Eurographics/ ACM SIGGRAPH Symposium on Computer Animation (2007)
D. Metaxas and J. Popovic (Editors)

Kinodynamic skinning using volume-preserving deformations

Alexis Angelidis[†] and Karan Singh[‡]

Department of Computer Science
University of Toronto

Abstract

We present a new approach to character skinning where divergence-free vector fields induced by skeletal motion, describe the velocity of skin deformation. The joint transformations for a pose relative to a rest pose create a bend deformation field, resulting in pose-dependent or kinematic skin deformations, varying smoothly across joints. The bend deformation parameters are interactively controlled to capture the varying deformability of bone and other anatomic tissue within an overall fold-over free and volume-preserving skin deformation. Subsequently, we represent the dynamics of skeletal motion, tissue elasticity, muscular tension and the environment as forces that are mapped to vortices at tissue interfaces. A simplified Biot-Savart law in the context of elastic deformation recovers a divergence-free velocity field from the vorticity. Finally, we apply a new stable technique to efficiently integrate points along their deformation trajectories. Adding these dynamic forces over a window of time prior to a given pose provides a continuum of user controllable kinodynamic skinning. A comprehensive implementation using a typical animator workflow in Maya *shows our approach to be effective for complex character skinning.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Character Skinning

1. Introduction

Character modeling and setup is a tedious process. Digital character geometry in a rest or bind pose is first created from artist sketches and maquettes. This geometry specifies both the external appearance of the character skin and implicitly embodies an underlying anatomy. The subsequent challenging problem of constructing a mapping from articulations of an underlying skeletal hierarchy to a deformation of the skin geometry is referred to as *character skinning* (see Figures 1,2). Skin deformations need to be generally smooth, intersection-free and computationally efficient. Furthermore, they need to capture variations in surface shape resulting from the dynamic shape and material properties of the implicit underlying anatomy. Bulging, creasing, sagging and dynamic secondary motion are examples of the visual impact underlying anatomy has on the surface of skin [CHP89]. A skinned character rigged for animation is ready to come to life in able hands. The skill of the animator, however, is transmitted through the character rig and can be completely destroyed by unconvincing skin deformations.

A popular technique in deformable object simulation is to separate the overall deformation into rigid and deformable components [PCLS05]. Such approaches are well-suited to character skins where the non-rigid deformation of the skin is small relative to the overall articulation of the character. Perhaps a more animator appropriate separation is based on differentials of character pose (see Figure 1). We refer to skin deformations that are purely a function of the skeletal pose as *kinematic* skinning. Deformations resulting from higher order time derivatives of skeletal pose and other dynamic forces are referred to as *dynamic* skinning. Kinematic skinning mostly comprises the articulated rigid body transformation of anatomy. We do, however, include pose dependent deformations such as bone protrusion, skin creasing around joints and the volume-preserving bulging resulting from active and passive contraction of tense muscle. Dynamic skinning captures the elastic tissue deformations generated by the character's skeletal motion and other environmental forces. In commercial practice, skinning is almost always kinematic since true dynamic skinning requires a temporal simulation to determine the skin shape and this impairs

[†] e-mail: silex@pixar.com

[‡] e-mail: karan@dgp.toronto.edu

SCA 2007, San Diego, California, August 04 - 05, 2007
© 2007 ACM 978-1-59593-624-4/07/0008 $ 5.00

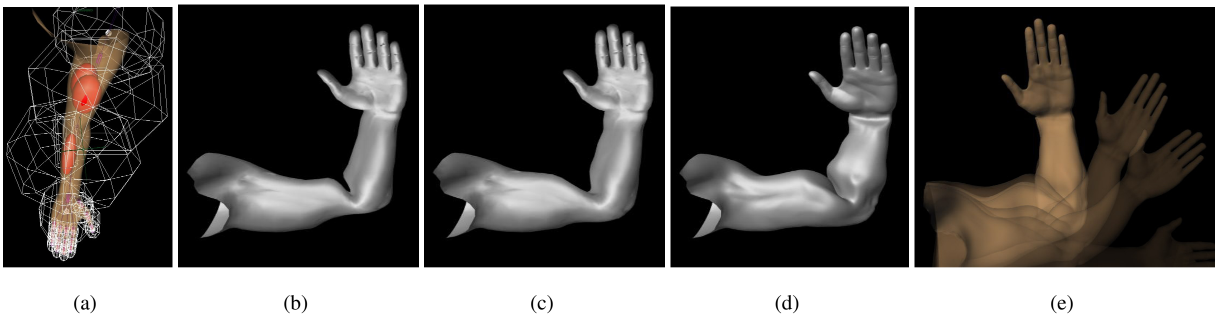


Figure 1: *Kinodynamic skinning: The undeformed skin setup in the bind pose (a), is kinematically skinned traditionally (b), with volume-preservation (c), and bone, tissue, and contracted muscle (d) and with elastic muscle dynamics (in flesh tone)(e).*

an animator's interactive control over the character [LCF00]. As a result dynamic skinning effects are usually added as needed post-hoc.

Motivation: A number of traditional deformation principles [TJ81] are applicable to character skinning. To a first approximation the skin behaves like an articulated rigid body. Particularly in the region around joints, however, it is visually important that the deformation is smooth and foldover-free (see Figure 1b). Clothing, fur and muscle dynamics are often modeled as physical simulations, for which self-intersection free skin geometry is a desirable property. Volume-controllable skin is also desirable since anatomic tissue and cavities, while largely volume-preserving are indeed selectively compressible. The skin should locally capture the deformation behavior of underlying bone, soft tissue (see Figure 1c) or contracting muscle (see Figure 1d). Finally, much of a character's visual appeal lies in the dynamic secondary motion of elastic anatomic tissue induced by skeletal motion and other external forces (see Figure 1e). We also observe that most dynamic skin deformations are localized in time around forces induced by skeletal motion due to strong elastic restorative forces and damping. This temporally local property of dynamic skinning makes it both possible and desirable to incorporate the effects of dynamic skinning in an animator interactive setting.

Approach: We first choose a deformation framework that addresses our character skinning goals. Divergence-free vector fields that define the deformation trajectory of points in space are foldover-free and volume-preserving [ACWK04], [vFTS06]. We thus use this fundamental property to specify all elements of skin deformation. The smooth, foldover-free, articulated rigid body like behavior is clearly the most significant aspect of skinning. We address this in the context of traditional linear-blend skinning by developing a closed form solution to the integration of deformation trajectories. Bending around joints induces a divergence-free deformation velocity as shown by [vFTS06]. We show how this deformation can be decomposed into our closed form skinning solution overlaid with a numerically integrated volume-preserving term. The bend deformation can be decomposed into the cross product of the gradient of two scalar fields [vFTS06]. We show how user control over the coefficients [Kot91] of this decomposition allows the local modeling of bone and soft tissue behavior, while preserving an overall divergence-free field. We then model anatomic tissue and muscle dynamics using a sampling of vorticity under the skin. The velocity field of these vortices is divergence-free by construction [ACWK04] and captures the animator desired effects of musculoskeletal dynamics. The overall system is interactive and homogenous in its approach to skin deformation and allows us to perform *kinodynamic* skinning, where dynamic effects resulting over a user defined window of time prior to the present can be incorporated into our kinematic skinning workflow in real-time (see Figure 4).

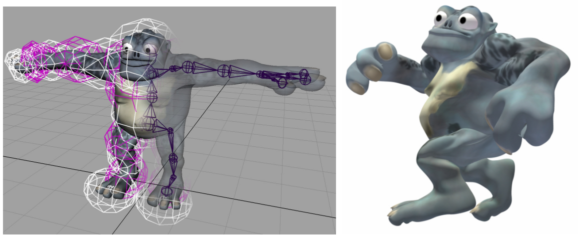


Figure 2: *A large implicit anatomy between skin and skeleton here makes skinning challenging. The troll is shown overlaid with ellipsoidal primitives capturing anatomic tissue in the bind pose and the subsequent skinned result.*

Contributions: The chief contribution of this paper is a character skinning system that is fold-over free and volume-preserving, captures anatomic tissue properties and interactively captures both kinematic and dynamic skinning with a traditional character rigging workflow. In the construction of this system we first describe a new analytic solution to fold-over free linear blend skinning. We also adapt the Clebsch decomposition of the bend deformation of [vFTS06] to account for the varying rigidity of bone and other anatomic tissue under the skin. We further describe a novel approach to dynamic volume-preserving object deformation based on vortex samples. We propose an efficient and stable integration of these vector fields that describe skin deformations discretized over pose-space, time or both in a kinodynamic setting. The techniques presented here are implemented as a *Maya* plug-in and seamlessly integrate into a typical character rigging process.

2. Related Work

Approaches to character skinning and more generally object deformation can be broadly classified as geometric or physically based. In the context of skinning many approaches further incorporate anatomic elements of varying sophistication.

Animators continue to prefer interactivity, precision and temporal independence of keyframing for expressive character animation and by extension prefer purely geometric approaches to character skinning [LCF00]. Geometric skinning approaches typically define the deformation at vertices of skin geometry as a weighted blend of skeletal joint transformations [MG03]. Vertex weights are determined by proximity to skeletal segments by default for a given rest or bind pose. This approach produces smooth deformation, but the results appear closer to a flexed pipe than to deforming tissue [LCF00]. Indirect skinning approaches improve provide richer behaviour and control [CHP89] [MT97] [SK00] [HYC*05]. Here skeletal articulations control spatial deformers, such as a free-form deformation lattice [SP86], which in turn deform the skin. The spatial deformers typically have a much lower resolution than skin geometry, making it easier to hand craft their skeletal mapping. Within the geometric domain, example based approaches have recently gained in popularity [LCF00] [SCFRC01] [ACP02] [KJP02] [WP02] [MG03]. Here the surface skin is sculpted in various poses, and deformation is defined as sparse data interpolation [LCF00] [ACP02] or by fitting to parameters of a simple deformation model [MG03]. Geometric approaches by themselves are efficient and provide direct animator control, but do not easily capture nuances of anatomic deformation [Sin95]. This has been highlighted by a large body of anatomy based skinning [SPCM97] [WG97] [AHS03] which wrap a skin around a given anatomic structure. Most notably, *outside-in* skinning [PCLS05] fits anatomy to a given skin geometry that subsequently augments traditional geometric skinning effectively. None of these approaches, unfortunately, specifically address geometric self-intersections or volume-preservation of the skin in a theoretically robust fashion. Such properties, however, have been successfully developed in the context of geometric modeling [ACWK04] [vFTS06] and we do indeed generalize the bend deformation presented in [vFTS06] to handle part of the skinning process.

Physically based simulation for object deformation is a rich area of ongoing research. Damped spring-mass systems [LTW95] [MHTG05] [LCA05], the finite element method (FEM) [CGC*02] [CBC*05] [TSIF05] and modal analysis [JP02] are among the popular approaches to capture the physics of deformable objects. We contribute a new interactive technique based on dynamic vortex samples to this body of research. Our approach is best suited to volume-preserving deformations of objects with layers of varying deformability. Our dynamic vortex samples can easily be created automatically or by a user arbitrarily in space and like [MHTG05] do not require explicit topological connectivity to each other.

In the context of skinning, a volumetric FEM mesh is used within a dynamic simulation to tie the surface skin to skeletal motion [CGC*02] or precomputed with modal analysis and added to kinematic skinning as a deformation texture [JP02]. Despite the availability of these approaches, physically based skinning has not been popular in commercial animation, due to a lack of efficiency and instant animator feedback during skeletal keyframing. We address this problem by proposing the idea of kinodynamic skinning where our dynamic skin deformations are accumulated over a discretization of pose-space and a finite window of time at interactive rates (see Figure 4). Our approach to dynamics has the advantages of a skin topology independent setup and inherent volume-preservation in an efficient and stable framework.

Finally, we note that the anatomic properties captured in this paper are coarse and intended for animator use, unlike research targeted at accurately modeling human musculoskeletal architecture [TSB*05].

3. Kinodynamic skinning

We begin by formally defining the problem of kinodynamic skinning (depicted visually in Figure 4). We represent a skeletal pose S, as the 4x4 parent relative transform matrices $< R_1, R_2, .., R_n >$ of a hierarchy of n joints. We also denote the world space joint transform for limb i in pose S by L_i. Hierarchically, $L_i = L_{pr(i)} R_i$, where $pr(i)$ is the parent of i (see Figure 3). The skin geometry resulting from the kinematic skinning of S is denoted by G_S. Kinematic skin deformation is strictly a function of skeletal pose and hence G_S is unique. The undeformed geometric skin G is associated with a special pose called the rest or bind pose B. The bind pose joint transforms for limb i are denoted as R_{Bi} and L_{Bi}. Kinematic skinning is the function that deforms G to G_S, based on the pose-space transformation from B to S (see Figure 4).

In traditional geometric skinning [LCF00] the position of deformed points is a function of this pose transformation. In our approach the pose transformation instead defines a *pose-space* velocity for the deformation of points in space. The pose transformation thus needs to be discretized to integrate the deformation of points in pose-space. Now consider some

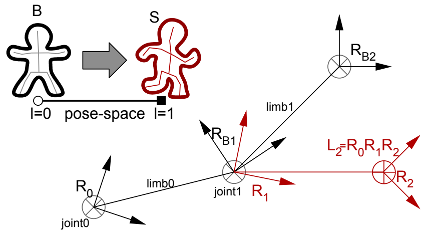


Figure 3: *Local R_i and world L_i joint transforms*

pose on the animation timeline S_t, t units of time prior to pose S. A similar temporal discretization would be necessary to solve for dynamic deformations from S_t to S. Given a time window of t units, kinodynamic skinning solves for skin deformation in pose-space from B to S_t and from S_t to S over time (see Figure 4). A window of size zero is kinematic skinning and increasing the window size incorporates more dynamism, with small window sizes ensuring a realtime response. Dynamic simulations over a small prior window of time can only be applied successfully in scenarios such as character skinning where the elastic restorative forces and damping are strong and the resulting dynamic deformations are transient. Note that the results over a finite window are not the same as running the entire simulation but that the results are close for short lived dynamic deformations.

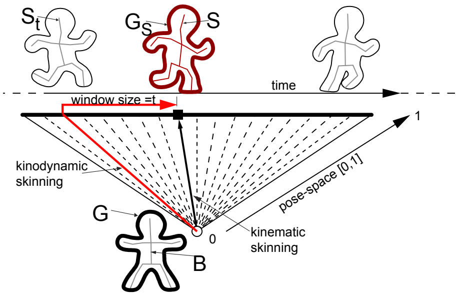


Figure 4: *Kinodynamic skinning*

4. Kinematic skinning details

In traditional linear blend skinning [LCF00], undeformed skin proximal to a limb i can be rigidly transformed from the bind pose B to a pose S by the transform $T_i = L_i L_{Bi}{}^{-1}$. Smooth transitions across joints are achieved by weight averaging these transforms, using a partition of unity weight vector $< w_1, w_2, .., w_n >$ defined for each skin point p. p is thus deformed to p_S in pose S as:

$$\boxed{p_S = \sum_{i=1}^{n} w_i T_i p} \qquad (1)$$

Equation 1, however, produces two undesirable effects around joints: the *collapsing elbow* and the *candy-wrapper* effect [MG03] (see Figure 7a). In both cases one typically observes self-intersecting geometry and an extreme loss of volume for large joint rotations. Although hand-crafted skin examples in extreme poses [LCF00] [SCFRC01] can alleviate the problem, they do not explicitly avoid self-intersecting configurations.

4.1. Fold-over free skinning

The Jacobian J_f of a spatial deformation function f represents the directional gradients of deformation and its determinant captures conditions for foldover-free ($Det(J_f) > 0$) and volume-preserving ($Det(J_f) = 1$) skinning in the neighborhood of a point. A divergence-free deformation velocity field is thus volume-preserving [ACWK04], [vFTS06].

Consider the continuous motion in pose-space from the bind to current pose. Equation 1 differentiated over a pose parameterization $l \in [0,1]$ from the bind pose B to current pose S gives us:

$$\boxed{\dot{p_S} = \sum_{i=1}^{n} w_i \dot{T_i} p} \qquad (2)$$

$\dot{T_i}$ is the sum of the world space velocities $\dot{L_j}$ induced by every joint j in the hierarchy from the root to i (denoted by the ancestor set $an(i)$). Formally:

$$\boxed{\dot{T_i} = \sum_{j \in an(i)} \dot{L_j}(L_{Bj}{}^{-1})} \qquad (3)$$

The world space velocity $\dot{L_k}$ induced by any limb k is simply its locally induced velocity $\dot{R_k}$, transformed to world space or $\dot{L_k} = L_{pr(k)} \dot{R_k}$.

Integrating this trajectory of p in Equation 2 can be done in discrete steps of dl, defined over a pose parameterization $l \in [0,1]$ from the bind pose B to current pose S:

$$\boxed{p(l + \mathrm{d}l) = p(l) + \mathrm{d}l\ \dot{p}(l)} \qquad (4)$$

The accuracy can be improved naively with more steps or with adaptive steps using a Runge-Kutta method [vFTS06].

Alternately, we can write Equation 2 in pose-space as:

$$\boxed{p(l + \mathrm{d}l) = \sum_{i=1}^{n} w_i\ T_i(l) p(l)} \qquad (5)$$

where $T_i(l) = L_i(l + \mathrm{d}l) L_i(l)^{-1}$ is a fractional transformation of limb i in pose-space corresponding to a step of length dl at a pose S, and $L_i(l) = \prod_{j\in an(i)}(R_jR_j^{-1})^lR_j$ defines a continuum of world space joint transforms. This hierarchical accumulation of the delta transformations can be computed numerically as shown by [Ale02] or more robustly using the closed form we derive in Appendix B. The step size provides a trade-off between speed and accuracy with a step of length 1 being equivalent to traditional smooth skinning Equation 1. We find in practice that a fixed number of 4-8 steps of Equation 5 produces foldover-free deformations at interactive rates (see Figure 5). The matrices $T_i(l)$, being independent of skin points, can be pre-computed once for the skeletal hierarchy at each discretized pose-space step. Equation 5, however, does not provide any control over the volume of the deforming skin.

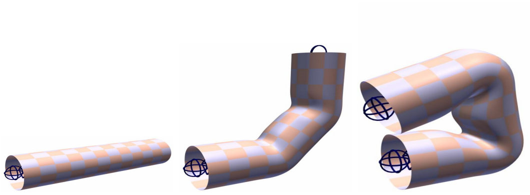


Figure 5: *Solving Equation 5 guarantees fold-over free deformations.*

4.2. Volume-preserving skinning

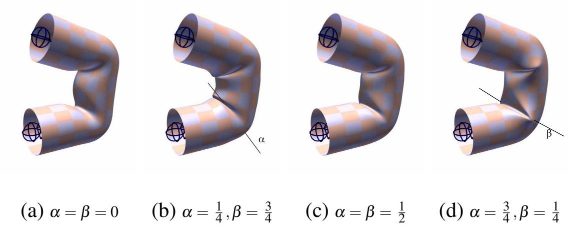
Let us now focus on the local rotational velocity $\dot{R}_k$ of joint k. Rotation is an inviscid motion and its velocity field can be decomposed into a pair of scalar fields (ϕ_k, ψ_k) (given in Appendix A) using a Clebsch decomposition [Kot91]. The cross product of the gradient of these fields produces a divergence free volume-preserving deformation field as shown by [vFTS06]. We substitute this decomposition into a combination of Equation 2 and Equation 3 to obtain Equation 6. First we replace T_i in Equation 2 by Equation 3. We then denote $p_j = L_Bj^{-1} \cdot p$ as the point defined local to limb j. We write $\dot{L_j}$ as $L_{pr(j)}\dot{R}_j$ and replace the rotational velocity $\dot{R}_j$ by its Clebsch decomposition. Finally we split weight w_i (a scalar field) among ϕ and ψ using α_i and β_i such that $\alpha_i + \beta_i = 1$. Denoting the gradient operator as ∇ we get:

$$\boxed{\begin{aligned}\dot{p} &= \sum_{i=1}^{n}\sum_{j\in an(i)} L_{pr(j)}(\nabla w_i^{\alpha_i}\phi_j)\times(\nabla w_i^{\beta_i}\psi_j)p_j \\ &= \sum_{i=1}^{n}\left(w_i\dot{T}_ip + \sum_{j\in an(i)} \nabla w_i \times (\alpha_i\phi_j\nabla\psi_j - \beta_i\psi_j\nabla\phi_j)\right)\end{aligned}} \tag{6}$$

Note that the first term is exactly Equation 2 and can be solved using the analytic solution in Appendix B. The second term makes the field solenoidal, and can be solved numerically. Thus:

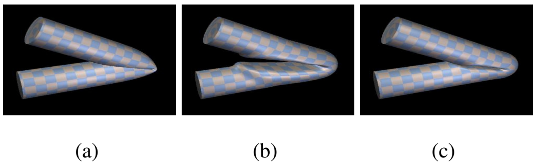
$$\boxed{\begin{aligned}p(l+\mathrm{d}l) &= \sum_{i=1}^{n} w_i\, T_i(l)p(l) + \\ \mathrm{d}l \sum_{j\in an(i)} &\nabla w_i \times (\alpha_i\phi_j\nabla\psi_j - \beta_i\psi_j\nabla\phi_j)\end{aligned}} \tag{7}$$

The solenoidal volume-preserving term for joint i is controlled using α_i and β_i and can range from Equation 5 ($\alpha_i = \beta_i = 0$) to more or less deformation along the axes shown in Figure 6. Under extreme joint articulations (see Figure 7b) precise volume-preservation can result in undesirable artifacts. These artifacts can be alleviated by modulating α_i and β_i, which give the animator control over the change in volume upon deformation (see Figure 7c).



(a) $\alpha = \beta = 0$ (b) $\alpha = \frac{1}{4}, \beta = \frac{3}{4}$ (c) $\alpha = \beta = \frac{1}{2}$ (d) $\alpha = \frac{3}{4}, \beta = \frac{1}{4}$

Figure 6: *Volume is preserved when $\alpha + \beta = 1$, with control over the directional incompressibility shown in (b),(c),(d).*



(a) (b) (c)

Figure 7: *Skinning behavior under extreme joint articulation: (a) linear blend, (b) volume-preserving $\alpha = \beta = 0.5$ and (c) volume-controlled $\alpha = 0.1, \beta = 0.3$.*

4.3. Deformable tissue weights

The skin weights, w_i, associated with limb i define both the locality of the skin surface affected by the limb, and a simplified model of the anatomy under the skin. Users interactively paint weights on the control vertices of the skin geometry or specify these weights using implicit falloff functions (see Figure 2). Weight gradients ∇w_i, for user painted weights are computed from a scalar field defined as an offset volume around the surface i.e. the weight value at a point is the weight at the closest point on the skin surface attenuated by it's distance from the surface. In this paper user painted weights and implicit function primitives are used to define *bone weights* that cause regions to move rigidly with limbs, and *soft tissue* weights that capture varying rigidity.

Soft Tissue: is modeled by interactively placing an ellipsoidal primitive around a region of skin (see Figure 2a). To compute the tissue weight μ at a point p we first transform p

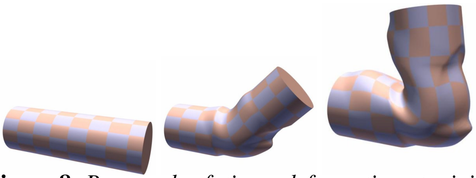


Figure 8: *Bone and soft tissue deformation at a joint.*

to a local point by the transform that maps the ellipsoid to a canonical sphere at the origin. We then apply a common radial falloff function $f(d)$ based on the distance d of the local point from the origin. f is monotonic, C^1, numerically close to C^2, almost antisymmetric about $1/2$, and uses the squared distance. For example we use:

$$\boxed{\mu(p) = f(d) = (d^2-1)^2, \text{if } d<1 \text{ and } 0 \text{ otherwise.}} \tag{8}$$

Bone: We define the bone weight χ at a point to be 1 for points inside the bone and 0 for points outside. In practice we ignore the large discontinuous gradient of χ at the bone surface and practically use a sharp distance falloff to 0 outside the bone.

In Equation 7, we now set $w_i = \chi_i + \mu_i$ and ∇w_i is simply $\nabla \mu_i$. We note here that χ_i and μ_i are defined in the local space of limb i and could be a sum of many implicit primitive functions.

The results of tissue variation can be seen in Figures 1 and 8. Note that by using arbitrarily shaped implicit function primitives [PCLS05], we can also model differently shaped underlying anatomic structures.

Having modeled the contributions of passive anatomic elements of kinematic skinning, we turn our attention to dynamic skinning effects.

5. Dynamic Skinning

We propose a new approach to modeling the dynamics of incompressible anatomic material: vortex deformers. Intuitively, a user creates shapes that capture a desired deformable surface (see Figure 9). We automatically create vortex samples that capture vorticity at points on these shapes. This vorticity is then dynamically updated by musculoskeletal motion and other environment forces. From these vortex samples we are able to recover a deformation velocity field which is divergence-free by construction and fits seamlessly with the velocity fields thus far that represent the kinematic skin of the character.

5.1. Vortex deformers

The motion of a 3D deformable material can be described by the motion of its velocity field v as discussed thus far or its vorticity field w. The velocity and vorticity at a point p can be converted into each other through a pair of mutually inverse relations, the *curl* $(\nabla\times)$ and the *Biot-Savart law*:

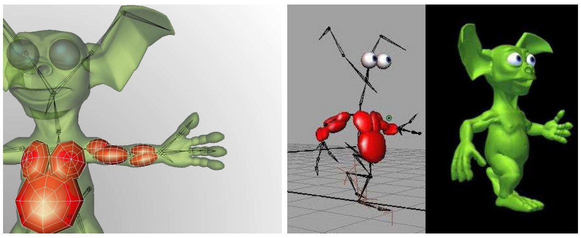


Figure 9: *The muscle shapes are placed in the skeleton's hierarchy. Deformed proxy muscles are not required but are useful in visualizing the dynamics of the character.*

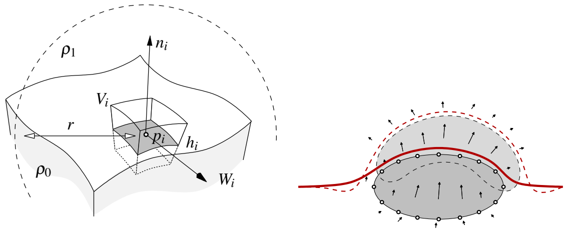
$$\boxed{\begin{array}{rcl} w(p) & = & (\nabla \times v)(p) \\ v(p) & = & \displaystyle\iiint_{x\in\mathbb{R}^3} \frac{w(x)\times(p-x)}{4\pi\|p-x\|^3}\,\mathrm{d}x \end{array}} \tag{9}$$

The Biot-Savart law gives the simplest inverse of $\nabla\times$. Others are obtained by adding a curl-free field to the velocity. A difference between the above equations is that the computation of $w(p)$ requires v at a point, while the computation of $v(p)$ requires w in $\mathbb{R}^3$. The integrand of the Biot-Savart law has a geometric interpretation: the velocity induced by one vorticity vector $w(x)$ is that of a rotation whose magnitude decreases away from its center x according to the radial function $\frac{1}{4\pi\|d\|^3}$. Given that the sum of rotational velocity fields is an incompressible velocity field [ACWK04], the velocity defined by the Biot-Savart law clearly preserves volume. Our deformable elastic material model is composed of two main parts: a motion equation for the vorticity, and an approximated Biot-Savart law to recover the velocity.

Let us first develop the latter. Consider a material M_0 of density ρ_0 embedded in a medium of density ρ_1 (see Figure 10a). The motion of M_0 can be defined at the interface between the two media only (see justification in Appendix C). If we consider an element of volume V intersecting the interface on an area h, with normal n pointing outwards from M_0 as shown in Figure 10a, then the variation over time of the vorticity $W = \iiint_{x\in V} w\ \mathrm{d}V$ carried by the element of volume V is defined as

$$\boxed{\frac{\mathrm{d}W}{\mathrm{d}t} = 2h\frac{\rho_1-\rho_0}{\rho_1+\rho_0} n\times(F-a)} \tag{10}$$

where a is the acceleration and F is the sum of the forces. Since we are interested in materials much heavier than air (the medium in which M_0 is embedded), we typically set $\rho_1 = 0$.



(a)

(b)

Figure 10: *Vortex deformers: (a) a single vortex sample defined by $\{p_i, n_i, a_i, h_i, W_i, u_i\}$ and (b) a deformation (in red) defined by a velocity field induced by* fixed samples *on the surface of an undeformed shape (in gray).*

5.2. Vorticity induced velocity field

We now define the animated velocity field that describes the deformation. To animate a deformable object, we discretize the continuum's variables at elements of the object's surface: a set of positions p_i placed on the initial undeformed surface have an area h_i, normal n_i, surface vorticity W_i and acceleration a_i (see Figure 10). To define the velocity field induced by the elements of surface, we approximate the Biot-Savart law in Equation 9 by replacing the falloff function $\frac{1}{4\pi d^3}$ with the finite extent falloff function f used in Equation 8, so that:

$$\boxed{\dot{p} = \sum_i f((\|p - p_i\|/r)^2)\; W_i \times (p - p_i)} \qquad (11)$$

In the context of muscle modeling a finite falloff function f is good for two reasons. First, the distance between the muscle vortices and the skin over them is typically small and second, unlike vortices in fluids we do not want a muscle vortex to influence distant skin points that likely belong to another part of the body. An exception to this is when body parts collide and then the colliding skin points are close to the muscle vortices. The default falloff distance r is thus a multiple of the distance between a muscle vortex and the proximal skin surface above it but is ultimately under animator control. The positions p_i are held fixed in the object's local frame (see Figure 10b). Thus only W_i is numerically animated, with Equation 10. To simulate multiple objects, the total velocity v_{total} is obtained by summing all the velocity fields or by combining the velocities in a more controlled manner (see Section 5.4). The new position p_t of a point p after δt seconds is defined with a forward integration $p_t = p + \delta t v_{total}(p)$ similar to the pose-space Equation 4.

5.3. Forces

We now model the necessary forces that would update the vorticity in Equation 10. The assumptions for our deformable framework are elasticity and incompressibility. The latter is captured implicitly by our vortex samples. To animate compressible material or very large scale deformations, it would be necessary to both assign new positions to the existing vortex samples p_i and to resample the vortices on the deformable interface for an adequate discretization. For elastic objects or those with small scale deformations we can afford the approximation of keeping the sampling and locations p_i of vortices constant in the local frame of the objects. Thus, unlike other forces, a restorative elastic force is required in our approach.

Stable Elasticity: The various assumptions and approximations of our model (see Appendix C) make it difficult to express the elastic force, which is not necessarily curl-free, in a similar way to the forces F of Equation 10. However, since we know the object is at rest when all vorticity $W_i = 0$, we define a linear force that pulls W_i towards 0, as opposed to a force that would pull deformed pointd p'_i towards their rest state p_i [MHTG05]. The elastic force F_e is
$F_e(p_i) = -k_s W_i$,

where k_s is the stiffness of the material. As this force is applied to W_i, we need to represent the *vortical velocity* with a vector u_i. The force F_e defines the new vortical velocity u'_i and surface vorticity W'_i at the next time step:

$$\boxed{u'_i = u_i + \delta t (F_e - k_d u_i)} \qquad (12)$$

$$\boxed{W'_i = W_i + \delta t \|u'_i\| \frac{W_i}{\|W_i\|}} \qquad (13)$$

Where $k_d u_i$ is a classic damping term. Note that this scheme preserves the direction of W_i, and of the various schemes we have tried it is the most stable.

Shear Strain: The magnitude of elastic force is based on the material's strain. Let us denote the displacement gradient tensor by $J = I + \delta t \nabla v(p)$. The strain tensor $\varepsilon = J^{\mathrm{T}} \cdot J - I$ is central to modeling accurately elastic forces [NMK*05], and measures two kinds of strain: *shear strain* and *compression strain*. Since our equations model an incompressible velocity field, there is no strain due to compression. The shear strain is the change in angle of previously orthogonal directions, and for a pair of unit orthogonal vectors τ_0 and τ_1 it is measured as the dot product $(J\tau_0) \cdot (J\tau_1)$. Since we consider isotropic materials, we do not privilege any direction. By integrating all possible pairs of unit orthogonal directions over a sphere, the shear strain is:

$$\boxed{\begin{aligned} S_i = \frac{2\pi}{30}\Big(& (\varepsilon_{00} - \varepsilon_{11})^2 + (\varepsilon_{11} - \varepsilon_{22})^2 + (\varepsilon_{22} - \varepsilon_{00})^2 \\ & + 6\,(\varepsilon_{01}^2 + \varepsilon_{12}^2 + \varepsilon_{20}^2)\Big) \end{aligned}} \qquad (14)$$

where ε_{ij} are elements of the strain tensor ε. This strain, evaluated at each location p_i is multiplied to the elastic force F_e. Shear strain computation requires W_i, making the complexity of computing strain for n surface elements $O(n^2)$. Enabling shear strain thus diminishes interactivity and is also not visually significant in many cases and thus can typically be disabled.

Plasticity: While muscles are elastic they are active elements that when internally contracted in a state of equilibrium can be conceived as having been plastically deformed. We can define this material state with rest state vorticity vectors W_i other than 0. Clearly specifying such vectors would be an impossible user task. The user instead specifies a desired a plastic shape (by displacing the vortices of the original shape) and we compute the appropriate W_i as follows. Since Equation 11 expresses the velocity V as a function of W_j and can be written in the form $V(p_{i\in[1,n]}) = \sum_j M_{i,j} \cdot W_j$, we structure the equations as a linear system:

$$\boxed{\begin{pmatrix} V(p_1) \\ \vdots \\ V(p_n) \end{pmatrix} = \begin{pmatrix} M_{1,1} & \cdots & M_{1,n} \\ \vdots & \ddots & \vdots \\ M_{n,1} & \cdots & M_{n,n} \end{pmatrix} \begin{pmatrix} W_1 \\ \vdots \\ W_n \end{pmatrix}} \tag{15}$$

where $V(p_i)$ is the desired displacements of p_i. The M matrix may be singular since the user is likely to specify a target shape with a different volume from the initial one. We therefore use the Moore-Penrose pseudo-inverse, which provides the least square solution for the W_i.

Gravity: and other curl-free environment forces can be easily added to our system $F_g(p_i) = g$. Note the strong persistant environment forces can cause the kinodynamic simulation over a small window of time to differ signicantly from the true dynamic simulation.

Squash-and-Stretch: is a desirable property for character animation [TJ81]. In our approach squash-and-stretch can be achieved conveniently by defining a force F_{ss} that pushes or pulls the deformable shape along an axis n_{ss}, with respect to a center $\tilde{c}$ (see Figure 11a,b):
$F_{ss}(p_i) = ((p_i - \tilde{c}) \cdot n_{ss})\, k_{ss}\, n_{ss}$.
A negative k_{ss} squashes the object, while a positive k_{ss} stretches the object.

Life: In reality active muscles produce joint motion. Animators, however, directly specify joint motion that muscles and skin follow passively. The *life* force causes muscles to deform as though they were actively causing the joint motion that has been authored by the animator. We show how adding the acceleration as an external force F_l makes the material actively deform:
$F_l(p_i) = k_l a_i$
Note that for $k_l = 1$, the acceleration component disappears completely and thus the overall dynamics can be *smoothly switched off*. As k_l is increased, the object becomes more of an actor.

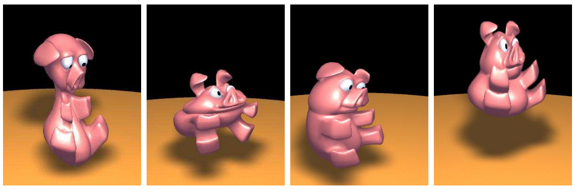

(a) (b) (c) (d)

Figure 11: *Animating squash-stretch (a)(b), and life (c)(d).*

5.4. Dynamic skinning workflow

The dynamic skinning workflow builds upon the traditional kinematic approach we presented in Section 4. The user places local muscle shapes under the skin and binds them to the skeletal structure, as is typical [PCLS05]. We automatically create vortex samples on the surface of muscle shapes (say at every control vertex) and their positions p_i are kept constant in a local skeletal frame. While they play no part in the actual skin deformation we find it useful to define proxy muscle shapes to interactively visualize how the muscle shape would deform. If we denote by V_j the velocity induced by the j^{th} muscle, a straightforward way to compute the new position of vortex point on the proxy muscle $p'_k = p_k + \delta t \sum_j V_j(q_k)$. In general vortex deformers define a dynamic 3D deformation velcity field that is added to the kinematically controlled velocity field in Equation 4 (parameterized over time instead of pose-space). The user can specify per muscle weights on a skin surface in a fashion identical to the deformable tissue weights in Section 4.3. If a skin vertex q_k, has a per muscle weight $m_{j,k}$ then deformation over δt is:

$$\boxed{\mathrm{q}'_k = \mathrm{q}_k + \delta t \sum_j m_{j,k}\, \mathrm{V}_j(\mathrm{q}_k)} \tag{16}$$

We now put the kinematic and dynamic skinning back together as outlined in Figure 4.

6. Kinodynamic skinning algorithm

Given a pose S and time window t, the skin is first kinematically deformed to the pose S_t, t units of time prior to the present ($S_t = S$ if no animation has been specified on the timeline). This skinning solution is a combination of a fold-over free deformation computed in closed form and a volume-preserving deformation velocity term computed numerically. A dynamic simulation is then run from pose S_t to S over time t. The kinematic deformation trajectories over time are simply added to those defined by the vortex dynamics. We integrate the trajectories using a simple explicit Euler solver. Since we work in a vorticity space, we can bound the values without affecting the visual quality, making our method stable for arbitrary long time steps. A step in this simulation is as follows:

1. Each vortex sample i $\{p_i, n_i, a_i, h_i, W_i, u_i\}$ (see Figure 10a) is initialized. p_i and n_i rigidly track the skeletal pose and the areas h_i of each sample is kept constant. W_i, a_i and u_i are initialized to 0.
2. Given the current skeletal pose compute the current position p_i, normal n_i and acceleration a_i in world coordinates for all vortex samples i.
3. Compute the external forces F_i (gravity, squash-and-stretch and life as described in Section 5.3. Using the external forces F_i and the acceleration a_i, update W_i with Equation 10.
4. Compute the elastic force $F_e(p_i)$, and its effect by updating u_i and to W_i, using Equation 12. Use the shear if enabled.
5. Add the velocity defined in Equation 16 to the pose-space kinematic skinning velocity defined in Equation 4 (as the skeleton transforms from s_t to S).
6. Numerically update the skin vertex positions and proxy muscles if desired, given the overall skin deformation velocity field.

7. Implementation and results

Our deformation approach has been implemented as a 3D deformer plug-in in *Maya* and can be procedurally combined with other deformers like lattices [SP86] or sculpt objects [PCLS05] for the general modeling and deformation of incompressible, elastic objects. As is standard practice in object skinning or deformation, a user can selectively associate skin points to joints or muscles in Maya. The advantages are improved efficiency and more precise association of anatomy to skin that when using simple falloff function shapes. The disadvantages are that forced association if not performed carefully can destroy the volume controlled and globally foldover-free surface deformations we set out to capture.

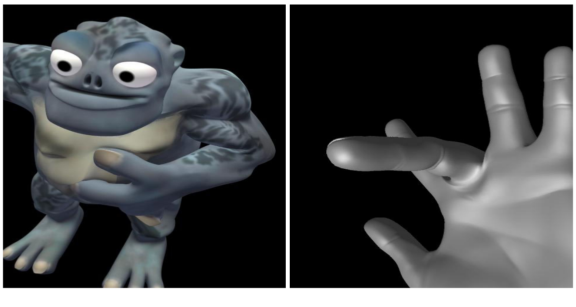

(a) (b)

Figure 12: *Extreme articulations: (a) Global fold-over free skinning shown as the hand pushes into the belly of the troll, (b) Metacarpophalangeal joint flexion.*

Animator control: Our formulation builds upon typical linear blend skinning (see Equation 1). Equation 5 discretizes the rotation of a joint i in Equation 1 into small steps, recomputing w_i for skin points at each step. This results in globally foldover-free skinning as seen in Figure 12a, where points on the trollŠs belly acquire a non-zero weight with respect to arm joints as the arm moves into the belly. Since the weights w_i are recomputed they need to be defined in 3D space. We accomplish this using ellipsoidal and arbitrary star shaped [PCLS05] falloff functions and user painted surface weights that falloff with distance from the surface. We provide two different falloff functions that capture bone or soft tissue behavior.

Volume-preservation is further built upon the foldover-free skinning of Equation 5 using a Clebsch decomposition (see Appendix A or [vFTS06]) of the rotation around a joint i (see Equation 7). The user can control the amount of volume conservation in two orthogonal directions using weights α_i and β_i. These weights scale ϕ_i and ψ_i by ${w_i}^{\alpha_i}$ and ${w_i}^{\beta_i}$ prior to computing their gradients and subsequent cross product, so that the final vector field remains divergence-free. While volume-preservation ($\alpha_i + \beta_i = 1$) is desirable it can be excessive in extreme articulation (see Figure 7b) and users can modulate it by reducing α_i and β_i for large rotations (see Figure 7c and Figure 12b). Indeed anatomic structures and cavities have some compressibility and animators can qualitatively control this in our approach. Further control over where the volume goes comes from the fact that only soft tissue functions (bones have a zero weight gradient) participate in the volume preservation term of Equation 7.

User control over dynamics is simpler. The user simply positions muscle primitives under the skin (see Figure 9) similar to many anatomic skinning approaches and specifies any external forces as described in section 5.3. The radius of influence r in Equation 11 allows a user to control the locality of muscle deformation. The dynamic window and physical muscle attributes of density ρ, stiffness k_s and damping k_d are also under interactive animator control. The quality of the motion can be adjusted and visualized at a high frame rate by displaying the deformed muscles (see Figure 9b).

Efficiency: The refresh rates of our technique within Maya (including Maya's display and dependency graph evaluation overhead) on models (see Figure 9) of around 8000 vertices are around:
30 fps for Maya smooth skinning (linear blend skinning).
5 fps for the Clebsch decomposition based kinematic skinning.
12 fps for the vortex deformer based dynamics with 12 muscles.
4 fps for the complete kinodynamic solution.

Theoretically the complexity scales linearly with both the number of vertices and the number of muscle vortices (without modeling material shear). In practice, for the complete kinodynamic solution we observe 4, 9, 24 and 42 fps for 8000, 4000, 2000 and 1000 vertices and 12 muscles, measured on a Pentium®4–2.40GHz–512 Mb RAM.

Refresh rates of the vortex dynamics examples within our standalone program are much better, > 90 fps, showing the promise of better performance if these techniques are implemented directly within a commercial animation system.

8. Conclusion

We have presented a number of concepts in this paper. Firstly we find the idea of kinodynamic skinning to be worthwhile in the context of transient dynamic deformations and may be applicable to other combinations of geometric and physically based simulations. Our kinematic skinning approach is effective and the closed form fold-over free term which has by far the largest deformation magnitude (see Appendix B), makes the overall approach very stable. We also extend the bend deformation of [vFTS06] to add a volume-preserving term to the kinematic skin (see Appendix A). Finally we show vortex deformers to be a compelling technique for modeling the dynamics of elastic incompressible materials. Most importantly, all these techniques work well together within a typical interactive character skinning framework (see Figure 13).

Popular example based skinning approaches that use scattered data interpolation can be thought of as having an example set of bind poses with corresponding skin geometry [LCF00] [WP02], [MG03]. We note here that our kinematic skinning is constructed from a single bind pose. Using multiple example skins to be combined in a volume-preserving fashion is subject to future work.

Our dynamics approach makes a number of approximations and assumptions tailored to expressive character animation and is thus not suited to applications where the precise modeling of anatomic tissue is required [TSB*05]. The acceleration and velocity of vortex samples are strictly controlled by the skeletal frame they are bound to and not induced in terms of vorticity w. Our muscle model for is thus a mixture of the vorticity and velocity equation of motion. Although our muscles primitives preserve volume, the skin surface is an approximation to the muscles. Finally, in our current implementation the dynamic interaction between two muscles from different parts of a character falls off continuously with the distance between them. In reality these muscles only interact if and when these body parts make contact. We acknowledge these limitations of our model as subject to future work.

Our emphasis at present has been an advance in the quality and control of kinodynamic skin deformation. Although we do not compete with linear blend skinning on efficiency for games and highly interactive applications, for high-end animation authoring the rates are adequately interactive. Furthermore, the use of low-res skin geometry and proxy muscles is common in commercial animation when character complexity hinders animator interactivity.

Despite some of these shortcomings our approach makes a number of contributions to the current state of the art of character skinning.

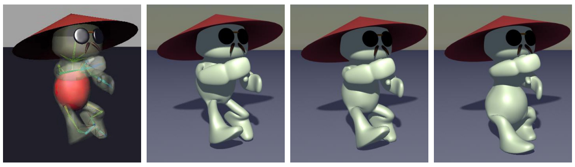


Figure 13: *A stomach muscle adds dynamism to Master Pai.*

Appendix A: Clebsch Decomposition of a Rotation

Let us look at the velocity of a rotation at a point p around an axis n (see Figure 14):

$$v(p) = \theta\, n \times (p-c) + t\, n \qquad (17)$$

Let $q = p - c$. Vectors $a = n \times q \times n$, $b = n \times q$ and n form a right-handed orthonormal basis . Since a rotation is a solenoidal or divergence-free flow, it can be decomposed in two irrotational or curl-free vector fields $\nabla\phi$ and $\nabla\psi$ orthogonal to v [Rut89]:

$$\begin{array}{l} v = \nabla\phi \times \nabla\psi \\ \text{where} \left\{ \begin{array}{lcl} \nabla\phi & = & \theta n - t\frac{b}{(b\cdot b)} \\ \nabla\psi & = & a \end{array} \right. \end{array} \qquad (18)$$

The vectors of the field $\nabla\psi$ are contained in planes perpendicular to n. The integration of $\nabla\phi$ and $\nabla\psi$ gives:

$$\begin{array}{lcl} \phi & = & \theta\, q\cdot n - t \arctan(q\cdot a,\, q\cdot b) \\ \psi & = & \frac{1}{2}(b\cdot b) \end{array}, \qquad (19)$$

The pair (ϕ, ψ) is the Clebsch decomposition of the rotation.

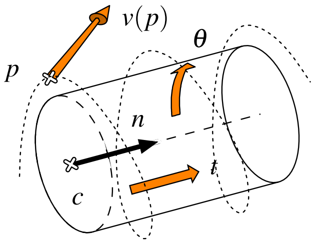


Figure 14: *Variables defining the velocity field of a rotation. Any point on the line (c,n) may be used instead of c. Note that n does not have to correspond to the axis of a joint.*

Appendix B: Fraction of a Matrix

We denote with a non-rational exponent the fraction of a matrix: M^l. If our method is implemented in an animation package, this can be done by using the package's matrix parameterization if available, or with $M^l = \exp(l \log M)$ in the more general case, where:

$$\log(M) = -\sum_{k=1}^{\infty} \frac{(\mathrm{I}-M)^k}{k} \qquad \exp(N) = \sum_{k=0}^{\infty} \frac{N^k}{k!} \qquad (20)$$

There is a stable way to evaluate these series numerically [Ale02], and in the case where M is the matrix of a twist, there is a closed-form which we provide here. First, the matrix M of the twist must be expressed in terms of (t,θ,n,c). Let us write M in terms of a 3×3 submatrix R and a point h:

$$M = \begin{pmatrix} & R & & h \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad (21)$$

If $R = \mathrm{I}$, then M is a translation, and:

$$\begin{array}{rclcrcl} t & = & \|h\| & \quad & n & = & h/t \\ \theta & = & 0 & & c & = & 0 \end{array} \qquad (22)$$

Otherwise, n is the eigenvector associated to eigenvalue 1 of the matrix R, $t = h\cdot n$, c is found by solving $(R-\mathrm{I})\cdot c = t\ n - h$, and given any unit vector u orthogonal to n, $\theta = \arctan(u\cdot(M\cdot u), (n\times u)\cdot(M\cdot u))$. This decomposition of M provides the closed-form of $\log M$:

$$\log M = \begin{pmatrix} 0 & -w_z & w_y & m_x \\ w_z & 0 & -w_x & m_y \\ -w_y & w_x & 0 & m_z \\ 0 & 0 & 0 & 0 \end{pmatrix} \qquad (23)$$

$$\text{where } w = \theta\ n \text{ and } m = c\times w + t\ n$$

And the closed-form for $\exp N$ is:

$$\exp N = \mathrm{I} + N + \frac{1-\cos\theta}{\theta^2} N^2 + \frac{\theta - \sin\theta}{\theta^3} N^3 \qquad (24)$$

Note that for a translation, $\lim_{\theta\to 0} \exp N = \mathrm{I} + N$.

Appendix C: Discontinuity Model

Let us consider the equations that describes the motion of the velocity field of an inviscid incompressible fluid:

$$\left\{ \begin{array}{rcl} \frac{\mathrm{d}v}{\mathrm{d}t} & = & -\frac{1}{\rho}\nabla p + F \\ \nabla\cdot v & = & 0 \end{array} \right. \qquad (25)$$

where p is the pressure, ρ is the density and F is the sum of the external forces per unit mass. Let us multiplying the first equation by ρ, apply to it the curl operator $\nabla\times$, and assume that the external forces is a curl-free field $\nabla\times F = 0$. We obtain the equation of the motion of w:

$$\frac{\mathrm{d}w}{\mathrm{d}t} = \frac{\nabla\rho}{\rho} \times (F-a) + w\cdot\nabla v \qquad (26)$$

The last term means that w is carried by material lines that stretch with the fluid. Since we focus on elastic material close to their equilibrium state, we assume that $w\cdot\nabla v = 0$. To define ρ, we assume the density ρ_0 and ρ_1 of the two continua constant, thus the function ρ is discontinuous at the continua interface. Let us define the signed distance to the interface as negative in the material M_0 and positive in M_1, and use the Heaviside step function H:

$$\rho(d) = \rho_1 H(d) + \rho_0 H(-d) \qquad (27)$$

$$\text{where} \quad H(d) = \left\{ \begin{array}{ll} 0 & \text{if } d < 0 \\ \frac{1}{2} & \text{if } d = 0 \\ 1 & \text{otherwise} \end{array} \right.$$

Dealing with the discontinuity is cumbersome. Fortunately, the analytical integration of Equation 26 in a small volume V makes the function H disappear:

$$\iiint_V \frac{\mathrm{d}w}{\mathrm{d}t}\,\mathrm{d}V = \left\{ \begin{array}{l} 0 \text{ if } V \text{ does not intersects the interface} \\ 2h\frac{\rho_1-\rho_0}{\rho_1+\rho_0} n\times(F-a) \text{ otherwise} \end{array} \right.$$

Finally, using Reynolds' transport theorem for incompressible materials, $\iiint_V \frac{\mathrm{d}w}{\mathrm{d}t}\,\mathrm{d}V = \frac{\mathrm{d}}{\mathrm{d}t}\iiint_V w\,\mathrm{d}V$ [Rut89]. Thus the above describes the variation of vorticity for elements of volume near the interface.

References

[ACP02] ALLEN B., CURLESS B., POPOVI&#263; Z.: Articulated body deformation from range scan data. *ACM Trans. Graph. 21*, 3 (2002), 612–619.

[ACWK04] ANGELIDIS A., CANI M.-P., WYVILL G., KING S.: Swirling-Sweepers: Constant-Volume Modeling. In *Pacific Graphics 2004* (Oct 2004), pp. 10–15.

[AHS03] ALBRECHT I., HABER J., SEIDEL H.-P.: Construction and animation of anatomically based human hand models. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer animation* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 98–109.

[Ale02] ALEXA M.: Linear Combination of Transformations. *ACM Trans. Graph. 21*, 3 (Jul 2002), 380–387.

[CBC*05] CAPELL S., BURKHART M., CURLESS B., DUCHAMP T., POPOVIĆ Z.: Physically Based Rigging for Deformable Characters. In *SCA'05: Proc. of the 2005 Symposium on Computer Animation* (Jul 2005), pp. 301–310.

[CGC*02] CAPELL S., GREEN S., CURLESS B., DUCHAMP T., POPOVIĆ Z.: Interactive Skeleton-Driven Dynamic Deformations. *ACM Trans. Graph. 21*, 3 (Jul 2002), 586–593.

[CHP89] CHADWICK J. E., HAUMANN D. R., PARENT R. E.: Layered construction for deformable animated characters. In *SIGGRAPH '89: Proceedings of the 16th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1989), ACM Press, pp. 243–252.

[HYC*05] HYUN D., YOON S., CHANG J., SEONG J., KIM M., JUTTLER B.: Sweep-based Human Deformation. In *Pacific Graphics* (Oct 2005), pp. 542–550.

[JP02] JAMES D. L., PAI D. K.: DyRT: Dynamic Response Textures for Real Time Deformation Simulation with Graphics Hardware. *ACM Trans. Graph. 21*, 3 (Jul 2002), 582–585.

[KJP02] KRY P. G., JAMES D. L., PAI D. K.: Eigenskin: real time large deformation character skinning in hardware. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2002), ACM Press, pp. 153–159.

[Kot91] KOTIUGA P.: Clebsch potentials and the visualization of three-dimensional solenoidal vector fields. *IEEE Transactions on Magnetics 27(5)* (Sep 1991), 3986–3989.

[LCA05] LARBOULETTE C., CANI M.-P., ARNALDI B.: Dynamic skinning: adding real-time dynamic effects to an existing character animation. In *SCCG '05: Proceedings of the 21st spring conference on Computer graphics* (New York, NY, USA, 2005), ACM Press, pp. 87–93.

[LCF00] LEWIS J., CORDNER M., FONG N.: Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation. In *ACM Trans. Graph. (Proc of SIGGRAPH'99)* (Jul 2000), pp. 165–172.

[LTW95] LEE Y., TERZOPOULOS D., WALTERS K.: Realistic modeling for facial animation. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1995), ACM Press, pp. 55–62.

[MG03] MOHR A., GLEICHER M.: Building Efficient, Accurate Character Skins from Examples. *ACM Trans. Graph. 22*, 3 (Jul 2003), 562–568.

[MHTG05] MULLER M., HEIDELBERGER B., TESCHNER M., GROSS M.: Meshless deformations based on shape matching. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), ACM Press, pp. 471–478.

[MT97] MOCCOZET L., THALMANN N. M.: Dirichlet free-form deformations and their application to hand simulation. In *CA '97: Proceedings of the Computer Animation* (Washington, DC, USA, 1997), IEEE Computer Society, p. 93.

[NMK*05] NEALEN A., MALLER M., KEISER R., BOXERMAN E., CARLSON M.: Physically based deformable models in computer graphics. In *Eurographics 2005 State of the Art Report* (Sep 2005).

[PCLS05] PRATSCHER M., COLEMAN P., LASZLO J., SINGH K.: Outside-In Anatomy Based Character Rigging. In *SCA'05: Proc. of the 2005 Symposium on Computer Animation* (Jul 2005), pp. 329–338.

[Rut89] RUTHERFORD A.: *Vectors, Tensors, and the Basic Equations of Fluid Mechanics*. Dover Publications, Inc, 1989.

[SCFRC01] SLOAN P.-P. J., CHARLES F. ROSE I., COHEN M. F.: Shape by example. In *SI3D '01: Proceedings of the 2001 symposium on Interactive 3D graphics* (New York, NY, USA, 2001), ACM Press, pp. 135–143.

[Sin95] SINGH K.: *Realistic Human Figure Synthesis and Animation for VR Applications*. PhD thesis, The Ohio State University, 1995.

[SK00] SINGH K., KOKKEVIS E.: Skinning Characters using Surface-Oriented Free-Form Deformations. In *Graphics Interface* (2000), pp. 35–42.

[SP86] SEDERBERG T., PARRY S.: Free-Form Deformation of Solid Geometric Models. In *ACM Trans. Graph. (Proc of SIGGRAPH'86)* (Aug 1986), pp. 151–160.

[SPCM97] SCHEEPERS F., PARENT R. E., CARLSON W. E., MAY S. F.: Anatomy-Based Modeling of the Human Musculature. In *ACM Trans. Graph. (Proc of SIGGRAPH'97)* (Aug 1997), pp. 163–172.

[TJ81] THOMAS F., JOHNSTON O.: *The illusion of life*. Hyperion, 1981.

[TSB*05] TERAN J., SIFAKIS E., BLEMKER S. S., NG-THOW-HING V., LAU C., FEDKIW R.: Creating and simulating skeletal muscle from the visible human data set. *IEEE Transactions on Visualization and Computer Graphics 11*, 3 (2005), 317–328.

[TSIF05] TERAN J., SIFAKIS E., IRVING G., FEDKIW R.: Robust Quasistatic Finite Elements and Flesh Simulation. In *SCA'05: Proc. of the 2005 Symposium on Computer Animation* (Jul 2005), pp. 181–190.

[vFTS06] VON FUNCK W., THEISEL H., SEIDEL H.-P.: Vector field based shape deformations. *ACM Trans. Graph. 25*, 3 (2006), 1118–1125.

[WG97] WILHELMS J., GELDER A. V.: Anatomically Based Modeling. In *ACM Trans. Graph. (Proc of SIGGRAPH'97)* (Aug 1997), pp. 173–180.

[WP02] WANG X. C., PHILLIPS C.: Multi-weight enveloping: least-squares approximation techniques for skin animation. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation* (New York, NY, USA, 2002), ACM Press, pp. 129–138.