

BENDING WITH LOCAL VOLUME PRESERVATION

A Thesis Proposal
Presented to
The Academic Faculty

by

Wei Zhuo

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in Computer Science in the
School of Interactive Computing

Georgia Institute of Technology
October 2013

BENDING WITH LOCAL VOLUME PRESERVATION

Approved by:

Professor Jarek Rossignac,
Committee Chair
School of Interactive Computing
Georgia Institute of Technology

Professor Jarek Rossignac, Advisor
School of Interactive Computing
Georgia Institute of Technology

Professor Stefanie Hahmann
Applied Mathematics
Grenoble Institute of Technology

Professor Karen Liu
School of Interactive Computing
Georgia Institute of Technology

Professor Greg Turk
School of Interactive Computing
Georgia Institute of Technology

Date Approved: TBD

PREFACE

ACKNOWLEDGEMENTS

TABLE OF CONTENTS

DEDICATION	iii
PREFACE	iv
ACKNOWLEDGEMENTS	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
SUMMARY	x
I INTRODUCTION	1
1.1 What is bending?	1
1.2 Applications of bending	4
1.2.1 Bending driven by a curve.	4
1.2.2 Bending driven by a surface.	4
1.3 Local volume preservation in bending	5
1.4 Problem formulation	7
II LITERATURE SURVEY	9
2.1 Prior art on bending	9
2.2 Existing techniques in global volume compensation	13
2.3 Approaches to local volume preservation	14
2.4 Variations of bending	16
III PLANS FOR REMAINING CONTRIBUTIONS	18
3.1 Sampling and approximation	18
3.1.1 Projection	18
3.1.2 Normal and curvature interpolation	18
3.1.3 Experiment with different levels of subdivisions of the spine .	19
3.2 Extend to stretchable spine	20
3.2.1 2D stretchable spine	20

3.2.2	3D stretchable spine	21
3.3	Extend to shell-based bending	23
3.4	Implementation Schedule	24
IV	SHORT SUMMARY	26
4.1	Bending driven by lower dimensional proxy	26
4.2	Volume preservation	26
4.3	Existing work and future plan	27
APPENDIX A	— PUBLICATIONS	28
REFERENCES	51

LIST OF TABLES

LIST OF FIGURES

1	An intuitive example of bending.	1
2	Models suitable for bending driven by a spine curve.	3
3	Two versions of planar bending.	6
4	The effect of local area preservation.	7
5	Barr 1984, Global and local deformation of solid primitives	10
6	Hsu, Lee and Wiseman 1984, Skeletal strokes	10
7	Llamas et al. 2005, Bender: A virtual ribbon for deforming 3D shapes in biomedical and styling applications	11
8	Rohmer, Hahmann and Cani 2008, Local volume preservation for skinned characters	15
9	Wang, Jüttler, Zheng and Liu 2008, Computation of Rotation Minimizing Frames	16
10	Implementation schedule of future work	25

SUMMARY

A bending is the effect of applying a continuous, non-affine transformation to a solid. It is controlled by a low dimensional proxy, which can be a spine curve or surface. This thesis addresses a suite of problems in bending with local volume preservation, meaning that the transformation preserves the volume of any subset of the solid. Our bending framework adopts the following assumptions in mechanics:

- Planes normal to the spine curve remain normal to the spine curve after bending.
- Lines normal to the spine surface remain normal to the spine surface after bending.

With these assumptions, we derive a family of closed-form solutions for bending with local volume preservation. The derivation exploits the constraint of unit Jacobian determinant everywhere. We compute the solutions in real-time using analytic geometry, and discuss anticipated applications in shape manipulation or animation, as well as geometric modeling of machining or deposition process.

CHAPTER I

INTRODUCTION

1.1 *What is bending?*

A bending is the effect of applying a continuous, spatially-varying transformation to a solid. It is controlled by a low dimensional proxy, such as a spine or a shell. For example, one may interactively stretch or curve a spine, usually through the control points of the spine. The deformation of the solid is then completely defined by the deformation of the spine. Such effects happen often in shape-editing operations when the designers or specialists manipulate a curve or a thin plate in manufacturing, medical or aesthetics applications.

Consider that the designer starts with a shape S_0 , as shown in Figure 1, and specifies an initial spine C_0 that is green and goes through S_0 . The spine needs to be a smooth curve that may pierce the solid S_0 or not. In bending, the designer specifies the positions and time-evolution of the control points of the spine curve C , such that C changes from C_0 to C_1 .

Assume that $C_0(s)$ is a point on the initial version C_0 of the spine. Here s is the

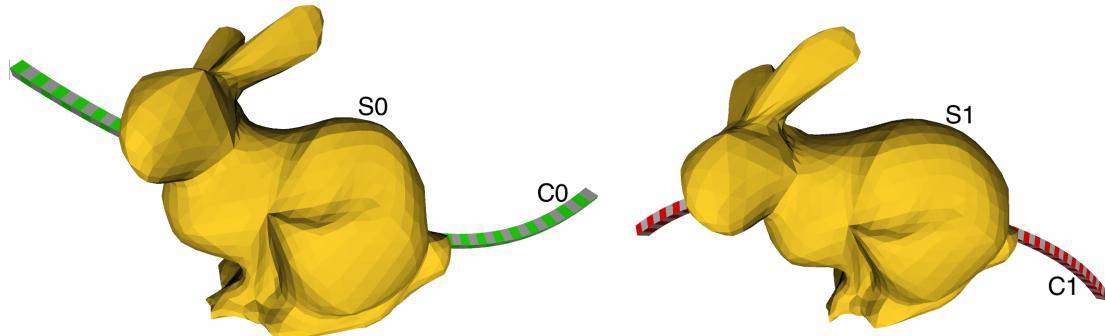


Figure 1: An intuitive example of bending.

parameter along the spine. We associate the parameter s with a point P_0 of S_0 such that $C_0(s)$ is the closest projection of P_0 onto C_0 . All points of S_0 associated with a particular parameter s of are called a cross-section, which is formally defined as the point set $\{P_0, P_0 \in S_0, \arg \min dis(C_0(s), P_0) = s\}$. Note that each cross-section is planar. During bending, points of the same cross-section will remain in the cross-section associated with the same parameter s . Note that the shapes of the initial and deformed cross-sections may be different, but they are both planar. To sum, the bending driven by a curve has the following assumptions: First, planes normal to the curve, or cross-sections remain normal to the curve after bending. Second, the parameter s of any cross-section remain the same during bending.

Often, the spine may represent the central axis of an elongated part and the key idea is that the deformation of the solid is completely defined by the deformation of the spine. Hence, the designer defines the initial spine (maybe by placing a few initial control points for it) and then defines its deformation over time (maybe by specifying a few key positions for each control point that will be interpolated by the motion of that control point). Then for each time t , the bending algorithm computes the current position of each vertex of the solid and displays the resulting triangulated surface.

Similarly, in bending control by a base surface, the deformation of the solid is completely defined by the deformation of the base surface. The designer defines the initial base surface, which may be a torus patch or a spline surface, and then specifies its deformation over time. Then for each time, the bending algorithm computes the current position of each vertex of the solid and display the result.

Specifically, assume that $C_0(u, v)$ is a point on the initial spine surface. Here we need two parameters u, v for surface parameterization. We associate the parameters u, v with a point P_0 of S_0 such that $C_0(u, v)$ is the closest projection of P_0 onto C_0 . All points of S_0 associated with a set particular parameters u, v form a line segment

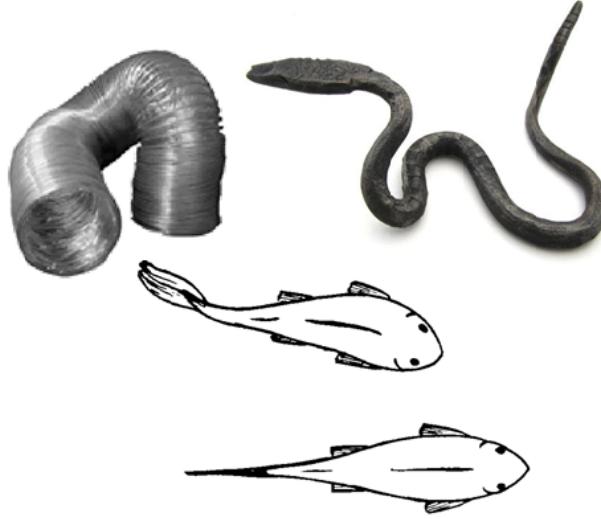


Figure 2: Models suitable for bending driven by a spine curve.

normal to C_0 at $C_0(u, v)$. During bending, points of a line segment will remain in the line associated with the same parameters. To sum, the bending driven by a surface has the following assumptions: First, lines normal to the base surface remain normal to the surface after bending. Second, the parameters u, v of any normal line segment remain the same during bending.

To define bending formally, one needs to provide a mathematical formulation of the time-parameterized mapping from the initial to the deformed position of each vertex. Note that this mapping is not an affine transformation as a triangle does not map into a triangle by bending. We address this problem by using a subdivision that produces a fine triangulation, so that the error between the image of a triangle (by bending) and the triangle spanning the images of its 3 vertices is small.

A key contribution reported in this thesis, is the mathematical definition of a family of mappings that all satisfy the requirements of preserving planes or straight lines normal to the spine or the surface, as well as preserving the local volume. We give the full list of requirements in Section 1.4.

1.2 Applications of bending

1.2.1 Bending driven by a curve.

Bending by a spine curve is motivated by applications in modeling tube-like structures, such as hoses, wires, ducts, and also deformations of a trunk, a snake, or a tongue as shown in Figure 2. In these models, it is intuitive to specify a curve, which may represent the central axis of an elongated part.

In shape-editing operations, the designer may manipulate the spine using two frames, each controlled by a tracker in a different hand [23]. The orientations of the trackers define the end-tangent directions to the spine. The total torsion along the spine is controlled by the rotations of the trackers around the corresponding tangents. In animation, the designer defines the initial spine and its evolution over time, as mentioned in Section 1.1. To do so, she would specify only a few control points of the spine and a few key positions for each control point that will be interpolated by the motion of that control point. Then for each time t , the bending algorithm computes the current position of each vertex of the solid and displays the resulting deformed solid.

1.2.2 Bending driven by a surface.

Bending by a spine surface is in part motivated by the studies in mechanics on thin plate bending. Instead of focusing on the deformation of the thin plate surface, we are interested in the answer to this question: How does the thickness of the “meat” attached to the surface change due to surface bending. For example, if the “meat” is a volume-preserving finite element mesh. Assume that the volume of each cell in the mesh is incompressible. Our bending algorithm computes the exact shape of each cell during bending while the cell volume remains a constant.

In machining or deposition process, the spine surface is static and we refer to it as the base (backbone) surface. The material removal rate at a point on the

base surface is characterized by the ratio of the removed volume to the local surface area at the point. For constant material removal rate, the milling depth is not a constant and should adapt to the local curvature of the base surface. Hence, if the base surface is developable, the milling depth can be computed by a locally volume-preserving bending of a plane into the base surface. Similarly, the deposition amount is characterized by the increased volume to the local surface area. We show that the thickness of the deposited layer can be computed by a special form of the curvature-sensitive formula in our bending framework.

1.3 *Local volume preservation in bending*

Physically plausible simulations that involve biological creatures or deformable shapes made of incompressible materials require that the volume be preserved.

For example, during an animation where no external forces or torques are exerted on a body, the momentum and kinetic energy are preserved. Both depend on the mass and hence of the volume (if one assumes constant mass). Hence, changes of volume during an animation will result in surprising changes of velocity.

It is much simpler to preserve global volume than local volume. For example, one may dilate the entire solid by a specific amount to compensate for the undesired volume gain or loss. In fact, one of the contribution of this thesis is to provide a simple formula for computing that dilation amount for arbitrary (not necessarily convex) solids. Unfortunately, preserving the global volume is not sufficient for a physically plausible behavior.

Consider the two-dimensional version of this problem shown in Figure 3. The blue curve represents the spine. We show (black outline) a slice of the solid. The center of mass of the slice is shown as the yellow point, which is initially on the spine for clarity. Assume that the user bend the spine downward. We show two versions of bending on the left. In 1, the offset distance on each side of the spine remains the same, so the

center of mass tends to move above the spine. In 2, the offset distance increase on the concave side and decrease on the convex side of the spine, so the center of mass tends to move below the spine.

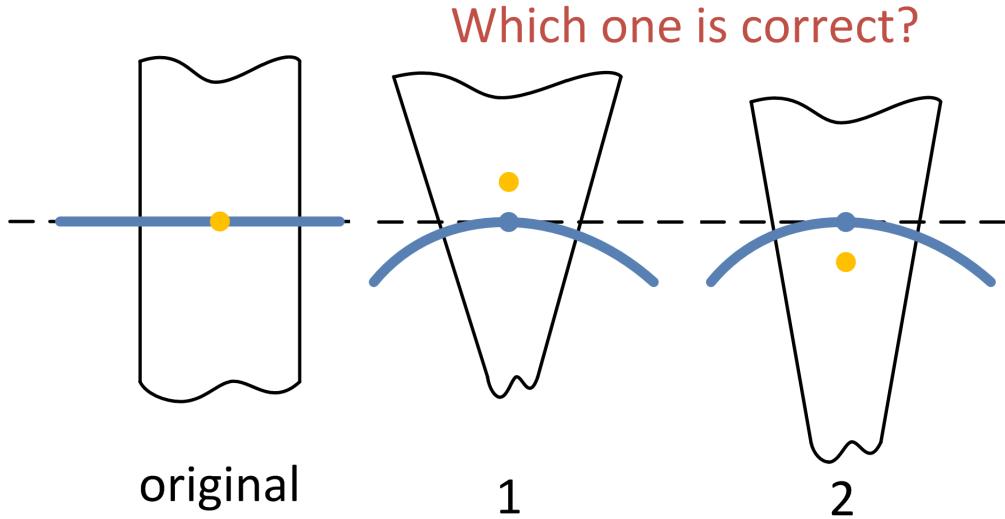


Figure 3: Two versions of planar bending.

Naive spine bending is shown in 1, where the area of a region above the spine is enlarged while the area of a region below is reduced. Hence, the center of mass is above the spine. Therefore, it appears that some area that was initially below the spine has magically moved above the spine.

When local area is preserved (shown in 2), the portion above the spine are stretched along the spine and therefore become narrower, hence closer to the spine. Portions below are subject to the inverse effect: they are pushed away from the spine. The center of mass tends to move below the spine.

Figure 4 further explains the effect of local preservation. Imagine the slice has a multiple layers along the spine. Each layer originally has the same area. After spine bending, the area of each layer should remain the same. Therefore, the layers above the spine gets stretched and become thinner. The layers below the spine gets compressed and become thicker.

To formalize the notion of local volume preservation, one must introduce the

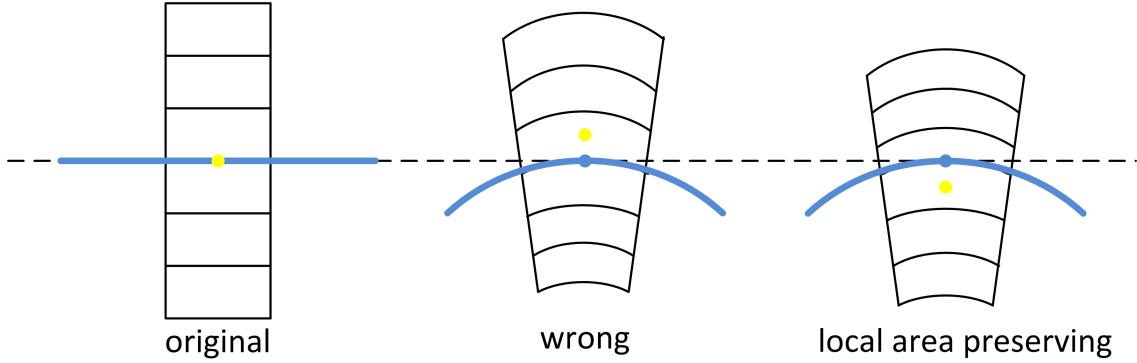


Figure 4: The effect of local area preservation.

measure of the local volume change. We identify that the determinant of the Jacobian of the transformation is a good measure of the local volume change. The Jacobian of the transformation is $J = \frac{\partial P_1}{\partial P_0}$, where P_1 represent any vertex of S_1 after bending; P_0 represent the corresponding vertex of S_0 before bending. There is a local expansion if the determinant, $\det(J)$, is larger than 1. The reverse is local contraction if $\det(J)$ is smaller than 1. For local volume preservation, the determinant should equal to 1 exactly.

1.4 Problem formulation

After having introduced the framework of bending, we now turn to the formulation. Recall that the designer starts with a shape S_0 and specifies an initial spine C_0 , which is a smooth curve that may pierce the solid S_0 or not. Then the designer deforms C_0 to C_1 . The solution for any shape to maintain its original volume during deformation is to obtain a mapping $M : S_0 \rightarrow S_1$, such that M preserves volume locally (i.e., $\text{vol}(U) = \text{vol}(M(U))$, for any subset U of S_0). A list of requirements for M to be valid and producing plausible results is the following.

1. Topology-independent: M should operate on any shape topology and independent of S_0 . Hence, M is fully defined by C_0 and C_1 .
2. Homeomorphism: M should be a homeomorphism between S_0 and S_1 . This

is important because we want the mapping to be invertible: $M^{-1}(X_1) = X_0$, where M^{-1} is defined by the initial spine as C_1 and the final spine as C_0 .

3. M should preserve the parameter on the spine. In bending driven by a curve, the parameter s of any point of the solid should remain the same. In bending driven by a surface, the parameters u, v remain the same.
4. Locally volume preserving: Last but most important, M preserves volume locally (i.e., $\text{vol}(U) = \text{vol}(M(U))$ for any subset U of S_0). This is important for the physical plausibility of digital simulations, especially when they involve interactions between evolving solids (swimming creature) and surrounding, incompressible fluids.

CHAPTER II

LITERATURE SURVEY

2.1 *Prior art on bending*

We present here an expository account of related work on bending. In particular, we present them in this section in a chronological order of their emergence, which also happens to reflect the degree of complexity of each model.

Barr [6] presents a bending model that simulates global linear bends along an axis. In this model, the length of the spine does not change during bending. The bending angle changes linearly in the bent region as shown in Figure 5. So the curvature k is constant in the bent region. The offset distance from the spine does not change during bending. This leads to the first version of bending in Figure 3.

Hsu, Lee and Wiseman apply the bending model to graphics design [17]. In this work, the spine is a user-specified planar curve, representing an artistic brush stroke, rendered with textures. They draw the textures using the normal to the spine as the local y-axis. To deal with sharp bending, local self-intersections are trimmed as shown in Figure 6.

The two approaches mentioned above do not preserve area or volume. A drawback of the bending model described so far is that it can not produce the correct result for simulating the bending of incompressible material. On bending a physical object, the material on the concave side of the spine will be compressed while that on the convex side stretched. The way and amount in which the material shrinks or expands should be local volume preserving.

Chirikjian [10] presents a mathematically precise approach for 2D bending with local area preservation. The solution is to update the offset distance based on the

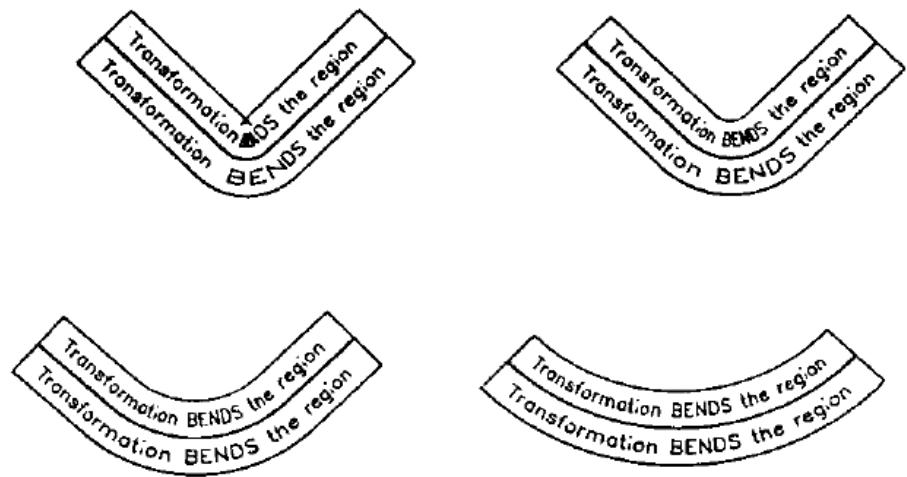


Figure 5: Barr 1984, Global and local deformation of solid primitives

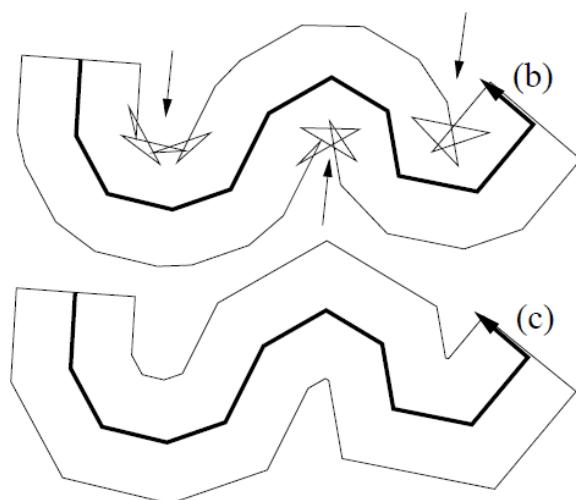


Figure 6: Hsu, Lee and Wiseman 1984, Skeletal strokes

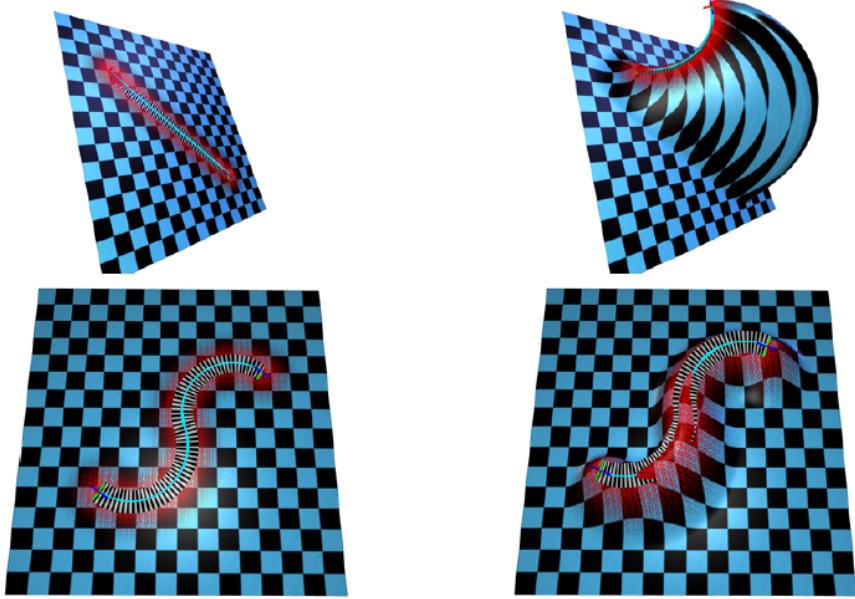


Figure 7: Llamas et al. 2005, Bender: A virtual ribbon for deforming 3D shapes in biomedical and styling applications

curvature of the spine: $h_1 - \frac{kh_1^2}{2} = h_0$, where h_1 and h_0 are the updated and the original offset distances; k is the curvature. We use this formula for bending an image with local area preservation [39]. To alleviate the drawback of insufficient sampling, we use the spine-aligned grid. The deformed image is a texture mapping of the original image with the deformed grid.

In machining with constant material removal rate, Moon [24] identifies the same quadratic formula for offsetting backbone curves with uniform flux: The increased area is evenly distributed along the boundary. The formula is $h - \frac{kh^2}{2} = r$, where h is the milling depth and r is the material removal rate. In [39], we show how to generate a series of contours of this curvature-aware offsetting. Directly offsetting according to the formula exhibits an increasing amount of discontinuities where the curvature of the previous offset changes rapidly. We propose to use the combination of curvature-aware offsetting and selective smoothing to produce concentric offset contours that are smooth and approach a constant area-to-length ratio.

Llamas, Powell, Rossignac and Shaw present the Bender tool [23], which allows the user manipulates the spine using two frames, each controlled by a tracker in a different hand. The orientations of the trackers define the end-tangent directions to the spine. The spine is modeled as a bi-arc curve [32]. The designer presses buttons that have been engineered on the trackers to indicate the moment where the current shape of the spine and of the torsion should be registered as the grab ribbon. Then, as the designer manipulates the two trackers, the current ribbon is computed at each frame.

In Bender, the mapping of the vertices of the solid is performed as follows. For each vertex P_0 of the solid, they compute the parameter s of the closest projection $C_0(s)$ of P_0 onto the spine of the grab ribbon. P_0 is expressed in the local frame at $C_0(s)$, $P_0 = C_0(s) + xT_0(s) + yN_0(s) + zB_0(s)$. They also compute the distance d between P_0 and $C_0(s)$. To compute the mapping P_1 , they identify the corresponding frame $T_1(s), N_1(s), B_1(s)$ on the current ribbon. However, instead of mapping P_0 to $\bar{P}_1 = C_1(s) + xT_1(s) + yN_1(s) + zB_1(s)$, they compute the screw motion M such that $M(0)$ is identity and $M(1)$ maps P_0 to \bar{P}_1 . Then, they apply a fraction $M(f(d))$ of that screw motion to P_0 and obtain $P_1 = M(f(d))P_0$, where $f(d)$ is a decay function modeled using a cosine square expression.

Their approach is designed to support local tweaks, where the effect of the tweak blends smoothly with the unchanged surrounding, as shown in Figure 7. Specifically, to produce useful bending of tubular parts, they change the f function to give it a plateau region. In this case, there is no attenuation and the effect of their mapping is similar to the one proposed here with two differences: (1) They can support an unnatural twist designed by the operator and distributed uniformly along the spine. (2) Though within the plateau region, their bending does not preserve the local volume.

2.2 Existing techniques in global volume compensation

Maintaining the volume is important for modeling deformations where the volume occupied by the shape remains constant, and in physics-based simulations where material incompressibility matters [21]. In general, volume can be efficiently corrected by uniform scaling the shape by $s = \sqrt[3]{\frac{V_t}{V_0}}$, where V_0 , V_t are current and target volumes [11]. However, uniform scaling may produce unbounded Hausdorff error between the original shape and the scaled shape, especially when the shape contains parts that are long and thin [39].

As a post-processing step, area or volume preservation has been studied for multi-level shape editing. Hahmann, Sauvage and Bonneau [13] present multiresolution deformation of curves which satisfy the bilinear constraint of constant enclosed area. In order for the volume to converge to the target value, they evaluate the current area at each iteration, and adjust the control vertices. Since the cost of volume evaluation is proportional to the number of vertices, Hirota, Maheshwari and Lin [15] compute the volume at multiple levels so that the volume does not need to be evaluated exactly at initial steps.

Angelidis, Cani, Wyvill and King [1] combine multiple volume-preserving shape operations, called swirls, to form a complex deformation that is also volume-preserving. They define the basic operation, called a swirl, that locally twists the space around a axis. By arranging multiple swirls in a circle such that the twist axes of these swirls are coplanar and radially outward, they can achieve the effect of pulling along the direction normal to the twist axes.

In differential geometry, a classical theorem due to Steiner [35] establishes the differential relationship between the surface properties and the volume enclosed. To preserve the total volume, some authors grow or shrink the shape uniformly (via constant distance normal offsetting rather than global scaling) based on global curvatures in one step without iteration.

For example, Moon [25]’s solution is to compute the variable offset distance from the base surface that makes the deposition amount locally proportional to the surface area. They verify the computational results on cylindrical, ellipsoidal and catenoid surfaces. For general surfaces, their approach requires evaluating the Gaussian and the mean curvatures everywhere on the surface. In comparison, we compute the constant offset distance from the base surface that regain the target global volume. Instead of evaluating the curvatures everywhere on the surface, we define and evaluate the global curvatures to compute the constant offset distance. We verify our results on triangle and quad meshes of various shapes.

2.3 Approaches to local volume preservation

In addition to keep the global volume of the object constant, local volume preservation is essential to a more natural, physically plausible behavior of the deformation.

In deformation driven by a base surface, local volume preservation aims at preserving the local volume distribution between the base surface and the offset surface. Botsch and Kobbelt [8] explore the degrees of freedom in the position for a offset point to satisfy the local volume preserving constraint: They do not require the offset direction to be normal to the base surface. This is different in classical theory of bending [27], where assumption is that lines remain normal to the base surface after bending. Moon’s approach to equivolumetric offset has the same assumption [25]. However, it does not applicable to surface bending with local volume preservation as it assumes a static base surface, which is not subject to any deformation. Even if the author parameterized the base surface and assumed that it could deform, the formula is not directly applicable as it does not take any local surface stretching or compression factor into account.

In deformation driven by external forces, local volume-preserving deformation of a object aims at obtaining a divergence-free displacement field for all points of the

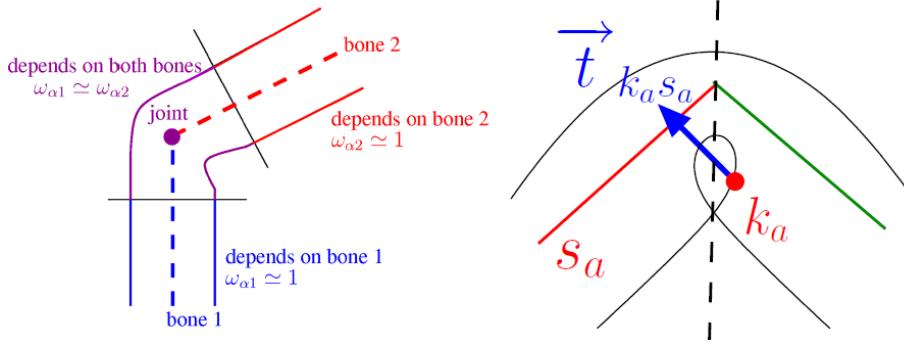


Figure 8: Rohmer, Hahmann and Cani 2008, Local volume preservation for skinned characters

object: $\nabla \cdot V = 0$, where V is the vector-valued function denoting the displacement vector defined everywhere within and on the object. In finite element simulations [7], the displacement field is computed by time integration. Each time step consists of: (1) Evaluating the strain and stiffness tensors from the object geometry and material property (2) Computing the force field everywhere within and on the object from the evaluated strain and stiffness (3) use the force field to update the velocity field, which is corrected to have vanishing divergence.

In skeleton-driven deformation with local volume preservation, Rohmer, Hahmann and Cani [29] localize the volume correction on the corresponding part of the skin mesh. They use a correction map associated with each region. To correct the volume, they offset each point by a amount proportional to the correction map at each point. To avoid local self-intersections, they detect if an offset point is within its region determined by automatic segmentation. If a point is not within its region, they translate the point until it reaches the border of its associated region, as illustrated in Figure 8. In [30], they further show that a stylized deformation, such as isotropic inflation, bulging, or rippling effects, is possible by using 1D profile curves to control the correction map.

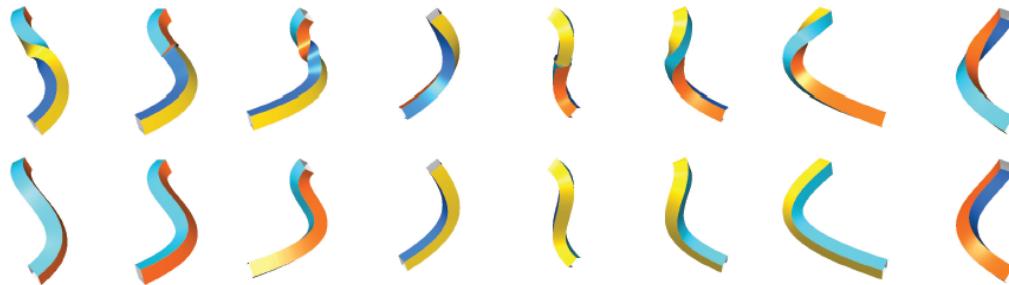


Figure 9: Wang, Jüttler, Zheng and Liu 2008, Computation of Rotation Minimizing Frames

2.4 Variations of bending

After discussed prior work and techniques on spine-driven deformation with local volume preservation, we review here some of the variations of spine-based models in other contexts. We have already mentioned that stroke design can be automated by curve bending [22]. In addition, they extend the bending model to animating 2.5D cartoon [16]. They anchor different parts (which may have overlaps) of a image to a spine. The user can twist, bend or stretch the spine for deforming the parts and generate a animation.

Spine-based models in object recovery proves useful in vision research [26]. Various types of objects are of structures of generalized cylinder, which is the result of a possibly varying cross section along a path specified by a spine which may be an arbitrary space curve. The cross section needs not be connected so as to have bifurcations. In [3], the authors extract such structures for blood vessels reconstruction and meshing from MR angiography. They also compute the central paths and maximal inscribed balls in the vessel for vessel surface analysis.

Twist compensation. Surface parameterization is necessary to register each point on the tubular surface with the spine. It is not trivial to determine the local frame on the 3D spine for tubular surface parameterization. Due to that the Frenet normal may be undefined at the inflection point, using the Frenet frame introduces

unwanted twists as show in the top row of Figure 9. Wang, Jüttler, Zheng and Liu compute the rotation minimized frame as the better alternative to the Frenet frame. Hanson and Ma [14] introduce the concept of parallel transport. Their concept is to generate an parallel vector that turns as much as is necessary for it to remain normal to the 3D curve. They present the parallel transport algorithm that computes a smoothly varying frame consisting of a pair of parallel vectors orthogonal to each other. Their algorithm makes use of the rotation matrix [12] for generating the parallel transport frame along a piecewise linear approximation of the 3D curve.

CHAPTER III

PLANS FOR REMAINING CONTRIBUTIONS

In this chapter we discuss the additional work, which will be included in the final thesis.

3.1 *Sampling and approximation*

One impetus for future work is dissatisfaction with the accuracy in the current implementation for spine-driven bending. We plan to use more accurate projection for model registration, interpolate the normal and the curvature at vertices of the spine, and experiment with different levels of subdivisions.

3.1.1 Projection

For each point P_0 on the original shape S_0 , we compute its closest projection on the spine curve C_0 . We identify two neighboring vertices C_0^k and C_0^{k+1} of C_0 closest to P_0 . The projection of P_0 on C_0 is the projection of P_0 on the edge $C_0^kC_0^{k+1}$.

Let Q_0 be this projection and a measures how far Q_0 is from C_0^k :

$$Q_0 = C_0^k + aC_0^kC_0^{k+1}, 0 < a < 1.$$

Then parameters of Q_0 is k, a . We use the same parameters to compute Q_1 on C_1 : $Q_1 = C_1^k + aC_1^kC_1^{k+1}$, which is used as the anchor point for computing P_1 in both 2D and 3D spine-driven bending.

3.1.2 Normal and curvature interpolation

Since we compute the projection (Section 3.1.1) as: $Q = C^k + aC^kC^{k+1}$, the normal N and the curvature K at Q should be interpolated from the normals and the curvatures at C^k and C^{k+1} .

For the normals, we want to interpolate their orientations as a linear function of a . Hence, we may use a SLERP or a more explicit formulation as follows: Let θ denote the angle between N^k and N^{k+1} . Then, the interpolated normal should have angle $a\theta$ from itself to N_{k+1} . θ is computed as:

$$N_\perp^k = (N^k \times N^{k+1}) \times N^k$$

$$\theta = \tan^{-1}(N_\perp^k \cdot N^{k+1}, N^k \cdot N^{k+1})$$

$$N = N^k \cos(a\theta) + N_\perp^k \sin(a\theta)$$

For the curvature, we can simply use linear interpolation:

$$K = (1 - a)K^k + aK^{k+1}.$$

Note that if the spine is a Euler spiral or a clothoid, linear interpolation computes the curvature precisely, as the clothoid has linearly varying curvature everywhere. If the spine is a smooth curve that does not have linearly varying curvature, linear interpolation is still an accurate approximation as it is possible to locally approximate any smooth curve with clothoids [5].

3.1.3 Experiment with different levels of subdivisions of the spine

We start with a piecewise linear representation of the spine curve and use $J_{\frac{3}{2}}$ subdivision which converges to a quintic spline [31]:

$$P_{2j}^{k+1} = (\frac{3}{2}P_{j-1}^k + 5P_j^k + \frac{3}{2}P_{j+1}^k)/8$$

$$P_{2j+1}^{k+1} = (\frac{1}{2}P_{j-1}^k + \frac{15}{2}P_{j+1}^k + \frac{1}{2}P_{j+2}^k)/16$$

where k is the level of subdivision. By retrofitting, the subdivision curve can interpolates the original control points P^0 .

We plan to investigate how the error scales with the number of curve points. The error is measured as the root mean square of local volume deviations in the bending

result [39]. All computations will be in double-precision floating point formats. The guess is that the error decreases for the first few levels of subdivisions as more accurate approximation of the spine is used. However, the levels of subdivision can not be unbounded as numerical errors (in curvature, normal estimation) accumulate by subdivision and eventually cause larger local volume deviations. To show how levels of subdivision affect the result, we will plot the root mean square of local volume deviations with the number of uniform subdivisions. Then we conclude the optimal level of subdivisions in terms of the minimized RMS volumetric errors.

3.2 Extend to stretchable spine

We present the mathematical formulation of volume-preserving bending with stretchable spine. Non-stretchable spine, or length-preserving bending with local volume preservation has been studied in 2D and 3D before [10] [39]. To the author's knowledge, the extension to the stretchable spine presented here is novel.

3.2.1 2D stretchable spine

Let $C(s)$ represent a planar curve in space, where s may not be the arc-length parameter. Also, let l be the arc-length parameter such that $l(s)$ is the arc-length of C from its origin to $C(s)$. The relationship between s and l is,

$$l(s) = \int_{s_0}^s |C'(s)|ds$$

Hence the derivative relationship between s and t is

$$dl = |C'(s)|ds.$$

$T(s)$, $N(s)$ are vectors representing the unit tangent, normal at $C(s)$. Let P denote a offset point near C , we have

$$P(s, h) = C(s) + hN(s).$$

Therefore,

$$\begin{aligned}\frac{dP}{ds} &= C'(s) + h \frac{dN(s)}{ds} = T(s)(1 + hk)|C'(s)|, \\ \frac{dP}{dh} &= N(s).\end{aligned}$$

In the deformation driven by planar curve, $C_0 \rightarrow C_1$, we allow the parameter h to change from h_0 to h_1 . The Jacobian determinant of the transformation is

$$\det\left(\frac{\partial P_1}{\partial P_0}\right) = (1 - k_1 x_1)|C'_1(s)|dh_1 / (1 - k_0 x_0)|C'_0(s)|dh_0.$$

For locally volume-preserving transformation, $\det\left(\frac{\partial P_1}{\partial P_0}\right) = 1$, therefore,

$$(1 - k_1 h_1)|C'_1(s)|dh_1 = (1 - k_0 h_0)|C'_0(s)|dh_0.$$

Integrate on both sides, we get

$$(h_1 - \frac{1}{2}k_1 h_1^2)|C'_1(s)| = (h_0 - \frac{1}{2}k_0 h_0^2)|C'_0(s)|$$

Note that the formula is similar to the non-stretchable spine except that we need to take the magnitude of local tangent $|C'_1(s)|$ and $|C'_0(s)|$ into consideration.

3.2.2 3D stretchable spine

Let $C(s)$ represent a 3D curve in space. s may not be the arc-length parameter. Again, the derivative relationship between s and the arclength l is

$$dl = |C'(s)|ds.$$

Let $T(s)$, $N(s)$ and $B(s)$ be unit vectors representing the Frenet tangent, normal and binormal at $C(s)$. Let P denote an offset point near C , such that,

$$P = C(s) + xN(s) + yB(s).$$

Therefore,

$$\frac{dP}{ds} = C'(s) + x \frac{dN(s)}{ds} + y \frac{dB(s)}{ds},$$

or,

$$\frac{dP}{dt} = T(t)|C'(t)| + x \frac{dN(s)}{ds}|C'(t)| + y \frac{dB(s)}{ds}|C'(t)|$$

according to the derivative relationship.

On the other hand, Frenet Serret formulae give that,

$$\frac{dN}{dl} = -\kappa T + \tau B, \frac{dB}{dl} = -\tau N.$$

Therefore,

$$\begin{aligned} \frac{dP}{ds} &= (1 - kx)|C'(s)|T - \tau y|C'(s)|N + \tau x|C'(s)|B \\ \frac{\partial P}{\partial(s, x, y)} &= \begin{bmatrix} (1 - kx)|C'(s)| & -\tau y|C'(s)|N & \tau x|C'(s)|B \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T \\ N \\ B \end{bmatrix} \end{aligned}$$

Normal. In normal fleshing, we allow the parameter x to change from x_0 to x_1 .

Hence the Jacobian determinant of the transformation is

$$\det\left(\frac{\partial P_1}{\partial P_0}\right) = (1 - k_1 x_1)|C'_1(s)|dx_1 / (1 - k_0 x_0)|C'_0(s)|dx_0$$

For locally volume-preserving transformation, $\det\left(\frac{\partial P_1}{\partial P_0}\right) = 1$, therefore,

$$(1 - k_1 x_1)|C'_1(s)|dx_1 = (1 - k_0 x_0)|C'_0(s)|dx_0$$

Integrate on both sides, we get

$$(x_1 - k_1 x_1^2/2)|C'_1(s)| = (x_0 - k_0 x_0^2/2)|C'_0(s)|$$

The solution x_1 is the quadratic root of the above equation.

Binormal. In binormal fleshing, we allow the parameter y to change from y_0 to y_1 . The Jacobian determinant of the transformation is

$$\det\left(\frac{\partial P_1}{\partial P_0}\right) = (1 - k_1 x_1)|C'_1(s)|dy_1 / (1 - k_0 x_0)|C'_0(s)|dy_0$$

For locally volume-preserving transformation, $\det\left(\frac{\partial P_1}{\partial P_0}\right) = 1$, therefore,

$$(1 - k_1 x_1)|C'_1(s)|dy_1 = (1 - k_0 x_0)|C'_0(s)|dy_0$$

Integrate on both sides, we get

$$(1 - k_1 x_1) |C'_1(s)| y_1 = (1 - k_0 x_0) |C'_0(s)| y_0$$

The solution y_1 is linearly related to y_0 .

Radial. Radial fleshing is a combination of normal and binormal fleshing. A point P near the spine is expressed as,

$$P = C(s) + r \cos \theta N(s) + r \sin \theta B(s)$$

In the radial solution, the offset distance from the spine is adjusted from r_0 to r_1 .

The Jacobian determinant of the transformation is then expressed (in r_0 and r_1) as,

$$\det\left(\frac{\partial P_1}{\partial P_0}\right) = \frac{r_1 dr_1 (1 - k_1 r_1 \cos \theta_1) |C'_1(t)|}{r_0 dr_0 (1 - k_0 r_0 \cos \theta_0) |C'_0(t)|}.$$

Let $\det\left(\frac{\partial P_1}{\partial P_0}\right) = 1$. Then solve for r_1 , we get

$$-\frac{2}{3} k_1 \cos \theta_1 r_1^3 + r_1^2 = \frac{|C'_1(t)|}{|C'_0(t)|} \left(-\frac{2}{3} k_0 \cos \theta_0 r_0^3 + r_0^2 \right)$$

The solution r_1 is a cubic root of the above equation.

3.3 Extend to shell-based bending

Let $S(u, v)$ denote a two-dimensional sub-manifold, parameterized by u, v , of three-dimensional Euclidean space. Let P denote a offset point from S ,

$$P(u, v, h) = S(u, v) + hN(u, v).$$

Therefore,

$$\frac{\partial P}{\partial(u, v, h)} = \begin{bmatrix} S_u(u, v) + hN_u(u, v) \\ S_v(u, v) + hN_v(u, v) \\ N(u, v) \end{bmatrix}$$

The determinant of the above Jacobian is,

$$\det\left(\frac{\partial P}{\partial(u, v, h)}\right) = (S_u(u, v) + hN_u(u, v)) \times (S_v(u, v) + hN_v(u, v)) \cdot N(u, v).$$

Due to that the mean curvature $H(u, v)$ is the divergence of the normal, and the Gaussian curvature $G(u, v)$ is the cross product of the Hessian of the normal. The above equation can be reduced to the following:

$$\det\left(\frac{\partial P}{\partial(h, u, v)}\right) = (1 - 2hH + h^2G)|S_u \times S_v \cdot N|$$

Note that $(1 - 2hH + h^2G)$ can be replaced by $(1 - hK_1)(1 - hK_2)$ where K_1 and K_2 are principle curvatures.

We allow the parameter h to change from h_0 to h_1 during shell-based deformation. In order for the deformation to be locally volume preserving, we have,

$$\det\left(\frac{\partial P_1}{\partial P_0}\right) = 1$$

Therefore,

$$(1 - 2h_1H_1 + h_1^2G_1)|S_{1u} \times S_{1v} \cdot N_1|dh_1 = (1 - 2h_0H_0 + h_0^2G_0)|S_{0u} \times S_{0v} \cdot N_0|dh_0$$

Integrate on both sides, we get

$$h_1 - h_1^2H_1 + \frac{h_1^3}{3}G_1|S_{1u} \times S_{1v} \cdot N_1| = h_0 - h_0^2H_0 + \frac{h_0^3}{3}G_0|S_{0u} \times S_{0v} \cdot N_0|$$

Therefore, the updated offset distance h_1 is the solution of the above cubic equation.

3.4 Implementation Schedule

We plan to finish the thesis and write a paper on finishing the remaining work. Fig 10 shows the implementation plan (after submitting the proposal): The more accurate implementation and stretchable spine will take three weeks respectively. We plan to take four weeks experimenting with surface-driven bending. In parallel, we will be writing a paper, which would increase an overhead of two weeks on the schedule.

The estimation of running computer experiments is up to three months, which is reasonable time period that fits the schedule of graduating in Spring 2014 and

Month	Nov, 2013					Dec, 2013				Jan, 2014		
Week	1	2	3	4	5	6	7	8	9	10	11	12
3.1												
3.1.1												
3.1.2												
3.1.3												
3.2												
3.2.1												
3.2.2												
3.3												
write up paper												

Figure 10: Implementation schedule of future work

getting prepared for the job market before graduation. After January, we will be focusing on writing the thesis, applying for jobs, preparing interview slides, traveling for interviews and preparing the dissertation defense.

CHAPTER IV

SHORT SUMMARY

4.1 Bending driven by lower dimensional proxy

We have proposed several transformation frameworks in which the object transformation is driven by a lower dimensional proxy, which can be a planar curve, a 3D spine, or a shell surface. The advantage of using a lower dimension proxy in specifying shape deformation is that no parameter is required for modeling a collection of complex transformations consisting of bending, stretching, and twisting. Our approach first computes the closest projection of each point (the registration step), then the parameter of the closest projection on the lower dimensional proxy is preserved during deformation. In order for the deformation to simulate correct material behavior, we solve the problem of local volume preservation with closed-form formulas for each kind of proxy.

4.2 Volume preservation

Volume preservation is important to the modeling of incompressible deformation, such as the transformation of shapes filled with clay or water, and animating them with scientific correctness. We identify a family of closed-form solutions that preserve the local and global volume. The solutions are derived precisely from the constraint that the determinant of the Jacobian of the transformation should equal to 1. We show that our solution can be computed using analytic geometry in real-time.

4.3 Existing work and future plan

We attach here our two publications on existing work. The first paper [39], “Cuvature-based offset distance”, presents solutions on three different problems: (1) global volume compensation with minimized Hausdorff distance; (2) spine-driven bending with local area preservation; (3) machining with constant material removal rate. The second paper, “Fleshing”, extends the 2D solution of spine-driven bending to 3D, and presents a family of solutions for stylized deformations with local volume preservation. We also discuss the existence conditions, numerical errors and their treatments in these two papers.

Chapter 3 proposes future work. Specifically, we plan to use more accurate registration by computing the closest projection exactly, and interpolating the normal and curvature. We present our newly formulated mathematical frameworks that feature stretchable spines and shells in Section 3.2 and 3.3. We hope that the proposal for future work will get supported so that we can proceed to the implementation according to the schedule in Section 3.4.

APPENDIX A

PUBLICATIONS

We include here the two publications [39] [40] of the existing work discussed in this thesis proposal.

Curvature-based Offset Distance: Implementations and Applications

Wei Zhuo^a, Jarek Rossignac^a

^awzhuo3, jarek@cc.gatech.edu
College of Computing, Georgia Institute of Technology

Abstract

We address three related problems. The first problem is to change the volume of a solid by a prescribed amount, while minimizing Hausdorff error. This is important for compensating volume change due to smoothing, subdivision, or advection. The second problem is to preserve the individual areas of infinitely small chunks of a planar shape, as the shape is deformed to follow the gentle bending of a smooth spine (backbone) curve. This is important for bending or animating textured regions. The third problem is to generate consecutive offsets, where each unit element of the boundary sweeps the same region. This is important for constant material-removal rate during numerically controlled (NC) machining. For all three problems, we advocate a solution based on normal offsetting, where the offset distance is a function of local or global curvature measures. We discuss accuracy and smoothness of these solutions for models represented by triangle or quad meshes or, in 2D, by spine-aligned planar quads. We also explore the combination of local distance offsetting with a new selective smoothing process that reduces discontinuities and preserves curvature sign.

1. Introduction

In this paper, we discuss the use of normal offsetting [1] for volume or area preservation, where the offset distance is computed globally or locally from curvature measures. Specifically, we address the following three problems.

1.1. Adjust volume while minimizing Hausdorff error

We are given a base solid P with volume V_P . Typically, P is obtained by applying a small deformation to some starting solid S , which has volume V_S . The deformation may be the result of subdivision [2], smoothing [3], or advection of a fluid/swimmer interaction [4]. We want to obtain an offset solid O that is similar to P , but has volume V_S . Specifically, we define O as the shape that minimizes the Hausdorff distance, $\delta(P, O)$, between P and O , with O constrained to having volume V_S . Maintaining the volume is important in manufacturing applications where weight matters [5] and in physically based simulations where incompressibility matters [6]. The solution proposed here defines O as the constant distance offset (CDO) of P : $O = P^h$. We explain how to compute the correct distance h , both in two and three dimensions. We discuss accuracy in cases where P and O are represented by piecewise linear boundaries. In Fig. 1, we compare this solution to

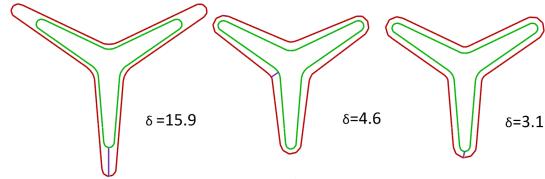


Figure 1: The original 3-branch-star base shape P (green) is shown with three offset shapes O (red) that enclose regions of the same area: global scaling (left), variable distance offsetting (center), and constant distance offsetting (right). The respective Hausdorff distances are: 15.9, 4.6, and 3.1. A line segment connecting P and Q indicate where the Hausdorff distance is reached. On the right, all points are at the Hausdorff distance from the other set.

global scaling and to variable distance normal offsetting (discussed in Sec. 1.3).

1.2. Preserve local area during spine bending

We are given a portion of a image R . R roughly aligned along a smooth spine curve P . Note that P does not need to be the medial axis of R and that the width of R may vary along P . We are also given a bent version \bar{P} of P . We assume that \bar{P} and P have identical length and are both parameterized by arc-length. Assume that each point O of R has a unique closest projection on P . We want a locally area-preserving homeomorphism H that maps point $O = P(s)+rN(s)$ to point $\bar{P}(s)+h\bar{N}(s)$, where

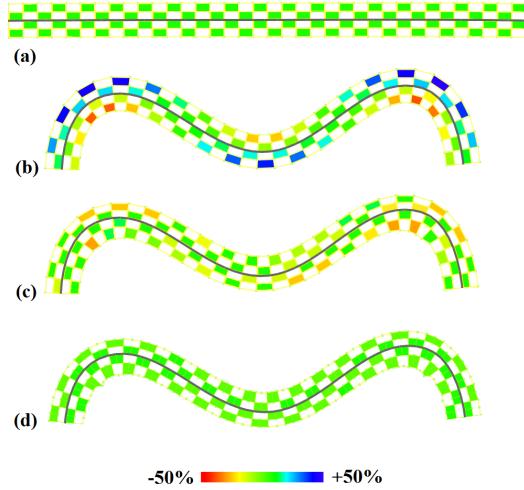


Figure 2: On the top (a), we show a texture region painted with an axis-aligned checkerboard pattern along a straight spine curve P . Below (b), we show a deformed version \bar{P} of the spine and the result of a mapping where $h = r$. The squares of the checkerboard are colored to indicate area preservation (more green), compression (more red), or dilation (more blue). Below (c), we show the proposed corrected mapping. At the bottom (d), we show the proposed corrected mapping while doubling the sampling density. Notice that this increased sampling reduces area errors significantly.

$N(s)$ is the normal to P at $P(s)$ and $\bar{N}(s)$ is the normal to \bar{P} at $\bar{P}(s)$. By locally area-preserving, we mean that any subset Q of R has same area as its image $H(Q)$.

The approach that we advocate here defines h in terms of r and the curvature $k(s)$ of P at $P(s)$ and the curvature $\bar{k}(s)$ of \bar{P} at $\bar{P}(s)$. For an exact solution h to exist, r must fall within a specific range defined by $k(s)$ and $\bar{k}(s)$. In Fig. 2, we compare this “fleshing” solution to the common skinning solution with $h = r$. We also discuss the computational and accuracy advantages of the spine-aligned grid, as shown in Fig. 2, over an axis aligned grid.

1.3. Generate contours for constant material removal

We are given the planar boundary P of a pocket to be machined, and we want to compute a series, $\{O_j\}$, of concentric variable-distance normal offset contours. For each contour, we want to adjust the offset distance locally, so that the area of a segment of the corridor between two consecutive contours is proportional to the length of that segment. More precisely, consider an animation that moves all points of O_j along their normal until they reach their offset point on O_{j+1} . For any connected subset S of O_j , let u denote its length. Our objective is to ensure that the region swept by S during this

animation has area ur , where r is a given nominal depth. This is important because NC machining is most efficient when the cutter advances at constant speed (tangentially along a contour O_i) and removes a constant amount of material per unit of time [7]. Our solution combines two steps: (1) a variable distance offset where the local offset distance h is computed from the nominal distance r and the local curvature k of O_j using a simple variation of the formulation discussed above, and (2) a selective smoothing, which reduces the sharp features introduced by step (1) and ensures that the curvature at a point does not change sign during offsetting. In Fig. 3, we compare constant distance offsetting, variable distance offsetting, and the proposed solution which combines steps (1) and (2).

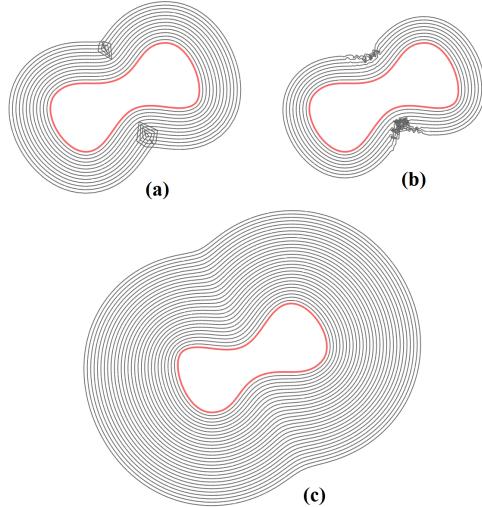


Figure 3: We show a series of contours produced by constant distance offsetting (a), curvature-based distance offsetting (b), and curvature-based distance offsetting with selective smoothing (c). The successive constant distance offsets (a) do not preserve a constant area-to-length ratio and produce self-intersections for larger offset distances. Successive curvature-based offsets (b) preserve that ratio, but exhibit an increasing amount of discontinuities where the curvature of the previous offset changes rapidly (we only render the first few contours). The proposed combination of curvature-aware offsetting and selective smoothing (c) produces concentric offset contours that are smooth and approach a constant area-to-length ratio. The selective smoothing ensures that the curvature at each point maintains its sign or becomes zero. Hence, the process converges towards a convex shape, as can be extrapolated from the drawing.

1.4. Summary of contributions

The solutions to all three problems are based on a curvature-based distance correction, which maps a

nominal distance r to a distance h . In two dimensions, assuming that k is the curvature, h is a specific root of

$$\frac{1}{2}kh^2 + h - r = 0 \quad (1)$$

In three dimensions, assuming that g is the Gaussian and m the mean curvature, h is a specific root of

$$\frac{1}{3}gh^3 + mh^2 + h - r = 0 \quad (2)$$

The derivation of these equations and their prior use for constant area or volume offsetting is discussed in the next section. Our contributions comprise the following:

- 1) To solve the first problem of constant distance offsetting for a desired volume change, we generalize the Steiner formula [8] for the volume change under constant distance offsetting to non-convex solids as well as to higher genus solids, and we describe an efficient implementation. We also analyze the error sensitivity of our formula, study the impact of sampling density on its accuracy, and report the results on benchmark curves and surfaces.
- 2) To solve the second problem of local area preservation during skeletal bending, we have adapted the formulation (Equ. 1) originally developed by Chirikjian [9] for locally area-preserving bending. Chirikjian discusses divergence-free deformation for continuous models. We explore its use for deforming discrete, texture-mapped quads to follow the bending of a polygonal spine. Specifically, we propose the use of a spine-aligned grid, and argue its advantages over axis-aligned grids.
- 3) To address the third problem of constant material removal modeling, we build upon the solution proposed by Moon [7], but show that it produces sharp discontinuities of the offset curve near concave features. We propose a novel selective smoothing technique which eliminates these sharp features while preserving the curvature sign between the original points and their offsets.

2. Prior Art

In this section, we discuss relevant prior work in constant distance offsetting, variable distance offsetting, volume correction, and skeleton-driven shape deformations.

2.1. Constant distance offsetting

The constant-distance offset (CDO) S' of a solid S by distance r [10], also called dilation, is formulated as

the Minkowski sum [11] of S with a ball of radius r centered at the origin. It may also be expressed as the union of all balls of radius r with center in S . S' contains all points at distance r or less from S . Steiner [8] has derived formulae for the area change and volume change under constant distance offsetting for the special cases of convex sets of genus zero. Here in Sec. 4 we prove its generalization to non-convex solids and to higher genus solids.

CDO operations are important in planning and simulating NC-machining processes [12], where they are used to generate constant thickness layers of material to be removed by successive machining passes, and for creating fillets and blends [13] by offsetting the solid and then its complement or vice versa. In 2D, CDO preserves the domain of shapes bounded by piecewise-circular curves [14]. In 3D, we obtain our approximation by offsetting each vertex by a constant distance along an estimated vertex normal. Numerical and topological accuracy issues of CDOs of solids bounded by triangle meshes and polyhedral surfaces have been investigated in various applications [12] [15].

2.2. Variable distance offsetting

Variable-distance offsetting (VDO) is specified by assigning a distance $h(s)$ to each point $P(s)$ of the base shape P (curve in 2D or surface in 3D). Three different interpretations of this specification have been compared in [16]. The radial offset is the union of balls $(P(s), h(s))$. The ball offset [17] is the union of balls of diameter $h(s)$ that are tangent to P at $P(s)$. Finally, the normal offset [1] is the union of all line segments of length $h(s)$ that are normal to P at $P(s)$. In all three cases, under sufficient assumptions on the smoothness and curvature of P , there is a bijective mapping between P and a portion of the boundary of the offset shape, which may be formulated as an envelope of a set of line segments or balls. (Note that each formulation imposes a different set of constraints on the relation between the offset distance function and the curvature of P [1].) The shape and curvature of these envelopes may be computed efficiently [16]. Here, we restrict our attention to the normal offset, hoping that the other two interpretations will be investigated later. One issue addressed in this paper is the computation of the offset distance field $h(s)$ that distributes the “invaded” space uniformly. Let P be a surface in 3D. Let, Q be a subset of P , and R be the region swept by Q during the offset. We want to compute a variable offset distance function $h(s)$ such that the ratio r of R ’s volume over the area of Q is a constant. If P is a curve in 2D, r is the ratio of the area of R over the arc-length of Q . This equi-volumetric offsetting

has been investigated by Moon [7] [18] for NC machining, so as to ensure a constant material-removal rate, rather than constant depth of removal. Moon has shown that, in valid situations where the curvature is smaller than some limit defined in terms of r , $h(s)$ may be formulated as the root of a quadratic equation, for the 2D case, and of a cubic equation, for the 3D case. Specifically, in 2D, h is the root of $\frac{1}{2}k(s)h(s)^2 + h(s) - r = 0$, where $k(s)$ is the curvature of P at $P(s)$. In 3D, $h(s)$ is the root of $\frac{1}{3}g(s)h(s)^3 + m(s)h(s)^2 + h(s) - r = 0$ where $g(s)$ is the local Gaussian curvature and $m(s)$ is the local mean curvature of P at $P(s)$. These curvature based distance functions have been studied by Hagen and Hahmann as generalized focal surfaces [19] as a tool for surface interrogation. We build our local offsetting solutions to the volume compensation and to the area-preserving bending on these equations.

2.3. Skeleton-driven deformations

Consider the planar shape S to be the union of an infinite set of disjoint line segments intersected at their midpoints by a continuous spine P . Let $2h(s)$ and $a(s)$ define the length of the line segment and its angle to the tangent to P at $P(s)$. Cavalieri's principle [20] implies that, when bending P , the area of the convex hull of two infinitely close line segments remains constant regardless of the shape of S , as long as we preserve $h(s)$ and $a(s)$ and do not bend $P(s)$ excessively (ensuring that the radius of curvature at $P(s)$ does not exceed $h(s)$). Although this solution preserves the area of each convex hull of consecutive two line segments, it does not preserve the local area on each side of the spine, as discussed in the introduction. Several approaches have been proposed to maintain a constant local area of a region as its spine is bent. Chirikjian [9] has derived the quadratic equation mentioned above by constraining the Jacobian of the deformation to be 1, so as to make it locally area preserving. When the spine bend exceeds the local limit, the normal offsetting is no longer appropriate. More general techniques for skinning and fleshing with locally-preserving bending have been proposed by Rohmer and colleagues [21]. They adjust both the direction and distance of the offsetting and solve for an optimal solution that favors locality.

3. Curvature-based Offset Distance Computation

In this section, we discuss implementation and accuracy issues of computing the curvature-based offset distance. For implementation simplicity, we define a function f in 2D and in 3D, which returns the proper offset

distance, when it exists within the allowable range, or the appropriate range bound otherwise. We use a subscript (f_{2D} and f_{3D}) to distinguish the 2D and 3D versions of f . We also discuss how to select the proper root in each case.

3.1. Function interface and capping

f_{2D} takes as input the signed curvature k and the reference distance r respectively. The output $h = f_{2D}(k, r)$ is the quadratic root $\frac{-1+\sqrt{1+2kr}}{k}$ of Equ. 1 when $1+2kr > 0$. Otherwise, f caps the value of h and returns the limit $-1/k$ so as to prevent a local self-intersection.

f_{3D} takes as input the signed Gaussian curvature g , the mean curvature m and the reference distance r . The output $h = f_{3D}(g, m, r)$ is the valid cubic root of Equ. 2. Notice that if $g = 0$, then h is computed via the 2D solution discussed above, as $f_{2D}(2m, r)$. Otherwise, we need to select the proper real root and to ensure that the solution is capped to an allowable bound. Moon [18] has derived the existence condition and the monotonic region where the valid root exists. In our implementation, we use a change of variables: $h^* = \frac{h}{r}$, $g^* = gr^2$ and $m^* = mr$. Then if $2\sqrt{m^{*2}-g^{*2}} - m^* > 3(m^* - \sqrt{m^{*2}-g^{*2}})$, there is a unique positive real root in $[0, \frac{1}{\sqrt{m^{*2}-g^{*2}-m^*}}]$. Otherwise, no valid real root exists and we output the maximum offset distance that is free from a local self intersection.

3.2. Error Sensitivities

Estimating curvature from a sampling of a smooth curve will produce an incorrect offset distance. Below we show that the error in h is a linear function of the errors in the curvature estimation, both in 2D and in 3D.

Let ϵ_x represent a small variation in the variable x . Assume that r is a constant. For 2D, we take the derivative of Equ. 1 and arrive at

$$\frac{h^2}{2}\epsilon_k + kh\epsilon_h + \epsilon_h = 0$$

From this, we conclude that ϵ_h is proportional (\propto) to ϵ_k :

$$\epsilon_h \propto \frac{h^2}{1+kh}\epsilon_k$$

Similar for 3D, we take the derivative of Equ. 2 and obtain

$$\epsilon_h \propto \frac{\epsilon_g h^3 + \epsilon_m h^2}{1+2mh+gh^2}$$

Therefore, the numerical error in the output of f is linear in the errors of its inputs when $kh > 0$ in 2D, or $2mh + gh^2 > 0$ in 3D.

3.3. Curvature approximation

Densely sampled polylines and polygonal meshes are often used in modeling solids with smooth boundaries whose parametric expression may not be conveniently available. Hence, we adopt discrete formulas to evaluate the curvatures.

3.3.1. Local curvatures

Let P denote a watertight quad or triangle mesh and P_i a vertex of P . The local curvature at P_i can be evaluated from its one-ring neighbors $\{Q_j\}$. In 2D, the discrete curvature k_i may be conveniently calculated by fitting a parabola to P_i and its neighbors. In 3D, we use the discrete formulas proposed by Meyer, et. al. [22]. Specifically, the local area A_i associated with P_i is approximated by the area sum of incident Voronoi cells. The gradient of A_i with respect to P_i , also known as the discrete Laplace Beltrami operator, has the following closed form [23]:

$$\nabla A_i = \frac{1}{2} \sum_j \left(\frac{P_i Q_{j-1} \cdot Q_{j-1} Q_j}{|P_i Q_{j-1} \times Q_{j-1} Q_j|} + \frac{P_i Q_{j+1} \cdot Q_{j+1} Q_j}{|P_i Q_{j+1} \times Q_{j+1} Q_j|} \right) P_i Q_j \quad (3)$$

Then, the local mean curvature is approximated by a scaled dot product of ∇A_i with the unit normal at P_i . The local Gaussian curvature is approximated by the angle deficit at P_i [22].

3.3.2. Global curvatures

Let A_P denote the total surface area of P . We refer to the surface integral of Gaussian curvature divided by A_P as the global Gaussian curvature (g_P) and the surface integral of mean curvature divided by A_P as the global mean curvature (m_P). The integrated Gaussian is intrinsic to P and equals $2\pi\chi_P$, where χ_P is the Euler characteristic of P . ($\chi_P = V - E + F$ where V, E, F are numbers of vertices, edges and faces.) Therefore,

$$g_P = \frac{2\pi\chi_P}{A_P} \quad (4)$$

The surface integral of mean curvature is related to the bending energy [24], which we denote as E_P . Note that E_P can be approximated by the scaled sum of $|\nabla A_i|$ at each vertex. Therefore,

$$m_P = \frac{E_P}{A_P} \quad (5)$$

In 2D when P denotes a Jordan curve, its integrated curvature is intrinsic and equals 2π [25]. Let L_P denote the length of P . The global curvature of P , k_P , is defined as

$$k_P = \frac{2\pi}{L_P} \quad (6)$$

Note that a global curvature has the same unit as its local counterpart.

4. Dilation with Prescribed Volume Change

Consider a 3D shape P with volume V_P . We want to compute O from P by a single step of dilation, so that the enclosed volume is increased by a prescribed amount ΔV . We first discuss methods that are not based on curvature measures. Then we present our solution.

4.1. Uniform scaling

The work of Desbrun et. al. [23] introduces a simple approach of rescaling P around its barycenter C by a uniform amount s :

$$O = C + s(P - C) \quad (7)$$

where $s = \sqrt[3]{\frac{V_p + \Delta V}{V_p}}$. Uniform scaling guarantees that the enclosed volume is increased exactly by ΔV . However, this approach generates unbounded Hausdorff error between O and P (Fig. 1).

4.2. Linearized solution

In contrast, when a constant distance normal offset by a distance h is used, the Hausdorff error is exactly h (assuming that h is smaller than the least feature size of the shape). When using a constant distance offset (CDO), to increase the volume of a solid by ΔV , one must compute the proper offset distance h . One approach [21] is to use $h = \frac{\Delta V}{A_P}$. We compare below this approximate solution to the one proposed here.

4.3. Normal offset based on the global curvature

The correct solution defines h as the appropriate root computed by f_{2D} or f_{3D} as explained earlier in Sec. 3.1. We include below the derivation of this result.

4.3.1. 2D

Let P denote a Jordan curve of length L_P . Let, $k(s)$ and $N(s)$ be the signed curvature and the unit normal of P at $P(s)$. The curvature $k(s)$ is the derivative of the unit normal. Hence, we have the following expression of the area increase ΔA associated with offsetting P by a constant distance h :

$$\begin{aligned} \Delta A &= \iint_{\gamma \in [0, h]} \left| \frac{\partial(P(s) + \gamma N(s))}{\partial s} \right| dy ds \\ &= h L_P + \frac{h^2}{2} \int k(s) ds \end{aligned}$$

By the Total Curvature Theorem [25], we have

$$\int k(s)ds = 2\pi$$

Therefore we arrive at,

$$\frac{\pi}{L_P} h^2 + h - \frac{\Delta A}{L_P} = 0 \quad (8)$$

Hence to compensate for the area change ΔA , we need to offset the curve P by a constant distance h computed by $h = f_{2D}(\frac{2\pi}{L_P}, \frac{\Delta A}{L_P})$. Or equivalently, $h = f_{2D}(k_P, \frac{\Delta A}{L_P})$ using the global curvature defined in Equ. 6.

4.3.2. 3D

Let $P(u, v)$ denote a point on a surface P parameterized by u and v . We derive the exact expression of the volume increase when offsetting $P(u, v)$ by a constant distance h . Let $m(u, v)$ and $g(u, v)$ represent the local mean and Gaussian curvature of P at (u, v) . Since the mean curvature is the divergence of the unit normal and the Gaussian curvature is the determinant of its Hessian, the volume increase ΔV can be expressed as follows:

$$\begin{aligned} \Delta V &= \iiint_{y \in [0, h]} |\nabla(P(u, v) + \gamma N(u, v))| dy dudv \\ &= h \iint |\nabla P| dv du + \frac{1}{2} h^2 \iint \nabla \cdot N dv du \\ &\quad + \frac{1}{3} h^3 \iint |\nabla N| dv du \\ &= h A_P + h^2 \iint m(u, v) dv du \\ &\quad + \frac{1}{3} h^3 \iint g(u, v) dv du \end{aligned}$$

By the Gauss-Bonnet Theorem [25], we have

$$\iint g(u, v) dv du = 2\pi \chi_P$$

where χ_P is the Euler characteristic of P which is $2 - g$ for a genus- g surface. The other integral term is the total integral of the mean curvature: $E_P = \iint m(u, v) dv du$. Therefore, we arrive at:

$$\frac{2\pi\chi_P}{3A_P} h^3 + \frac{E_P}{A_P} h^2 + h - \frac{\Delta V}{A_P} = 0 \quad (9)$$

Hence to increase the current volume by ΔV , we offset P by $h = f_{3D}(\frac{2\pi\chi_P}{A_P}, \frac{E_P}{A_P}, \frac{\Delta V}{A_P})$. Notice that the definition of global curvatures in Equ.4 and Equ. 5, the solution can also be written as $h = f_{3D}(g_P, m_P, \frac{\Delta V}{A_P})$.

4.4. Proof of minimizing Hausdorff error

Let P , O and Q either be regularized planar regions or solids. Assume that $O = P^d$ for some positive distance d . (If instead we want a negative d , the argument below will hold for the complements of P , O and Q and still support our conclusion.) We will prove that $\forall Q \neq O$, $V_Q = V_O \Rightarrow H(Q, P) > H(O, P)$, where H defines Hausdorff distance and V_X denotes the area or volume of X .

Assume that $V_Q = V_O$. First, we note that Q cannot be a proper subset of O , otherwise we would have $V_Q < V_O$. Second, we note that Q cannot contain any point q outside of O , otherwise we would have the distance from q to P , $d(q, P) > d$ (Since O includes all points at distances less or equal to d from P) and hence $H(Q, P) > d$. From these two observations (Q is not a proper subset of O and Q is a subset of O), we conclude that if $Q \neq O$ then $H(Q, P) > H(O, P)$. Hence, O is Hausdorff distance minimized. \square

4.5. Implementation

We have implemented the three volume correction schemes (Uniform scaling, Linearized, and Curvature-based solutions) on quad as well as triangle meshes. Our implementation uses a Corner Table [26] representation and the associated corner operators. The whole process is only a few lines of code. First, to compute the global mean curvature m_P we sum the area gradient at each vertex and divide it by 3 for triangle meshes or 2 for quad meshes. Then, the normal at each vertex is the weighted sum of the normals of the incident triangles scaled by their areas. Then, we compute the surface area A_P of P (as the sum of triangle areas), the volume V_P (as a sum of signed volumes of the tetrahedron formed by each triangle with the origin). For a quad mesh, we treat each quad face as a bi-linear patch interpolating the four face vertices. We compute the total volume and the total surface area as the sums of the sub-volume and the sub-area associated with each bi-linear patch, using formulae presented in [21]. The extraction of the proper root of the cubic polynomial was discussed in Sec. 3.1. Although we have not optimized the code, the whole process of computing the corrected offset distance and of performing the offsetting is instantaneous (it takes a very small fraction of a second for all models tested).

We evaluate P 's barycenter C as the area-weighted sum of geometric centers of all faces of P divided by A_P . The Hausdorff distance between P and O is approximated by

$$\max\{\max\{d(p, O), p \in P\}, \max\{d(o, P), o \in O\}\}$$

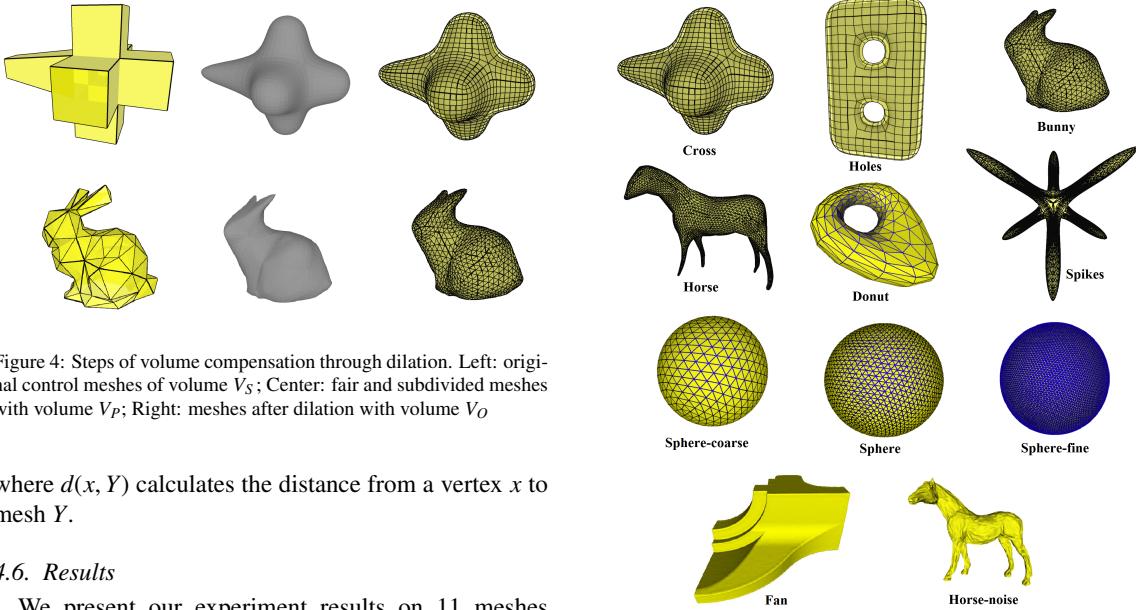


Figure 4: Steps of volume compensation through dilation. Left: original control meshes of volume V_S ; Center: fair and subdivided meshes with volume V_P ; Right: meshes after dilation with volume V_O

where $d(x, Y)$ calculates the distance from a vertex x to mesh Y .

4.6. Results

We present our experiment results on 11 meshes shown in Fig. 5. In these examples, 9 (Cross, Holes, Bunny, Horse, Donut, Spikes, Sphere-(coarse, fine)) are obtained from coarse solids by Catmull [2] or Butter-fly [27] subdivision and smoothing [3] steps shown in Fig. 4. Mesh ‘‘Horse-noise’’ is obtained by adding random noises to the subdivided horse model. We prescribe the desired volume change ΔV , and want to offset P to produce a solid $O = P^h$ with volume $V_P + \Delta V$. We report in Tab. 1 the number of vertices n_V , volume V_P , area A_P for each mesh P . The volume after correction is denoted as V_O . It is not exactly $V_P + \Delta V$ due to numerical errors. We measure the discrepancies between $V_P + \Delta V$ and V_O in terms of ϵ defined as follows:

$$\epsilon = \frac{|V_P + \Delta V - V_O|}{V_P} \quad (10)$$

Tab. 1 shows the errors of the linearized solution (ϵ_{linear}) where $h = \frac{\Delta V}{A}$ and the errors of our solution based on the global curvatures ($\epsilon_{\text{curv.}}$). The results show that in general the curvature-based solution is about 3 times more accurate than the linearized solution. We also report the Hausdorff error between P and O . For meshes that contain parts that are long and thin, the Hausdorff error (δ_{scaling}) produced by uniform scaling is much larger than the Hausdorff error ($\delta_{\text{curv.}}$) produced by our solution based on global curvatures. For spheres, δ_{scaling} and $\delta_{\text{curv.}}$ are roughly the same. We also observe that for all models tested, repeating the offsetting with the correct solution (Equ. 9) for h (each time using the remaining volume error as inputs) three or four times reduces the relative error to 0.00003% or less.

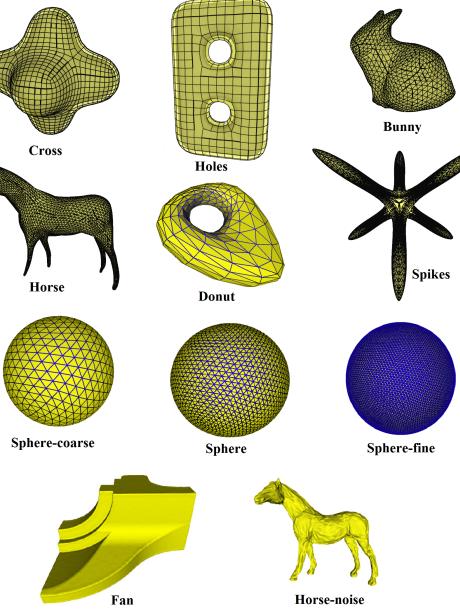


Figure 5: Mesh models used in our experiments: Cross, Holes, Bunny, Horse, Donut, Spikes, Sphere-noise, Sphere-fine, Fan, Horse-noise

5. Spine Bending with Local Area Preservation

Volume and area preserving deformation are often keys to simulations with physical realism. The fundamental idea for locally volume/area-preservation is to make the deformation field divergence-free, which implies that the Jacobian determinant is 1.

Here we consider the problem in 2D. The spine is represented by a polygonal curve produced by subdivision or by a dense sampling of a smooth curve. Applications of bending curves range from rendering brush strokes with variable thickness and textures [28] to image and shape manipulation [29]. We notice that a ribbon-style framework suitable for bending an open continuous curve was first proposed by Alan Barr [30]. The framework provides an efficient method for a planar deformation controlled by a skeletal curve. We present below a locally area-preserving shape manipulation application based on this framework.

5.1. Continuous model

We include here a derivation of Equ. 1 for bending with a continuous curve. Given a skeletal curve which we denote as $P(s)$, a nearby point O is expressed as:

$$O(s, r) = P(s) + rN(s)$$

Model	n_V	V_P	A_P	ϵ_{linear}	$\epsilon_{curv.}$	$\delta_{scaling}$	$\delta_{curv.}$
Cross	3198	2.15e7	7.05e5	2.5%	0.12%	12.5	2.8
Holes	1922	4.26e7	7.26e5	1.9%	0.11%	14.8	5.7
Bunny	1522	5.19e6	1.73e5	2.4%	0.065%	6.3	2.9
Horse	4002	8.29e6	3.12e5	2.95%	0.21%	12.3	2.5
Donut	256	1.43e7	4.03e5	2.2%	0.18%	7.4	4.8
Spikes	3842	7.69e6	5.165e5	3.20%	0.65%	17.6	1.4
Sphere-coarse	194	3.01e7	4.73e5	3.1%	0.62%	5.8	5.7
Sphere	770	3.08e7	4.78e5	3.8%	0.83%	6.4	6.4
Sphere-fine	3074	3.08e7	4.78e5	3.9%	1.0%	6.4	6.4
Fan	25895	5.022e7	1.07e6	2.2%	1.3%	8.7	4.5
Horse-noise	4002	1.21e7	4.69e6	2.7%	0.96%	11.5	2.3

Table 1: Mesh statistics and results of different volume-correction schemes corresponding to the models in Fig. 5

here r is the distance from O to its orthogonal projection on P . We denote the skeletal curve after length-preserving bending as \bar{P} with its unit normal and curvature denoted as \bar{N} and \bar{k} . The deformed position \bar{O} is then:

$$\bar{O}(s, r) = \bar{P}(s) + h\bar{N}(s)$$

Setting $h = r$ produces an approximate solution as previously discussed in Sec. 1.2. However, the deformation is not locally area-preserving as the local rate of expansion varies depending on the curvatures at $P(s)$ and $\bar{P}(s)$. Hence, $h \neq r$. By the chain rule, we have:

$$\frac{\partial \bar{O}}{\partial O} = \frac{\partial \bar{O}}{\partial (s, h)} \frac{\partial (s, h)}{\partial (s, r)} \frac{\partial (s, r)}{\partial O}$$

By setting the determinant of the above transformation to 1, we have:

$$\frac{dh}{dr}(1 + h\bar{k}(s))(1 + rk(s))^{-1} = 1$$

Therefore,

$$\frac{\bar{k}(s)}{2}h^2 + h - (r + \frac{r^2}{2}k(s)) = 0$$

The solution for h is a curvature-based distance which can be computed by $h = f_{2D}(\bar{k}(s), r + \frac{k(s)}{2}r^2)$.

5.2. Discretization

To bend an image, the designer specifies the initial and final spine curves. We use a grid of quads and paint the bent image as a texture onto the deformed quads. One could do this using an axis-aligned grid, but such an approach has two drawbacks: (1) there is an expense of computing the closest projection of each grid point onto the initial spine curve, and (2) aliasing artifacts occur

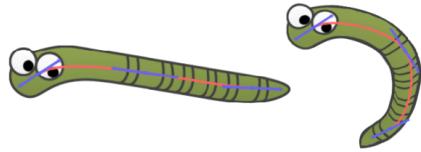


Figure 6: The user draw a initial curve (left) over an image and a deformed curve (right). The deformed image is rendered as a texture mapping over the spine-aligned grid. We preserve the length of the spine by keeping the number of samples and the distance between consecutive samples as constants, when sampling from a curve manipulated by the user.

when the spine curve is not sufficiently sampled, as several grid points that would project on different points of a continuous spine may have, as closest projection, the same vertex of a polygonal approximation. To alleviate these drawbacks, we advocate using a spine-aligned grid, as shown in Fig. 2. For simplicity, we sample the smooth spine curve so that all edges of its polygonal approximation have the same length. We generate the initial grid by estimating the normal at each vertex P_i of the initial spine (as being orthogonal to the line passing by its neighbors) and by generating offset points in both directions by jr , with j being an integer in some desired range. At each such grid-point, we record its coordinates in the image as texture coordinates. To display the deformed image, we use the same process to establish the normal at each vertex of the bent spine, and generate the corresponding grid points, but instead of offsetting them by jr , we offset them by $f_{2D}(\bar{k}, jr + \frac{k}{2}(jr)^2)$, where k and \bar{k} are the local curvatures before and after bending. Then we render the grid quads with texture mapping. An example of this bending process is shown in Fig. 6.

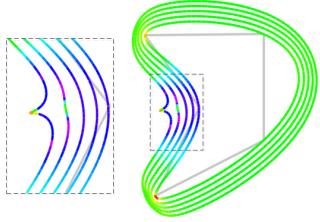


Figure 7: A family of curvature-based distance offsets. Notice that the offset curve may contain sharp pointy protrusion at concave side of the spine curve when r approaches the limit $-\frac{1}{2k(s)}$.

5.3. Limitations

As the half-width of the grid approaches the validity limit discussed above, the corrected offset distance increases rapidly, creating a spike, as shown in Fig. 7. Hence, in practice, we must limit the width of the area of the picture upon which we operate or the amount of curvature change at every point between the initial and final spines. Specifically, we limit $|k|$ to $[0, \frac{1}{2r}]$ where k is the local curvature and r is the half-width of the grid. In practice, to avoid spikes, we limit $|k|$ to $[0, \frac{1}{2.5r}]$.

6. Constant Material Removal Rate

6.1. Machining

We recall the quadratic formula proposed by Hwan Pyo Moon [7] in the context of machining:

$$\frac{1}{2}k(s)h(s)^2 + h(s) - r = 0$$

where $k(s)$ is the local curvature of the progenitor curve P , $h(s)$ is the depth of cut, and r is the material removal rate to feedrate ratio. Moon argues its importance in NC milling with constant power consumption. General milling tools have sufficient degrees of freedom which allow them to follow arbitrary planar paths. One of the challenges is to define a tool path that lead to constant material removal rate in milling for a target shape modeled by P . Since we want to keep the translational speed of the milling tool as constant as possible, the removed area per unit length should also be constant in order to achieve stable power consumption. Let this constant be r , solving the above equation gives the offset distance that defines the tool path with removed area per unit length equal to r .

6.2. Successive offsets

In practice, the tool path could consist of a set of concentric offsets from P . They form a set of *successive*

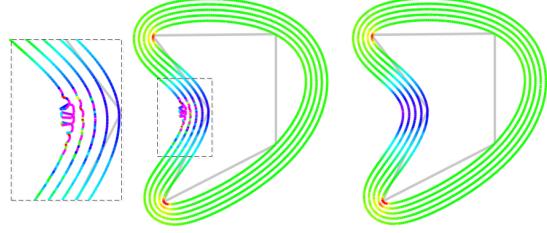


Figure 8: A set of successive curvature-based distance offsets. Left: direct offset curves without fairing; Right: the same set of offsets with selective smoothing

offsets $\{O^j\}$, $j = 1, 2, \dots$ from P :

$$\begin{aligned} O^1(s) &= P(s) + f(k_{P(s)}, r)N_{P(s)} \\ O^{j+1}(s) &= O^j(s) + f(k_{O^j(s)}, r)N_{O^j(s)} \end{aligned}$$

6.3. Loss of smoothness

It is known in differential geometry that the curvature transformation $k_{P(s)}$ is a second-order operator on the parametric curve $P(s)$. Naturally, the curvature-based distance function $f(k_{P(s)}, r)$ is second order as well. Hence only C^{d-2} continuity is observed in the offset when $P(s)$ is C^d continuous. To verify this loss of smoothness when P is approximated by dense polyloop, we show a set of successive offsets on a dense polyloop P produced by the $J_{1,5}$ subdivision scheme [31] whose limiting curve is of C^4 continuity.

Fig. 8 (Left) shows the result of directly applying f_{2D} to discrete curvatures evaluated at points of P and $\{O^j\}$. The first two offset curves appear smooth. However, the third appears jaggy and the fourth contains self-intersections. These discontinuities result from large differences of curvature estimates between neighboring vertices. Variances in evaluating the discrete curvatures could cause the offset to contain unwanted local convexities and concavities, and further increase the curvature variances in the offset curve. Therefore, we propose below an iterative algorithm, *selective smoothing*, for successively generating visually smooth offset curves.

6.4. Selective Smoothing

We observe that changes in the sign of the curvature are undesirable in generating a smooth offset curve. Hence, our smoothing strategy focuses on producing a curvature-compatible offset curve, where a point with non-negative curvature is mapped to a offset point with non-negative curvature, and the same for non-positive curvature.

Selective Smoothing is similar to the Laplacian smoothing except that only points with non-compatible

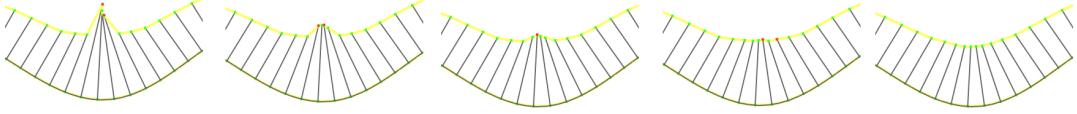


Figure 9: The 1st, 3rd, 6th, 12th, 14th iteration of selective smoothing. Points with incompatible curvatures are shown in red.

curvatures are subject to the operation. It consists of two steps in each iteration (Fig. 9): *Select* and *Smoothen*. Let k_i^o denote the discrete curvature at the i -th vertex on the offset curve O ; k_i and N_i denote the signed curvature and the unit normal at P .

- **Select:** Check each vertex O_i in O and put i into a smoothing list L if k_i and k_i^o are of different signs.
- **Smoothen:** Compute a list of Laplacian vectors $\{V_i\}$ at vertices of L ; Move each vertex of L along the unit normal N_i by the dot product of V_i and N_i .

Typically, as shown in Fig. 9, there are only a few incompatible points along the initial offset curve. As these are made compatible by a step of the selective smoothing, some of their immediate neighbors may become incompatible. However, the process converges rapidly. Fig. 8 (Right) and 3 (Right) show results of applying selective smoothing: in Fig. 8, unwanted noise is smoothed out while the rest of a curve is not modified; in Fig 3, we are able to generate a large series of consecutive offsets using this combination of curvature-based distance and selective smoothing.

6.5. Discussion and limitations

Consider now selective smoothing as a separate process. It could be used to smoothen a polygonal curve so that each vertex is either flat (has zero curvature) or has a curvature with a prescribed sign. Selective smoothing identifies incompatible vertices—those where the curve makes the wrong turn—and moves them to the average of their immediate neighbors. When a chain of incompatible vertices has the same prescribed curvature sign, repeating the process is essentially equivalent to Laplacian smoothing and converges to a straight line. However, selective smoothing can fail if the curve becomes self-crossing. When used as a smoothing to curvature-based normal offsetting, we restrict the motion of each vertex to be along the normal to the original curve. Furthermore, the extent of that motion is constrained by the cap on the corrected offset value ($|h|$ is confined to $[0, -1/k]$ if $k < 0$). Hence, allowable vertex

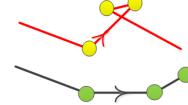


Figure 10: The yellow vertices are having compatible curvature signs with the green vertices on the black curve.

motions cannot create local loops. Therefore, we conjecture that our Selective Smoothing process will converge to a compatible curve. Of course, the offset curve may exhibit global self-intersections, which can be detected and should be prevented or resolved by trimming, if topological changes are desired. But such a global post-processing is necessary regardless of the smoothing step.

Finally, due to the discretization and numerical errors when evaluating k , an offset contour may still contain a local self-intersecting loop (Fig. 10). To detect these situations, we detect self-crossing along the offset curve and flag, as incompatible, all vertices between two consecutive self-crossing points. This heuristic works correctly only when the loops are isolated.

7. Discussion

This section discusses the impact of sampling density on the accuracy of locally area/volume distribution computed by the curvature-based normal offset. We compute variable distance normal offset from prototypical curve and surface patch (denoted as P). In order to show the error on both local and global scales, we divide P into a constant number of portions and define the following measures:

In 2D, we compute the sub-area a_k swept by offsetting the k th portion of P with length l_k . The local relative error for each portion is defined as $\delta_k = \frac{a_k}{l_k r} - 1$. We report the maximum absolute value, δ_{max} , and the mean absolute value, δ_{mean} , of the local relative errors for all portions of P . We also report the global relative error as $\delta_{global} = \frac{\sum_k a_k}{r \sum_k l_k} - 1$. δ_{global} measures the relative difference from the total-increased-area to perimeter ratio from the user-input reference distance r . In 3D, we

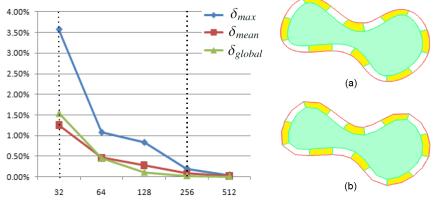


Figure 11: Dependence of the local and the global error on sampling density: (a) finely sampled curve that consists of 256 points. (b) coarsely sampled curve that consists of 32 points.

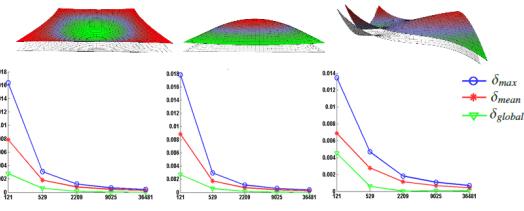


Figure 12: Dependence of the local and the global error on mesh resolutions

define similar measures which we use to analyze the errors associated with different types of surface patches. Fig. 11 shows values of δ_{max} , δ_{mean} and δ_{global} at 5 different sampling densities of a polygonal curve. Both the local and the global relative errors converge to zero as the subdivision depth increases. For example, the relative errors are less than 0.5% when there are 256 sample points on P . Fig. 12 shows values of δ_{max} , δ_{mean} and δ_{global} at different subdivision levels of bi-cubic surface patches. We collect statistics from three types of surface patches to avoid biases. Again, both the global and the local relative errors fall quickly as the sampling density increases. The relative errors are less than 0.5% when there are 529 sample points on each surface patch.

These results show that in general, the accuracy of even-area/volume distribution can be improved by increasing the sampling density.

8. Conclusion

In this paper, we have presented our study and implementation on the curvature-based offset distance for several applications. Specifically, we present a simple formulation of the offset distance and discuss its accuracy and smoothness, when computed on discrete models. We provide an exact formulation of the offset distance for adjusting the offset of 3D shapes by a constant distance offset. Our solution generalizes prior art which was limited to convex, zero-genus shapes. For bend-

ing images, we propose the use of an axis-aligned grid and the formulation of the offset mapping between two curved spines. Finally, for machining, we propose combining curvature-based local offsetting with an iterative selective smoothing process.

Acknowledgment

This project was funded by NSF grant #0811485, Collaborative Rsch: CPA-G&V-T: Aquatic Propulsion Lab. The authors would like to thank PhD students Mark Luffel, Jason Williams and Topraj Gurung for their help, Professors David Rosen and Yingjie Liu for their guidance and reviewers for their suggestions.

References

- [1] F. Chazal, A. Lieutier, J. Rossignac, Normal-map between normal-compatible manifolds, *Int. J. Comput. Geometry Appl.* 17 (5) (2007) 403–421.
- [2] E. Catmull, J. Clark, Recursively generated b-spline surfaces on arbitrary topological meshes, *Computer-Aided Design* 10 (6) (1978) 350–355.
- [3] G. Taubin, A signal processing approach to fair surface design, in: *SIGGRAPH*, 1995, pp. 351–358.
- [4] J. Tan, Y. Gu, G. Turk, C. K. Liu, Articulated swimming creatures, in: *ACM SIGGRAPH 2011 papers*, *SIGGRAPH '11*, 2011, pp. 58:1–58:12.
- [5] W. Nadir, I. Y. Kim, Structural shape optimization considering both performance and manufacturing cost, in: *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA '04, 2004, p. 4593.
- [6] B. Kim, Y. Liu, I. Llamas, X. Jiao, J. Rossignac, Simulation of bubbles in foam with the volume control method, *ACM Trans. Graph.* 26 (3) (2007) 1919–1927.
- [7] H. P. Moon, Equivolumetric offsets for 2d machining with constant material removal rate, *Computer Aided Geometric Design* 25 (6) (2008) 397 – 410.
- [8] J. Steiner, über parallele flächen, *Monatsberichte der Akademie der Wissenschaft zu Berlin (Monthly Report of the Academy of Sciences, Berlin)* (1840) 114–118.
- [9] G. S. Chirikjian, Closed-form primitives for generating locally volume preserving deformations, *Journal of Mechanical Design* 117 (3) (1995) 347–354.
- [10] J. Rossignac, A. A. G. Requicha, Offsetting operations in solid modelling, *Computer Aided Geometric Design* 3 (2) (1986) 129–148.
- [11] J. Serra, *Image Analysis and Mathematical Morphology*, Ac. Press, 1988.
- [12] T. Maekawa, An overview of offset curves and surfaces, *Computer-Aided Design* 31 (3) (1999) 165 – 173.
- [13] J. Rossignac, A. Requicha, Constant-radius blending in solid modeling, *ASME Computers In Mechanical Engineering (CIME)* 3 (1984) 65–73.
- [14] J. Rossignac, A. Requicha, Piecewise-circular curves for geometric modeling, *IBM Journal of Research and Development* 3 (1987) 129–148.
- [15] Y. Chen, H. Wang, D. W. Rosen, J. Rossignac, A point-based offsetting method of polygonal meshes (2005).
- [16] J. Rossignac, Ball-based shape processing, in: *DGCI*, 2011, pp. 13–34.

- [17] F. Chazal, A. Lieutier, J. Rossignac, B. Whited, Ball-map: Homeomorphism between compatible surfaces, *Int. J. Comput. Geometry Appl.* 20 (3) (2010) 285–306.
- [18] H. P. Moon, Equivolumetric offset surfaces, *Computer Aided Geometric Design* 26 (2009) 17 – 36.
- [19] H. Hagen, S. Hahmann, Generalized focal surfaces: A new method for surface interrogation, in: *IEEE Visualization*, 1992, pp. 70–76.
- [20] R. L. Foote, The volume swept out by a moving planar region, *Mathematics Magazine* 79 (2006) 289–297.
- [21] D. Rohmer, S. Hahmann, M.-P. Cani, Local volume preservation for skinned characters, *Computer Graphics Forum* 27 (7) (2008) 1919–1927, proceedings of Pacific Graphics.
- [22] M. Meyer, M. Desbrun, P. Schröder, A. H. Barr, Discrete differential-geometry operators for triangulated 2-manifolds (2002).
- [23] M. Desbrun, M. Meyer, P. Schröder, A. H. Barr, Implicit fairing of irregular meshes using diffusion and curvature flow, in: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, SIGGRAPH '99, 1999, pp. 317–324.
- [24] D. Zorin, Curvature-based energy for simulation and variational modeling, in: *SMI*, 2005, pp. 198–206.
- [25] J. M. Lee, *Riemannian Manifolds: An Introduction to Curvature*, Springer, 1997.
- [26] J. Rossignac, A. Safonova, A. Szymczak, 3d compression made simple: Edgebreaker on a corner-table, in: *Shape Modeling International Conference*, 2001, pp. 278–283.
- [27] N. Dyn, D. Levine, J. A. Gregory, A butterfly subdivision scheme for surface interpolation with tension control, *ACM Trans. Graph.* 9 (1990) 160–169.
- [28] S. C. Hsu, I. H. H. Lee, N. E. Wiseman, Skeletal strokes, in: Proceedings of the 6th annual ACM symposium on User interface software and technology, UIST '93, 1993, pp. 197–206.
- [29] S. Schaefer, T. McPhail, J. Warren, Image deformation using moving least squares, in: *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, 2006, pp. 533–540.
- [30] A. H. Barr, Global and local deformations of solid primitives, in: Proceedings of the 11th annual conference on Computer graphics and interactive techniques, SIGGRAPH '84, 1984, pp. 21–30.
- [31] J. Rossignac, S. Schaefer, J-splines, *Computer Aided Design* 40 (2008) 1024 – 1032.

Fleshing: Spine-driven Bending with Local Volume Preservation

Wei Zhuo* and Jarek Rossignac *

* Graphics Visualization and Usability Center, College of Computing, Georgia Tech

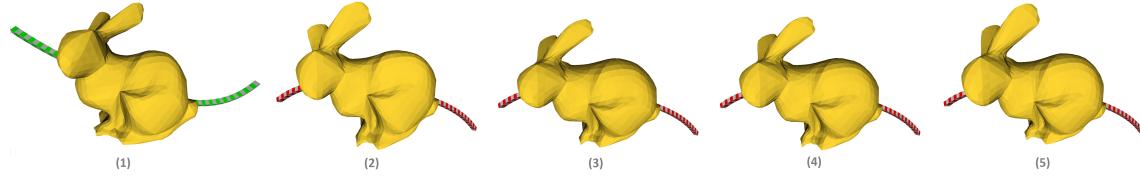


Figure 1: (1) Original bunny and the initial spine. (2) Result of bending the spine obtained using the standard skinning without volume-preserving correction for which the total volume change is 9%. (3) (4) (5) are the results produced by our three fleshing solutions: radial, normal, and binormal, for which the total volume change (due to sampling and round off errors) is less than 0.3%

Abstract

Several design and animation techniques use a one-dimensional proxy C (a spine curve in 3D) to control the deformation or behavior of a digital model of a 3D shape S . We propose a modification of these “skinning” techniques that ensures local volume preservation, which is important for the physical plausibility of digital simulations. In the proposed “fleshing” techniques, as input, we consider a smooth spine C_0 , a model S_0 of a solid that lies “sufficiently close” to C_0 , and a deformed version C_1 of C_0 that is “not overly bent”. (We provide a precise characterization of these restrictions.) As output, we produce a bijective mapping M , that maps any point X of S onto a point $M(X)$ of $M(S)$. M satisfies two properties: (1) The closest projection of X on C_0 and of $M(X)$ on C_1 have the same arc length parameter. (2) U and $M(U)$ have the same volume, where U is any subset of S . We provide three different closed form expressions for radial, normal and binormal and discuss the details of their practical real-time implementation.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric transformations F.2.2 [Theory of Computation]: Nonnumerical Algorithms and Problems—Geometrical problems and computations

1. Introduction

Often, the design of a three-dimensional model or of its animation involves bending elongated parts. Models of humans and of various animals are often defined in terms of an articulated skeleton with a few rigid bones connected at (possibly spherical) joints. As the joint angles change, points on the surface or inside the model are moved so as to preserve their relative position with respect to nearby bones. Such “skin-

ning” techniques typically use the arc-length of the closest projection onto individual bones to define relative coordinates and track bone twists (around the bone axis) to fully define the new location of a point after skeletal bending. Points close to a joint may project on more than one bone. The displacements suggested by these different bones are often blended using weights and linear combinations of locations or weighted combinations of rigid motions [LCF00].

Successful techniques have been proposed to increase physical realism of the skin deformations near the joints, so as to more accurately reflect the behavior of skin during the bending of a human elbow [KCZ07]. Some strive to preserve the total volume of the solid near the joint. Other strive to preserve the volume of each slice by possibly tilting the cross-sections [RHC08].

We focus here on deformations that follow the gentle three-dimensional bending of a “spine” that is a smooth curve. Such a tool seems appropriate for bending models of tubes, hoses, wires, ducts, and for approximating the spinal bending of vertebrates (reptiles, fish), and the bending of muscles with no skeletal support, such as mammalian tongues, elephant trunks, octopus arms, or nautilus tentacles. Although such “spine-driven bending” may not be sufficient to model exactly the physically correct behavior of these vertebrae and muscles, it provides an important tool that facilitates the design of useful approximations of their behavior and may have computational advantages over more expensive finite element simulations.

Our main contribution is to propose an explicit mathematical model of spine-driven bending that preserves local volume exactly. By “local” we mean that any solid portion of the initial shape preserves its volume during bending. This objective is more challenging than the preservation of the overall volume (which may for example be achieved by a global scaling or constant distance offsetting [ZR12]) and the preservation of the overall volume of each arbitrary cross-sectional slice using an extension of the Cavalier’s Principle [HS97]. Unfortunately, such global or per-slice volume preservation approaches do not provide a volume preserving mapping (homeomorphism) from the initial shape to the final shape. Hence, in these prior approaches, either it is not clear where exactly in a slice a particular chunk of muscle of the initial shape will end up in the bent model, or, when an exact mapping is defined, the volume of the chunk is not preserved.

When the shape is planar and the bending is in that plane, the 2D problem amounts to preserving the local area. An exact solution to local area preserving bending in 2D has been proposed in [ZR12]. It is based on a local correction, which, after the standard bending, adjusts the normal offset (from the spine) of a point based on the curvatures of the initial and the bent versions of the spine at the corresponding (closest projection) point. Our contribution is to extend this prior 2D solution to 3D, where the spine is a possibly non-planar curve and where the goal is to preserve the local volume instead of the area. The extension to 3D is far from trivial. As illustrated in Fig. 2, the solution proposed in [ZR12] is only valid for a point X_0 that lies in the osculating plane of the closest projection Q_0 of X_0 on the initial spine C_0 . The solution proposed here does not have this limitation.

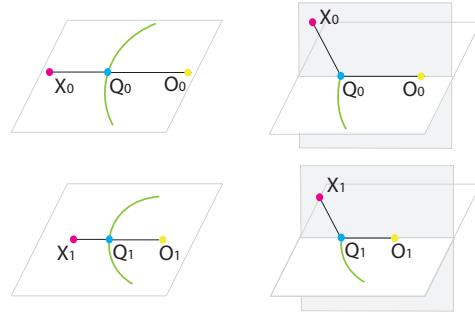


Figure 2: On the left, we show the planar bending proposed in [ZR12] where Q_0 is the closest projection of X_0 onto spine C_0 , O_i is the curvature center of C_i at Q_i . X_i is restricted to lie on the line passing O_i and Q_i . On the right, we show the 3D version of this problem, where X_0 is no longer restricted to lie on the line through O_0 and Q_0 . It can be anywhere on the cross-sectional plane orthogonal to C_0 at Q_0 .

1.1. Problem statement and our solutions

The designer starts with a shape S_0 in 3D. Our solution is a mapping from a subset of three-space to another and, as such, it operates on any shape (point cloud, bundle of curves, surfaces, solids, meshes, or cell complexes). Still, because our focus is on volume preservation, for clarity, we say that S_0 is a solid.

The designer first specifies an initial spine C_0 . The spine is a smooth curve in 3D that may pierce the solid S_0 or not. In fact, an important benefit of our solution is that the spine may be positioned fully outside of the initial solid. Then the designer specifies a new (“bent”) version C_1 of the spine.

We restrict our attentions to formulations that are defined by a mapping M which maps each point X_0 of S_0 to its image $X_1 = M(X_0)$ in S_1 . We say that M is a “fleshing” if it satisfies the following conditions:

1. M is fully defined by C_0 and C_1 , and hence independent of S_0 . This is essential in applications where different versions of S_0 may be used with different resolutions or levels of detail.
2. M is an homeomorphism between S_0 and S_1 . This is important because we want the mapping to be invertible: $M^{-1}(X_1) = X_0$, where M^{-1} is defined by the initial spine as C_1 and the final spine as C_0 in M .
3. M maps C_0 to C_1 (i.e., $M(C_0) = C_1$). The spines can be adjusted precisely by the user and a solution that ignores such a constraint may be surprising and unnatural.
4. M preserves the arc length along the spine of the closest projection (i.e., $s_0 = s_1$, where s_i ($i = 0, 1$) is the arc-length parameter of the closest projection of P_i onto curve C_i .) This constraint restricts the mapping to respect cross-sections. Although this constraint seems natural, it may

- not be physically correct for some materials that are capable of stretching along the spine [ACWK04]. Nevertheless, this constraint is key to the effectiveness of our solution and a proper assumption if the spine is rigid (only capable of bending and twisting, but not stretching).
5. M preserves volume locally (i.e., $\text{vol}(U) = \text{vol}(M(U))$ for any subset U of S_0). This is important for the physical plausibility of digital simulations, especially when they involve interactions between evolving solids (swimming creature) and surrounding, incompressible fluids.

1.2. Contributions

We propose three different fleshings that satisfy all of the constraints defined in the previous subsection. We call them “radial”, “normal”, and “bi-normal”. We provide the explicit and mathematically exact expression for each one of these fleshings and explain its derivation.

During interactive manipulation or animation, these fleshings may be computed in real time, at each frame, and animated as the user manipulates the spines. Hence, we advocate their use for gaming and medical simulations where live animation of bending shapes are desired.

Our three solutions produce results that are qualitatively different. To clearly illustrate these differences and help the reader decide which one is appropriate for a particular application, we show in Fig. 4 5 comparisons of their effects when S_0 is a tube or extruded cross-section around C_0 . We also show in Fig. 1 7 6 8 their effect on a solid bounded by an arbitrary triangle or quad meshes. We require that C_0 and C_1 be smooth. We provide, for each fleshing, the precise formulation of a valid space in which S_0 must be contained for M to exist as a valid fleshing.

2. Prior Art

The basic deformation operations proposed by Barr [Bar84] extend the conventional operations of affine transformation and CSG to include planar curve-based bending, which preserves the normal offset distance from the spine curve. The resulting mapping is not locally volume preserving as there are local expansion on the convex side and contraction on the concave side of the bent spine. To address this shortcoming, Chrikjian [Chi95] presents a mathematically precise, closed-form solution: for locally volume preserving bending in 2D, the offset distance is computed as a root of a quadratic equation with curvature-based coefficients. This variable offset distance allows the shape in the concave side of the spine to grow in the normal direction in order to compensate for the area loss. Moon [Moo08] derives the same quadratic formula for milling with constant material removal rate.

In character animation, Lewis et al. [LCF00] propose generalized forms of skeleton-driven deformations as scattered interpolations. Kavan and colleagues [KCZ007] present the

dual quaternion blending as an effective approach to preserve the skinning mesh’s rigidity and roughly its local volume around the joint. Constant volume deformations are driven by divergence-free vector field of the boundary points as shown in the work of von Funck et al. [vFTS06]. Angelidis and Singh [AS07] present the computation of divergence-free vector fields induced by skeletal motion. Their framework requires time integration as physically based rigging [CBC*07] and may have computational disadvantages for high resolution meshes. Rohmer and colleagues [RHC08] compute the offset distance scaled by the skinning weight based on affinity and bone-length. To avoid self-intersection, they detect if an offset point is within its region determined by automatic segmentation. Their subsequent work [RHC09] allows the user to specify the locality of the compensation through 1D profile curves that represent isotropic inflation, bulging, or rippling effects.

A classic theorem due to Steiner [Ste40] establishes the relationship between the differential properties of the surface and the volume enclosed. Thus, if we wish to preserve the total volume, we can grow or shrink the shape uniformly (via constant distance normal offsetting rather than global scaling) in one step (without iteration) [ZR12]. Note that this approach minimizes Hausdorff error and may hence be preferred over global rescaling [DMSB99]. It provides an more efficient algorithm for preserving the total volume of a solid undergoing free-form deformation [HML99], or for compensating the volume change due to advection [KLL*07]. To preserve the details of a shape during deformation, one may use registration with the extracted skeleton [STG*97], or with a lower level subdivision model or base surface. Botsch and Kobbelt [BK03] propose to keep the displacement volumes locally constant through relaxation during a deformation of the base surface. Moon [Moo09] presents a closed-form solution for the variable offset distance from a surface that preserves the local volume.

3. Preliminaries

3.1. Locally Volume-preserving Mapping

We consider a bijective mapping $M : X_0 \rightarrow X_1$ that maps any point $X_0 = P_0(x_0)$ onto $X_1 = P_1(x_1)$, where P_0 and P_1 are themselves mappings from local parameters x_0 and x_1 onto Cartesian space. (We use P^{-1} to denote the inverse of a mapping P). The local parameters can be the arc length, radial offset distance and the angle between the offset direction and the Frenet normal. The mapping M is volume-preserving (i.e. divergence-free) if the Jacobian determinant, $\det(J(M))$, equals 1 [Chi95]. We compute the Jacobian of M by the following equation:

$$J(M) = \frac{\partial X_1}{\partial X_0} = \frac{\partial P_1}{\partial x_1} \frac{\partial x_1}{\partial x_0} \frac{\partial x_0}{\partial P_0}. \quad (1)$$

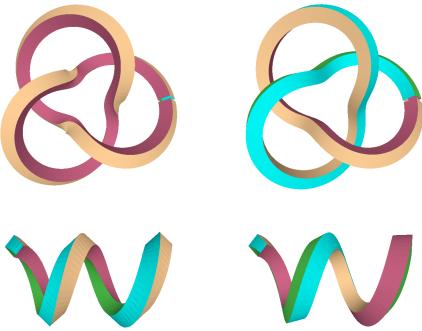


Figure 3: Left: skinning according to the Frenet frame on a trefoil knot (top) and a helix (bottom). Right: skinning using the normal propagated frame is twist-minimized.

3.2. Frenet versus twist-compensated local frames

In order to perform skinning, one computes the local frame F_0 of C_0 at the point $C_0(s)$ that is closest to X_0 , registers X_0 to F_0 , which amounts to computing the local coordinates, and then computes the local frame F_1 of C_1 at the point $C_1(s)$ and constructs X_1 from $C_1(s)$ using the local coordinates. Typically, we select frames that are aligned with the tangent to the spine. Hence, we pick $T_0(s)$ as the tangent to C_0 at $C_0(s)$ and $T_1(s)$ as the tangent of C_1 at $C_1(s)$. The remaining issue is how to determine the other two basis vectors, or their twist around the tangent. A natural candidate for the local frame is the Frenet frame $\{T(s), N(s), B(s)\}$ at $C(s)$ where $N(s)$ and $B(s)$ are the normal and binormal. By Frenet-Serret theorem [dC92], the derivative of the Frenet frame at $C(s)$ is related to the frame itself through the curvature κ and the torsion τ at $C(s)$,

$$\begin{bmatrix} T'(s) \\ N'(s) \\ B'(s) \end{bmatrix} = \begin{bmatrix} 0 & \kappa & 0 \\ -\kappa & 0 & \tau \\ 0 & -\tau & 0 \end{bmatrix} \begin{bmatrix} T(s) \\ N(s) \\ B(s) \end{bmatrix}. \quad (2)$$

Although the Frenet frame provides a convenient local frame along the curve, it is not appropriate as the tool for skinning, because it contains undesired twists, as shown in Fig. 3 (left). For example, the Frenet frame has an orientation discontinuity along a piecewise circular curve [RR87] at the C^1 continuous junction between two adjacent, but not coplanar circular arcs.

To address this problem, we use a “twist-compensated local frame”, as shown in Fig. 3 (right). Its rotation with respect to the Frenet frame is defined by the integral of the torsion [Sal] [Far03]. We construct the twist-compensated normal $I(s+ds)$ at $C(s+ds)$ by projecting $I(s)$ to the normal plane of $C(s+ds)$. Therefore, given an initial normal $I(0)$, the twist-compensated normal $I(s)$ is obtained by propagation from $I(0)$. Then for each point X_0 of S_0 , we register it

with the twist-compensated frame $W_0(s) = \{I_0(s), J_0(s)\}$ on $C_0(s)$.

3.3. Overview of the fleshing algorithm

We are given a solid S_0 , an initial spine C_0 , and a final spine C_1 . We are also given an initial normal vector $I_0(0)$ to C_0 at $C_0(0)$ and an initial normal vector $I_1(0)$ to C_1 at $C_1(0)$. Alternatively, we compute $I_0(0)$ and $I_1(0)$ automatically, using an agreed upon rule for generating a vector normal to a tangent direction, and let the designer control the global twist angle w which we use to adjust $I_1(0)$ by rotating it around the tangent to C_1 at $C_1(0)$. We assume that each point of S_0 has a unique closest projection on C_0 and that C_1 satisfies our validity conditions.

We compute the bent version S_1 of S_0 by applying a fleshings to every vertex or control point X_0 of S_0 to obtain its image $X_1 = M(X_0)$. Our approach involves the following steps:

1. *Spine preparation:* Compute parameter s such that $Q_0 = C_0(s)$ is the closest projection of X_0 onto C_0 , and $Q_1 = C_1(s)$. The corresponding Frenet frame and curvature at Q_i ($i = 0, 1$) are $F_i = \{N_i, B_i\}$ and k_i ; the corresponding twist-compensated frame is $W_i = \{I_i, J_i\}$.
2. *Unbending:* Compute X_u , the unbend mapping of X_0 by applying one of our unbend fleshing mappings.
3. *Rotation:* Compute X_r from X_u by applying a change of basis from Q_0, W_0 to Q_1, W_1 so that X_r represents the unbend point in the frame of C_1 .
4. *Bending:* Compute X_1 , the bending mapping of X_r by applying one of our bending fleshing mappings.

We give the derivation and implementation of the three fleshing mappings in Sec. 4.

3.4. Validity conditions

To express the validity conditions under which our approach produces a fleshing, we define a valid space $S(C_0, C_1)$ which must contain S_0 . To do so, we define the “reach” $R(C)$ of a curve C as the locus of all points that have a unique normal projection onto that curve. The reach may be computed as the space obtained by radially inflating the curve at each point and in all orthogonal directions until we reach the corresponding curvature axis (which is the axis of the osculating circle). We define $S(C_0, C_1)$ as the intersection $R(C_0) \cap M^{-1}(R(C_1))$ of the reach of C_0 with the pre-image of the reach of C_1 . In Sec. 4, we provide explicit formulae for testing, during unbending and bending whether a point X_0 , is in the valid space.

4. Fleshings

To simplify their formulation, we express each one as a composition of unbending which straightens C_0 with a subsequent bending to C_1 .

4.1. Radial Fleshting

Radial fleshing transformation is denoted by M_r . We start with the point X parameterized by (s, r, θ) . s is the arc length parameter of the closest projection $C(s)$ of X onto C . r and θ are the polar coordinates of X on the normal plane of $C(s)$:

$$X = P(s, r, \theta) = C(s) + r \cos \theta N(s) + r \sin \theta B(s).$$

We take the derivative of X with respect to its parameters and substitute T' , N' and B' by using Frenet-Serret equation (Eq. 2), and reduce the result to

$$\frac{\partial X}{\partial(s, r, \theta)} = \begin{bmatrix} (1 - \kappa r \cos \theta) & -r \sin \theta & r \cos \theta \\ 0 & \cos \theta & \sin \theta \\ 0 & -r \sin \theta & r \cos \theta \end{bmatrix} \begin{bmatrix} T(s) \\ N(s) \\ B(s) \end{bmatrix}. \quad (3)$$

Therefore, we have:

$$\det\left(\frac{\partial X}{\partial(s, r, \theta)}\right) = r(1 - \kappa r \cos \theta). \quad (4)$$

In $M_r : X_0 \rightarrow X_1$, we assume that only the radial offset distance r is updated from r_0 to r_1 while other parameters remain the same. We solve for r_1 under the constraint that $\det(J(M_r)) = 1$. We next show that there exists an closed-form solution for r_1 , and hence an analytic solution for M_r . Specifically, from Eq. 1 we have:

$$J(M_r) = \frac{\partial X_1}{\partial(s, r_1, \theta_1)} \frac{\partial(s, r_1, \theta_1)}{\partial(s, r_0, \theta_0)} \left(\frac{\partial X_0}{\partial(s, r_0, \theta_0)} \right)^{-1}. \quad (5)$$

Given Eq. 4, we compute the determinant of the Jacobian in Eq. 5 as:

$$\begin{aligned} \det(J(M_r)) &= \det\left(\frac{\partial X_1}{\partial(s, r_1, \theta_1)}\right) \frac{dr_1}{dr_0} / \det\left(\frac{\partial X_0}{\partial(s, r_0, \theta_0)}\right) \\ &= \frac{r_1(1 - \kappa_1 r_1 \cos \theta_1)}{r_0(1 - \kappa_0 r_0 \cos \theta_0)} \frac{dr_1}{dr_0}. \end{aligned}$$

In order to let $\det(J(M_r)) = 1$, we solve the following ODE:

$$r_1 dr_1 - \kappa_1 r_1^2 dr_1 \cos \theta_1 = r_0 dr_0 - \kappa_0 r_0^2 dr_0 \cos \theta_0,$$

and integrate from 0 to r_i on both sides of the above equation to obtain:

$$-\frac{2}{3} \kappa_1 \cos \theta_1 r_1^3 + r_1^2 = -\frac{2}{3} \kappa_0 \cos \theta_0 r_0^3 + r_0^2. \quad (6)$$

Therefore, r_1 is a cubic root of Eq. 6 with coefficients specified by r , κ_0 , κ_1 and $\cos \theta$. The solution provided by Eq. 6 assumes that the bending (change of curvature) does not change the local Frénet frame. To support more general bending, as explained in Sec. 3.3, we split the fleshing into several steps which include *unbending* (locally at $C(s)$, the spine becomes a straight line) and *bending* (the spine becomes curved again with the new curvature and normal). Here, we include the formulae for the unbend and bend mappings.

Radial Unbending: We first assume that $C_0(s)$ is straight-

ened into a line and solve for a temporary value r_* :

$$r_* = r_0 \sqrt{1 - \frac{2}{3} \kappa_0 \cos \theta_0 r_0}. \quad (7)$$

In order for r_* to exist, $\frac{2}{3} \kappa_0 \cos \theta_0 r_0 < 1$. As $\cos \theta$ varies in $[-1, 1]$, an sufficient condition for r_* to exist is $|\kappa_0 r_0| \leq \frac{3}{2}$. *Radial Bending:* We then bend the straight spine into C_1 and solve for r_1 using r_* :

$$-\frac{2}{3} \kappa_1 \cos \theta_1 r_1^3 + r_1^2 = r_*^2. \quad (8)$$

We normalize the unknown and the coefficients in Eq. 8. Specifically, let $\lambda = \frac{r_1}{r_*}$ and $\alpha = -\frac{2}{3} \kappa_1 r_* \cos \theta$, then Eq. 8 becomes $\alpha \lambda^3 + \lambda^2 = 1$. Let $f(\lambda) = \alpha \lambda^3 + \lambda^2 - 1$, which has two local extrema (minimum at $\lambda_1 = 1$ and maximum at $\lambda_2 = -\frac{2}{3\alpha}$). If $\alpha > 0, \lambda_2 < 0$, then $f(0)f(1) < 0$ and $f' > 0 \in [0, 1]$, and hence there exists a valid solution in $[0, 1]$. If $\alpha > 0, \lambda_2 > 0$, then a valid solution exists only if $f(\lambda_2) > 0$, or equivalently $\alpha^2 < \frac{4}{27}$. Again since $\cos \theta$ varies in $[-1, 1]$, an sufficient condition for r_1 to exist is $|\alpha| < \frac{2}{3\sqrt{3}}$, or

$$|\kappa_1 r_*| \leq \frac{1}{\sqrt{3}}, \quad (9)$$

and when κ_1 reaches this curvature limit, $r_1 = \sqrt{3}r_*$.

Fig. 4 illustrates the bending of a straight spine into a circular arc with length preservation. Fig. 4 (a) is the original spine with two layers of cylindrical tube surfaces. Fig. 4 (b) shows the application of Radial Bend to the original tube. Intuitively, the radial distance increases for points on the inner side of the curved spine in order to compensate for local compression. Excessive bending leads to self-intersection of the tube surface and the curvature limit in Eq. 9.

4.2. Normal Fleshting

Here, we define the normal fleshing M_n . We consider expressing a point X in the local Frenet frame as follows:

$$X = P(s, x, y) = C(s) + xN(s) + yB(s).$$

We take the derivative of P with respect to its parameters and substitute the derivatives using Eq. 2:

$$\frac{\partial X}{\partial(s, x, y)} = \begin{bmatrix} (1 - \kappa x) & -\tau y & \tau x \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T(s) \\ N(s) \\ B(s) \end{bmatrix}.$$

Therefore,

$$\det\left(\frac{\partial X}{\partial(s, x, y)}\right) = 1 - \kappa x.$$

During normal fleshing, we change the parameter x from x_0 to x_1 while keeping s and y constant. Given κ_0 , κ_1 , s , y and x_0 , we solve for x_1 under the constraint $\det(J(M_n)) = 1$.

Specifically, we have:

$$\begin{aligned} \det(J(M_n)) &= \det\left(\frac{\partial P_1}{\partial(s, x_1, y)}\right) dx_1 / \det\left(\frac{\partial P_0}{\partial(s, x_0, y)}\right) \\ &= \frac{(1 - \kappa_1 x_1) dx_1}{(1 - \kappa_0 x_0) dx_0}. \end{aligned}$$

Setting $\det(J(M_n)) = 1$ gives that:

$$dx_1 - \kappa_1 x_1 dx_1 = dx_0 - \kappa_0 x_0 dx_0.$$

Integrate on both sides of the above equation and we have:

$$x_1 - \frac{1}{2} \kappa_1 x_1^2 = x_0 - \frac{1}{2} \kappa_0 x_0^2. \quad (10)$$

Therefore, x_1 is a quadratic root of Eq. 10 with coefficients specified by x_0 , κ_0 , κ_1 .

As for Normal Fleshing, Eq. 10 is limited to cases where the local curvature is changed, but the Frenet frame remains constant. To support more general fleshing, as explained in Sec. 4.1, we provide below its decomposition into normal unbending and bending maps, which may be combined with the twist-compensated rotation, as discussed in Sec. 3.3. To solve x_1 , we break Eq. 10 into two steps:

Normal Unbending: Assume that $C_0(s)$ is first straightened ($\kappa_1 = 0$) and we solve for a temporary value x_* ,

$$x_* = x_0 (1 - \frac{1}{2} \kappa_0 x_0). \quad (11)$$

As $\frac{x_*}{x_0} \geq 0$, the condition for a valid solution of r_* to exist is $|\kappa_0 x_0| \leq 2$.

Normal Bending: We then bend the straight spine into C_1 and solve for x_1 using x_* :

$$-\frac{1}{2} \kappa_1 x_1^2 + x_1 = x_*. \quad (12)$$

Hence, the closed-form solution for x_1 is

$$x_1 = \frac{1 - \sqrt{1 - 2\kappa_1 x_*}}{\kappa_1}.$$

In order for x_1 to be valid, we have:

$$\kappa_1 x_* \leq \frac{1}{2}, \quad (13)$$

and when κ_1 reaches this curvature limit, $x_1 = 2x_*$.

Fig. 4 (c) shows the application of Normal Bend to the cylindrical tube surfaces in Fig. 4 (a). As shown in the cross-sectional plot, M_n only slides points in the direction normal to the osculating plane. Intuitively, the tube surface stretches towards the inner side and shrinks from the outer side of the circular spine in order to compensate for local compression and expansion. When reaching the curvature limit in Eq. 13, the tube surface starts to intersect itself. Note that M_n has a more stringent curvature limit than M_r for the same initial tube surface.

4.3. Binormal Fleshing

During the Binormal Fleshing, we adjust the coordinate from y_0 to y_1 while keeping s and x constant. We then solve for y_1 under the constraint $\det(J(M_b)) = 1$:

$$\det(J(M_b)) = \frac{(1 - \kappa_1 x) dy_1}{(1 - \kappa_0 x) dy_0}.$$

We set $\det(J(M_b)) = 1$ to obtain,

$$(1 - \kappa_1 x) dy_1 = (1 - \kappa_0 x) dy_0.$$

Therefore,

$$(1 - \kappa_1 x) y_1 = (1 - \kappa_0 x) y_0. \quad (14)$$

This result shows that y_1 is linearly related to y_0 with the coefficient defined by κ_0 , κ_1 and x .

As for the other two Fleshings, we provide here the unbend and bend versions.

Binormal Unbending: Let $\kappa_1 = 0$, $y_* = y_1$ and we have,

$$y_* = (1 - \kappa_0 x) y_0. \quad (15)$$

In order for y_* to be valid, we have $\kappa_0 x \leq 1$.

Binormal Bending: Let $\kappa_0 = 0$, $y_0 = y_*$ and we have,

$$y_1 = \frac{1}{1 - \kappa_1 x} y_*. \quad (16)$$

In order for y_1 to be valid, we have

$$\kappa_1 x < 1. \quad (17)$$

When κ_1 reaches this curvature limit, y_1 becomes unbounded.

Fig. 4 (d) shows the application of Binormal Bend to the cylindrical tube surfaces in Fig. 4 (a). As shown in the cross-sectional plot, M_B only allows stratification in the binormal direction: points on the tube surface expand or shrink bilaterally on the inner side or the outer side of the circular spine. When reaching the curvature limit in Eq. 17, the tube surface becomes flat on the inner side. Note that M_b has the least stringent curvature limit among the three solutions.

Discussion: Note that Eq. 6, Eq. 10 and Eq. 14 are symmetric in the initial and final states of the spine and the space point. Hence, the mappings are homeomorphisms between S_0 and S_1 . Fig. 4 presents a qualitative comparison of the three fleshing solutions by showing their effects on tubular surfaces: Bi-normal fleshing is closest to what happens when a tube is bent horizontally a bit too much: the flesh is pushed vertically up or down (as in the crack of a bent elbow). The normal fleshing is the reverse: the flesh moves horizontally, hence it moves more quickly in the direction of the center of curvature. The radial is a compromise, the flesh moves radially away or towards the spine.

5. Experiments and Results

This section shows the results of our three fleshing solutions.

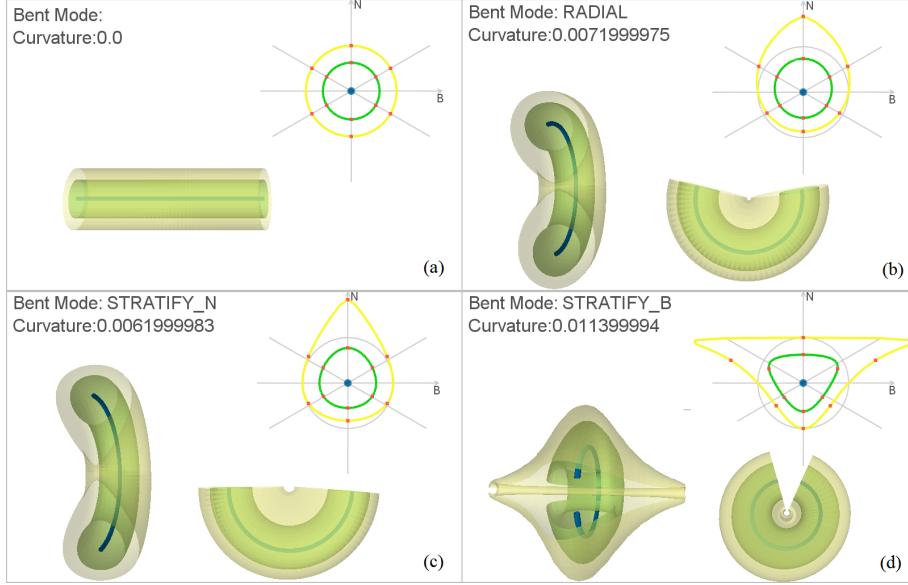


Figure 4: The deformation of two layers of tube surfaces driven by bending a straight spine into a circular arc. We show three types of fleshing to the original tube surfaces in (b) (c) (d). Each of them shows the transverse (left) and the frontal (right) views of the bent tubes, and the cross section is dynamically plotted on the top-right. The red tick marks show the mapping trajectories of points on tube surfaces.

5.1. Extrusion Models

We first present the application of fleshing to models of solids produced by sweeping a user specified planar cross-section along a smooth 3D spine curve. To better show the different effects of our three solutions, we apply them to extrusions along spines that are circular arcs. In the 2D view as shown in Fig. 5, the user draws a contour and indicates the point $Q = Q_0 = Q_1$ at which the initial and the final spines are aligned and have the same tangent, but different radii and osculating planes. The centers of the arcs are specified by locations O_0 and O_1 . We show the initial cross-section in blue, then for each fleshing, we show the result of unbending in green and in red the result of bending the green in Fig. 5(b). The vectors O_0Q and O_1Q define the Frenet frames and curvatures. We assume here no twist compensation (i.e., $\theta_1 = \theta_0$). Notice that the radial fleshing nearly preserves straight lines (even though it is not an affine map). In Fig. 5(c), we show the results in 3D and the corresponding statistics in Tab. 1.

We compute the exact volumes of the extrusion models in all cases using the following approach. Let W be the centroid of a planar region R and let Q be the point passed by the arc C with length l and center O . Then the volume of the solid S swept out by R along C is computed as [Foo06],

$$\text{vol}(S) = \text{area}(R)\rho l$$

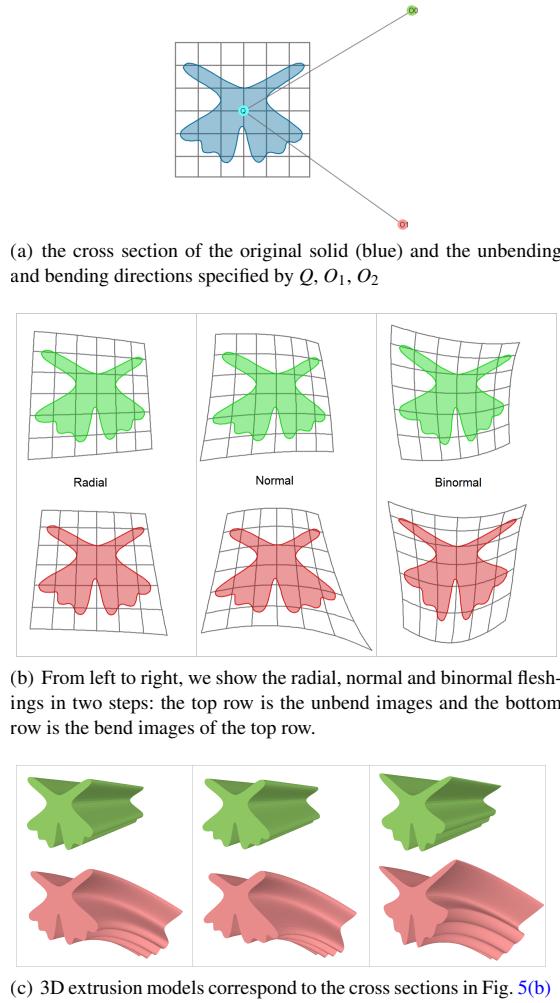
	$\text{area}(R)$	ρ	$\text{vol}(S)$	ϵ
Original	0.278	1.032	1.436	0
Unbend	0.278	1	1.391	-0.031
Bend	0.278	0.982	1.366	-0.049
Radial Unbend	0.287	1	1.436	-1.36E-5
Radial Bend	0.296	0.969	1.436	-3.97E-6
Normal Unbend	0.287	1	1.436	-5.63E-5
Normal Bend	0.298	0.963	1.436	-1.03E-4
Binormal Unbend	0.287	1	1.436	1.17E-4
Binormal Bend	0.293	0.979	1.437	2.83E-4

Table 1: Statistics of the cross-sectional areas, ratios of the centroid traveled distance to l , solid volumes and their relative errors for extruded models in Fig. 5(c).

where $\rho = \frac{\text{OW} \cdot \text{OQ}}{|\text{OQ}|^2}$ is the ratio of the actual distance traveled by the centroid and l . If C is a line, $\rho = 1$. We compute the relative error ϵ of the solid S as

$$\epsilon = \frac{\text{vol}(S_1) - \text{vol}(S_0)}{\text{vol}(S_0)}$$

where S_i is the solid swept out by R_i along C_i , $i = 0, 1$. As shown in Tab. 1, models without fleshing have relatively large volumetric errors (3%-5%). The other 6 models with

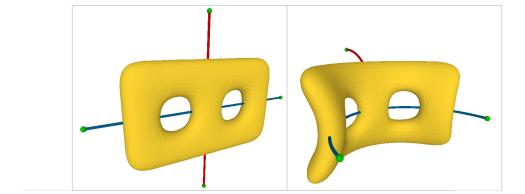
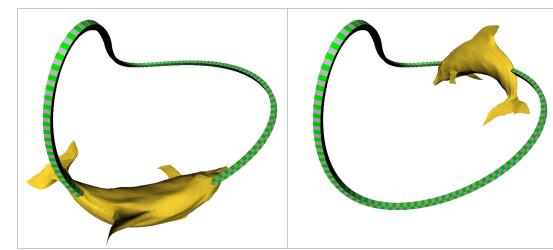
**Figure 5:** *Fleshing mappings of an extrusion model*

fleshing have nearly the same volume with very small (less than 0.03%) volumetric errors.

5.2. Quad and Triangle Meshes

We show fleshing applications to general shapes and report total volume changes of less than 0.3% in real time. Fig. 1 shows bending a triangle mesh, bunny, with a interpolating spline with control points. As the initial spine (green) deforms into the final spine (red), fleshing mappings help preserve the total volume that would otherwise be subjected to a large change by skinning. Similar to skinning, fleshing preserves smoothness, and hence also sharp features, such as the bunny's ears.

Fig. 6 shows bending a genus-2 quad mesh first with a

**Figure 6:** *successively bending a mesh***Figure 7:** *Two frames in an animation of a dolphin sliding along a 3D curve: by using the radial fleshing, the volume deviations of the mesh model are 0.12% and 0.08% in these two positions. In comparison, the volume deviations are 6.75% and 3.47% without fleshing.*

vertical spine into the frontal plane, then with a horizontal spine out of the frontal plane. Initially, the red and the blue spines are on the plane that divides S_0 into identical halves. Without the volume-preserving mapping, the volume remain unchanged after the first bending, but increases by 9% after the second bending. In comparison, the volume deviates little (<0.1%) from the orginal one if using fleshing. However, the binormal stretch (Fig. 6(b) right) causes the mesh to grow unexpectedly in the horizontal direction. Fig. 7 shows

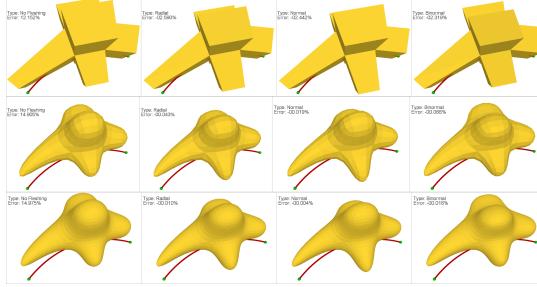


Figure 8: From left to right we show the bending results of skinning, radial, normal and binormal on different levels of a subdivision mesh. From top to bottom, the number of vertices are 32, 482, 1922.

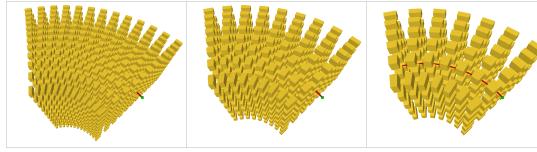


Figure 9: Bending a cloud of cubes of uniform size. From left to right, the original cube sizes are 15, 22, 30.

an application in animation, where a dolphin mesh is sliding along a 3D curve with its orientation and deformation determined by the curve. Here the curve, C_1 , represents a animator-specified path for the digital model to follow. The path may be curved so that the model may bend. Without fleshing, this causes unexpected changes of volume perceivable by the viewer. The volume of the digital model is preserved by one of our fleshing mappings. As shown in the figure, the volume deviation is above 5% without fleshing and reduced to 0.01% with the radial method.

5.3. Resolution and Accuracy

This section discusses the impact of sampling density on the accuracy of (local) volume preservation implemented by fleshing. Fig. 8 shows bending a subdivision mesh at different resolutions. The increase of the subdivision depth greatly decrease the relative volume errors of the three fleshing mappings (from 2.5% to 0.001%). On the contrary for skinning without fleshing, the relative volume error increases slightly (from 12% to 15%). This shows that fleshing mappings give accurate total volume-preserving results for high resolution meshes. In fact, our objective is not only to preserve the total volume, but to preserve the local volume for each small chunk of the solid. Hence, the proper measure of volume error that should be used to demonstrate the benefit of fleshing over skinning is to report the average of the

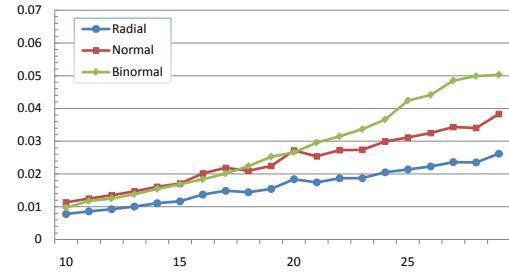


Figure 10: plot of the percentage mean absolute relative error versus the cube size

volume errors of the small chunks. Fig. 9 shows bending a cloud of cubes at different sizes. We compute the volume of each cube deformed by the spine. The relative error for each cube is computed as $\epsilon = (v - v_0)/v_0$. We report the mean absolute value, ϵ_{mean} , of the relative errors for all cubes. Fig. 10 shows the plot of ϵ_{mean} versus the cube size. In general, the relative error scales with the cube size in all three fleshing mappings. This indicates that fleshing mappings give more accurate local volume-preserving results for deforming a higher resolution volumetric model.

6. Discussion

Fleshing presents three closed-form volume preserving mappings, which depends on a proper local parameterization of the 3D shape along the 1D spine. The spine may have a simple parametric expression, such as circular or helical arc. Then the closed-form parameterization along the curve is easy to obtain. However, this limits the designer's ability to bend the spine by manipulating a few control points. If we represent the spine by a interpolating polynomial, we must be able to compute the arc-length parameter of the closest projection of a point onto the spine.

The choice of representation for the spine is orthogonal to our contribution. Nevertheless, we support two formulations for the spine C : (1) a low degree, interpolating polynomial, which we evaluate using Neville's algorithm, as shown in Fig. 1, and (2) quintic NURBS, which we evaluate using de Casteljau's algorithm, as shown in Fig. 7. Assume that the approximating polylines are C_0 and C_1 . We sample them so that they have the same edge length, dl . Again, let Q_0 be the closest projection of X_0 on C_0 . Assume that Q_0 is on the edge $C_0[k]C_0[k+1]$:

$$Q_0 = C_0[k] + aC_0[k]C_0[k+1], \quad 0 < a < 1.$$

The arc-length parameter s of Q_0 is $s = (k+a)dl$. We use the same arc-length parameter to compute Q_1 on C_1 : $Q_1 = C_1[k] + aC_1[k]C_1[k+1]$, which is used as the anchor point for computing X_1 , as described in the overall fleshing

transformation. The influence of this polyline approximation depends on the sampling step size (dl). When dl is not overly small, decreasing dl improves the precision for locally volume preservation. However, we found that further increasing the sampling density causes larger error as vertices of C are not accurate enough due to numerical errors, which propagate to the normal and curvature evaluation, and to the overall fleshing transformation. Angular distortion exists in fleshing mappings due to that the mappings are not conformal. Also, it is not possible to deform B-spline surfaces with locally volume preservation by applying the mapping only to its control points as the mappings are not affine. However, they preserve smoothness, and hence also sharp features. In fact, the radial fleshing is nearly line preserving, as shown in Fig. 5(b) (left). The result is guaranteed to be free from self-crossing (when we are within the validity conditions), and hence it will not produce new sharp features.

7. Conclusion

We have proposed three formulations for deforming a shape driven by bending a spine with which the shape is registered. Our fleshing solutions ensure that the local volume of any subset of a valid space is preserved during the bending. Furthermore, our solution is based on a closed form mapping of space and depends neither on the initial shape nor on the given global coordinate system. Hence the fleshing may be applied to any shape topology (point cloud, watertight surfaces or cell complexes). Furthermore, we extend our approach to spines that need not be circular, by propagating a twist compensated local frame and by letting the user or an application control the bending and twisting. We hope that the simplicity, accuracy and performance of the proposed Fleshing approach will make it a standard bending tool for many modeling and animation applications where local volume preservation is desired. In the future, we plan to extend the methods to handle multiple spines.

References

- [ACWK04] ANGELIDIS A., CANIF M., WYVILL G., KING S.: Swirling-sweepers: constant-volume modeling. In *Computer Graphics and Applications, 2004. PG 2004. Proceedings. 12th Pacific Conference on* (2004), pp. 10–15. 3
- [AS07] ANGELIDIS A., SINGH K.: Kinodynamic skinning using volume-preserving deformations. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation* (2007), SCA '07. 3
- [Bar84] BARR A. H.: Global and local deformations of solid primitives. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques* (1984), SIGGRAPH '84, pp. 21–30. 3
- [BKB03] BOTSCHE M., KOBBELT L.: Multiresolution surface representation based on displacement volumes. *Comput. Graph. Forum* 22, 3 (2003), 483–492. 3
- [CBC*07] CAPELL S., BURKHART M., CURLESS B., DUCHAMP T., POPOVIĆ Z.: Physically based rigging for deformable characters. *Graph. Models* 69, 1 (2007), 71–87. 3
- [Chi95] CHIRIKJIAN G. S.: Closed-form primitives for generating locally volume preserving deformations. *Journal of Mechanical Design* 117, 3 (1995), 347–354. 3
- [dC92] DO CARMO M. P.: *Riemannian Geometry*, 1st ed. Birkhäuser, 1992. 4
- [DMSB99] DESBRUN M., MEYER M., SCHRÖDER P., BARR A. H.: Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), SIGGRAPH '99. 3
- [Far03] RATIONAL APPROXIMATION SCHEMES FOR ROTATION-MINIMIZING FRAMES ON PYTHAGOREAN-HODOGRAPH CURVES. *Computer Aided Geometric Design* 20, 7 (2003), 435–454. 4
- [Foo06] FOOTE R. L.: The volume swept out by a moving planar region. *Mathematics Magazine* 79 (2006), 289–297. 7
- [HML99] HIROTA G., MAHESHWARI R., LIN M. C.: Fast volume-preserving free form deformation using multi-level optimization. In *Proceedings of the fifth ACM symposium on Solid modeling and applications* (1999), SMA '99. 3
- [HS97] HARRIS J. W., STOCKER H.: *The Handbook of Mathematics and Computational Science*, 1st ed. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1997. 2
- [KCZ007] KAVAN L., COLLINS S., ZÁRA J., O'SULLIVAN C.: Skinning with dual quaternions. In *SI3D* (2007), pp. 39–46. 2, 3
- [KLL*07] KIM B., LIU Y., LLAMAS I., JIAO X., ROSSIGNAC J.: Simulation of bubbles in foam with the volume control method. *ACM Trans. Graph.* 26, 3 (2007). 3
- [LCF00] LEWIS J. P., CORDNER M., FONG N.: Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* (2000), SIGGRAPH '00. 1, 3
- [Moo08] MOON H. P.: Equivolumetric offsets for 2d machining with constant material removal rate. *Computer Aided Geometric Design* 25, 6 (2008), 397–410. 3
- [Moo09] MOON H. P.: Equivolumetric offset surfaces. *Computer Aided Geometric Design* 26 (2009), 17–36. 3
- [RHC08] ROHMER D., HAHMANN S., CANI M.-P.: Local volume preservation for skinned characters. *Comput. Graph. Forum* 27, 7 (2008), 1919–1927. 2, 3
- [RHC09] ROHMER D., HAHMANN S., CANI M.-P.: Exact volume preserving skinning with shape control. In *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2009), SCA '09. 3
- [RR87] ROSSIGNAC J., REQUICHA A.: Piecewise-circular curves for geometric modeling. *IBM Journal of Research and Development* 3 (1987), 129–148. 4
- [Sal] SALOMON D.: 4
- [Ste40] STEINER J.: über parallele flächen. *Monatsberichte der Akademie der Wissenschaft zu Berlin (Monthly Report of the Academy of Sciences, Berlin)* (1840), 114–118. 3
- [STG*97] STORTI D. W., TURKIYYAH G. M., GANTER M. A., LIM C. T., STAL D. M.: Skeleton-based modeling operations on solids. In *Proceedings of the fourth ACM symposium on Solid modeling and applications* (1997), SMA '97. 3
- [vFTS06] VON FUNCK W., THEISEL H., SEIDEL H.-P.: Vector field based shape deformations. In *ACM SIGGRAPH 2006 Papers* (2006), SIGGRAPH '06. 3
- [ZR12] ZHUO W., ROSSIGNAC J.: Curvature-based offset distance: Implementations and applications. *Computers & Graphics* 36, 5 (2012), 445–454. 2, 3

REFERENCES

- [1] ANGELIDIS, A., CANI, M.-P., WYVILL, G., and KING, S., “Swirling-sweepers: Constant-volume modeling,” *Graphical Models*, vol. 68, no. 4, pp. 324–332, 2006.
- [2] ANGELIDIS, A. and SINGH, K., “Kinodynamic skinning using volume-preserving deformations,” in *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 129–140, Eurographics Association, 2007.
- [3] ANTIGA, L., ENE-IORDACHE, B., and REMUZZI, A., “Computational geometry for patient-specific reconstruction and meshing of blood vessels from mr and ct angiography,” *Medical Imaging, IEEE Transactions on*, vol. 22, no. 5, pp. 674–684, 2003.
- [4] AUBERT, F. and BECHMANN, D., “Volume-preserving space deformation,” *Computers & Graphics*, vol. 21, no. 5, pp. 625–639, 1997.
- [5] BARAN, I., LEHTINEN, J., and POPOVIĆ, J., “Sketching clothoid splines using shortest paths,” in *Computer Graphics Forum*, vol. 29, pp. 655–664, Wiley Online Library, 2010.
- [6] BARR, A. H., “Global and local deformations of solid primitives,” in *ACM Siggraph Computer Graphics*, vol. 18, pp. 21–30, ACM, 1984.
- [7] BATHE, K.-J., *Finite element procedures*, vol. 2. Prentice hall Englewood Cliffs, 1996.
- [8] BOTSCHE, M. and KOBELT, L., “Multiresolution surface representation based on displacement volumes,” in *Computer Graphics Forum*, vol. 22, pp. 483–491, Wiley Online Library, 2003.
- [9] CAPELL, S., BURKHART, M., CURLESS, B., DUCHAMP, T., and POPOVIĆ, Z., “Physically based rigging for deformable characters,” in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 301–310, ACM, 2005.
- [10] CHIRIKJIAN, G. S., “Closed-form primitives for generating locally volume preserving deformations,” *Journal of Mechanical Design*, vol. 117, p. 347, 1995.
- [11] DESBRUN, M., MEYER, M., SCHRÖDER, P., and BARR, A. H., “Implicit fairing of irregular meshes using diffusion and curvature flow,” in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 317–324, ACM Press/Addison-Wesley Publishing Co., 1999.

- [12] FOLEY, J. D., *Computer graphics: Principles and practice, in C*, vol. 12110. Addison-Wesley Professional, 1996.
- [13] HAHMANN, S., SAUVAGE, B., and BONNEAU, G.-P., “Area preserving deformation of multiresolution curves,” *Computer aided geometric design*, vol. 22, no. 4, pp. 349–367, 2005.
- [14] HANSON, A. J. and MA, H., “Parallel transport approach to curve framing,” *Indiana University, Techreports-TR425*, vol. 11, pp. 3–7, 1995.
- [15] HIROTA, G., MAHESHWARI, R., and LIN, M. C., “Fast volume-preserving free-form deformation using multi-level optimization,” *Computer-Aided Design*, vol. 32, no. 8, pp. 499–512, 2000.
- [16] HSU, S. C. and LEE, I. H., “Drawing and animation using skeletal strokes,” in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pp. 109–118, ACM, 1994.
- [17] HSU, S. C., LEE, I. H., and WISEMAN, N. E., “Skeletal strokes,” in *Proceedings of the 6th annual ACM symposium on User interface software and technology*, pp. 197–206, ACM, 1993.
- [18] IRVING, G., SCHROEDER, C., and FEDKIW, R., “Volume conserving finite element simulations of deformable models,” *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, p. 13, 2007.
- [19] KÄLBERER, F., NIESER, M., and POLTHIER, K., “Stripe parameterization of tubular surfaces,” in *Topological Methods in Data Analysis and Visualization*, pp. 13–26, Springer, 2011.
- [20] KASS, M. and ANDERSON, J., “Animating oscillatory motion with overlap: wiggly splines,” in *ACM Transactions on Graphics (TOG)*, vol. 27, p. 28, ACM, 2008.
- [21] KIM, B., LIU, Y., LLAMAS, I., JIAO, X., and ROSSIGNAC, J., “Simulation of bubbles in foam with the volume control method,” in *ACM Transactions on Graphics (TOG)*, vol. 26, p. 98, ACM, 2007.
- [22] KIM, M.-S., PARK, E.-J., and LIM, S.-B., “Approximation of variable-radius offset curves and its application to bezier brush-stroke design,” *Computer-Aided Design*, vol. 25, no. 11, pp. 684–698, 1993.
- [23] LLAMAS, I., POWELL, A., ROSSIGNAC, J., and SHAW, C. D., “Bender: a virtual ribbon for deforming 3d shapes in biomedical and styling applications,” in *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pp. 89–99, ACM, 2005.

- [24] MOON, H. P., “Equivolumetric offsets for 2d machining with constant material removal rate,” *Computer Aided Geometric Design*, vol. 25, no. 6, pp. 397–410, 2008.
- [25] MOON, H. P., “Equivolumetric offset surfaces,” *Computer Aided Geometric Design*, vol. 26, no. 1, pp. 17–36, 2009.
- [26] O’DONNELL, T., BOULT, T. E., FANG, X.-S., and GUPTA, A., “The extruded generalized cylinder: A deformable model for object recovery,” in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR’94., 1994 IEEE Computer Society Conference on*, pp. 174–181, IEEE, 1994.
- [27] REDDY, J. N., *Theory and analysis of elastic plates and shells*. CRC press, 2007.
- [28] REUTER, M., BIASOTTI, S., GIORGI, D., PATANÈ, G., and SPAGNUOLO, M., “Discrete laplace–beltrami operators for shape analysis and segmentation,” *Computers & Graphics*, vol. 33, no. 3, pp. 381–390, 2009.
- [29] ROHMER, D., HAHMANN, S., and CANI, M.-P., “Local volume preservation for skinned characters,” in *Computer Graphics Forum*, vol. 27, pp. 1919–1927, Wiley Online Library, 2008.
- [30] ROHMER, D., HAHMANN, S., and CANI, M.-P., “Exact volume preserving skinning with shape control,” in *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 83–92, ACM, 2009.
- [31] ROSSIGNAC, J. and SCHAEFER, S., “J-splines,” *Computer-Aided Design*, vol. 40, no. 10, pp. 1024–1032, 2008.
- [32] ROSSIGNAC, J. R. and REQUICHA, A. A., “Piecewise-circular curves for geometric modeling,” *IBM Journal of Research and Development*, vol. 31, no. 3, pp. 296–313, 1987.
- [33] SCHAEFER, S., MCPHAIL, T., and WARREN, J., “Image deformation using moving least squares,” in *ACM Transactions on Graphics (TOG)*, vol. 25, pp. 533–540, ACM, 2006.
- [34] SEDERBERG, T. W. and PARRY, S. R., “Free-form deformation of solid geometric models,” in *ACM Siggraph Computer Graphics*, vol. 20, pp. 151–160, ACM, 1986.
- [35] STEINER, J., “über parallele flächen,” *Monatsberichte der Akademie der Wissenschaft zu Berlin (Monthly Report of the Academy of Sciences, Berlin)*, pp. 114–118, 1840.

- [36] STORTI, D. W., TURKIYYAH, G. M., GANTER, M. A., LIM, C. T., and STAL, D. M., “Skeleton-based modeling operations on solids,” in *Proceedings of the fourth ACM symposium on Solid modeling and applications*, pp. 141–154, ACM, 1997.
- [37] VON FUNCK, W., THEISEL, H., and SEIDEL, H.-P., “Vector field based shape deformations,” *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 1118–1125, 2006.
- [38] WANG, W., JÜTTLER, B., ZHENG, D., and LIU, Y., “Computation of rotation minimizing frames,” *ACM Transactions on Graphics (TOG)*, vol. 27, no. 1, p. 2, 2008.
- [39] ZHUO, W. and ROSSIGNAC, J., “Curvature-based offset distance: Implementations and applications,” *Computers & Graphics*, vol. 36, no. 5, pp. 445–454, 2012.
- [40] ZHUO, W. and ROSSIGNAC, J., “Fleshing: Spine-driven bending with local volume preservation,” in *Computer Graphics Forum*, vol. 32, pp. 295–304, Wiley Online Library, 2013.