()
Translate pullback information into motion in 3D (embedding space).....tubi.measurePIV3d()

Refined processing of surface dynamics
Recommended: smooth the resulting 3D dynamics along material pathlines..tubi.timeAverageVelocities()
Compute material pathlines in 3D.....tubi.measurePullbackPathlines()
Generate pullbacks of material pathlines.....tubi.generateCurrentPullbacks()
Measure rate of strain over the surface at each timepoint.....tubi.measureStrainRate()
Query the rate of strain along pathlines.....tubi.measurePathlineStrainRate()
Measure the integrated strain along pathlines.....tubi.measurePathlineStrain()

Interpretation and mode decomposition
Visualize flows.....tubi.plotTimeAvgVelocities()
Decompose into divergence and curl, scalar potential fields.....tubi.helmholtzHodge()
Compute areal rate of change.....tubi.measureMetricKinematics()
Mesaure anisotropic strain using Beltrami coefficient.....tubi.measureBeltramiCoefficient()
Decompose into Principal Component Analysis.....tubi.getPCAoverTime()
Decompose into eigenmodes of the Laplace-Beltrami operator.....tubi.getLBSoverTime()
```

FIG. S3. **Example high-level pipeline for data analysis using TubULAR passes through constrained parameterization, measurement of surface dynamics, refinement, and steps to interpret the results.** Descriptions of typical steps in a TubULAR pipeline are listed on the left, with the corresponding class method calls for each goal on the right.

ternatively, users may leverage any of ImSAnE's surface detection methods before passing the result to TubULAR so long as the detection results in a series of meshes, named as indicated in the metadata passed to TubULAR. See the online documentation for further details.

B. Upgrades to ImSAnE integrate TubULAR's functionality

ImSAnE workflows proceed in two parts: surface detection and surface fitting. ImSAnE now has an `integralDetector` method for surface detection which mirrors TubULAR's `getMeshes` method. After surface detection, the `tubularFitter` creates a TubULAR instance and carries this object as a property. All functions can be accessed within the ImSAnE `Experiment` instance's surfaceFitter.

IV. VALIDATION USING A SYNTHETIC DATASET

We validated the performance of TubULAR using synthetic data of a monolayer tissue arranged in a tube that coils and uncoils. Maximum intensity projections of the 'nuclei'-like and 'membrane'-like channels of this data are shown in Fig. S7A.

A. Dataset generation

To generate the dataset, the surface geometry was encoded numerically by specifying a noisy centerline that coils with an amplitude that varies along the curve and that varies over time. The centerline was defined as

$$x(\zeta) = \frac{R_0}{2}\cos(2\pi\zeta)(1-\cos(2\pi\zeta)) + \eta_x(\zeta) \quad (1)$$

$$y(\zeta) = \frac{R_0}{2}\sin(2\pi\zeta)(1-\cos(2\pi\zeta)) + \eta_y(\zeta) \quad (2)$$

$$z(\zeta) = 5((2\zeta-1)^3+1) + \frac{R_0}{2}\sin(2\pi\zeta), \quad (3)$$

where $\eta_x, \eta_y \in [0,1]$ add noise to the curve, and $R_0 = (1+3\sin[2\pi(t-1)/20 - 0.25])/2$ represents the signed radius of the tube's coil. The width of the tube was chosen to vary as $\max(0.02, 0.25 + R_1[\cos(\pi(\zeta+0.5)) + 0.1])$, where $\zeta \in [0,1]$ parameterizes the centerline curve.

We placed 120 nuclei-like blobs of intensity centered at locations across the surface. Locations were chosen as a solution to an iterative farthest-point search, so that nuclei are well-spaced from each other. We then performed

FIG. S4. **The TubULAR class and the DiscreteExteriorCalculus class within `DECLab` carry properties and methods, listed here for reference.**

a Voronoi tessellation to create a channel mimicking 'cell-cell junctions' (green in Fig. S7). The nuclei sizes were determined based on the distance of each nucleus to the nearest membrane location.

Fig. S7B shows the data projected on extracted surfaces for all timepoints.

B. Validation of tracked cell positions over time

How well can the TubULAR workflow capture positions of cells in this synthetic data? After extracting surfaces shown in Fig. S7B, pulling back to the $s\phi$ coordinate system, and tiling the pullback image along the $\hat{\phi}$ direction, we identify nucleus locations by threshold $s\phi$ pullback images using Otsu's method [14] and computing centroids of each segmented object.

We then track the resulting positions, linking sets of nucleus locations into trajectories using two separate approaches: (1) the Crocker-Grier algorithm [13] ('CG' method) or (2) ascribing the nearest nucleus in timepoint t_{i+1} to each identified nucleus in timepoint t_i ('Nearest-Neighbor' or 'NN method'). In the Nearest-Neighbor method, if two cells at timepoint t_i have the same nearest neighbor in timepoint t_{i+1}, then we choose to advance only the closer of the two cells to that neighbor's position. The track of the cell which was farther away will therefore terminate at this timepoint. Results of these two methods are shown in Fig. S8. Finally, we push those tracked positions back into 3D onto the dynamic surfaces. This process is illustrated in Fig. S9A.

Fig. S9B shows the positional error of identified nuclei obtained from tracking nuclei (via the Crocker-Grier method) in the $s\phi$ pullback images and pushing tracked positions forward into 3D. This error is measured against the known locations of nuclei input into the dataset creation process. Typical errors are below the width of a single pixel. This error includes positional error introduced from surface detection, potential errors during object tracking, and systematic errors from object detection. Only tracks that span all 20 timepoints of the dataset are included (111 out of 120 possible particles). As shown in Fig. S9C-E, the measured cell coordinates match the true cell coordinates, with only a few errant tracked positions.

C. TubULAR workflow improves tracking performance

We compared our method of tracking objects in the $s\phi$ coordinate system, which is largely stabilized against material motion, compared to directly tracking the true object positions in 3D space. We find that by first tracking in 2D with TubULAR using either the Crocker-Grier [13] or a Nearest-Neighbor algorithm then pushing tracks into 3D, we greatly improve both the duration of individual tracks that are automatically detected and the number

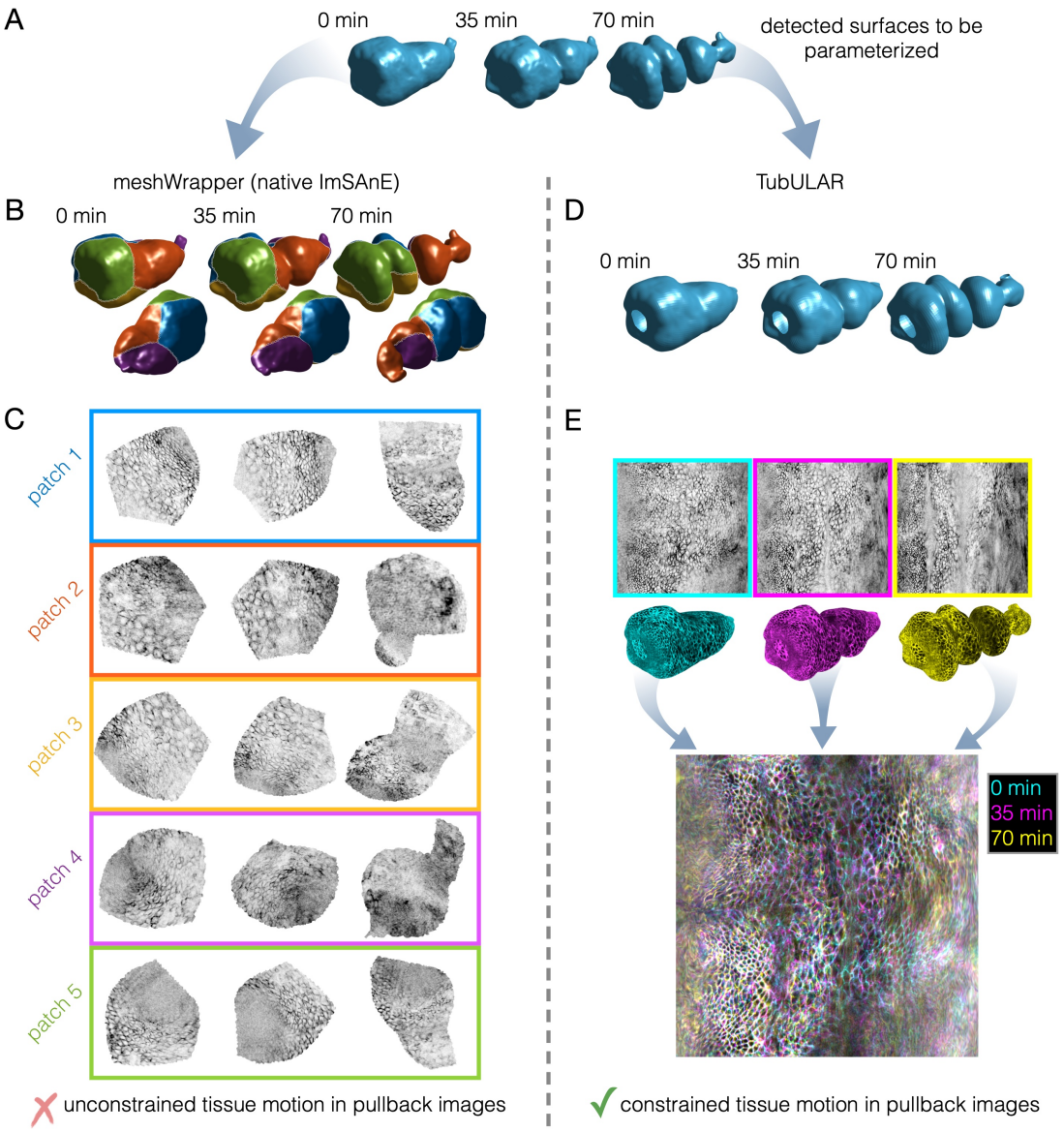

FIG. S5. **Previous methods fail to follow tissue motion, here illustrated by using the `meshWrapper` fitter native to ImSAnE.** *(A)* A detected surface results in a surface mesh triangulation to be parameterized. *(B)* Fitting the surface using ImSAnE's `meshWrapper` maps patches of the surface to the plane via a conformal map. *(C)* Pullback images from `meshWrapper` reflect large tissue motions over time, precluding cell tracking, tissue tracking, or measurements of tissue velocity. The overall image change, shape, and orientation varies from timepoint to timepoint. *(D)* In contrast, TubULAR parameterizes the full surface in a single chart, modulo two endcaps which are cut from the surface at the anterior and posterior ends to ensure a cylindrical topology. *(E)* Crucially, the resulting pullback images from TubULAR remain nearly stationary in the pullback frame due to stabilization. Here we overlay the textured surfaces (cyan, magenta, and yellow) in the material frame. The largely white cell membrane signal in the image reflects the stationary orientation of tissue in the pullback plane. Maps to the material frame from different timepoints are here taken as the fully stabilized maps $\varphi = J \circ \Phi \circ Z \circ f$.

of tracks that successfully identify nuclei across all timepoints (Fig. S10). Both 2D segmentation and 3D segmentation used Otsu's method [14] on the nuclear channel only (magenta in Fig. S7). Fig. S10 shows benchmarks of the tracking results. To link positions into tracks, we used the same two tracking algorithms in 2D vs 3D. This implies that compared to our method, tracking the 3D positions of cells requires more advanced algorithms, more manual corrections, or both. We also note here that to interpret isolated 3D tracks as tissue deformations requires additional analysis approaches, and these methods are already handled in our toolkit.

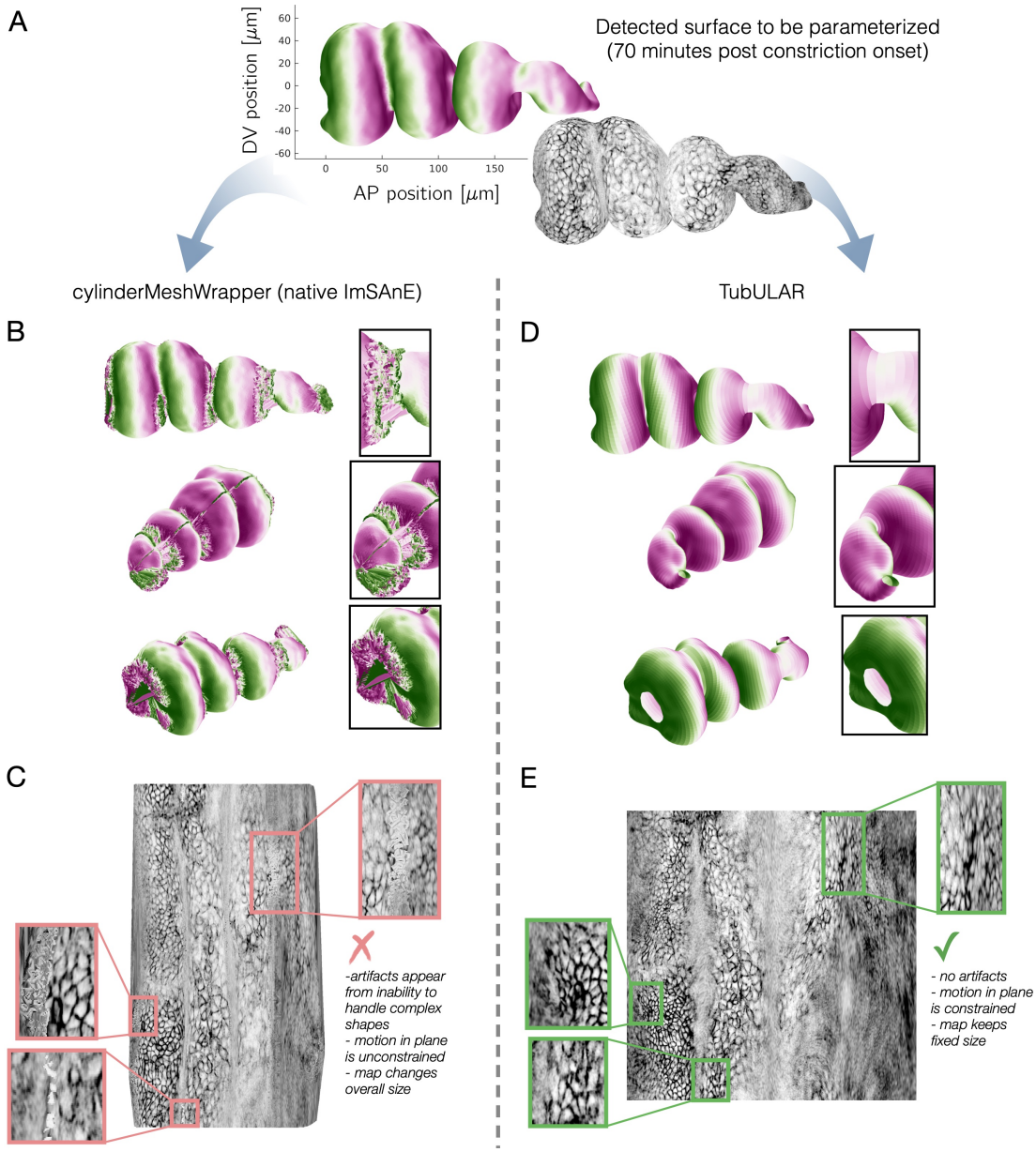

FIG. S6. **Previous methods fail to parameterize complex surface geometries, here illustrated by using the `cylinderMeshWrapper` fitter native to ImSAnE.** *(A)* A detected surface results in a surface mesh triangulation to be parameterized by an ImSAnE fitter. *(B)* Fitting the surface using `cylinderMeshWrapper` results in parameterization issues because of build-in assumptions about the simplicity of the underlying cylinder-like surface. The surface is colored by the normal vector of each face projected onto the anterior-posterior axis, so that green is pointing anteriorly and purple is pointing posteriorly. *(C)* Pullback images from `cylinderMeshWrapper` reflect the artifacts from poor parameterization near deep folds and surface overhangs. *(D)* The parameterized surface using TubULAR preserves surface normals and preserves mesh geometry. The surface is colored as in panel (B). *(E)* The resulting pullback images from TubULAR do not have parameterization artifacts.

D. Validation of cell velocities over time

Finally, we validated cell velocities by comparing the difference in tracked positions over time with known cell velocities of the synthetic dataset. We compared these velocities against the known motion of nuclei used in the dataset generation. As shown in Fig. S11, measured cell velocities agree with true cell velocities within a few percent, and the root-mean-squared (RMS) error of the velocities is small compared to the magnitude of the motion.

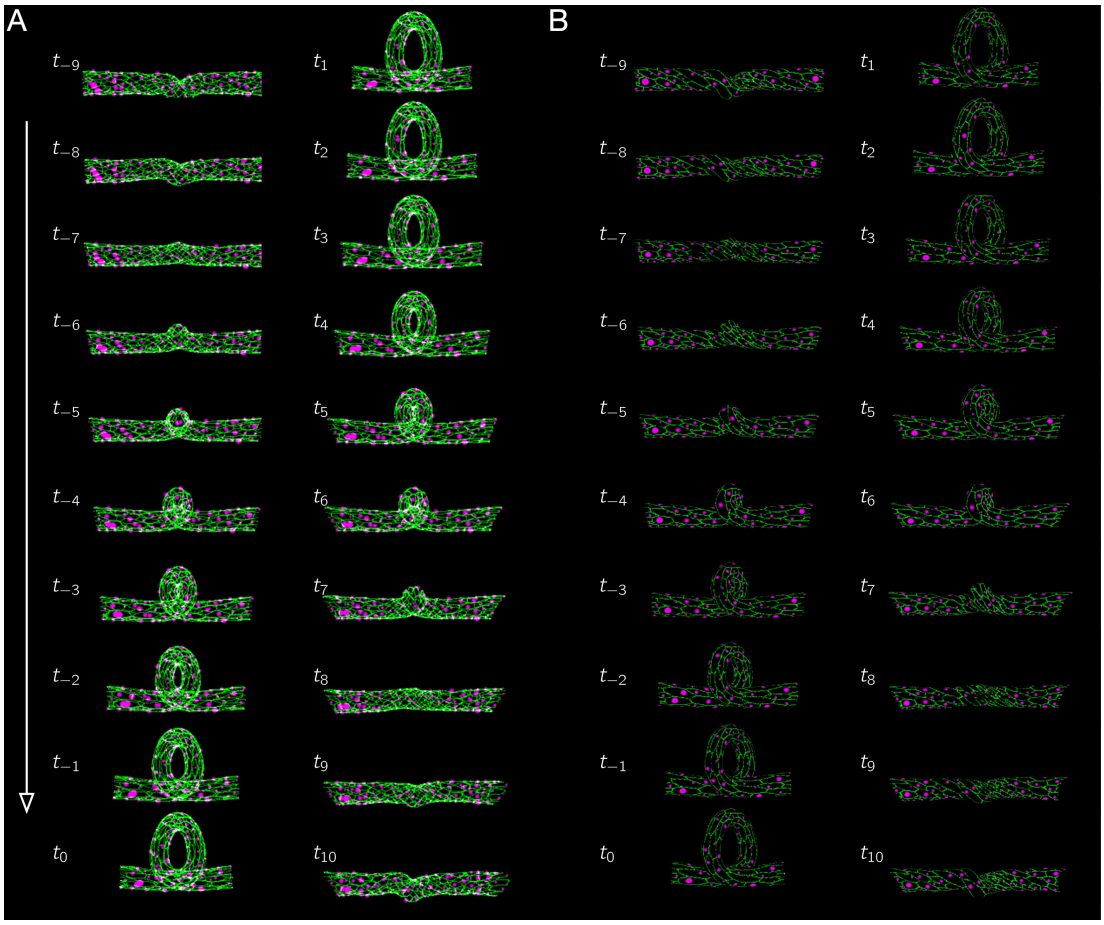

FIG. S7. **A synthetic dataset features a tube with dramatic changes in geometry, and its projection onto a surface generated in TubULAR captures the changes in shape.** *(A)* The maximum intensity projection of 'nuclear' (magenta) and 'membrane' (green) channels for 20 timepoints show a tube coiling and uncoiling. Nuclear positions corresponded to a farthest-point search of 120 locations on a prescribed surface with a time-dependent centerline and a (time-dependent and space-dependent) radius around that centerline. The membrane channel was created by building a Voronoi tesselation of the nuclear positions within the prescribed surface. Nuclei are given sizes based on their distance from the nearest 'cell-cell junction.' The prescribed surface is distinct from the extracted surface, but the extracted surface accurately captures the prescribed surface. *(B)* The projection of the dataset onto extracted surfaces shows both the changes in shape over time and the movement of cells on the dynamic surface.

V. SURFACE EXTRACTION USING LEVEL SETS

We now address details of steps that would appear in a TubULAR workflow. In this section, we begin with surface extraction. To extract whole-organ surfaces, we use a level sets approach, combined with marching cubes [15] and Laplacian smoothing. While the literature on level sets segmentation is vast [16], we give a brief overview of the relevant method here.

The process of surface detection is mapped onto an optimization problem by defining a physics-inspired cost functional [17, 18]:

$$F[c_1, c_2, \mathcal{S}] = \mu \int_{\mathcal{S}} \mathrm{d}s + \nu \int_{\Omega} \mathrm{d}^3\mathbf{x} \tag{4}$$

$$+ \lambda_1 \int_{u>0} |I(\mathbf{x}) - c_1| \mathrm{d}\mathbf{x} + \lambda_2 \int_{u<0} |I(\mathbf{x}) - c_2| \mathrm{d}\mathbf{x}, \tag{5}$$

where c_1 and c_2 are the average values of the data inside and outside, respectively:

$$c_1(\mathcal{S}) = \langle I(\mathbf{x}) \rangle_{\text{inside}} \tag{6}$$

$$c_2(\mathcal{S}) = \langle I(\mathbf{x}) \rangle_{\text{outside}}. \tag{7}$$

The first term is a surface tension that tries to smooth out ruggedness in the surface. The second term is an effective pressure that penalizes blow up in the enclosed volume of the segmented regions. The final two attachment terms incorporate the actual measured data into

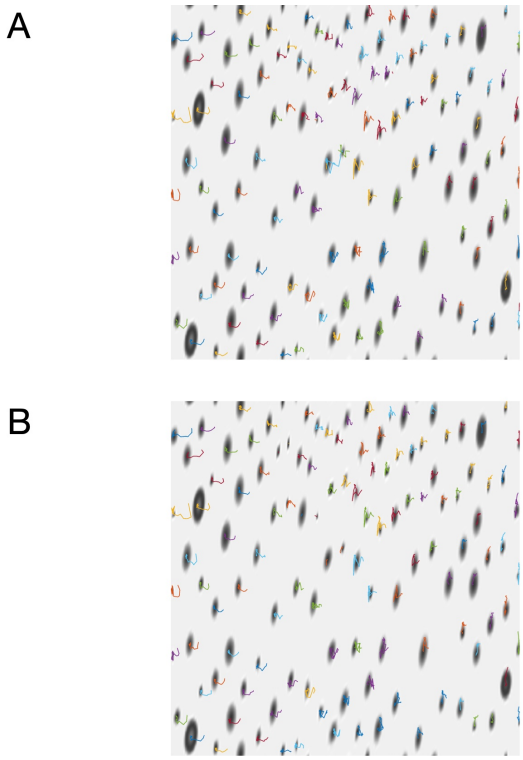

FIG. S8. **Constrained parameterization, here shown in the $s\phi$ coordinate system, aids in tracking.** Different tracking results across all 20 timepoints (colored curves) show almost no motion of nuclei despite large 3D motions of each nucleus. The tracks overlie intensity data for an example timepoint. Tracks obtained in the pullback plane via Crocker-Grier [13] algorithm in *(A)* and using a nearest-neighbor approach in *(B)*.

the optimization procedure. The third term attempts to homogenize the intensities of voxels included in the interior of the segmented volume and the fourth term attempts to homogenize the intensities of the excluded voxels by adjusting the surface position. Fig. S12 shows an example segmentation.

A. Segmentation of the midgut and synthetic datasets

To segment the surfaces of the midgut, synthetic datasets, and droplets shown in Fig. S1G, we generate a level set solution for a contiguous volume enclosing the interior of the tube, partitioning space into a topological sphere (the 'filled' tube) and its exterior, as shown in Fig. S12. TubULAR next removes endcaps of the mesh to create a topological tube (see later discussion). Before turning to endcap selection, we first note a variation on the TubULAR strategy that we employed for the zebrafish heart.

B. Segmentation of the zebrafish heart

The zebrafish heart dataset posed particular challenges requiring the application of some non-standard segmentation procedures, which we summarize here. Rather than attempt to segment the heart tissue by encapsulating the enclosed space with a level set, we used the level set methods to segment only the heart tissue. This resulted in a binary level set solution of toroidal topology. We applied a homotopic thinning procedure slice-by-slice along the length of the tube to produce a point cloud approximating the mid-surface of the heart tissue. We then fed a smoothed, up-sampled version of this point cloud into our Poisson surface reconstruction algorithm to produce a closed, sphere-like mesh of the heart. The results at this point in the segmentation process were structurally identical to those produced in a typical pipeline.

VI. ENDCAP SELECTION

Given extracted surfaces with sphere-like surface topology (rather than a cylinder-like surface topology) from the previous step, we remove two 'endcaps' and define a virtual seam along which to unwrap a cylinder into the plane. Endcaps are illustrated by red and green curves in Extended Data Fig. 1, and the virtual seam is a dashed purple curve. We define an endcap as a collection of contiguous faces (at least one face) that are removed from the mesh in order to map free boundaries to the plane.

As shown in Fig. S13, while removing a larger region from each endcap leads to larger "blind spots" where the surface is not parameterized, the resulting distortion in the mapped image may be smaller. This can have some practical advantages while segmenting or tracking cells in the 2D pullback parameterization. The minimum size of a removed endcap is a single face of the mesh triangulation, in which case the triangular boundary of the face would be mapped to $u = 0$ or $u = 1$. Removing a single vertex would lead to its immediate neighborhood (typically a hexagon) being mapped to $u = 0$ or $u = 1$.

By default, the locations of endcaps can be chosen manually with an interactive point-and-click method in a MATLAB Figure window. Subsequent timepoints will point-match the chosen vertex onto the subsequent timepoint's mesh triangulation, and this point-matching repeats through time.

Alternatively, the user can define a probability cloud identifying a region of the 3D data that should be considered an endcap for each timepoint using iLastik. The 'center of mass,' defined as the weighted centroid of the largest component of the probability cloud above a provided threshold, supplies a location to identify the endpoint. The mesh vertex nearest this center of mass is then defined as the endpoint.

A mesh region near each endpoint is removed from the mesh, as depicted in Fig. S13. The default behavior is to remove a patch that lies within some (geodesic) dis-

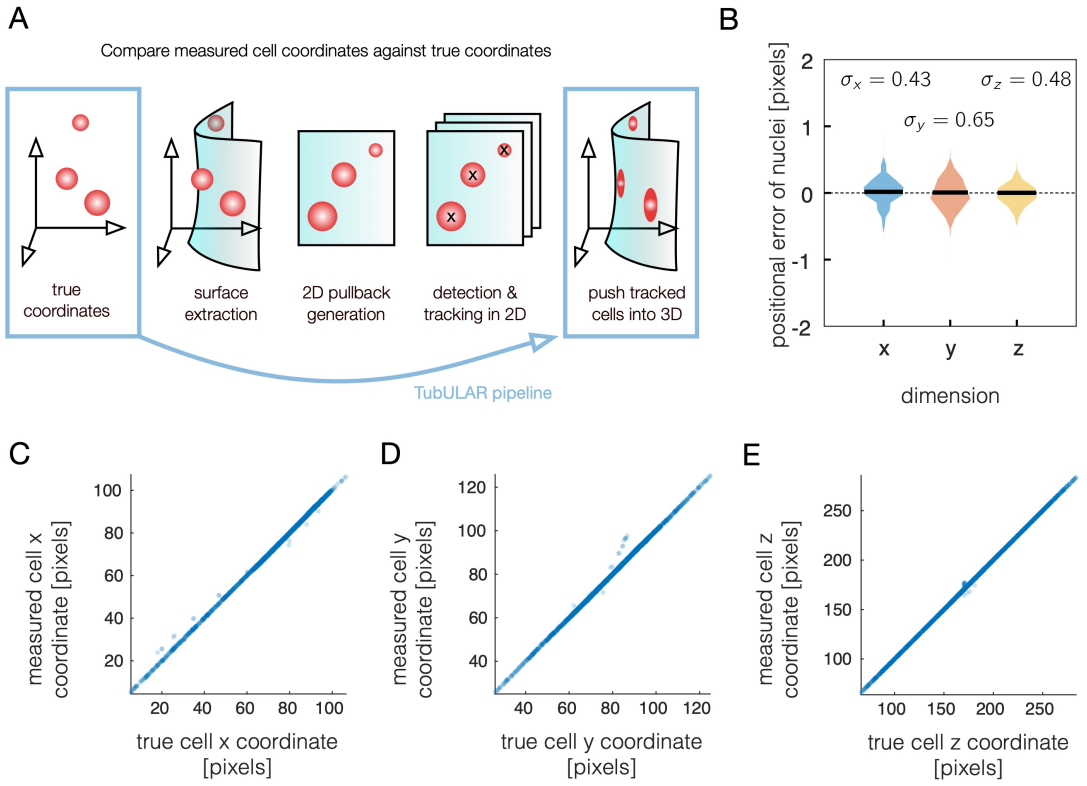

FIG. S9. **The first step of method validation compares measured nuclear positions on the extracted surface to the known input positions of nuclei in the synthetic dataset.** *(A)* Schematic representation shows nuclear positions in true coordinates, which are then intersected by a detected surface. A projection to 2D captures the intersection of the data with the surface and maps those intensity values to the plane. We detect nuclei in the plane and track them across timepoints. Pushing these trajectories into 3D on the dynamic surface provides the measured cell coordinates over time. *(B)* The measured cell coordinates differ from the true (known) coordinates by a value that is typically far less than a pixel width. Median differences for x, y, and z positions are shown by a black bar. *(C-E)* Measured cell positions across tracked cells match the true input coordinates in each spatial dimension. For each plot, each blue datapoint represents a tracked cell (nucleus) position at a single timepoint, and we include positions from all timepoints.

tance threshold on the surface from each endpoint. Other methods are permitted, such as removing the largest connected component lying within a ball of provided radius, or removing a component of the mesh lying beyond a user-specified x, y, or z plane.

VII. CONSTRAINED MAPPING TO THE PLANE FOLLOWS TISSUE MOTION

To follow tissue surfaces as they deform, we begin with an initial map at a reference timepoint that defines the material coordinates, then construct maps to minimize subsequent tissue motion in the pullback plane, and finally generate material pathlines in 3D. We denote the dynamic map from the evolving surface to a fixed 2D material coordinate system as $\varphi(t)$. This dynamic map to a fixed material coordinate system is built via a sequence of four steps: $\varphi(t) \equiv J \circ \Phi \circ Z \circ f$, each of which is detailed in this section. Briefly, $f : \mathcal{S}(\mathbf{x}) \to (u, v)$ is a conformal map of the surface $\mathcal{S}$ to the unit square (via Ricci flow or Dirichlet energy minimization). $Z : u \to s$ maps each longitudinal coordinate $u(t)$ to proper length $s(u(t))$ along the long axis of the tube-like surface. $\Phi : v \to \phi$ stabilizes motion of the tissue along the circumferential axis. Finally, $J : (s, \phi) \to (s_0, \phi_0)$ removes any residual motion of the material in the pullback plane by subtracting optical flow (obtained via particle image velocimetry). Let us turn to each component in turn.

A. Initial conformal map f

The first step is to build a conformal map to the plane. A conformal map preserves angles but changes local areas, resulting in an image of the surface with smoothly varying dilation of the tissue. We conformally flatten the 3D surface to the plane using one of two methods. The first option is Ricci flow, which results in a precisely conformal output at the cost of being slow, while the sec-

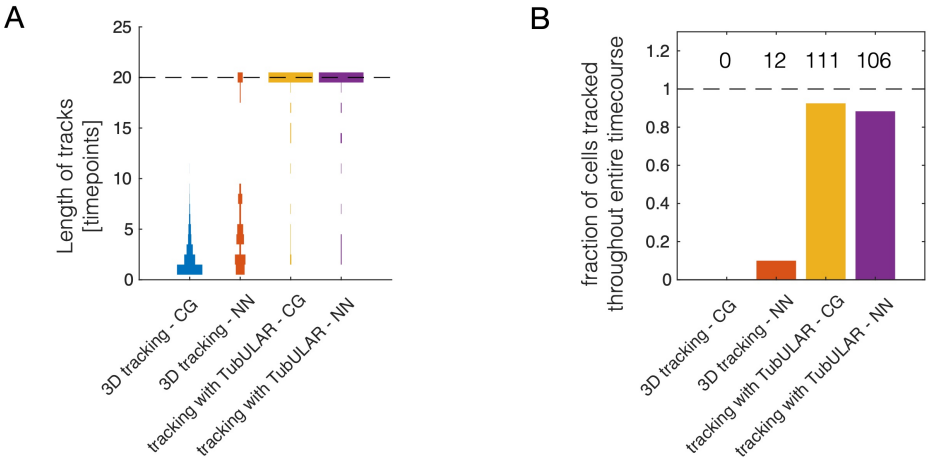

FIG. S10. **TubULAR aids automatic tracking and improves the fidelity of track trajectories.** *(A)* In a synthetic dataset of 20 timepoints, few tracks (obtained by linking 3D positions of nuclei) connected objects across more than ~ 5 timepoints. In contrast, tracking with TubULAR resulted in most automatically-computed tracks connecting nuclei across all 20 timepoints. Dashed line denotes a track duration of 20 timepoints. *(B)* Similarly, we plot the fraction of tracks that span all 20 timepoints, highlighting that while few 3D tracks connect across the full dataset, most tracks generated in the stabilized 2D pullback images and pushed into 3D span all 20 timepoints. The total number of tracks that span all 20 timepoints is printed above each bar. The dataset had a total of 120 nuclei. 'CG' denotes 'Crocker-Grier' method for particle tracking with the largest cutoff distance permitted by the method and 'NN' denotes 'Nearest-Neighbor' method, in which each identified nucleus in time t_i is connected only with the closest identified nucleus in time t_{i+1} in 3D space (for 3D tracking) or in the 2D $s\phi$ pullback coordinate space (for TubULAR tracking).

ond option is using a map minimizing a Dirichlet energy, which in general is faster, but produces less conformal results.

1. *Ricci flow*

Originally introduced by Hamilton in the context of geometric topology, Ricci flow is a tool that enables the design of (Riemannian) metrics with prescribed curvatures. A metric tensor $\mathbf{g}$ is a structure defined on the surface (or 'manifold') that defines distances and angles. Ricci flow deforms a metric proportionally to its intrinsic curvature, such that the curvature evolves according to a nonlinear heat diffusion process and eventually becomes constant everywhere. In the continuous setting, Ricci flow on 2D surfaces can be defined as

$$\frac{d\overline{\overline{\mathbf{g}}}(t')}{dt'} = -2(K(t') - \bar{K})\,\overline{\overline{\mathbf{g}}}(t'), \tag{8}$$

where $\overline{\overline{\mathbf{g}}}(t')$ is the metric generated by the flow at pseudotime t', $K(t')$ is the associated Gaussian curvature, and $\bar{K}$ is the target Gaussian curvature. Note that the pseudotime t' is simply a parameter defining the extent to which the geometric flow has been allowed to evolve and should not be confused with any physical time t related to the actual dynamics of the system (e.g. the time evolution of physical shape during organ morphogenesis). To create a map to the plane via Ricci flow, we set $\bar{K} = 0$; the target metric is flat. The formal solution to Eq. (8) is

$$\overline{\overline{\mathbf{g}}}(t') = e^{-2\int_0^{t'}(K(\tau)-\bar{K})\,\mathrm{d}\tau}\,\overline{\overline{\mathbf{g}}}(0) = e^{\Omega(t')}\,\overline{\overline{\mathbf{g}}}(0), \tag{9}$$

where the quantity $e^{\Omega(t')}$, called the conformal factor, is a scalar field that isotropically shrinks or dilates patches of the surface as the flow unfolds. It is apparent from Eq. (9) that surface Ricci flow is conformal, i.e. preserves angles defined by $\overline{\overline{\mathbf{g}}}(0)$. Explicitly, for two tangent vectors $\mathbf{u}$ and $\mathbf{v}$,

$$\begin{aligned}
\frac{\langle \mathbf{u},\mathbf{v}\rangle_{\overline{\overline{\mathbf{g}}}(t')}}{\sqrt{\langle \mathbf{u},\mathbf{u}\rangle_{\overline{\overline{\mathbf{g}}}(t')}\,\langle \mathbf{v},\mathbf{v}\rangle_{\overline{\overline{\mathbf{g}}}(t')}}} &= \frac{g_{ij}(t')\,u^i v^j}{\sqrt{(g_{ij}(t')\,u^i v^j)\,(g_{ij}(t')\,u^i v^j)}} \\
&= \frac{e^{\Omega(t')}\,g_{ij}(0)\,u^i v^j}{\sqrt{(e^{\Omega(t')}\,g_{ij}(0)\,u^i v^j)\,(e^{\Omega(t')}\,g_{ij}(0)\,u^i v^j)}} \\
&= \frac{g_{ij}(0)\,u^i v^j}{\sqrt{(g_{ij}(0)\,u^i v^j)\,(g_{ij}(0)\,u^i v^j)}} \\
&= \frac{\langle \mathbf{u},\mathbf{v}\rangle_{\overline{\overline{\mathbf{g}}}(0)}}{\sqrt{\langle \mathbf{u},\mathbf{u}\rangle_{\overline{\overline{\mathbf{g}}}(0)}\,\langle \mathbf{v},\mathbf{v}\rangle_{\overline{\overline{\mathbf{g}}}(0)}}}.
\end{aligned} \tag{10}$$

It has also been demonstrated that the Gaussian curvature during the flow always remains bounded [19].

In practice, we work with surfaces represented by discrete meshes rather than continuous functions. Surface Ricci flow has intuitive geometric interpretations, which directly inform the design of data structures in the discrete setting. The surface can be represented as a mesh

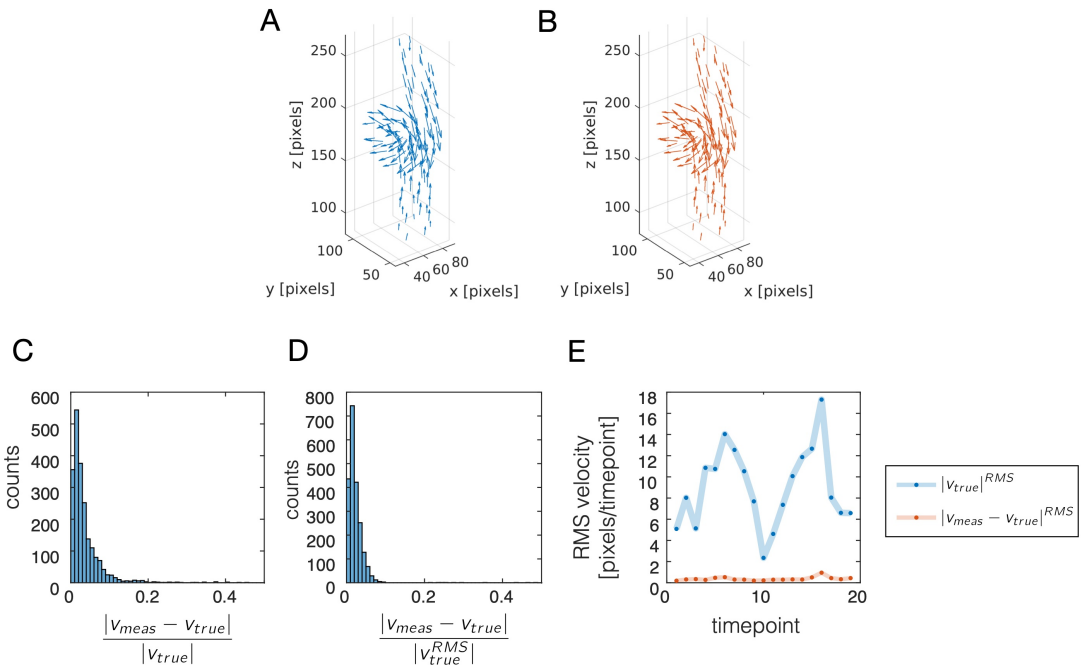

FIG. S11. **Measured velocities using TubULAR agree with true velocities of the synthetic dataset.** *(A)* An example of true cell velocities for a sample timepoint in the synthetic dataset show the difference in cell positions between the subsequent and current timepoint. *(B)* Measured cell trajectories evaluated at each tracked cell location match the true cell velocities. *(C)* The difference (normed vector difference in 3D) between measured and true cell velocities is typically less than 10% of the true velocity magnitude. The RMS fractional error of vector differences divided by the velocity across all trajectories and all timepoints is 6%. *(D)* The difference (normed vector difference in 3D) between measured and true cell velocities is typically less than 5% of the RMS value of the true speed. The RMS fractional error of vector differences divided by the RMS velocity across all trajectories and all timepoints is 4.1%. Here $|v_{\text{true}}^{\text{RMS}}| = \sqrt{\langle |v_{\text{true}}|^2 \rangle}$, where the average is taken over all trajectories and all timepoints. *(E)* The RMS error in measured velocities (orange curve) is far smaller than the true RMS velocity (blue curve) for any given timepoint. Here, $|v_{\text{true}}|^{\text{RMS}} = \sqrt{\langle |v_{\text{true}}|^2 \rangle}$, where the average is taken over all trajectories within a given timepoint.

triangulation, and the metric tensor can be represented as a set of positive edge lengths on this triangulation satisfying the triangle inequality. To then flow the discrete mesh triangulation to the plane, we follow a method reviewed in [19]. Briefly, we reformulate discrete surface Ricci flow as a convex optimization problem over the space of discrete metrics, which has a unique minimum and can be solved efficiently using Newton's method. Intuitively, given an initial metric, the method first constructs a 'circle-packing metric,' meaning it instantiates a set of circles living at mesh vertices with consistent conditions on how the circles intersect. Each variational step in the minimization process flows the discrete metric closer to the target curvature (zero in our case) in a way that always maintains the integrity of the circle packing. The conformality of the discrete flow is ensured by construction since it always maps (discrete) infinitesimal circles into (discrete) infinitesimal circles.

An implementation of the discrete Ricci flow is included as a custom package within TubULAR and also as a fully independent, standalone package called `RicciFlow_MATLAB`. This method is slower but more precisely conformal than the second method we offer, discussed next. Note also that this method fixes the u position of the inner and outer boundary, but fixes the v location of only one point shown as a black circle in Extended Data Fig. 1.

2. *Annular orbifold map*

The annular orbifold map is a second option for obtaining f that is typically faster than Ricci flow but less precisely conformal. This approach follows the formalism of Aigerman & Lipman [20], but applies these ideas to a topological cylinder mapped to a rectangle that is periodic along one dimension. Essentially, we produce a parameterization that minimizes the Dirichlet energy, enforcing certain boundary conditions so that the resulting parameterization is both seamless and globally bijective.

For a parameterization of the surface $\mathbf{u} = (u, v)$, the Dirichlet energy is defined as

$$E_D = \frac{1}{2} \int ||\nabla \mathbf{u}||^2 \mathrm{d}A, \tag{11}$$

where ∇ is a gradient operator defined on the surface

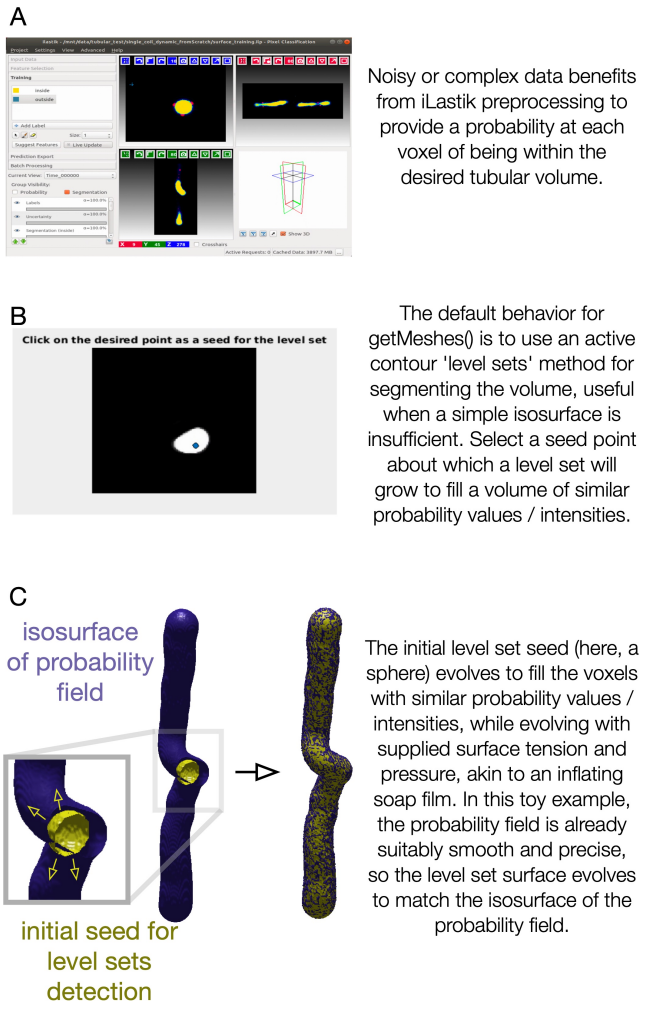

FIG. S12. **Surface extraction using level sets aids in capturing noisy or complex geometries.** *(A)* In this toy example, we segment a looped tube by first training in an iLastik workflow to create a probability field recognizing inside (yellow denotes high probability) versus outside. *(B)* In the default workflow, TubULAR generates meshes via an 'active contour' (level sets) approach. To initialize a 3D sphere to seed the growth of the estimated surface, the user can click on a location in a 2D cross-section that lies within the surface. *(C)* The resulting seed grows to fill the volume. In this toy example, the isosurface (purple) already well approximates the target surface. The initial seed (yellow sphere shown in cutaway) grows to fill the volume enclosed by the isosurface because those voxels contain similar (high) probability from the step shown in panel (A).

in 3D. This energy penalizes ruggedness in $\mathbf{u}$ – that is, ruggedness with respect to position on the surface in 3D space. As a result, its minimizers are smooth, well-behaved parameterizations. Extrema of this energy functional can be found according to a variational principle by solving the associated Euler-Lagrange equation:

$$\nabla^2 \mathbf{u} = 0. \tag{12}$$

This is the Laplace equation. The associated solutions are called *harmonic maps*. In practice, since $\mathbf{u}$ is unknown, the operator ∇^2 is taken to be the Laplace-Beltrami operator defined on the input 3D surface. Additionally, we demand that the anterior and posterior endcaps of the cylindrical mesh are mapped to vertical lines $u = 0$ and $u = 1$. We also enforce the 'orbifold' condition that ensures the v direction has the same periodicity regardless of u position. Conceptually, for the 'cut path' shown as the purple dashed curve in Extended Data Fig. 1, each point on the path is represented by two identified vertices in the flat uv plane – one on the 'upper' boundary and one on the 'lower' boundary in the uv plane. The pair of vertices representing a given point on the cut path must map to the same point in the plane, modulo a vertical offset corresponding to the periodic spacing along the azimuthal direction of the tube. These boundary constraints ensure that the planar parameterization retains the cylindrical topology of the 3D surface under proper tiling of the plane. This entire problem, including the constraints, can be cast as a single linear system of the form

$$A\mathbf{u} = b. \tag{13}$$

Any conformal parameterization of the surface will be a minimizer of the Dirichlet energy. To see this, let us specialize for a moment to the case of planar transformations $f : \mathbb{C} \to \mathbb{C}$. Let $z = x + i\,y$ denote a complex coordinate defined over the input domain being parameterized (the analogue of the input 3D surface) and let $f = u + i\,v$ be a complex representation of the parameterization. A conformal parameterization must satisfy the Cauchy-Riemann condition

$$\partial_{\bar{z}} f = 0. \tag{14}$$

It is therefore clear that any conformal mapping must also be harmonic since $\nabla^2 f = 4\partial_z \partial_{\bar{z}} f = 0$. Similar generalized arguments prove the harmonicity of 2D conformal parameterizations of 3D surfaces. In practice, the boundary conditions we enforce preclude the possibility of a truly conformal mapping. However, the results are generally a good approximation of a discrete conformal mapping, especially in the bulk away from the mesh boundaries. The high quality of Dirichlet energy minima as approximations to discrete conformal mappings has also been observed in a variety of settings [20]. We note that the conformality of the parameterizations can be improved by replacing the Dirichlet energy with the so-called *conformal energy*, $E_C = E_D - A(\mathbf{u})$, where $A(\mathbf{u})$ denotes the area of the domain of parameterization [21], though such methods are not currently implemented in TubULAR. This improvement in conformality comes at the expense of increased computational cost and usually do not preserve angles as well the conformal Ricci maps anyway.

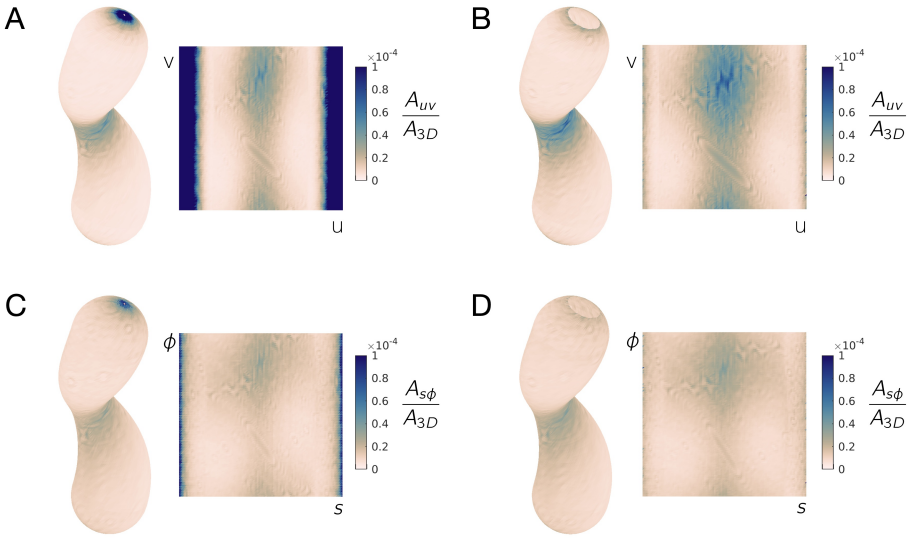

FIG. S13. **Removing endcaps from the tube facilitates mapping to the plane, and larger endcaps can lead to smaller areal distortion in the mapping at the expense of a less complete surface parameterization.** To map a tube-like object to the plane, we first remove small endcaps, defined on either end as regions within a given distance from specified endpoints. This provides two non-periodic boundaries which are mapped to $u = 0$ and $u = 1$. The size of the removed endcap can affect the distortion of the image in the map. This can be a problem if the distortion is so large as to affect the quality of cell segmentation or the fidelity of velocimetry measurements in the 2D pullback projections. *(A)* Removing a small portion of the tube at each end leads to large distortions near the endcaps. The distortion map is shown in color both on the 3D surface and on the 2D uv conformal parameterization. A_{uv} denotes the area of a given triangle in the discrete mesh triangulation in the uv plane, while A_{3D} is the area of the corresponding triangle in the embedding space. *(B)* Removing a larger portion of the tube at each end leads to lower distortions near the endcaps for this example surface in the uv parameterization. *(C)* Similarly, in the $s\phi$ parameterization, the example surface shows largest area distortion near the endcaps. $A_{s\phi}$ denotes the area of a given triangle of the mesh in the $s\phi$ pullback parameterization. *(D)* Increasing the size of endcap likewise reduces areal distortion in the mapping.

3. *Independence of mapping on choice of longitudinal seam*

The position of the mapped surface coordinates should intuitively be independent of the choice of 'cut path' along the long axis of the organ for unrolling the cylinder. For the Ricci flow case, this is true by construction: we map to an annular domain (with one endcap boundary mapped to the unit circle and the other endcap boundary mapped to a circle near the origin), then take a complex logarithm of these coordinates to acquire a rectilinear representation (Extended Data Fig. 1). For the annular orbifold map, we must enforce this path independence.

Inspired by the topology-preserving orbifold mappings of spherical surfaces [20], we enforce a set of boundary conditions to ensure that the parameterization in the plane respects the cylindrical topology of the 3D surface. Namely, we demand continuity in the boundary components of the 2D meshes associated with the virtual seam, such that, if one were to tile these meshes in the plane, moving across this boundary from one tile into another would be physically indistinguishable from crossing over the virtual seam on the 3D cylindrical surface. This mapping ensures that the edges composing the cut path takes on a unique shape in the domain of parameterization, and the shape of these edges would be identical whether or not the edges are components of the cut path.

There is one caveat to this independence: the cut path will change the output pullback map if it winds around the tube relative to the centerline with a different winding number (Fig. S14). Therefore, we take measures to enforce this topological constraint. For the first timepoint mapped to the plane (t_0), we choose a cut path to be the geodesic connecting the two endcaps. For subsequent timepoints (and previous timepoints, if any), we ensure the winding number does not change relative to this path's winding (Fig. S14).

While several options are available for enforcing this constraint, the default behavior in TubULAR uses an approximation to the centerline and measures the winding of the cut path around this centerline curve. The details of this centerline construction are not particularly important, since the curve's purpose is to provide a topological constraint – not a geometric one – on the cut path chosen, whose own geometry is immaterial so long as it does not wind around the centerline. Nevertheless, we give a brief description here: for each timepoint, we measure an approximate centerline via fast marching (see Section VIII). We connect the endpoints by a curve that spans the interior of the mesh found by minimizing the 'time of travel' with a speed of travel through any given

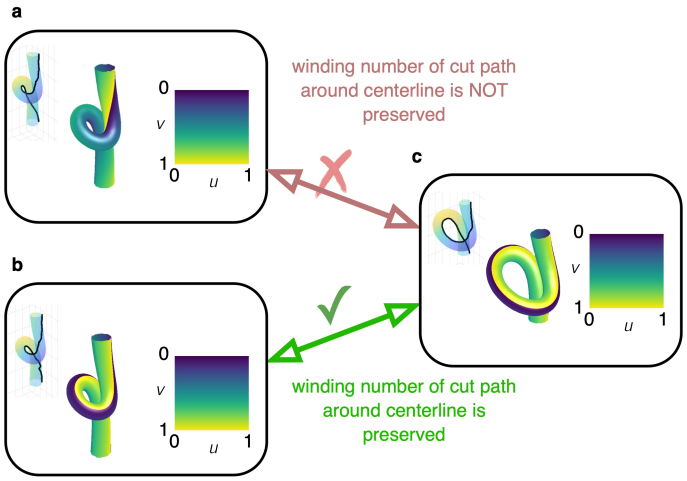

FIG. S14. **Our parameterization scheme is independent of the choice of longitudinal seam (cut path), but only given the proper topological constraint on the winding of that seam.** A potential complication in a TubULAR pipeline may arise if a tube is undergoing substantial coiling. The longitudinal seam is first chosen as a geodesic connecting two endpoints (or a piecewise geodesic path for $t \neq t_0$ if complex winding of the path is detected), as shown in small insets in the top left of panels (a), (b), and (c). These longitudinal seams (or 'cut paths') need to exhibit a winding number about the centerline curve that matches the winding of the cut paths for adjacent timepoints, or else the map to the pullback plane will be sheared along the v direction. In this example for a synthetic dataset, the reference timepoint is given in panel (c). There, a geodesic defines the phase v around the tube in the (u, v) plane. In panel (a), a geodesic path for a different timepoint does not satisfy the same winding around the centerline. The default behavior in TubULAR is then to seek a different path that does match the topology of the adjacent timepoint. In panel (b), a piecewise geodesic path, which appears subtly different but winds differently around the centerline, does enable a conformal map with the same topology as that of panel (c).

voxel determined by the distance transform of the segmented data volume [22]. If a geodesic path is found to change its winding from one timepoint to the next, the path is perturbed to more closely match the previous one in space until the topology is preserved. We found this 'trial and error' approach to be far faster than constructing an explicitly-topologically-equivalent curve (for example, after Ricci flow to an annular domain). The relevant TubULAR methods provide options for choosing different approaches if needed, including the explicit construction method.

B. Quasiconformal map $\Phi \circ Z$ to (s, ϕ) coordinates

We then introduce a further coordinate transformation which we found aids in surface stabilization. We found that directly measuring optical flow in (u, v) coordinates is impeded by large jitter in the mapped tissue image from timepoint to timepoint. Furthermore, the potentially large variations in dilation across the conformal map leads to nonuniform sampling of tissue motion across the embedding surface. For example, mapping a tube with a constriction will lead to large dilation of the constricted region, with the rest of the tube compressed to smaller areas in the uv plane. We therefore built a more constrained approach to remove motion, which we found to aid in subsequent refinement of the tissue stabilization and more evenly sample the tissue in the pullback plane. We denote this second planar parameterization (s, ϕ), and we compare this parameterization to the (u, v) parameterization found previously in Fig. S15. We find this (s, ϕ, t) parameterization aids in both visualization and enables more accurate velocimetry measurements than other choices we considered, particularly when large variations appear in the effective radius of the surface along its long axis. This second map, which we denote $\Phi \circ Z$, is a quasi-conformal transformation (i.e. a smooth transformation with finite anisotropic distortion [23]) of the initial (u, v) coordinates.

For the reference timepoint t_0 considered first, the coordinate directions $\hat{s}$ and $\hat{\phi}$ are the same as the $\hat{u}$ and $\hat{v}$ directions from the conformal mapping to the plane. Furthermore, at this initial timepoint, $\phi = v$ is identical to the intrinsic circumferential axis of the conformal map. The sole difference is that s parameterizes a longitudinal position along the long axis of the organ at $t = t_0$. In particular, we compute s as the average geodesic length along the surface from the anterior endcap to a set of uniformly-sampled points with fixed horizontal coordinate in the conformal pullback space (Fig. 2A-C of the main text). Intuitively then, s this is the average path length required to travel on the surface from the anterior face to a given location along curves of constant ϕ.

In more detail, we define the (s, ϕ) domain of parameterization as follows. The $s\phi$ parameterization is less conformal but which more equally represents different patches of tissue that initially experience different dila-

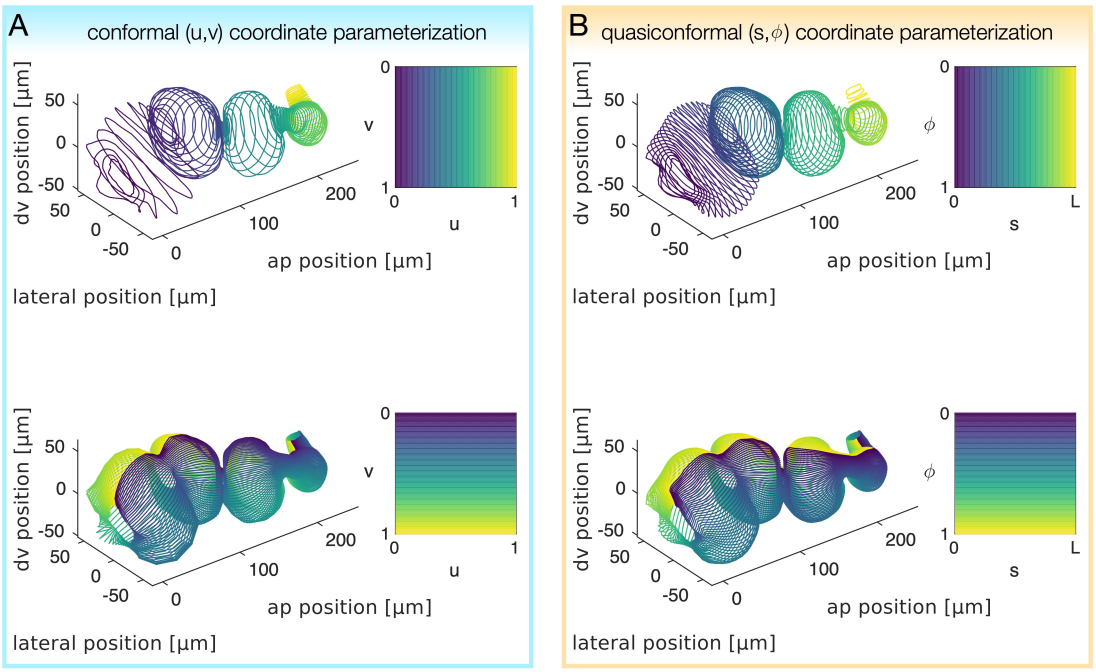

FIG. S15. **A quasiconformal mapping from (u, v) to (s, ϕ) aids in spatially uniform surface sampling and velocimetry for refined tissue tracking.** *(A)* The constrained conformal map to the plane at a time 90 minutes after the onset of constrictions ($t = 90$ min) demonstrates large variations in the sampling density along the longitudinal direction (top). Note the high density of circumferential hoops near constrictions and near the posterior tip. *(B)* After creating the conformal mapping to (u, v), the TubULAR workflow constructs an second map to a parameterization in which the surface is more evenly sampled along the longitudinal axis and in which motion in the v direction is subtracted off. Note the more uniform longitudinal density of circumferential hoops (top) and the difference in the curve $\phi = 0$ at this later timepoint, which matches the previous position of the material along $\phi = 0$ at earlier timepoints. At time $t = t_0 = 0$ min, $\phi = v$, while at this later timepoint, $\phi \neq v$.

tion in the map f from 3D embedding $\mathbf{x}$ to 2D pullback $\mathbf{u} = (u, v)$. This empirically improves measurements of tissue velocity in plane for our shapes, and we expect the additional transformation will improve other tissues that are elongated in quasi-axisymmetric geometries. Circumferential 'hoops' of tissue surrounding the centerline that are equally sampled *along* the centerline will be equally spaced in the pullback coordinates (Fig. 2C of the main text and Fig. S15). The map $\Phi \circ Z$ from the previous (conformal) frame (u, v) to the (s, ϕ) coordinate system is defined by

$$s(u) = \int_0^u \left\langle \frac{\mathrm{d}s(u', v)}{\mathrm{d}u'} \right\rangle_v \mathrm{d}u', \tag{15}$$

and

$$\phi(u, v) = v - \phi_0(u). \tag{16}$$

Equation 15 ensures that circumferential hoops sampled at equally spaced distances (as measured by their average proper distance) along the longitudinal axis are equally spaced in pullback space.

Equation 16 removes tissue motion along ϕ at each longitudinal position s. The form of $\phi_0(u)$ is such that motion of the tissue along each hoop is cancelled out in the pullback to acheive a more Lagrangian parameterization. For $t = t_0$, $\phi_0(u) = 0$ and $(s, \phi) = (s, v)$ defines the material coordinate frame. For other timepoints, $\phi_0(u)$ is chosen to minimize the difference in positions of material points at the current timepoint relative to the previous (next) timepoint for $t > t_0$ ($t < t_0$). To minimize the difference in material positions, we consider each sampled value of s in turn, which represents a circumferential 'hoop' in 3D (see Fig. S15). We numerically minimize the sum of squared Euclidean distances of uniformly-sampled points along this circumferential hoop from the mapped 3D locations at a previously-solved timepoint closer to t_0. These hoops can be visualized in 2D as vertical strips in the (s, ϕ) pullback coordinates.

If the tissue already appears relatively stationary in the uv parameterization, then accumulated errors from repeated optical flow measurements between images of the tissue in the uv plane may be sufficiently small. In this case, the user may toggle the option for stabilization method so that $\phi_0(u)$ is defined by maximizing the cross correlation between the intensity data lying within each circumferential hoop ($s_i < s < s_j$) and the intensity data in the previous pullback lying near the same u coordinate. This option uses phase correlation of the pullback image itself and compares each strip in pullback space (corresponding to a hoop in 3D data space) to the previous image. If optical flow is an unreliable measure in this step, we have found that shifting these slices along

ϕ (which is a periodic dimension) to minimize the difference in 3D space defines $\phi_0(u)$ gives satisfactory results, and further processing in the next stages of the pipeline remove any residual material motion in the (s, ϕ) plane.

C. Refined Lagrangian parameterization of the surface, $\varphi = J \circ \Phi \circ Z \circ f$

After the previous constrained parameterization into the plane via $\Phi \circ Z \circ f$, which minimizes much of the tissue motion in the parameterization plane, we accomplish further refinement via computing pathlines in the domain of parameterization using particle image velocimetry (PIV) [24]. Advecting mesh vertices along these pathlines in the 2D pullback space, then pushing these 2D coordinates into 3D along $\varphi^{-1}(t)$ provides the positions of material points as they deform. This provides the surface shown in Fig. 1D of the main text, for instance. Pullback images in the material coordinates defined from the refined Lagrangian parameterization then show minimal tissue motion, as highlighted in Extended Data Fig. 2.

VIII. AUXILIARY GEOMETRIC DESCRIPTORS OF SURFACE DYNAMICS: CENTERLINE AND EFFECTIVE RADII

In addition to following tissue motion, our parameterization scheme naturally produces a unique route to compute a useful version of the organ's centerline. This definition of centerline has several advantages over other definitions, including the built-in association of each centerline point with a set of points on the organ surface (a set of mesh vertices) that span a circumferential 'hoop'. Because of this association, the notion of an 'effective radius' at each point along the centerline naturally follows, as shown in Fig. S16.

A. Constrained parameterization defines a system-spanning centerline of the surface

Centerline construction leverages the surface parameterization in 3D space already created from the previous step ($\varphi = J \circ \Phi \circ Z \circ f$). Hoops for which $s =$ constant define an effective circumferential hoop for increments along the length of the organ, and the average 3D position of each circumferential hoop defines its centerline point. Connecting mean points of adjacent hoops along the length of the organ defines a centerline of the object.

This construction offers several advantages to previous methods of centerline construction, as illustrated in Fig. S16. Homotopic thinning techniques lead to a curve that does not typically span the entire tube, and also results in branches where the surface features roughness or protrusions (Fig. S16A). Fast-marching based methods (used earlier to measure a crude approximation to the centerline) can deviate from the 'center' of a material cross-section if the tube has bulges or pinched geometry such as in Fig. S16B. Neither homotopic thinning nor fast marching immediately provide an association between surface points and points along the centerline. In contrast, our method provides a single (branch-free) system-spanning curve with built-in association between surface points and centerline points.

B. Constrained parameterization defines an effective radius along the surface

We define an effective radius as the average distance from each point in a uniform sampling of a ring of constant s (computed via the mapping to the pullback plane) to the centerline, which is composed of the mean positions of all circumferential rings. We note that this measurement of effective radius could be done on the (u, v) coordinate parameterization and would give identical results, since curves of constant u are also curves of constant s. The effective radii $r(s)$ are therefore equal to those indexed by u: $r(s(u)) = r(u)$.

For the midgut dataset, we used this feature to define t_0, which we set to be the onset of constrictions. We identified constriction locations as rings of constant s whose effective radii $r(s)$ are local minima (so that $\partial_s r(s) = 0$). The first timepoint exhibiting such a minimum defined t_0.

For the zebrafish heart, we used this feature in Fig. 4D of the main text.

IX. EXAMPLE OF INFERRING INTERCALATION RATES ('TISSUE TECTONICS') USING TUBULAR

Tissues can change shape by several mechanisms. Cells area changes, cell shape changes, cell rearrangements (intercalations), and cell divisions can all contribute [26]. Our approach aids in measuring these contributions. We demonstrate this by resolving tissue-scale convergent extension into cell shape change and oriented cell intercalation contributions in the gut, which exhibits no cell divisions. In convergent extension, different axes of the tissue converge and extend. An example of tracked cells demonstrating these both cell shape changes and cell rearrangements is shown in Extended Data Fig. 3. Note that each cell changes neighbors (through cell intercalations) and also changes its shape in a way to extend along the longitudinal (horizontal) axis and converge along the circumferential (vertical) axis.

We can separate contributions of cell shape change from contributions of cell rearrangements using our Lagrangian parameterization, as we did in [1]. First note that by imprinting the cell segmentation at time $t_0 = 0$ hr onto the surface and advecting the polygonal segmentation along material pathlines in 3D, we measure the

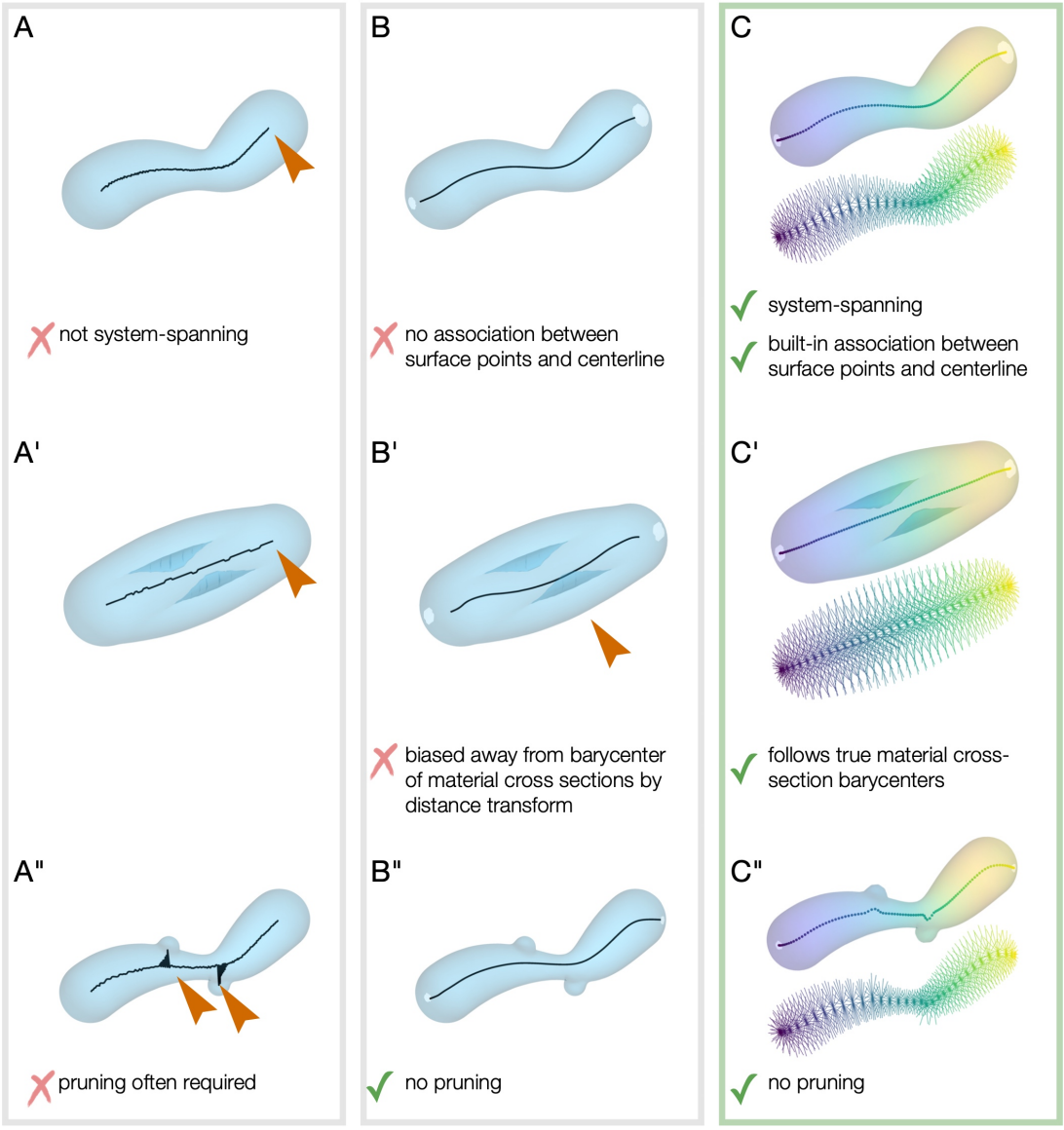

FIG. S16. **Our method provides a geometric route to centerline extraction which offers several benefits over traditional approaches.** *(A-A")* An approximate centerline can be built from homotopic thinning methods [25]. This simple method has several downsides, including that the curves do not span the whole system, and pruning of the curve is needed when the surface is not sufficiently smooth. Associating the mesh surface and the curve poses conceptual challenges as well. *(B-B")* An alternative – which is implemented in an auxiliary step for constraining surface parameterization in TubULAR – minimizes the 'time of travel' within the segmented volume from selected endpoints, where the speed of travel through a given voxel is weighted by its signed distance from the mesh surface [22]. While this offers system-spanning centerline curves, the curves deviate from the center of the object when the surface is puckered, as in panel B'. Additionally, no association between the mesh surface the the curve is given by this method. For contorted tubes such as the fly midgut, nearest distance matching gives spurious associations. *(C-C")* Constructing centerlines based on the coordinate parameterization to the material frame offers advantages for finding a single curve without branching with explicit associations between surface points and the centerline. This construction enables examination of the constriction cross-sections in the main text.

effect of tissue-scale motions on the advected imprinted virtual cells. This provides a measurement of virtual cell deformations at each point where a cell resided at t_0. Note that the material coodinates are constructed such that cell rearrangements that produce no net motion stay in place (see Section VII 'Constrained Mapping to the Plane Follows Tissue Motion'). Advection our virtual, imprinted cells is therefore given by the time-dependent pushforward of material coordinates of each virtual cell boundary. We can then directly compare the tissue-scale shear to the actual shape change of cells segmented at different timepoints. Without explicitly tracking cells,

the difference between the tissue-scale shear and the cell shape change is attributed to net oriented cell intercalations. Extended Data Fig. 3C shows the decomposition of tissue shear into cell shape change and inferred intercalations using this method.

Note that this method measures *net* intercalations. Therefore, one T1 event forming a new cell-cell junction aligned with the longitudinal direction can be canceled by a T1 event forming a new cell-cell junction aligned with the circumferential direction. Indeed, in the midgut many more cell intercalations occur than are measured by the net difference [1].

X. SURFACE VELOCITIES AND DISCRETE EXTERIOR CALCULUS

3D velocity vectors arise naturally from our approach via mapping the endpoints of 2D PIV vectors into their respective 3D surfaces. Geometrically, displacement vectors $\mathbf{v}$ extend from one coordinate $\mathbf{x}_0$ in 3D on the surface at time t_0 to a different coordinate $\mathbf{x}_1$ on the deformed surface at time t_1. When t_0 and t_1 are adjacent timepoints, this defines the 3D tissue velocity at t_0 as $\mathbf{v} = (\mathbf{x}_1 - \mathbf{x}_0)/(t_1 - t_0)$. This is shown in Fig. 1B of the main text. We then decompose the velocity into a component tangential to the surface $\mathbf{v}_\parallel$ and a normal component v_n.

A. Analyzing tangential velocity fields with discrete exterior calculus

The tangential velocity fields $\mathbf{v}_\parallel$ can then be further analyzed using our implementation of discrete exterior calculus (DEC). DEC discretizes the methods of exterior calculus in the continuous setting for application on simplicial complexes such as mesh triangulations [27]. DEC is built using a straightforward set of discrete differential forms, defined on mesh vertices, edges, and faces. On a (curved) 2D surface, the only such forms are 0-forms (scalars), 1-forms (analogues of vector fields), and 2-forms (oriented areas). The DEC also defines representations of the exterior derivative d and the Hodge star $\star$ in terms of simple linear operations. These elemental operations are maps between the different spaces of k-forms on the mesh ($k \in \{0, 1, 2\}$). Roughly, the exterior derivative d acting on a k-form ω yields a $(k+1)$-form $\beta = d\omega$ that encapsulates how rapidly ω changes in every possible direction. By definition, successive application of the exterior derivative $d^2\omega = 0$ for all k-forms ω. Note that this is important for the Helmholtz-Hodge decomposition discussed later. Since there are no 3-forms on a surface, $d\beta = 0$ for any 2-form β. The Hodge star $\star$ acting on a k-form ω generates a $(2-k)$-form that is 'dual' to ω in the sense that they can be combined together to form an oriented 2D patch whose area is the "size" of ω. This technology can be exploited for a wide variety of applications in discrete geometry processing. In particular, it allows us to easily compute gradients of the velocity field (or other vector/tensor fields) on curved surfaces.

For a given mesh, instantiating `DECLab`'s `DiscreteExteriorCalculus` class generates the elemental operators, d and $\star$, for each possible pairing of k-form types, i.e. for a 1-form ω the operation $d\omega = \beta$ generates a 2-form β. When strung together, these operators generate familiar mathematical operations such as the divergence, curl, and Laplacian – except now these operators take care to incorporate the curvature and geometry of the triangulated surface. For completeness, we enumerate some familiar differential operations in the language of exterior calculus. Let φ denote a scalar field (0-form), let $\mathbf{v}_\parallel$ denote a tangential vector field, and let $\flat/\sharp$ denote the musical isomorphisms that transform vector fields into 1-form fields and 1-form fields into vector fields, respectively. Then, common differential operations in the language of exterior calculus are:

$$\nabla\varphi \to (d\varphi)^\sharp \tag{17a}$$

$$\nabla^2\varphi \to \star d \star d\varphi \tag{17b}$$

$$\nabla \cdot \mathbf{v}_\parallel \to \star d \star (\mathbf{v}_\parallel^\flat) \tag{17c}$$

$$\nabla \times \mathbf{v}_\parallel \to (\star d(\mathbf{v}_\parallel^\flat))^\sharp \tag{17d}$$

$$\nabla^2\mathbf{v}_\parallel = ((\star d \star d + d \star d\star)\, \mathbf{v}_\parallel^\flat)^\sharp. \tag{17e}$$

B. `DECLab` validation

Fig. S4 shows the properties and methods of this class, and the online GitHub documentation provides example usage and benchmarks for accuracy, also shown in Fig. S17 and in Fig. S18. Fig. S17 demonstrates the accuracy of derivatives of surface quantities calculated using DEC relative to analytic results on a triangulation of the unit sphere with 32,768 faces. Fig. S18 shows the accuracy of DEC derivatives as a function of mesh resolution on triangulations of the unit sphere with increasing numbers of faces. We see that triangulations with ∼1,000 faces (a relatively coarse approximation) already compute discrete derivatives in high agreement with analytic results ($\lesssim 5\%$ median fractional error). The scalar field used for this comparison was

$$S = \cos(4\phi)\cos\theta\sin\theta^4 - \cos\theta^2, \tag{18}$$

where (θ, ϕ) are the usual spherical angles and the tangential vector field was

$$\mathbf{v} = \begin{pmatrix} xz(z^2 - 1/4) - y \\ yz(z^2 - 1/4) + x \\ (x^2 + y^2)(1/4 - z^2) \end{pmatrix} \tag{19}$$

where the 3D Cartesian coordinates (x, y, z) should be interpreted as being restricted to the surface of the unit sphere.

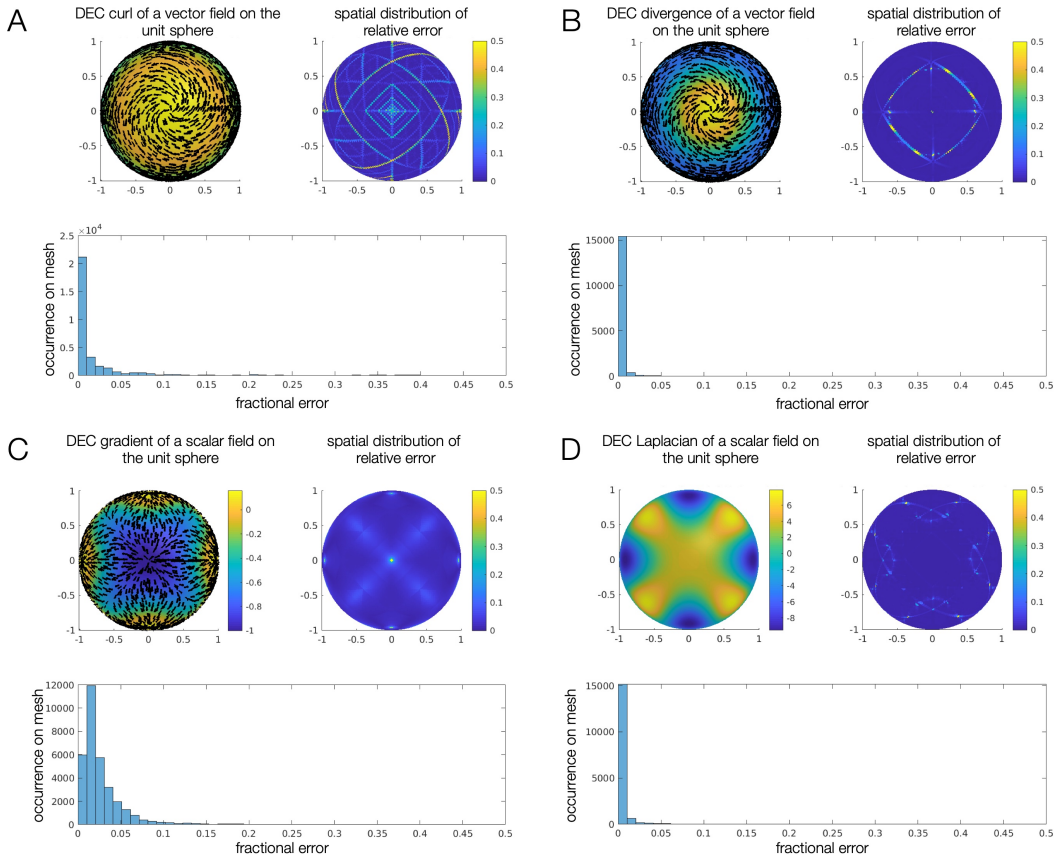

FIG. S17. **DECLab operators reproduce analytic results, with some mesh-dependent errors.** Examples of computed curl ($(\star d(\mathbf{v}^\flat))^\sharp$), divergence ($\star d \star (\mathbf{v}^\flat)$), gradient ($(d\varphi)^\sharp$), and Laplacian ($\star d \star d\varphi$) fields on the surface of a sphere show little relative error, plotted as histograms for each case. Values are compared with analytic results for the tested vector fields $\mathbf{v}$ in (A) and (B) and scalar fields ϕ in (C) and (D). Further details and example code are available on the GitHub documentation page.

Note that the implementation of the gradient operator given in Eq. (17a) is actually identical to the familiar gradient operator derived from piecewise-constant linear finite elements [28], i.e.

$$(\nabla\varphi)_T = (\varphi_j - \varphi_i)\frac{(\mathbf{x}_i - \mathbf{x}_k)^\perp}{2A_T} + (\varphi_k - \varphi_i)\frac{(\mathbf{x}_j - \mathbf{x}_k)^\perp}{2A_T}, \quad (20)$$

where φ is some scalar function defined on the counter-clockwise ordered vertices $[\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_j]$ comprising a triangle T, A_T is the area of that triangle, and the symbol $\perp$ denotes a counter-clockwise rotation by 90° in the plane of the triangle. The Laplacian operator given in Eq. (17b) is identical to the cotangent construction of the Laplace-Beltrami operator with inverse Voronoi area weights [28].

C. Helmholtz-Hodge decomposition of vector fields on dynamic surfaces

Our `DECLab` implementation includes a simple interface to generate a Helmholtz-Hodge decomposition of tangential vector fields [27]. Briefly, a Helmholtz-Hodge decomposition of a tangential velocity vector field on a surface breaks up that vector field into three physically distinct classes of motion: dilational, rotational, and harmonic. The dilational (or "curl-free") velocity exclusively mediates local area change in the absence of out-of-plane deformation. The rotational (or "divergence-free") velocity encapsulates the extent to which in-plane motion will cause a material patch to circulate around a closed, oriented loop on the surface. Any swirling or vortical motion on the surface will be captured by this component. Finally, the harmonic velocity is, by construction, "everything else." Tubular geometries can support non-trivial harmonic velocities, such as uniform flows along or around the tube.

Let v denote the 1-form field associated with the tangential surface velocity. In general, any surface velocity

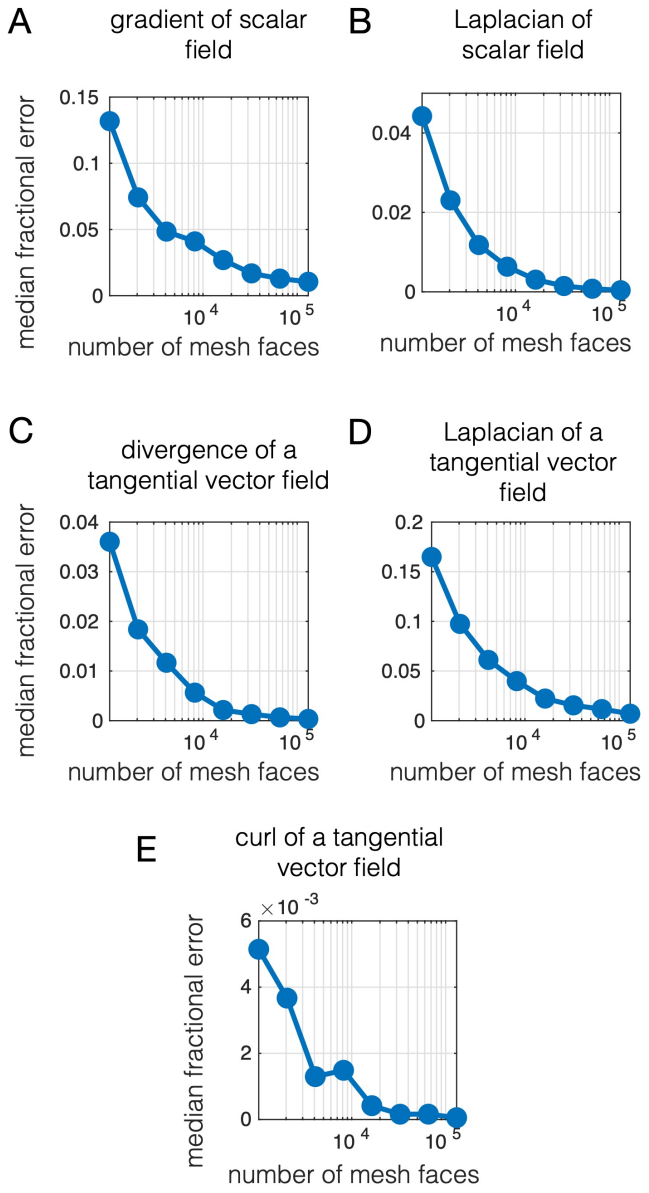

FIG. S18. **Increasing mesh resolution improves the quality of surface derivatives calculated using DEC.** We apply derivatives to scalar and vector fields defined on triangulations of the unit sphere with increasingly fine resolution (i.e. number of mesh faces). As resolution increases, median fractional error relative to analytic results diminishes. See Eq. (17) for definitions. *(A)* Error in the gradient of a scalar field relative to analytic values. *(B)* Error in the surface Laplacian of a scalar field. *(C)* Error in the divergence of a tangential vector field. *(D)* Error in the surface Laplacian of a tangential vector field. *(E)* Error in the curl of a tangential vector field.

field can be decomposed in the following way

$$v = d\varphi + \delta\beta + h, \tag{21}$$

where d is the exterior derivative, $\star$ is the Hodge star, and $\delta = \star d \star$ is the codifferential. The codifferential acting on a k-form β generates a $(k-1)$-form $\omega = \delta\beta$. To get a sense of the meaning of this operator, note that if we define an appropriate inner product on k-forms $\langle\langle u, v\rangle\rangle$ that quantifies the "overlap" between u and v (just like the familiar dot product of 3D vectors), then the overlap between $(k+1)$-forms $d\omega$ and β is equal to the overlap between k-forms ω and $\delta\beta$, i.e. $\langle\langle d\omega, \beta\rangle\rangle = \langle\langle \omega, \delta\beta\rangle\rangle$. Analogously to the exterior derivative d, one can show that $\delta^2\beta = 0$ identically. The dilatational part of the velocity field, $d\varphi$, is given by the exterior derivative of the scalar potential φ. One can show that this contribution must be curl-free by considering the form of the "curl" on surfaces given in Eq. (17d) and applying the identity $d^2\varphi = 0$. The rotational part of the velocity field, $\delta\beta$, is given by codifferential of the vector potential β (confusingly β is actually a 2-form despite the common naming convention). Similarly, one can show that this contribution must be divergence-free by considering the form of the divergence given in Eq. (17c) and applying the identity $\delta^2\beta = 0$. Finally, h is a harmonic 1-form (i.e. $(d\delta + \delta d)h = 0$) encompassing the remaining aspects of v that are neither dilatational nor rotational. By construction, there is no potential field (neither scalar nor vector) that can be associated to h – it is simply whatever is left over after subtracting away $d\varphi$ and $\delta\beta$ from v. Each term and potential function is given by the method `helmholtzHodgeDecomposition`.

D. Lagrangian measures of time-integrated tissue strain

Endowing the evolving surface with a set of Lagrangian coordinates enables the construction of a *material metric*. The metric tensor, $\overline{\overline{\mathbf{g}}}(t)$, is a geometric object enabling the measurement of distances and angles between nearby points on the surface. The rate-of-deformation tensor describes how lengths and angles change locally as the surface deforms in time:

$$\frac{dg_{ij}(t)}{dt} = \nabla_i v_{\|_j} + \nabla_j v_{\|_i} - 2\, v_n\, b_{ij}, \tag{22}$$

where $v_{\|i}$, $i \in \{1, 2\}$, and v_n denote the tangential and normal components, respectively, of the Lagrangian surface velocity, ∇_i denotes the covariant derivative with respect to the i^{th} tangential coordinate, and b_{ij} denote the components of the second fundamental form – another geometric tensor object that measures surface curvature. Essentially, Eq. (22) tells us that lengths and angles deform under the surface motion when there are gradients in the tangential velocity *and/or* when there is normal motion in curved regions of the tissue.

We can then integrate the rate-of-deformation tensor along pathlines to construct a Lagrangian measurement of cumulative tissue strain, i.e.

$$\overline{\overline{\boldsymbol{\varepsilon}}}(t) = \frac{1}{2}\int_{\tau=0}^{\tau=t} d\tau\, \frac{d\overline{\overline{\mathbf{g}}}(\tau)}{d\tau} = \frac{1}{2}\left(\overline{\overline{\mathbf{g}}}(t) - \overline{\overline{\mathbf{g}}}(t_0)\right). \tag{23}$$

In the language of geometric elasticity, this is equivalent to the Green-St. Venant strain tensor [29], defined relative to the 'undeformed' reference configuration at time $t = 0$. The strain tensor can be decomposed into a dilatational (isotropic) component

$$\frac{1}{2}\mathrm{Tr}\left[\overline{\overline{\mathbf{g}}}^{-1}(t_0)\overline{\overline{\boldsymbol{\varepsilon}}}(t)\right]\,\overline{\overline{\mathbf{g}}}(t_0), \tag{24}$$

and the deviatoric component

$$\mathrm{Dev}\left[\overline{\overline{\boldsymbol{\varepsilon}}}(t)\right] = \overline{\overline{\boldsymbol{\varepsilon}}}(t) - \mathrm{Tr}\left[\overline{\overline{\mathbf{g}}}^{-1}(t_0)\overline{\overline{\boldsymbol{\varepsilon}}}(t)\right]\overline{\overline{\mathbf{g}}}(t_0)/2. \tag{25}$$

XI. ANALYSIS OF BEATING ZEBRAFISH HEART DYNAMICS

In the absence of cell proliferation, the relationship between local tissue area rate of change, in-plane divergence, and out-of-plane motion is [30]:

$$\nabla\cdot\mathbf{v}_{\parallel} - 2Hv_n = \frac{1}{2}\mathrm{Tr}\left[\overline{\overline{\mathbf{g}}}^{-1}\dot{\overline{\overline{\mathbf{g}}}}\right], \tag{26}$$

where $\nabla\cdot\mathbf{v}_{\parallel}$ is the in-plane covariant divergence of the in-plane tissue velocities $\mathbf{v}_{\parallel}$, H is the mean curvature of the surface, v_n is the normal (out-of-plane) velocity, and $\mathrm{Tr}\left[\mathbf{g}^{-1}\dot{\mathbf{g}}\right]/2$ is the rate of local area change. We find the two terms on the left hand side are not equal, and in fact are anti-correlated. We measure their cross correlation between their circumferentially-averaged values, each of which is a function of the longitudinal coordinate, s, and of time, t: $\langle\nabla\cdot\mathbf{v}_{\parallel}\rangle_{\phi}(s,t)$ and $\langle 2Hv_n\rangle_{\phi}(s,t')$. Fixing the spatial coordinate but varying the time delay between measured values $\Delta = t - t'$ returns a sinusoidal correlation function parameterized by the time delay Δ. This curve fits well to

$$C(\Delta) \approx A\cos\left(2\pi(\Delta-\tilde{\Delta})/T\right). \tag{27}$$

The time shift corresponding to the maximum correlation – such that the in-plane and out-of-plane deformations would be in phase – is $\tilde{\Delta}$, which we report in the main text. This analysis gives insight into the kinematic properties of the tissue: tissue compressibility dominates the kinetics, prompting further modeling of the heart's mechanical cycle.

Fig. S19 shows details of the three highest PCA modes of the heart dynamics, expanding on the results presented in the main text.

XII. LIMITATIONS OF TUBULAR

While limitations are sketched in the main text, here we present a more detailed discussion on the limitations of TubULAR.

A. Topology

Our current implementation is focused on following single tubes. This means that the TubULAR functionality does not naturally handle tissue surfaces that dynamically branch, split, merge, or intersect. For instance, our methods would fail to follow portions of the tube that pinch off from the primary structure to form a separate surface. Systems with fixed, but non-cylindrical topology represent another class of systems that are not captured *in toto* using this implementation. For example, branching tubes in the circulatory system would require analyzing multiple segments independently. Handling complex networks or branched tubular structures represents a future challenge, as does handling topological changes over time.

B. Quality of volumetric intensity data

Our method is designed for 3D microscopy techniques such as confocal or light-sheet imaging with sufficient resolution to resolve features in the surface of interest. We note that some 3D imaging techniques such as (micro) CT scans may not provide sufficient texture (variation in intensity along the surface) to measure residual motion in the pullback images using PIV. For instance, a uniformly bright fluorescing tissue/surface could lead to limited information about in-plane motions. More specifically, if the (s,ϕ) parameterization output by $\Phi\circ Z\circ f$ acting on the curved surface is a featureless image, the map J cannot be determined from the data. Without J, the surface stabilization can rely only on surface geometry in 3D. Motions of cells or other objects within the surface that do not affect the 3D surface geometry would not be captured without measuring the map J (and/or measuring optical flow in the pullback plane using the `phiMethod='texture'` option) and interpreting the resulting 'velocity' fields without this step should be done with care.

C. Temporal resolution

Poor temporal resolution may result in large changes in geometry per timepoint. In such a case, mapping the surface at timepoint t_1 via the conformal map f or via $\Phi\circ Z\circ f$ may result in a $s\phi$ parameterization that is so far from the previous map of timepoint t_0 that connecting the material points between the two images may be difficult. Using landmarks to constrain the mapping to the plane for timepoints $t \neq t_0$ represents a natural extension of TubULAR's capabilities for such cases.

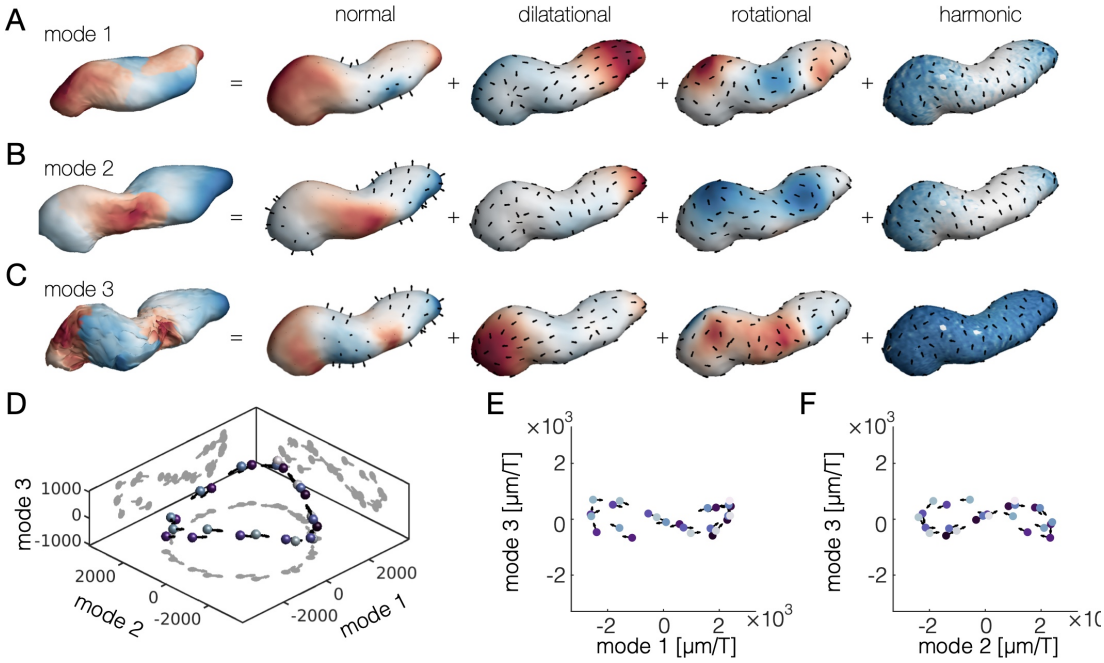

FIG. S19. **Higher order PCA modes contribute little to the description of the developing zebrafish heartbeat.** *(A-C)* Comparison of the first three PCA modes of heart beating dynamics shows that past the second mode, higher-order modes become noisy and do not capture directional pumping. *(D-F)* Projections of dynamics involving mode 3 (and higher modes) show nearly zero subtended area, unlike the dominant two modes, which sweep out a large area in mode space.

D. Extreme geometry changes

TubULAR was built to handle surfaces with exaggerated and dynamic geometries. That being said, it is worth highlighting that surfaces with extreme geometries may still pose inherent difficulties during analysis. Surfaces that are far from axisymmetric or with wild variation in their average radius along the centerline may exhibit more distortion in the 2D pullback space. This distortion can make tangential velocity extraction more challenging, especially in scenarios where the data has only coarse temporal resolution.

Another potentially problematic case may arise if the length of the centerline is far greater than (or far less than) the scale of the lateral dimensions of the tube. Such surfaces may yield parameterizations with extreme aspect ratios. Ultimately, it should be possible to successfully analyze such systems with proper care (the late-stage *Drosophila* midgut exhibits all of these extreme geometric features and has been successfully handled), however it may require non-default tuning of TubULAR parameters by the user. For example, adjustment of the aspect ratio via the `a_fixed` parameter of the TubULAR class instance can aid in appropriate scaling of the tissue images.

E. Winding

As discussed in Section VIIA3 'Independence of mapping on choice of longitudinal seam', dynamic tubes that coil (or uncoil) over time can be challenging to follow in a pullback frame due to the fact that the shortest path (geodesic) connecting a given set of anteroposterior endcap vertices may abruptly change its winding around the centerline as the coiling or uncoiling proceeds. Mathematically, this reflects the coupling between writhe and twist [31]. While we have implemented multiple approaches in TubULAR to overcome this challenge, the default behavior might not immediately track the correct windings, so that some tuning of the algorithm's parameters may be required. For example, while analyzing the synthetic dataset of Fig. S7, we chose $t_0 = 10$ to aid in preserving the winding for all previous and subsequent timepoints.

[1] N. P. Mitchell, D. J. Cislo, S. Shankar, Y. Lin, B. I. Shraiman, and S. J. Streichan, eLife **11**, e77355 (2022).
[2] K. G. Chan, S. J. Streichan, L. A. Trinh, and M. Liebling, IEEE Transactions on Computational Imaging **2**, 348 (2016).
[3] A. M. Tayar, F. Caballero, T. Anderberg, O. A. Saleh, M. Cristina Marchetti, and Z. Dogic, Nature Materials (2023), 10.1038/s41563-023-01660-8.

[4] J. Radon, BerSächsiche Akademie Der Wissenschaften **69** (1917).

[5] S. Deans, *The Radon Transform and Some of Its Applications*, Dover Books on Mathematics Series (Dover Publications, 2007).

[6] S. Biffi, R. Cerbino, F. Bomboi, E. M. Paraboschi, R. Asselta, F. Sciortino, and T. Bellini, Proceedings of the National Academy of Sciences **110**, 15633–15637 (2013).

[7] T. Sanchez, D. T. N. Chen, S. J. Decamp, M. Heymann, and Z. Dogic, Nature **491**, 431–434 (2012).

[8] S. J. Streichan, M. F. Lefebvre, N. Noll, E. F. Wieschaus, and B. I. Shraiman, eLife **7** (2018), 10.7554/elife.27454.

[9] P. Barbier de Reuille, A.-L. Routier-Kierzkowska, D. Kierzkowski, G. W. Bassel, T. Schüpbach, G. Tauriello, N. Bajpai, S. Strauss, A. Weber, A. Kiss, A. Burian, H. Hofhuis, A. Sapala, M. Lipowczan, M. B. Heimlicher, S. Robinson, E. M. Bayer, K. Basler, P. Koumoutsakos, A. H. Roeder, T. Aegerter-Wilmsen, N. Nakayama, M. Tsiantis, A. Hay, D. Kwiatkowska, I. Xenarios, C. Kuhlemeier, and R. S. Smith, eLife **4**, e05864 (2015).

[10] M. Saadaoui, D. Rocancourt, J. Roussel, F. Corson, and J. Gros, Science **367**, 453 (2020).

[11] N. Romeo, A. Hastewell, A. Mietke, and J. Dunkel, eLife **10** (2021), 10.7554/eLife.68679, 2103.08130.

[12] I. Heemskerk and S. J. Streichan, Nature Methods **12**, 1139 (2015).

[13] J. C. Crocker and D. G. Grier, Journal of Colloid and Interface Science **179**, 298 (1996).

[14] N. Otsu, IEEE Transactions on Systems, Man, and Cybernetics **9**, 62 (1979).

[15] W. E. Lorensen and H. E. Cline, in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH '87*, Vol. 21 (ACM Press, New York, New York, USA, 1987) pp. 163–169.

[16] S. J. Osher and R. Fedkiw, *Level set methods and dynamic implicit surfaces.*, Applied mathematical sciences, Vol. 153 (Springer, 2003) pp. I–XIII, 1–273.

[17] P. Marquez-Neila, L. Baumela, and L. Alvarez, IEEE Transactions on Pattern Analysis and Machine Intelligence **36**, 2 (2014).

[18] T. F. Chan and L. A. Vese, IEEE Transactions on Image Processing **10**, 266 (2001).

[19] W. Zeng and X. D. Gu, *Ricci Flow for Shape Analysis and Surface Registration*, SpringerBriefs in Mathematics (Springer New York, New York, NY, 2013).

[20] N. Aigerman and Y. Lipman, ACM Transactions on Graphics **34**, 190:1 (2015).

[21] U. Pinkall and K. Polthier, Experimental Mathematics **2** (1993), 10.1080/10586458.1993.10504266.

[22] N. D. Cornea, D. Silver, and P. Min, IEEE Transactions on visualization and computer graphics **13**, 530 (2007).

[23] L. V. Ahlfors, *Lectures on Quasiconformal Mappings*, University lecture series (American Mathematical Society, 2006).

[24] W. Thielicke and R. Sonntag, Journal of Open Research Software (2021), 10.5334/jors.334.

[25] T.-C. Lee, R. L. Kashyap, and C.-N. Chu, CVGIP: Graph. Models Image Process. **56**, 462–478 (1994).

[26] R. Etournay, M. Popović, M. Merkel, A. Nandi, C. Blasse, B. Aigouy, H. Brandl, G. Myers, G. Salbreux, F. Jülicher, and S. Eaton, eLife **4** (2015), 10.7554/elife.07090.

[27] K. Crane, F. de Goes, M. Desbrun, and P. Schröder, in *ACM SIGGRAPH 2013 courses*, SIGGRAPH '13 (ACM, New York, NY, USA, 2013).

[28] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Levy, *Polygon Mesh Processing*, Ak Peters Series (Taylor & Francis, 2010).

[29] E. Efrati, E. Sharon, and R. Kupferman, Journal of the Mechanics and Physics of Solids **57**, 762 (2009).

[30] M. Arroyo and A. Desimone, Physical Review E **79** (2009), 10.1103/physreve.79.031915.

[31] H. K. Moffatt and R. L. Ricca, Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences **439**, 411 (1992).