6.838: Shape Analysis, 2017
Instructor: Justin Solomon

Surface Deformation By Linear Reconstruction and Material Mapping

Hanjun Li

h1212@mit.edu

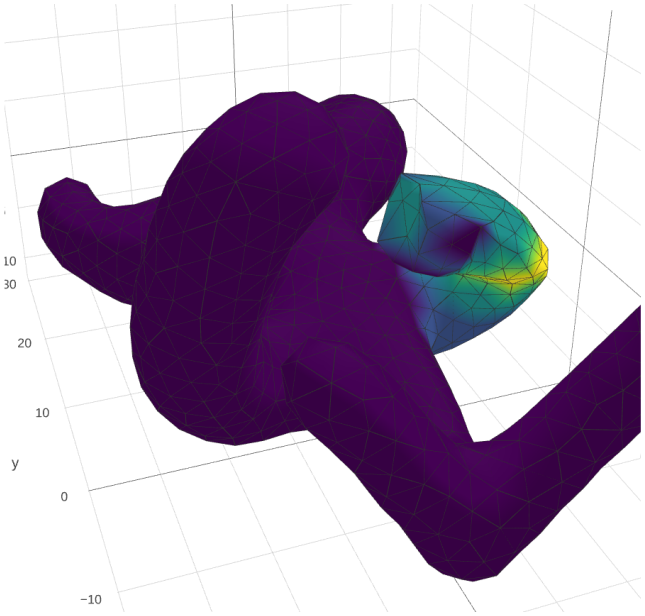

Figure 1: *A folded arm by mapping "soft" material to the elbow.*

Abstract

Linear methods produces fast and usually results for shape deformation tasks. In this paper, we implement the linear reconstruction technique proposed in [WLT12], and apply it to shape deformation tasks. As a linear algorithm, it fits the family of linear methods for shape deformation. We also propose a natural extension to the method to produce more flexible and natural looking deformation results by applying an additional per-edge material mapping. By specifying a material mapping, users can customize the deformation results by including non-geometric information to the deformation process.

1. Introduction

Surface deformation is a very common task for artists in many areas, such as computer graphics, animation, and architectural design. To aid the design process of the artists, it is therefore desirable to produce interactive response to user inputs. As the traditional non-linear methods for shape deformation are too computationally expensive for such objectives, linear approximation methods become more and more popular. This paper adopts one novel method for linearly reconstructing surfaces proposed by [WLT12], and applies it to surface deformation tasks. We also propose a natural extension to the algorithm to produce more flexible and natural looking deformation results. By applying a per-edge material mapping, users can specify the stiffness of different areas of a surface, and incorporate this non-geometric information into the deformation process.

2. Related Work

Surface deformation methods can be generally classified in to non-linear linear algorithms. While non-linear algorithms usually produce robust and natural looking results, they are also very computationally expensive. Linear methods perform well in many cases, but also causes unnatural looking artifacts in other cases. [BS08] provides a comprehensive survey of current linear methods. Since the method this paper focus on fits into the family of differential representation methods, we only discuss related methods in this category. As described in [BS08], there have been many proposed differential representations of surfaces, including gradient, laplacian, and local frame coordinates. In the discrete setting, all of those representations result in values stored on each face or edge of the surfaces. A naive approach would be to directly minimize the differences of those differential representations between the original surface S, and the deformed surface S'.

$$\min_{p'} \sum_i A_i ||D(p_i') - D(p_i)|| \tag{1}$$

Where p, p' are vertices of the original and deformed surfaces, and $D()$ is the differential operator considered by different methods. However, such an optimization doesn't necessarily minimize the differences between s, and s', because each differential value, either on an edge, or on a face is relative to a local frame. Minimizing Eq. 1 only make sense when the local frames are aligned between the surfaces.

To determine the transforms on each local frame, many methods have been proposed. By requiring users to input a local transform along with a set of constraint points, H, one can interpolate the transform by a smooth scalar field s. Possible definition of the scalar field s includes the geodesic distance function from H, the harmonic function that satisfies $Ls = 0$, or the result of the minimization problem $\min_s \sum_{j \in N_1(v_i)} \phi_{ij} ||s_i - s_j||^2$, where ϕ_{ij} is a per-edge weight, connecting face f_i and f_j. The last one is called material mapping because the per-edge weight ϕ_{ij} determines the stiffness of surface by specifying the objective difference of the between scalar field between adjacent faces f_i and f_j. If the difference is small, then the local transform T_i and T_j are similar, which corresponds to a stiff surface area. This idea of using per-edge weight to control the stiffness of surfaces is later applied to the linear reconstruction method as described in Section 3. After determined a set of local transform T_i, one can then minimize the modified version of Eq. 1:

$$\min_{p'} \sum_i A_i ||D(p_i') - T_i(D(p_i))|| \tag{2}$$

The technique proposed by [WLT12] similarly considers a differential representation of surfaces, but it solves for the local transforms T_i implicitly by proposing a discrete version of the fundamental theorem of surfaces, which combines and solves for (transformed) local frames and local coordinates together in a natural way.

3. Technical Approach

To implement the proposed method of [WLT12], we first need to understand the proposed discrete theorem of surfaces. In plain words, the theorem says that given the connectivity of a set of triangles, the edge lengths, as the discrete first fundamental form, and the dihedral angles between adjacent triangles, as the discrete second fundamental form, we can uniquely determine a surface mesh, up to translation and rotation. The theorem can be illustrated and understood by the following steps to reconstruct the surface.

1. First, we can use the law of cosines to compute the inner angles of each triangle face of the surface, from its edge lengths.
2. Second, we can assume a local frame $f_i = (e_{i1}, e_{i2}, e_{i3})$ of each triangle by using the normalized first edge as e_1, the face normal as e_3, and $e_3 \times e_1$ as e_2. The first edge can be consistently determined by the triangle connectivity data. (i.e. use the first vertex to second vertex of each triangle as the first edge.) Although we don't have a concrete description of each local frame, we can compute the coordinate of the edge vectors with respect to each local frame. (The first edge always has $(edge\_len, 0, 0)$ as its coordinate. The coordinates of the other 2 edges can be calculated as cosine and sine of the inner angles computed in step one.)
3. Third, we can calculate the rotation matrices R_{ij} that transform one local frame f_i to the ones of its adjacent triangles f_j, as a product of three rotations. The leftmost rotation rotates the local frame f_i of triangle i around e_{i3}, the normal direction, so that e_{i1} aligns with the edge E_{ij} it shares with triangle f_j. The middle rotation rotate the frame around e_{i1}, now aligned with the shared edge, so that e_{i3} aligns with the normal direction of triangle j. The right most rotation rotate the frame around e_{i3} axis, now aligned with the normal of j, so that the frame aligns with the local frame of triangle j, f_j.
4. Finally, fixing any orthonormal basis for one triangle, and a position for one of its vertex, we can construct the whole surface. The other 2 vertices are determined by their coordinates and the fixed local frame. We can further determine the local frames for the adjacent faces using the rotation matrices, and then, their vertices.

Therefore, the discrete fundamental forms, the edges and dihedral angles determines a surface, as stated in the theorem.

We can then implement the algorithm to linearly reconstruct a surface from its discrete first and second fundamental forms. According to the above steps, it's clear that for a consistent surface, we have the following frame equation:

$$\forall e_{ij} \in E, f_j = f_i R_{ij} \tag{3}$$

An illustration of the frame equation is shown in Figure 2. We similarly have the the following edge equations: (for any triangle T containing vertex n and m)

$$x_n - x_m = (a^1_{mn,T} e_{1,T} + a^2_{mn,T} e_{2,T}) \tag{4}$$

Where a are the edge coordinates in the local frame of T. An illustration of the edge equation is shown in Figure 3. After following the above described steps to calculate the rotations matrices R_{ij} and local edge coordinates a^1, a^2, we can solve Eq. 3 and Eq. 4 by minimizing the following energy term:

$$E_f(\mathbf{f}) = \frac{1}{2} \sum_{e \in E, e = T_i \cap T_j} w_e^{-1} ||f_j - f_i R_{ij}||^2 \tag{5}$$

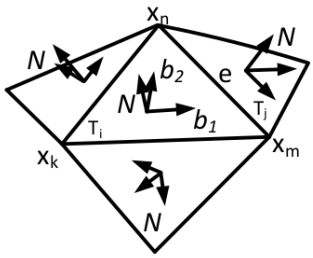

Figure 2: *This figure illustrate the frame equation, Eq. 3. We can see the to transform the frame $f_i = (b_{i1}, b_{i2}, N_i)$ to align with f_j, we first rotate it along N_i to align b_{i1} with edge mn. Then, we rotate it along the new b_{i1} vector, so that N_i aligns with N_j. We finally rotate it along the new $N_i = Nj$ so that b_{i1} is aligned with b_{j1}*

.

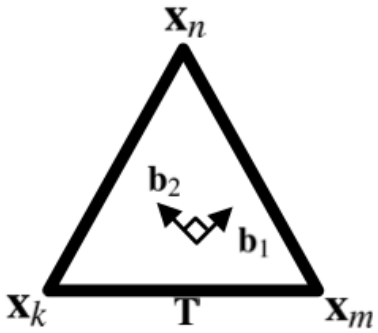

Figure 3: *This figure illustrate the edge equation, Eq. 4. We can see that vector $Xn - Xm = E_{mn}$ is determined by 2 coefficients with respect to the local frame $(b_2, b_1, (N))$*

.

$$E_x(\mathbf{x},\mathbf{f}) = \frac{1}{2}\sum_{e=v_m V_n}\sum_{T\ni e} w_{e,T}||x_n - x_m - f_T(a^1_{e,T}, a^2_{e,T}, 0)^T||^2 \quad (6)$$

$$E_M(\mathbf{x},\mathbf{f}) = wE_f + E_x \quad (7)$$

Where E_f corresponds to Eq. 3, and E_x corresponds to Eq. 4. w_e in the equations are the cotangent edge weights, given by:

$$w_{mn} = \cot(\alpha_{mn}) + \cot(\beta_{mn})$$

where α_{mn} and β_{mn} are the angles opposite to edge mn.

To minimize the energy terms, we first, rewrite the energy equations with corresponding matrix representations:

$$E_f = \frac{1}{2}||W_f^{1/2} R_3 \mathbf{f}'||^2$$

$$E_x = \frac{1}{2}||W_{x,l}^{1/2}(R_1\mathbf{x} - R_{2,l}\mathbf{f}')||^2 + \frac{1}{2}||W_{x,r}^{1/2}(R_1\mathbf{x} - R_{2,r}\mathbf{f}')||^2$$

Where W is the cotangent weight matrix, R_1, R_2, R_3 are calculated

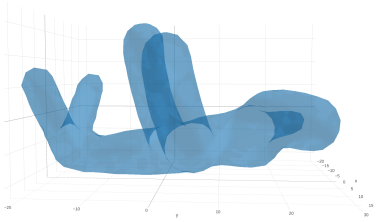

Figure 4: *The figures are produced using the provided data file for homework 2. This is a plot for the original mesh data.*

according to the corresponding definitions of E_f and E_x. Note that since E_x counts the two triangles connected by every edge, we separate it into two equations, with $W_l, M_{2,l}$ corresponds to the triangles on the left hand side of every edge. Now, we can derive and solve the following linear systems:

$$(wR_3^T W_f R_3 - R_{2,l}^T W_{x,l} R_{2,l} - R_{2,r}^T W_{x,r} R_{2,r})\mathbf{f}' + (R_{2,l}^T W_{x,l} R_1 + R_{2,r}^T W_{x,r} R_1)\mathbf{x} = 0$$

$$(-R_1^T W_{x,l} R_{2,l} - R_1^T W_{x,r} R_{2,r})\mathbf{f}' + (R_1^T W_{x,l} R_1 + R_1^T W_{x,r} R_1)\mathbf{x} = 0$$

This sparse linear equation can be solved with a seed vertex position that specifies the translation of the surface, and a seed local frame that specifies the rotation. Figure 5 shows an example of first building the fundamental form representation of a surface, and then solve the system with an arbitrary set of seed vertex and frame. In order to apply this reconstruction algorithm to surface deformation, we first determine a set of vertices as the handle H, and specify their desired positions as hard constraints to the linear system. Usually, we want to only deform a surface locally around some interesting features, but any deformation to a local handle will necessarily affect all other local frames. Therefore, in addition to the movable handle H, we also use a fixed handle H_f to fix the areas that we want to keep the surface unchanged. Figure 6 and Figure 7 shows an example of using movable and fixed handles to stretch an arm of the figure. The fixed handle is shown in dark purple, while the movable handle is shown in bright yellow. Although the shape deformation methods tries to produce natural looking surface deformations, sometimes the expected natural ways of deformation cannot be inferred solely from the geometry of the surface. For example, we may naturally view the deformed part in Figure 7 as something similar to human arms. That is, if we fold the "arm" inwards, we expect to see it bending more in the inner center area corresponding to human elbows. However, from the geometric point of view, there is no obvious clue that a cylinder looking surface should bend non-uniformly in certain areas. To make our deformation algorithm more flexible, we include this non geometric information by adding a material mapping to the surface, and incorporate it into the algorithm.

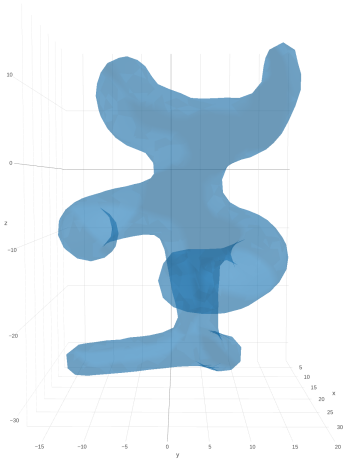

Figure 5: *This is the result of solving the linear system derived from the energy terms in Eq. 7, with an arbitrary* $\mathbf{f_0}$ *and the original* $\mathbf{x_0}$ *as the constraint (seed). Compare with Figure 4 to see the rotation and translation effect. This illustrate that the discrete fundamental form representation of the surface is rigid motion invariant.*

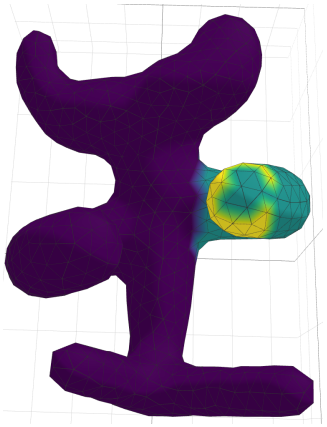

Figure 6: *The original mesh, with texture color showing the area of the fixed handle (dark purple), and the area of the movable handle (bright yellow).*

Note that in Eq. 5 and Eq. 6, E_f and E_x already include a cotangent edge weight w_e. We can naturally add our material related information into the energy by multiplying the edge weights with a per-edge scalar that accounts for the desired material properties. Intuitively, E_f tries to minimize the difference of cross edge rotation matrices R_{ij} between a deformed surface and the original surface. Since R_{ij} is determined by the dihedral angles of the surface, E_f corresponds to minimizing the bending of the deformed surface. Similarly, since E_x tries to minimized the difference of the local coordinates of every edge, it corresponds to minimizing the stretching of the deformed surface. Therefore, to make a certain area easier to fold, we can put a higher weight for the (approximately) bending energy E_f and a lower weight for the (approximately) stretching energy E_x for the corresponding edges. The low weight for E_x makes shrinking the area easy, while the higher weight for E_f enforces the curvature of the shrink area.

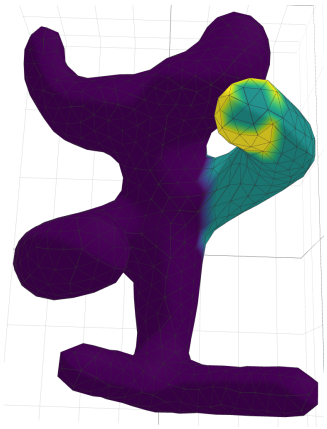

Figure 7: *This figure illustrates how to use fixed handle (dark purple area) and movable handles (bight yellow area) to locally deform a mesh around interesting feature points (the arm). Compare with Figure 6 to see the deformed arm.*

To implement this idea, we need a smooth material mapping that satisfies the required material property. Ideally, we would want some automated method that interpolates a smooth function on the surface requiring minimum user input. We can solve the harmonic function that is similar to the harmonic interpolation technique described in Section 2. However, due to the time constraint, in the experiment of this project, we manually assign a approximately smooth weight mapping on to the desired area by searching vertex positions for target areas. Continuing with the arm deformation example shown in Figure 8, Figure 9 and Figure 10, we map a soft material (i.e. easy to fold) to the elbow looking area (shown in bright yellow), and then fold the arm inward using the above described algorithm with a fixed handle (shown in dark purple) and a movable handle (shown also in dark purple, around the fist area of the deformed arm). Comparing Figure 9 and 10, we can see that the original algorithm produces unnatural widening of the arm area in an effort of trying to preserve edge lengths of compressed area. With material mapping applied, the algorithm let the inner elbow area naturally shrink, and produces a more natural looking folded arm.

Note that during experiment, it's observed that the algorithm requires the mapping to be smooth. If we apply a naive mapping like a step function, with lower weights in the elbow area, we would get highly discontinuous deformation shown in 11. Also, the relative weights to put on the areas also need careful calibration. It took several tries to get the good looking results shown in Figure 8 and Figure 9, while a bad choice of weight, (e.g. too low) produces results shown in 12.

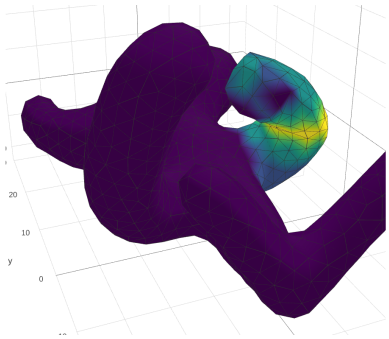

Figure 8: *The folded arm with material mapping. This figure illustrates the effect of applying the material mapping to the target area, to model a easier to fold surface area that is similar to a human elbow. The dark purple areas are the fixed and movable handle points, while the softness of the surface is color coded, with the bright yellow area being the softest.*

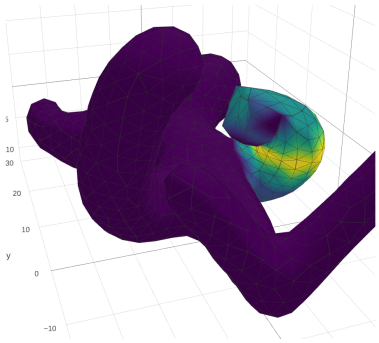

Figure 10: *The more folded arm without material mapping. We can observe a apparent distortion to the arm when it's deformed using the original algorithm, trying to preserve the edge lengths by widening the arm, while the algorithm that incorporates the material mapping avoids the unnatural widening, as shown in Figure 9*

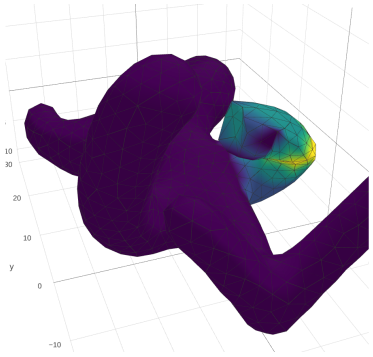

Figure 9: *The more folded arm with material mapping. This figure shows a more extreme example of folded arm. We can compare this with Figure 8 to see how the surface deform after material mapping. We can also compare with Figure 10 to see that with material mapping, the deformed surface looks more natural and has less distortion.*

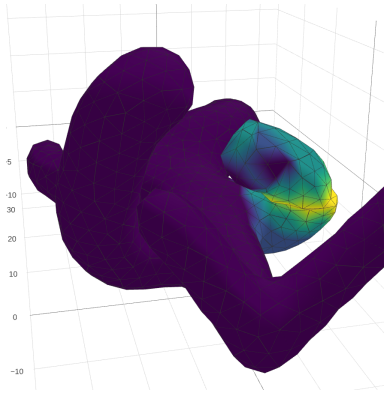

Figure 11: *The more folded arm using a not smooth material mapping. On the boundary of the soft material and the normal material, the deformed surface shows discontinuity due to large change of dihedral angle. This deformation uses the same constraint as Figure 8*

4. Conclusion and Future work

In this paper, we implement the linear surface reconstruction technique proposed by [WLT12] and apply it the surface deformation tasks. This method fits in the family of linear deformation algorithm, but has the advantage of combining the local frame transformation and local coordinate representation into a single linear optimization problem. Therefore, this method does not require users to input local transformations of the handle points, and solve for local transformations implicitly. We then propose to combine the material mapping technique for local transformation interpolation with the linear reconstruction algorithm, by simply multiplying the edge weight term w_e in Eq. 5 and Eq. 6 with a per edge weight that corresponds to a material mapping that controls the stiffness of surface areas under deformation. This gives a more flexible and more powerful algorithm that can incorporate non-geometric details into the deformation process.

However, determine the proper material mapping can be tricky. Firstly, it would ideally be an automated process of interpolating values according to a few user input points. We can treat this as a similar problem to local transformation interpolation, and apply the harmonic interpolation method. In the future, we can implement this idea and integrate it into our surface deformation pipeline. Secondly, the values of the material mapping needs careful design. This would require a very good understanding of the energy terms E_f and E_x defined in Eq. 5 and Eq. 6 from the user. The process of defining a good material mapping may require many experiments. A good thing is that once the mapping is determined, it is always valid for the corresponding surface, before and after deformations.

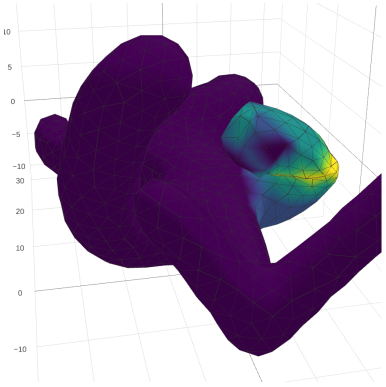

Figure 12: *The more folded arm using too soft a material. This also results in unnatural looking deformation. This deformation uses the same constraint as Figure 8*

References

[BS08] BOTSCH M., SORKINE O.: On linear variational surface deformation methods. *IEEE transactions on visualization and computer graphics 14*, 1 (2008), 213–230. 2

[WLT12] WANG Y., LIU B., TONG Y.: Linear surface reconstruction from discrete fundamental forms on triangle meshes. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 2277–2287. 1, 2, 5