Laplacian Surface Editing

吴松隐
李沛卓

Fitting transformed Laplacian coordinates

- Instead of using absolute coordinates V, we would like to describe the mesh geometry using a set of differentials $\Delta = \{\delta_i\}$.
- The δ_i definition is

$$\delta_i = \mathcal{L}(\mathbf{v}_i)$$

- Define $\mathcal{L}$ with uniform weights:

$$\mathcal{L}(\mathbf{v}_i) = \mathbf{v}_i - \frac{1}{d_i} \sum_{j \in \mathcal{N}_i} \mathbf{v}_j$$

.

Laplacian Coordinates

- *Laplacian coordinates* are invarient under tranlation. Since we only care about its shape, *Laplacian coordinates* is a better way to represent the mesh.

$$\Delta = LV$$

- we notice that the rank of L is $n - 1$. So if we fix one vertex, V can be recovered from Δ by fixing one vertex and solving a linear system.

Laplacian Coordinates

- Now we use *Laplacian coordinates* Δ to fix the absolute postion of several vertices

$$\mathbf{v}_i^{'} = \mathbf{u}_i, \;\; i \in \{m, \cdots, n\}, \;\; m < n$$

- Solve for the remaining vertices $\{\mathbf{v}^{'}\}, i \in \{1, \cdots, m-1\}$
- While *Laplacian coordinates* can preserve the shape, we want to remain the *Laplacian coordinates* as mush as possible.

Error Function

- If we give the constrains $\{u_i\}$, and we want to preserve the shape, in other word, to fit the Laplacian coordinates of V' to the given Laplacians Δ.
- It has been observed that the solution behaves better if the constraints $\{u_i\}$ are satisfied in a least squares senserather than exactly.
- So we get the Error function:

$$E(V') = \sum_{i=1}^{n} \left\| \delta_i - \mathcal{L}(\mathbf{v}_i') \right\|^2 + \sum_{i=m}^{n} \left\| \mathbf{v}_i' - \mathbf{u}_i \right\|^2$$

- The rationale of fitting given Laplacian coordinates is that details of the shape are preserved, as the relative location of vertices is encoded in Δ.

Improvement

- The detail structure of the shape can be translated, but not rotated or scaled.
- The main idea is to compute an appropriate transformation T_i for each vertex i based on the eventual new configuration of vertices V'.
- Thus, $T_i(V')$ is a function of V' and we formulate the error functional as

$$E(V') = \sum_{i=1}^{n} \left\| T_i(V')\delta_i - \mathcal{L}(\mathbf{v}_i') \right\|^2 + \sum_{i=m}^{n} \left\| \mathbf{v}_i' - \mathbf{u}_i \right\|^2$$

More about T_i

- We want to minimize error function, but both T_i and $V^{'}$ are unknown.
- If the coeffcients of T_i are a linear function in $V^{'}$, question are simpler.
- A basic idea is we want our transformation don't change the shape. In other word, T_i transform v_i to $v_i^{'}$ and also need to transform v_i's neighbors to $v_i^{'}$'s neighbors. So we define T_i:

$$T_i = \arg\min_{T_i} \left(\left\| T_i v_i - v_i^{'} \right\|^2 + \sum_{j \in \mathcal{N}_i} \left\| T_i v_j - v_j^{'} \right\|^2 \right)$$

More constrain

- If we don't constrain T_i, it may lose geometry detail. Thus, T i needs to be constrained in a reasonable way.
- Reasonable operation: rotation, isotropic scales, and translations.
- Unreasonable operation: like anisotropic.
- The class of matrices representing isotropic scales and rotation can be written as: $T = s\exp(H)$, where H is a skew-symmetric matrix.
- Drawing upon several other properties of 3×3 skew matrices, one can derive the following representation of the exponential above:

$$s\exp(H) = s(\alpha I + \beta H + \gamma \mathbf{h}^T \mathbf{h})$$

More constrain

- $\mathbf{h}^T\mathbf{h}$ is quadratic, so we delete it.
- So, we use s, I, H and translation matrix to define T_i:

$$T_i = \begin{pmatrix} s & -h_3 & h_2 & t_x \\ h_3 & s & -h_1 & t_y \\ -h_2 & h_1 & s & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- This matrix is a good linear approximation for rotations with small angles.

Calculate T_i

- Let $(s_i, \mathbf{h}_i, \mathbf{t}_i)^T$ be the vector of the unknowns in T_i. Then we wish to minimize:

$$\left\| A_i(s_i, \mathbf{h}_i, \mathbf{t}_i)^T - \mathbf{b}_i \right\|^2$$

- A_i contains the positions of v_i and its neighbors and $\mathbf{b}_i$ contains the position of $v_i^{'}$ and its neighbors.

$$A_i = \begin{pmatrix} v_{k_x} & 0 & v_{k_z} & -v_{k_y} & 1 & 0 & 0 \\ v_{k_y} & -v_{k_z} & 0 & v_{k_x} & 0 & 1 & 0 \\ v_{k_z} & v_{k_y} & -v_{k_x} & 0 & 0 & 0 & 1 \\ \vdots & & & & & & \end{pmatrix}, \; k \in \{i\} \cup \mathcal{N}_i$$

- and $\mathbf{b}_i$:

$$\mathbf{b}_i = \begin{pmatrix} v_{k_x}^{'} \\ v_{k_y}^{'} \\ v_{k_z}^{'} \\ \vdots \end{pmatrix}, \; k \in \{i\} \cup \mathcal{N}_i$$

Calculate T_i

- The linear least-squares problem above is solved by:

$$(s_i, \mathbf{h}_i, \mathbf{t}_i)^T = (A_i^T A_i)^{-1} A_i^T \mathbf{b}_i$$

- T_i is a linear function in V', as required.
- Minimize error function to find V':

$$E(V') = \sum_{i=1}^{n} \left\| T_i(V')\delta_i - \mathcal{L}(\mathbf{v}_i') \right\|^2 + \sum_{i=m}^{n} \left\| \mathbf{v}_i' - \mathbf{u}_i \right\|^2$$

Adjusting T_i

There are two exceptions that require adjusting the transformations:

- As mentioned, T_i does not exactly represent the class of isotropic scales and rotations. For large angles ϕ around the axis $\mathbf{h}/\|\mathbf{h}\|$ the space is scaled along $\mathbf{h}/\|\mathbf{h}\|$ with a factor of $\cos\phi$.
- Sometimes anisotropic scaling is the wanted free-form deformation.

Adjusting T_i

The current set of transformations T_i is computed from V and V'. Then each T_i is inspected, the corresponding Laplacian coordinate δ_i is updated appropriately depending on the cases above, and the system is solved again.

- Too large angles: first apply an approximated reconstruction using the method in [*Differential coordinates for interactive mesh editing*] and then refine it with our technique.
- the $\{\delta_i\}$ are scaled by the inverse of the scale implied by the constraints.

Mesh Editing

- Interactive detail-preserving surface editing:
  - like manipulating an object made of some physical soft material

1. Define the region of interest (ROI) for editing. The ROI is defined by the closed simple loop of its boundary edges.
2. Define *handle* inside the ROI.
3. Optionally define the amount of “padding” of the ROI by *stationary anchors*.

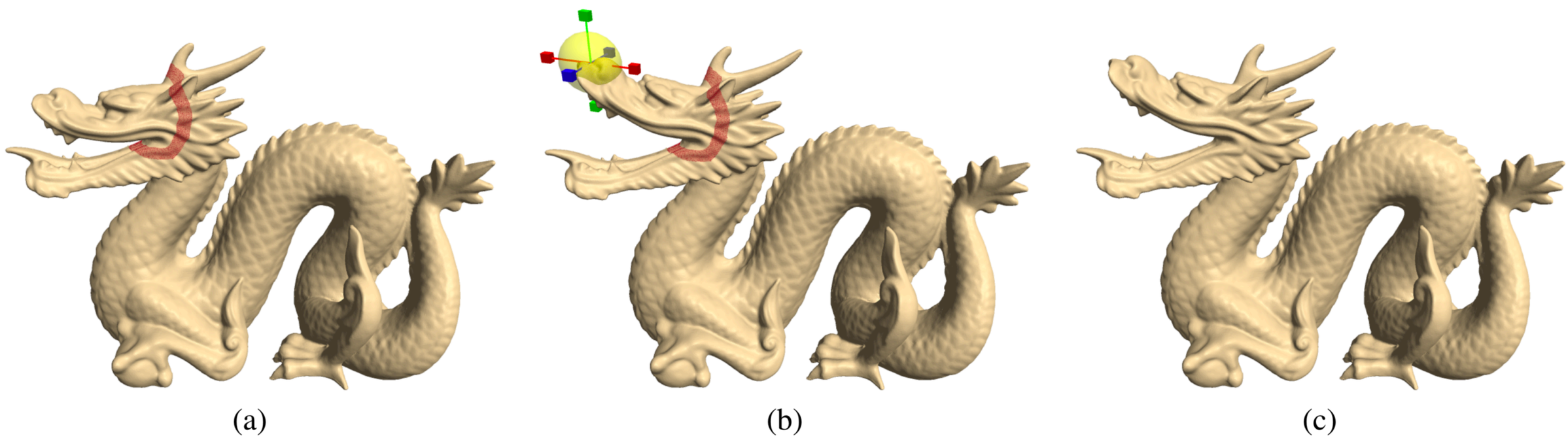


Figure 1: *The editing process. (a) The user selects the region of interest – the upper lip of the dragon, bounded by the belt of stationary anchors (in red). (b) The chosen handle (enclosed by the yellow sphere) is manipulated by the user: translated and rotated. (c) The editing result.*

Mesh Editing

- The positions of the handle vertices and the stationary anchors constrain the reconstruction and hence the shape of the resulting surface. $E(V') = \sum_{i=1}^{n} \left\| T_i(V')\delta_i - \mathcal{L}(\mathbf{v}_i') \right\|^2 + \sum_{i=m}^{n} \left\| \mathbf{v}_i' - \mathbf{u}_i \right\|^2$
- *Handle* for user control, *anchors* for transition from the ROI to the fixed part of the mesh
- Then reconstruction of the submesh by solving the linear least-squares system as described before

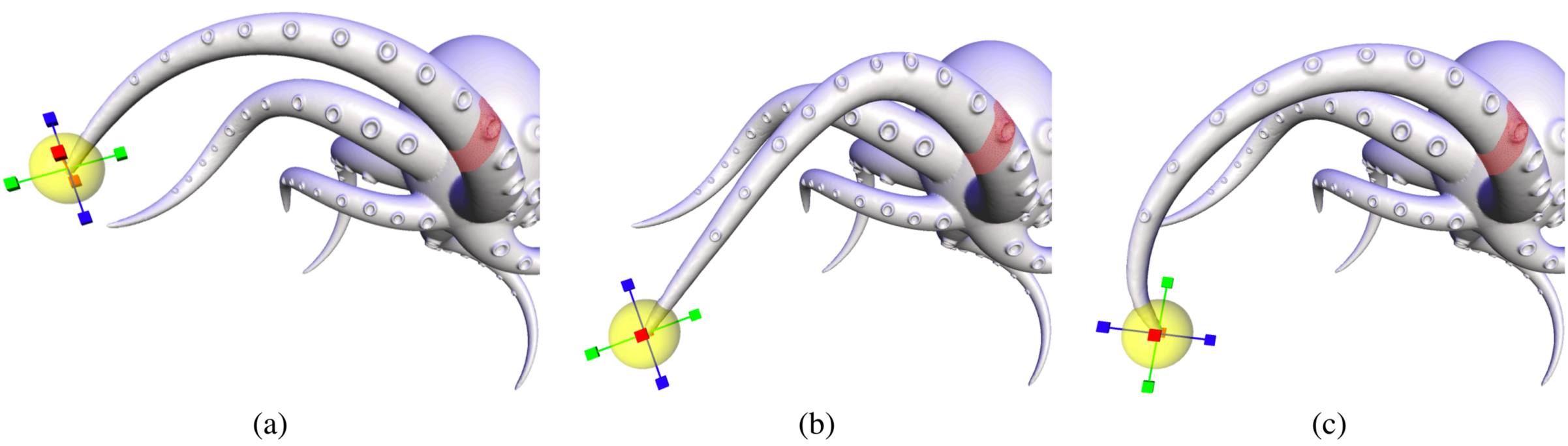


Figure 2: *Different handle manipulations. (a) The region of interest (arm), bounded by the belt of stationary anchors, and the handle. (b) Translation of the handle. (c) Subsequent handle rotation. Note that the detail is preserved in all the manipulations.*

Coating Transfer

- Peeling the coating of a *source* surface and transferring it onto a *target* surface
- *Coating* refers to the high-frequency surface details, which is defined as the difference between the original surface and a low-frequency band of the surface.

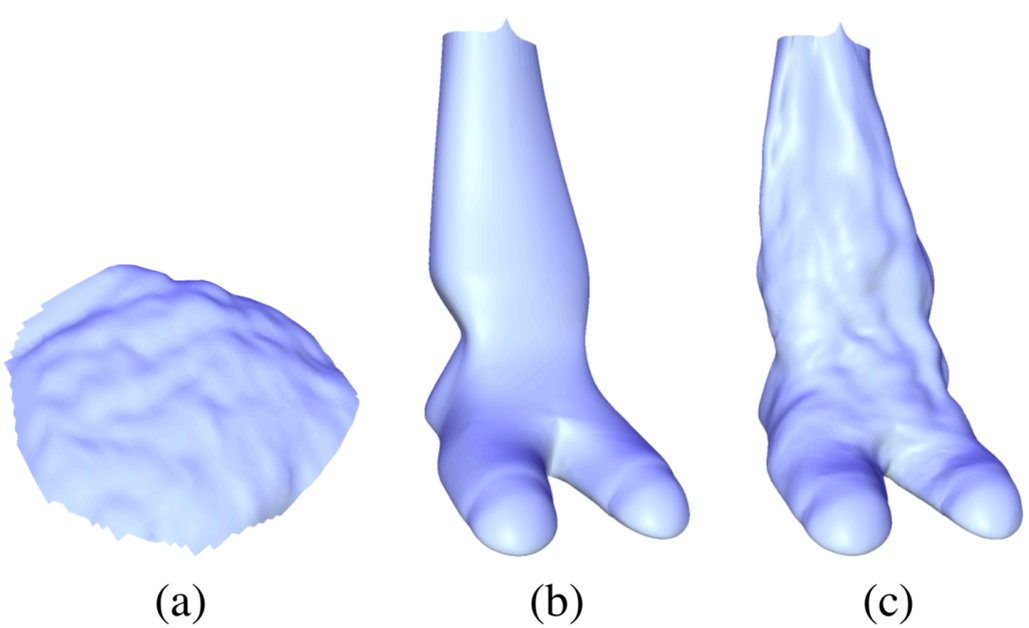


Figure 4: *Coating transfer; The coating of the Bunny (a) is transferred onto the mammal's leg (b) to yield (c).*

- Let $\tilde{S}$ be the smoothed version of S, $\delta_i, \tilde{\delta}_i$ be the Laplacian coordinates of the vertex *i* in S and $\tilde{S}$ and $\xi_i = \delta_i - \tilde{\delta}_i$ be the encoding of coating. We can easily construct S by $S = L^{-1}(\tilde{\delta} + \xi)$
- This is because the neighborhoods of the corresponding vertices *i* have the same orientation.

Coating Transfer

- Assume the source surface *S* and the target surface *U* sharing the same connectivity, but having different geometries, and that the correspondence between their vertices is given.
- The frame of vertex *i* in *S* is defined by its normal $\mathbf{n}_s$ and the normalized projection of some edge e_s emanating from *i* onto the tangent plane defined by $\mathbf{n}_s$ (the third vector is determined by the right-hand product of the first two)
- The orientation of a coating detail is defined by the local frame at the corresponding vertex in the low frequency surface
- Denote the rotated coating encoding of vertex *i* by $\xi_i' = R_i(\xi_i)$
- Now let $U' = L^{-1}(\Delta + \xi')$, where Δ denotes the Laplacian coordinates of the vertices of *U'*. Now the new surface *U'* has the coating of *U* .

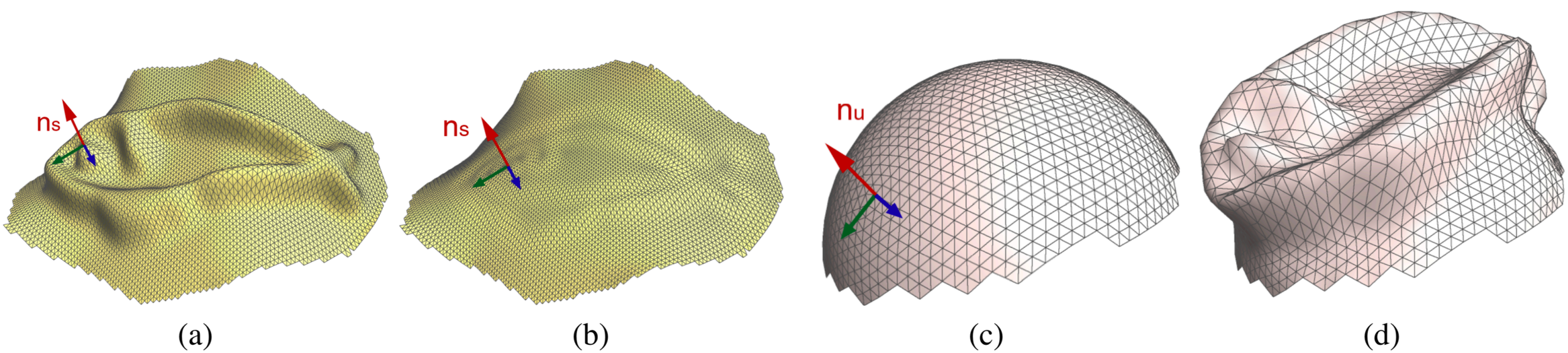


Figure 6: *Coating transfer. The orientation of a coating detail (a) is defined by the local frame at the corresponding vertex in the low frequency surface in (b). The transferred coating vector needs to be rotated to match the orientation of the corresponding point in (c) to reconstruct (d).*

Mapping and Resampling

- When *S* and *U* don't share the same connectivity, we need a mapping between them.
- Establish a mapping by parameterizing the meshes over a common domain.
- Both patches are assumed to be homeomorphic to a disk

- Fix the boundary conditions for the parameterization such that a correspondence between the source and target surfaces is achieved
- Some applications require a more careful correspondence
- The mapping between two faces should link relevant details like facial features

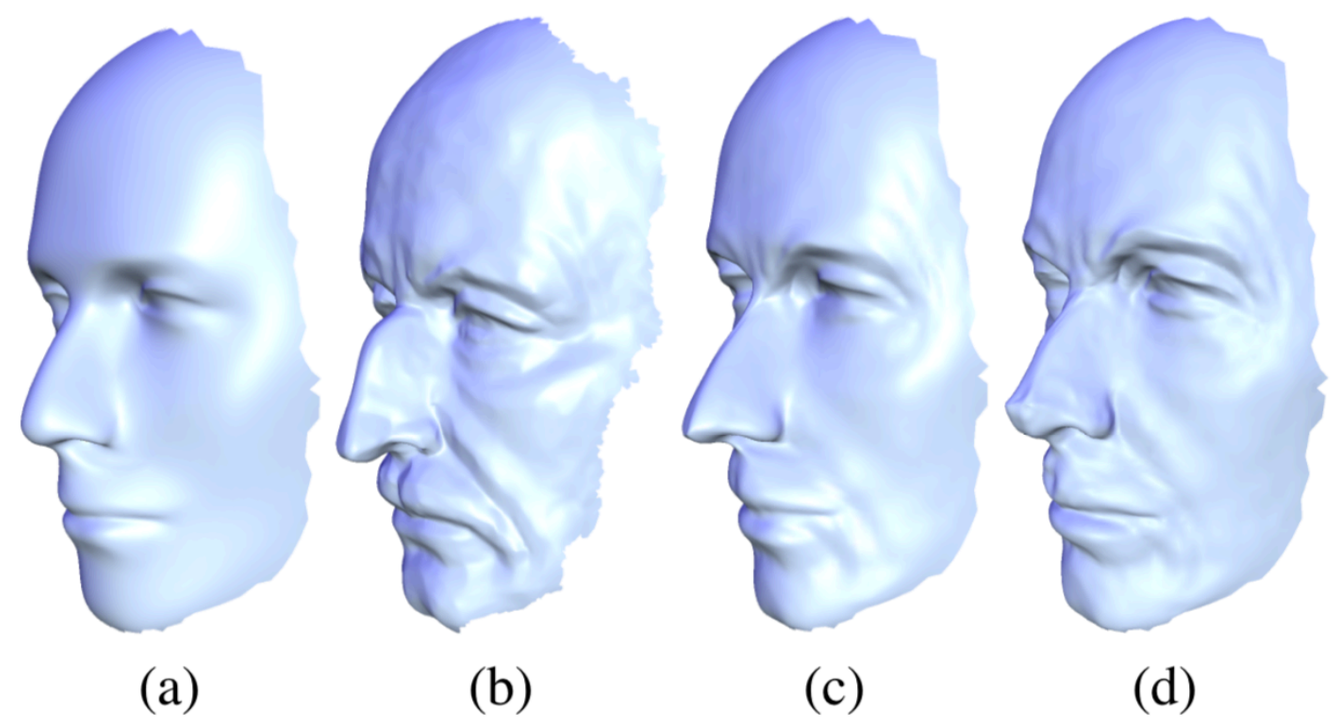


Figure 5: *The coating of the Max Planck is transferred onto the Mannequin. Different levels of smoothing were applied to the Max Planck model to peel the coating, yielding the results in (c) and (d).*

Mapping and Resampling

- In general, a vertex $i \in U$ is mapped to some arbitrary point inside a triangle $\tau \in S$.
- The best results are obtained by first mapping the 1-ring of i onto S using the parameterization, and then computing the Laplacian from this mapped 1-ring.

- Linear interpolation of the three Laplacian coordinates sampled at the vertices of the triangle τ is also ok
- To enable faithful resampling of the Laplacian coordinates, the tessellations of the surfaces U and S need to be "compatible"
- After the mapping between U and S has been established and the Laplacians have been sampled, the coating transfer proceeds as explained before.

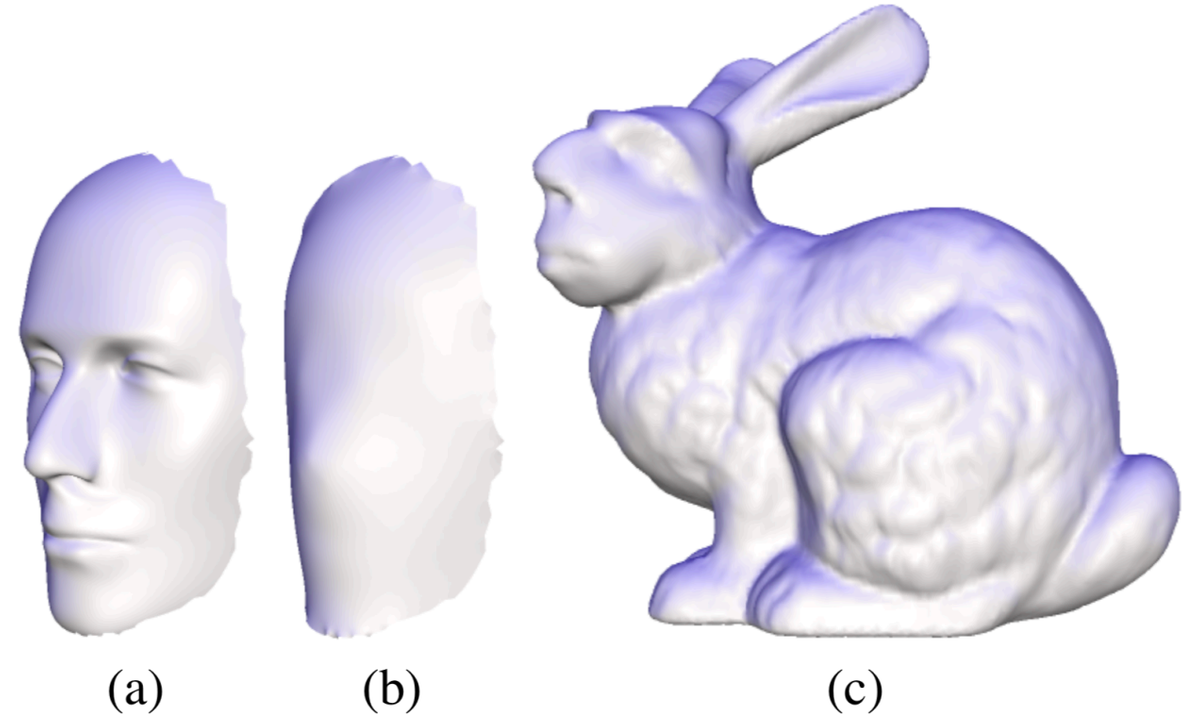
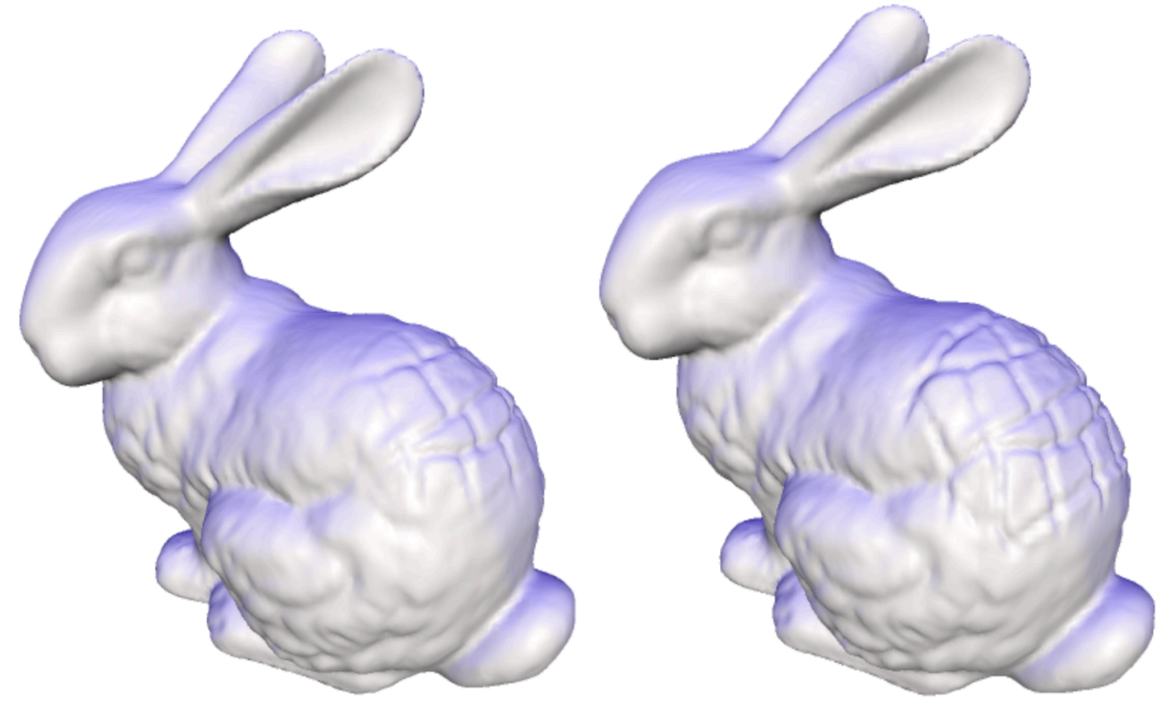


Figure 7: *Transferring the coating of the Mannequin onto the face of the Bunny. (a) The source surface S. It is significantly smoothed to peel the coating. (b) The smoothed surface $\tilde{S}$. (c) The result of coating transfer onto the Bunny.*

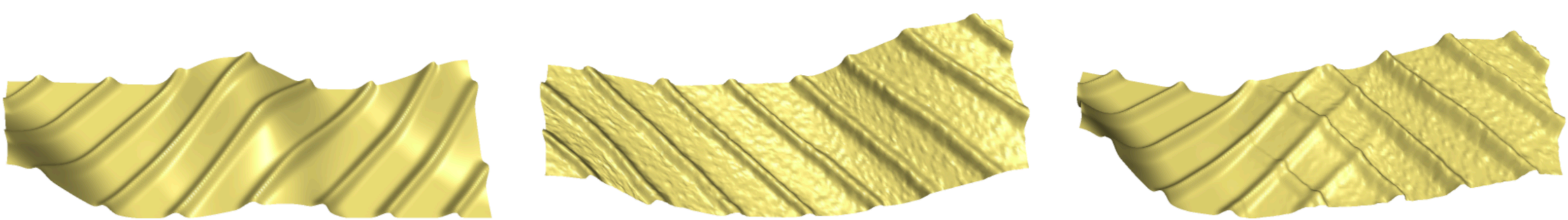
Mixing Details

- Given two meshes with different detail, above mechanism can be applied on a third target mesh from the two sources.
- Each vertex in the transitional region of the target mesh receives the linear interpolation of the corresponding Laplacian coordinates of the source meshes.



(a) (b)

Figure 9: *Transplanting the Armadillo's coating onto the Bunny's back with a soft transition (a) and a sharp transition (b) between the two types of details. The size of the transition area in which the Laplacians are blended is large in (a) and small in (b).*

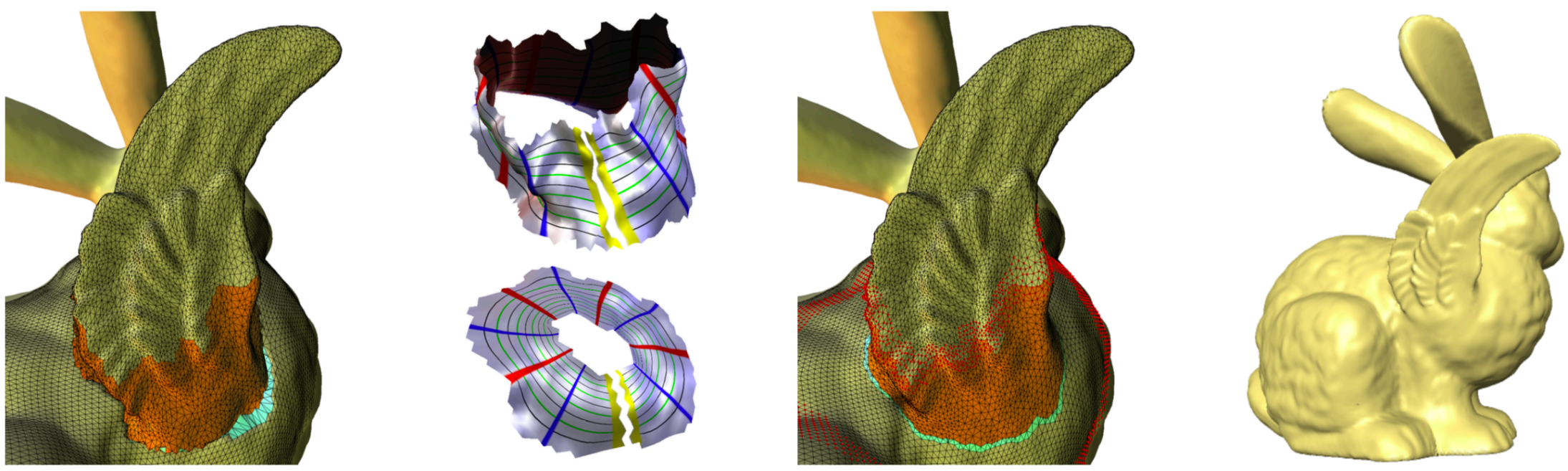


(a) (b) (c)

Figure 8: *Mixing details using Laplacian coordinates. The Laplacian coordinates of surfaces in (a) and (b) are linearly blended in the middle to yield the shape in (c).*

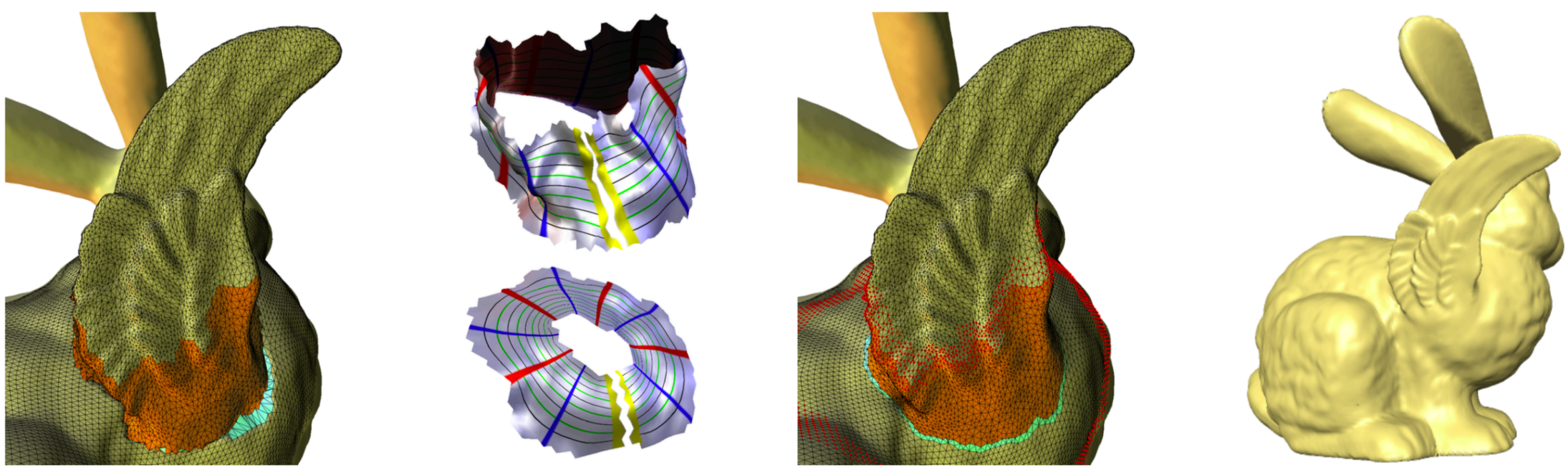
Transplanting Surface Patches

- The transplanting operation consists of two apparently independent classes of operations: topological and geometric.
- The latter operation is based on the Laplacian coordinates and the reconstruction mechanism.
- An example: right wing (S) of the *Feline* is transplanted onto the *Bunny* (U)
- The user selects a region U_0 of U onto which S will be transplanted(the boundary of U_0 is assumed to be homeomorphic to the boundary of S).
- After cutting U_0 off U , the two boundary loops are trivially zipped.



Transplanting Surface Patches

- The remaining transplanting algorithm is similar to details mixing.
- The transitional regions for resampling, S’ on S and U’ on U_0
- For sampling, we require a correspondence between the patches *S’* and *U’*. We parameterize both meshes over a common domain, e.g. a unit square parameter domain.
- Once the transitional regions and the mappings are defined
- In order to mix details, the corresponding samples from *S’*and *U’* are linearly interpolated with weights defined by their relative position in the unit square parameter domain.



Transplanting Surface Patches

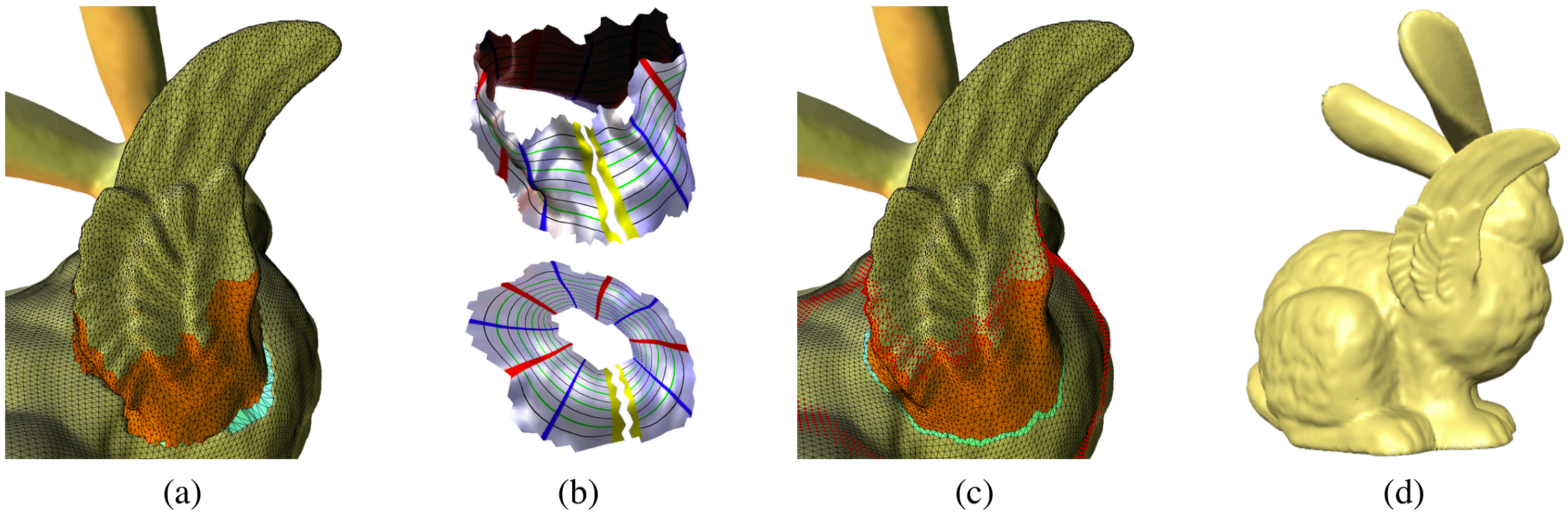


Figure 10: *Transplanting of Feline's wings onto the Bunny (see also the color section). (a) After cutting the parts and fixing the desired pose, the zipping (in green) defines the target connectivity D. The transitional region D' is marked red. (b) D' is sampled over the respective regions $U' \subset U_\circ$ ($U_\circ$ is the cut part of the Bunny's back) and S' (the bottom of the wing). The texture with uv-isolines visualizes the mapping over the unit square. The cut (in yellow) aligns the two maps. (c) The result of reconstruction. The reconstructed submesh is padded by a belt of anchors (red dots). Note the change of the zipping seam triangles (in green) and the gradual change and preservation details within the transition region (red). (d) The flying Bunny.*