Regularized Kelvinlets: Sculpting Brushes based on Fundamental Solutions of Elasticity

Pixar Technical Memo #17-03

FERNANDO DE GOES, Pixar Animation Studios
DOUG L. JAMES, Pixar Animation Studios, Stanford University

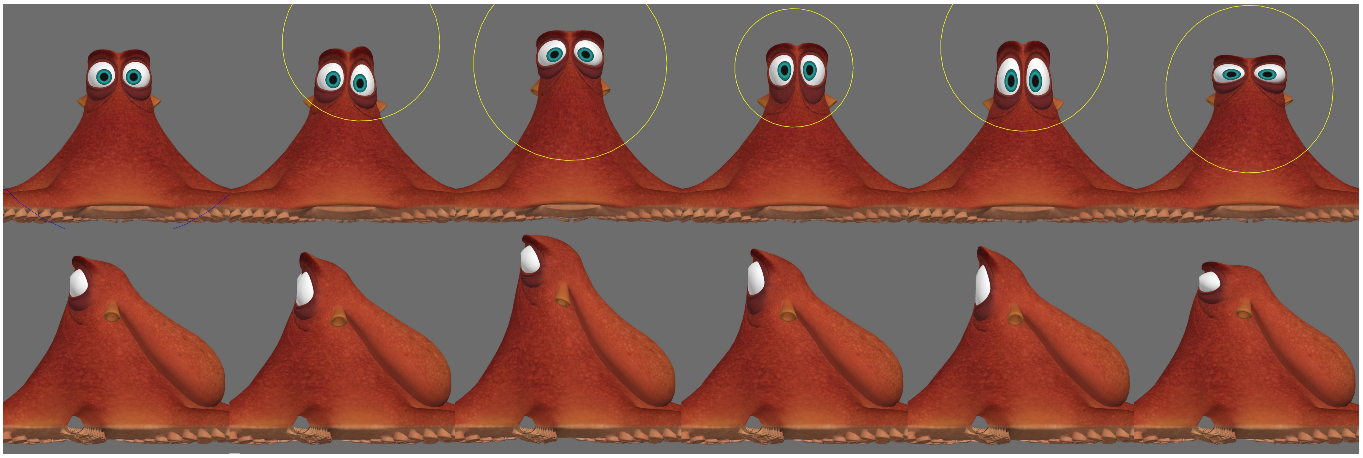


Fig. 1. **Regularized Kelvinlets:** Examples of expressions generated by regularized Kelvinlets, a novel approach for sculpting that provides space deformations with real-time feedback and interactive volume control based on closed-form analytical solutions of linear elasticity. ©Disney/Pixar

We introduce a new technique for real-time physically based volume sculpting of virtual elastic materials. Our formulation is based on the elastic response to localized force distributions associated with common modeling primitives such as grab, scale, twist, and pinch. The resulting brush-like displacements correspond to the regularization of fundamental solutions of linear elasticity in infinite 2D and 3D media. These deformations thus provide the realism and plausibility of volumetric elasticity, and the interactivity of closed-form analytical solutions. To finely control our elastic deformations, we also construct compound brushes with arbitrarily fast spatial decay. Furthermore, pointwise constraints can be imposed on the displacement field and its derivatives via a single linear solve. We demonstrate the versatility and efficiency of our method with multiple examples of volume sculpting and image editing.

CCS Concepts: • **Computing methodologies → Volumetric models**;

Additional Key Words and Phrases: sculpting brushes, linear elasticity.

ACM Reference format:
Fernando de Goes and Doug L. James. 2017. Regularized Kelvinlets: Sculpting Brushes based on Fundamental Solutions of Elasticity Pixar Technical Memo #17-03 . *ACM Trans. Graph.* 36, 4, Article 40 (July 2017), 11 pages.
DOI: http://dx.doi.org/10.1145/3072959.3073595

1 INTRODUCTION

Digital sculpting is a core component in modeling and animation packages, e.g., ZBrush, Scultpris, MudBox, Maya, or Modo. While purely geometric approaches are commonplace for shape editing, physically based deformers have long been sought to provide a more natural and effective sculpting tool for digital artists. However, existing physics-driven methods tend to have practical impediments for interactive design: they are inherently slow due to the need to numerically solve equations of the underlying discrete deformation model, and also involve tedious setup steps such as volumetric meshing, boundary condition specification, or preprocessing to accelerate the simulation.

In this paper, we introduce a novel sculpting tool that offers interactive and physically plausible deformations. Our approach is based on the regularization of the fundamental solutions to the equations of linear elasticity for specific brush-like forces (e.g., grab, twist, pinch) applied to a virtual infinite elastic space. Since the fundametal solutions of linear elasticity for singular loads are known as Kelvinlets [Phan-Thien and Kim 1994], we refer to our new solutions as *regularized Kelvinlets*. This formulation leads to analytical closed-form expressions, thus making our method free of any geometric discretization, computationally intensive solve, or large memory requirement. Instead, our brush-like deformations can be rapidly evaluated on the fly for sculpting sessions, with interactive control of the volume compression of the elastic material (Figure 1). To address the typical long-range falloffs of the fundamental solutions, we present a multi-scale extrapolation scheme that builds compound brushes with arbitrarily fast decay, thereby enabling highly localized edits. For more specialized control, we also propose a linear combination of brushes that imposes pointwise constraints on displacements and gradients through a single linear solve. In summary, regularized Kelvinlets offer volume-aware deformations that are efficient to evaluate, easy to parallelize, and complementary to existing modeling techniques.

2 RELATED WORK

Sculpting methods for digital shapes have been investigated extensively in graphics, see survey in [Cani and Angelidis 2006]. The most common sculpting tool is the family of grab/twist/scale/pinch brushes available in commercial modeling packages, e.g., [Autodesk 2016]. In these brushes, a single affine transformation is applied to every point in space and scaled by a predefined falloff. The simple setup and fast analytical evaluation make these tools suitable for highly detailed modeling, as well as animation and post-simulation edits. Motivated by these premises, our work augments existing brush deformations with interactive elastic response.

In order to provide volume-aware sculpts, Angelidis et al. [2004] introduced the notion of sweeper brushes, which deform the space with no self-intersection by advecting points along user-authored displacement fields. This technique was later extended to swirling-sweepers [Angelidis et al. 2006] that ensure constant-volume deformations by emulating the formation of vortex rings in incompressible fluids. Similarly, the work of von Funck et al. [2006] proposed deformations with divergence-free displacements generated by user-designed scalar fields. Our approach shares similar properties with sweeper brushes, in particular, it guarantees foldover-free sculpts guided by displacement fields. However, we model a wider family of brush deformations that incorporates volumetric responses to elastic materials with compression. Also, instead of crafting functions that produce incompressible flows, we design smooth force distributions and rely on analytical solutions to the equations of elasticity to produce physically based deformations.

Solid modeling and carving were also explored using voxel discretizations [Galyean and Hughes 1991], sampled distance functions [Frisken et al. 2000], and adaptive resolution [Ferley et al. 2001]. In [Dewaele and Cani 2004], physics-aware volume sculpting was addressed through voxel-based approximations of elastic and plastic deformations. However, these methods are restricted to grid-like data structures and tend to produce over-smoothed shapes. Other interactive techniques rely on user-controlled handles to drive volumetric deformations. This includes lattice deformers [Coquillart 1990; Sederberg and Parry 1986], cage-based generalized barycentric coordinates [Joshi et al. 2007; Lipman et al. 2008], linear blending skinning [Magnenat-Thalmann et al. 1988], wire curves [Singh and Fiume 1998], and sketch editing [Cordier et al. 2016], to cite a few. In [Jacobson et al. 2011], various handle types were supported by optimizing a unified linear blending scheme. In contrast to our work, these approaches require a careful placement of handles and a precomputation step that binds handles to the deforming objects. Moreover, the resulting deformations are purely geometric with limited volume control.

Aiming at physically plausible deformations, some methods combined handle-based systems with simulation techniques. For instance, Jacobson et al. [2012] posed skinned skeletons by minimizing an elastic potential energy, while Hahn et al. [2012] projected the equations of elasticity to the subspace spanned by animation rigs. In a similar way, the work of Ben-Chen et al. [2009] deformed a cage mesh by optimizing the rigidity of the associated deformation. In [Kavan and Sorkine 2012], an elasticity-inspired skinning was presented by constructing joint deformers. Physics-driven video editing was also considered by combining simulation with image aware constraints [Bazin et al. 2016]. It is also worth mentioning advances in elasticity solvers that speed up quasi-static simulations in 2D [Setaluri et al. 2014] and 3D [McAdams et al. 2011]. Model reduction can also make simulations orders of magnitude faster by precomputing deformation modes [Barbič and James 2005] and multi-domain subspaces [Barbič and Zhao 2011]. Surface-only methods were also proposed to approximate elastic materials using, e.g., rigidity penalties [Sorkine and Alexa 2007] and moving frames [Lipman et al. 2007]. Since these approaches rely on mesh discretizations and non-linear solvers, we consider them complementary to our mesh-free closed-form solutions.

Our work is closely related to the concept of regularized Stokeslets. Coined by Hancock [1953], Stokeslets are singular fundamental solutions of Stokes flows (i.e., incompressible fluid flows with low Reynolds number [Chwang and Wu 1975]) based on pointwise loads. To remove the singularity of these solutions, Cortez [2001] introduced regularized Stokeslets as an alternative that replaces impulses by smoothed force distributions. Extensions to doublets and the method of images were later derived in [Ainley et al. 2008; Cortez et al. 2005], and employed to model the trajectories of self-propelled micro-organisms. Instead, we introduce the regularization of the fundamental solutions of linear elasticity, thus generalizing regularized Stokeslets to arbitrary elastic materials. In particular, our formulation reproduces regularized Stokeslets in the case of incompressible elasticity. In addition, we present a new approach to design regularized solutions with faster decays.

Finally, we point out that fundamental solutions are central to integral formulations of elasticity and discretizations such as the boundary element method (BEM) [Brebbia et al. 2012]. In [James and Pai 1999], BEM was employed to deform surface meshes by approximating contact mechanics. However, this technique requires the precomputation and incremental updates of boundary conditions for displacements and surface tractions. This approach was later accelerated by compressing singular fundamental solutions of linear elasticity based on wavelets [James and Pai 2003]. In contrast, we advocate the use of regularized fundamental solutions for digital sculpting, with no boundary specification or mesh discretization.

3 BACKGROUND

Before presenting our new sculpting tools, we first review core concepts of linear elastostatics upon which our formulation is based. In the next sections, we focus our exposition on 3D linear elasticity and postpone the 2D case to Section 8, where the analogous 2D expressions are derived using our 3D results. For more details on linear elastostatics, we point the reader to [Slaughter 2002].

Elastostatics: In an infinite 3D continuum, the quasi-static equilibrium state of linear elasticity is determined by a displacement field $\boldsymbol{u} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ that minimizes the elastic potential energy

$$E(\boldsymbol{u}) = \frac{\mu}{2}\|\nabla \boldsymbol{u}\|^2 + \frac{\mu}{2(1-2\nu)}\|\nabla\cdot\boldsymbol{u}\|^2 - \langle \boldsymbol{b}, \boldsymbol{u}\rangle, \tag{1}$$

where $\|\cdot\|$ and $\langle\cdot,\cdot\rangle$ are integrated over the infinite elastic volume. The first term in (1) controls the smoothness of the displacement field, the second term penalizes infinitesimal volume change, and the last term indicates the external body forces $\boldsymbol{b}$ to be counteracted. In this formulation, linear homogeneous and isotropic material parameters are given by the elastic shear modulus μ and the Poisson ratio ν. Note that, while the former is a simple scaling factor indicating the material stiffness, the latter dictates the material compressibility. By computing the critical point of (1), one can associate the optimal displacement field $\boldsymbol{u}$ with the solution of the well-known Navier-Cauchy equation (see, e.g., [Slaughter 2002]):

$$\mu\Delta\boldsymbol{u} + \frac{\mu}{(1-2\nu)}\nabla(\nabla\cdot\boldsymbol{u}) + \boldsymbol{b} = 0. \tag{2}$$

Kelvinlets: In the case of a concentrated body load due to a force vector $\boldsymbol{f}$ at a point $\boldsymbol{x}_0$, i.e., $\boldsymbol{b}(\boldsymbol{x}) = \boldsymbol{f}\,\delta(\boldsymbol{x}-\boldsymbol{x}_0)$, the solution of (2) defines the (singular) fundamental solution of linear elasticity, also known as the *Kelvin's state* [Kelvin 1848] or *Kelvinlet* [Phan-Thien and Kim 1994], which can be written as

$$\boldsymbol{u}(\boldsymbol{r}) = \left[\frac{(a-b)}{r}\boldsymbol{I} + \frac{b}{r^3}\boldsymbol{r}\boldsymbol{r}^t\right]\boldsymbol{f} \equiv \boldsymbol{\mathcal{K}}(\boldsymbol{r})\,\boldsymbol{f}, \tag{3}$$

where $\boldsymbol{r} = \boldsymbol{x}-\boldsymbol{x}_0$ is the relative position vector from the load location $\boldsymbol{x}_0$ to an observation point $\boldsymbol{x}$, and $r = \|\boldsymbol{r}\|$ is its norm. The coefficients in (3) are of the form $a = 1/(4\pi\mu)$ and $b = a/[4(1-\nu)]$. Here, we refer to $\boldsymbol{\mathcal{K}}(\boldsymbol{r})$ as the Green's function for linear elasticity that defines a 3×3 matrix mapping the force vector $\boldsymbol{f}$ at $\boldsymbol{x}_0$ to the displacement vector $\boldsymbol{u}$ at a relative position $\boldsymbol{r}$. Notice that the first term in $\boldsymbol{\mathcal{K}}(\boldsymbol{r})$ resembles the Laplacian Green's function scaled by a constant, while the second term controls volume compression by modifying the elastic response based on the alignment of the relative position vector $\boldsymbol{r}$ to the force vector $\boldsymbol{f}$. Moreover, observe that no boundary conditions were needed for the definition of Kelvinlets, since these displacement fields are the minimizer of the elastic potential energy over the infinite space.

Stokeslets: An important special case of linear elastostatics occurs for incompressible materials, i.e., $\nu = 1/2$. In this case, the divergence penalty term in (1) becomes a hard constraint $\nabla\cdot\boldsymbol{u} = 0$ that ensures incompressible displacements. The solution for this constrained minimization can be computed via the Stokes equation:

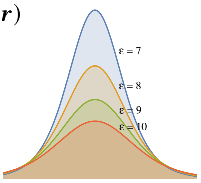
$$\mu\Delta\boldsymbol{u} + \boldsymbol{b} - \nabla p = 0 \quad \text{subject to} \quad \nabla\cdot\boldsymbol{u} = 0, \tag{4}$$

where p is the pressure scalar field acting as a Lagrangian multiplier that enforces the divergence-free constraint. For a pointwise load with force vector $\boldsymbol{f}$, the (singular) fundamental solution of (4) is known as the *Stokeslet* [Chwang and Wu 1975; Hancock 1953], and matches the expression in (3) with $\nu = 1/2$ (i.e., $b = a/2$).

Deformation Gradient: From a displacement field $\boldsymbol{u}(\boldsymbol{x}-\boldsymbol{x}_0)$, an arbitrary point $\boldsymbol{x}$ embedded in a linear elastic material is deformed to $\boldsymbol{x}+\boldsymbol{u}(\boldsymbol{x}-\boldsymbol{x}_0)$. The associated deformation gradient is then defined by a 3×3 matrix of the form $\boldsymbol{G}(\boldsymbol{x}-\boldsymbol{x}_0) = \boldsymbol{I}+\nabla\boldsymbol{u}(\boldsymbol{x}-\boldsymbol{x}_0)$. By analyzing this matrix $\boldsymbol{G}(\boldsymbol{r})$, we can obtain different properties of the displacement field $\boldsymbol{u}(\boldsymbol{r})$ (see, e.g., [Slaughter 2002]). For instance, the skew-symmetric part of $\nabla\boldsymbol{u}(\boldsymbol{r})$ indicates the infinitesimal rotation induced by $\boldsymbol{u}(\boldsymbol{r})$, while its symmetric part corresponds to the elastic strain and determines the infinitesimal stretching. The strain tensor can be also decomposed into a trace term that represents the uniform scaling of the volume of the elastic medium, and a traceless term that informs the undergoing pinching deformation. We will use these deformation gradient decompositions in Section 6 to construct twist, scale, and pinch elastic brushes.

4 3D REGULARIZED KELVINLETS

The concentrated body load at a single point $\boldsymbol{x}_0$ introduces a singularity to the Kelvinlet solution, making its displacements and derivatives indefinite nearby $\boldsymbol{x}_0$. For this reason, the singular Kelvinlet is numerically unsuitable for digital sculpting. To overcome this issue, we adopt the regularization scheme introduced in [Cortez 2001], and consider a smoothed body load $\boldsymbol{b}(\boldsymbol{r}) = \boldsymbol{f}\rho_\varepsilon(\boldsymbol{r})$ with force vector $\boldsymbol{f}$ and normalized density function $\rho_\varepsilon(\boldsymbol{r})$ distributed around $\boldsymbol{x}_0$ by a radial scale $\varepsilon > 0$. Similar to [Cortez et al. 2005], we define the regularized distance $r_\varepsilon = \sqrt{r^2+\varepsilon^2}$ and set the normalized density function to be of the form (see inset)



$$\rho_\varepsilon(\boldsymbol{r}) = \frac{15\varepsilon^4}{8\pi}\frac{1}{r_\varepsilon^7}. \tag{5}$$

As detailed in Appendix A, the solution to the elastostatics equation (2) associated with the load distribution (5) corresponds to

$$\boxed{\boldsymbol{u}_\varepsilon(\boldsymbol{r}) = \left[\frac{(a-b)}{r_\varepsilon}\boldsymbol{I} + \frac{b}{r_\varepsilon^3}\boldsymbol{r}\boldsymbol{r}^t + \frac{a}{2}\frac{\varepsilon^2}{r_\varepsilon^3}\boldsymbol{I}\right]\boldsymbol{f} \equiv \boldsymbol{\mathcal{K}}_\varepsilon(\boldsymbol{r})\boldsymbol{f}.} \tag{6}$$

This solution, which we name *regularized Kelvinlet*, forms the building block of our sculpting technique. Compared to (3), the regularized Kelvinlet includes an extra $\varepsilon^2/r_\varepsilon^3$ term, and reproduces the singular case as the radial scale ε approaches zero. Moreover, the undesirable $1/r$ singularity of Kelvinlets is now regularized at $r = 0$ using a $1/r_\varepsilon$ term, while keeping the $O(1/r)$ asymptotic decay as $r \longrightarrow \infty$. Therefore, the resulting displacements are finite, differentiable, and localized around $\boldsymbol{x}_0$. One can also verify that the deformation gradient at $\boldsymbol{x}_0$ is trivial, i.e., $\nabla\boldsymbol{u}_\varepsilon(0) = 0$, implying that the deformation at the point under the load is locally rigid. It is also worth mentioning that different density functions (and thereby regularized solutions) could be used, and our choice is based on the formulation of regularized Stokeslets [Cortez et al. 2005]. In fact, one can easily verify that the regularized Kelvinlets simplify to the regularized Stokeslets in the incompressible limit ($\nu = 1/2$).

Equipped with the regularized Kelvinlets, we can now construct a sculpting brush that assigns the load center $\boldsymbol{x}_0$ to the brush tip and sets the brush radius to ε. This indicates that, by increasing the brush size, the load distribution becomes wider. Given a force vector $\boldsymbol{f}$, we can then evaluate the displacement generated by the regularized Kelvinlet analytically for every point in $\mathbb{R}^3$. As a result, we can move points along this displacement field, thus defining our physically based space deformation. It is also convenient to parameterize the

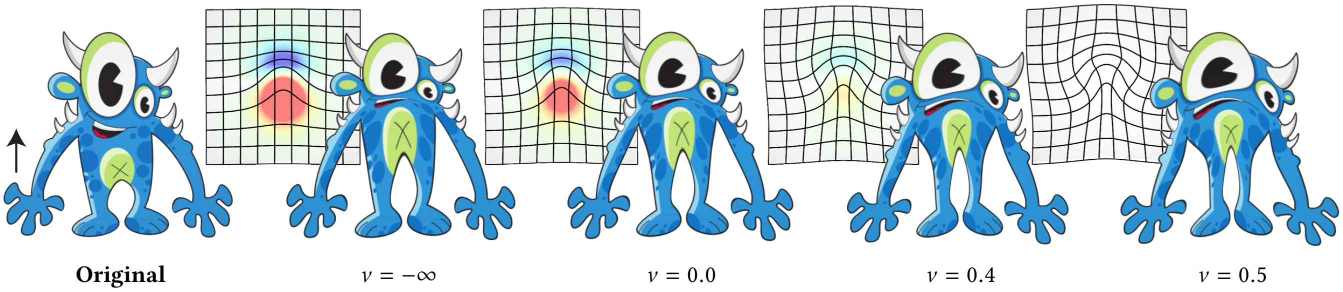


Fig. 2. **Elastic Grab Brush:** Regularized Kelvinlets offer interactive elastic response to grab displacements. This example shows the result of a bi-scale sculpting brush in 2D computed with a series of Poisson ratio values ν. Notice how larger values of ν lead to more bulging, e.g., along the body silhouette. The grid plots illustrate the local area loss and gain associated with different Poisson ratios. The pseudocolors encode a range from compression of 50% (blue) to dilation of 50% (red) relative to the rest area. Input image courtesy of Mirela Ben-Chen.

force vector $\boldsymbol{f}$ in terms of the brush tip displacement $\bar{\boldsymbol{u}}$. To this end, we expand (6) with the linear constraint $\boldsymbol{u}_\varepsilon(0) = \bar{\boldsymbol{u}}$, leading to

$$\boldsymbol{u}_\varepsilon(\boldsymbol{r}) = c\,\varepsilon\,\boldsymbol{\mathcal{K}}_\varepsilon(\boldsymbol{r})\,\bar{\boldsymbol{u}}, \tag{7}$$

where $c = 2/(3a - 2b)$. Note that this parameterization makes our brush independent of the shear modulus μ, so the only material parameter affecting the deformation is the Poisson ratio ν. Figure 2 shows regularized Kelvinlet deformations with varying Poisson ratio. In particular, observe the increasing bulging as the elastic material becomes incompressible. At last, we point out that, by setting the (unphysical but feasible) value $\nu = \pm\infty$, the second term in (6) is annihilated and our displacement field simplifies to a typical grab brush with radial falloff available in commertical packages [Autodesk 2016], thus confirming that regularized Kelvinlets enrich existing grab brushes with elastic response.

5 MULTI-SCALE EXTRAPOLATION

Sculpting brushes are commonly accompanied by falloff profiles that define the locality of the deformations. Generally, these functions are set independently of the brush operation and can even be adjusted by the user. In contrast, the spatial decay of regularized Kelvinlets is determined completely by the equations of linear elasticity. This fixed falloff may make our elastic deformations inadequate for fine sculpts in which a faster decay is demanded. To address this issue, we present next an extrapolation scheme that constructs physically based brushes with arbitrarily fast decays by linearly combining regularized Kelvinlets of different radial scales. Due to the superposition principle, these compound brushes form high-order regularized Kelvinlets that satisfy (2).

Asymptotic analysis: To better understand the far-field decay of a regularized Kelvinlet $\boldsymbol{u}_\varepsilon(\boldsymbol{r})$, we analyzed the asymptotic expansion of the matrix $\boldsymbol{\mathcal{K}}_\varepsilon(\boldsymbol{r})$ as $r \to \infty$, whose first terms are

$$C_1(\hat{\boldsymbol{r}})\left(1/r\right) + \varepsilon^2 C_3(\hat{\boldsymbol{r}})\left(1/r^3\right) + \varepsilon^4 C_5(\hat{\boldsymbol{r}})\left(1/r^5\right) + O\left(1/r^7\right),$$

where the coefficients $\boldsymbol{C}_i\,(\hat{\boldsymbol{r}})$ indicate 3×3 matrices in terms of the normalized relative position $\hat{\boldsymbol{r}} = \boldsymbol{r}/r$, and material parameters μ and ν. (Formulas for $\boldsymbol{C}_i(\hat{\boldsymbol{r}})$ are provided in the supplemental material.) This expansion reveals an $O(1/r)$ leading-order decay followed by odd powers of $1/r$ and even powers of ε. By properly combining n regularized Kelvinlet matrices $\boldsymbol{\mathcal{K}}_{\varepsilon_i}(\boldsymbol{r})$ with different ε_i values, we can cancel the leading $n - 1$ terms and hasten the displacement falloff. This process is analogous to classical Richardson extrapolation for eliminating leading-order error terms in numerical differentiation [Brezinski and Zaglia 2013].

Bi-scale extrapolation: The simplest case of extrapolation is the difference of two regularized Kelvinlets of radii $\varepsilon_1 < \varepsilon_2$:

$$\boldsymbol{u}_{\varepsilon_1,\varepsilon_2}(\boldsymbol{r}) = \left[\boldsymbol{\mathcal{K}}_{\varepsilon_1}(\boldsymbol{r}) - \boldsymbol{\mathcal{K}}_{\varepsilon_2}(\boldsymbol{r})\right]\boldsymbol{f}. \tag{8}$$

This solution corresponds to the physical response to two body loads applied at the same location, where the force pushing the smaller ε_1-region is compensated by an equal but opposite pull over a spatially larger ε_2-region. We can then consider the bi-scale extrapolation as an approximation of the first derivative of $\boldsymbol{u}_\varepsilon$ with respect to ε. From the asymptotic expansion, we observe that the first term cancels and the net result has a leading-order decay of $O(1/r^3)$, much faster than a single regularized Kelvinlet. We can also reproduce the tip displacement $\bar{\boldsymbol{u}}$ by parameterizing the force vector $\boldsymbol{f}$ via the linear constraint $\boldsymbol{u}_{\varepsilon_1,\varepsilon_2}(0) = \bar{\boldsymbol{u}}$, yielding

$$\boldsymbol{u}_{\varepsilon_1,\varepsilon_2}(\boldsymbol{r}) = c\,\left({}^{1}\!/\!{}_{\varepsilon_1} - {}^{1}\!/\!{}_{\varepsilon_2}\right)^{-1}\left[\boldsymbol{\mathcal{K}}_{\varepsilon_1}(\boldsymbol{r}) - \boldsymbol{\mathcal{K}}_{\varepsilon_2}(\boldsymbol{r})\right]\bar{\boldsymbol{u}}. \tag{9}$$

Tri-scale extrapolation: Even greater locality of order $O({}^{1}\!/\!{}_{r^5})$ can be generated by employing a tri-scale extrapolation:

$$\boldsymbol{u}_{\varepsilon_1,\varepsilon_2,\varepsilon_3}(\boldsymbol{r}) = \left[w_1\boldsymbol{\mathcal{K}}_{\varepsilon_1}(\boldsymbol{r}) + w_2\boldsymbol{\mathcal{K}}_{\varepsilon_2}(\boldsymbol{r}) + w_3\boldsymbol{\mathcal{K}}_{\varepsilon_3}(\boldsymbol{r})\right]\boldsymbol{f}, \tag{10}$$

with distinct radial scales $\varepsilon_1 < \varepsilon_2 < \varepsilon_3$ and weights w_1, w_2, w_3. By analyzing its asymptotic expansion, we obtain

$$\left(\sum\nolimits_i w_i\right) C_1(\hat{\boldsymbol{r}})\left({}^{1}\!/\!{}_{r}\right) + \left(\sum\nolimits_i w_i\,\varepsilon_i^2\right) C_3(\hat{\boldsymbol{r}})\left({}^{1}\!/\!{}_{r^3}\right) + O\left({}^{1}\!/\!{}_{r^5}\right).$$

We can then cancel the two leading terms by setting the weights to

$$w_1 = 1,\;\; w_2 = -\,{}^{(\varepsilon_3^2-\varepsilon_1^2)}\!/\!{}_{(\varepsilon_3^2-\varepsilon_2^2)},\;\; w_3 = {}^{(\varepsilon_2^2-\varepsilon_1^2)}\!/\!{}_{(\varepsilon_3^2-\varepsilon_2^2)}.$$

This solution provides an approximation of the second derivative of $\boldsymbol{u}_\varepsilon$ with respect to ε. As before, we can parameterize the tri-scale deformation in terms of tip displacement $\bar{\boldsymbol{u}}$, which takes the form

$$\boldsymbol{u}_{\varepsilon_1,\varepsilon_2,\varepsilon_3}(\boldsymbol{r}) = c\,\left(\sum\nolimits_i {}^{w_i}\!/\!{}_{\varepsilon_i}\right)^{-1}\left[\sum\nolimits_i w_i\,\boldsymbol{\mathcal{K}}_{\varepsilon_i}(\boldsymbol{r})\right]\bar{\boldsymbol{u}}. \tag{11}$$

Although a user may adjust each radial scale individually, setting ε_2 and ε_3 to small increments of ε_1 leads to a better approximation of the derivatives of $\boldsymbol{u}_\varepsilon$ with respect to ε, thus creating a faster falloff in

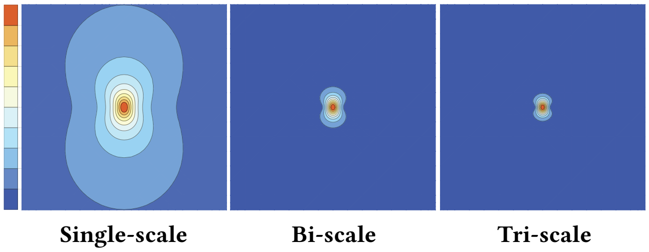


Fig. 3. **Multi-scale extrapolation for fast decay:** These plots display the contours of the norm of displacement fields generated by different multi-scale regularized Kelvinlets in 3D with $\nu = 1/2$. These brushes were parameterized by a unit displacement at the origin, thus leading to norm values in the inverval [0, 1]. Observe the long-range falloff of the single-scale brush, in contrast to the faster decays obtained by bi-scale and tri-scale brushes.

the proximity of the brush center and a high-order long-range decay. Therefore, we use $\varepsilon_{i+1} = 1.1\varepsilon_i$ as the default in our implementation, where ε_1 corresponds to the brush radius. Figure 3 illustrates the different far-field decays generated by single, bi-scale, and tri-scale regularized Kelvinlets, all computed with the same tip displacement $\overline{\boldsymbol{u}}$. Figure 4 shows a closeup comparing the displacement field generated by a single-scale brush before and after reducing its radius ε by 50%, versus the result of increasing the falloff order to a tri-scale brush. Observe the significantly different decays of these displacement fields, despite of their similarity near the brush center. Finally, we point out that the multi-scale extrapolation can be continued indefinitely to produce brushes of arbitrarily high order decay and increasing locality.

6 EXTENSION TO AFFINE LOADS

So far we showed how to use regularized Kelvinlets to construct grab-like deformations guided by the brush tip displacement. However, other sculpting operations assign an affine transformation, instead of a translation, to the brush. A typical example is the twist brush which defines a rotation centered at $\boldsymbol{x}_0$. Similarly, scale and pinch brushes are generated based on affine transformations. In order to augment these brushes with elastic response, we propose to extend the formulation of regularized Kelvinlets by replacing the vector-based load distribution with a matrix-based distribution.

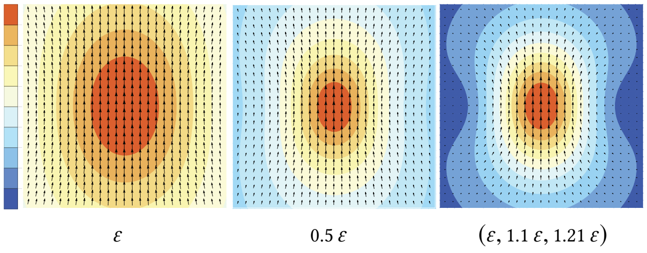


Fig. 4. **Smaller radius vs. multi-scale extrapolation:** A closeup of Figure 3 for a single-scale brush is shown overlaid by its displacement vectors (left). Reducing the brush radius ε makes the displacements more concentrated, but still with a long-range falloff (middle). In contrast, a tri-scale brush leads to localized displacements with a faster decay (right).

Our approach leverages the linearity of the elastostatics equation with respect to differentiation. More specifically, given a regularized Kelvinlet $\boldsymbol{u}_\varepsilon$ and its associated force vector $\boldsymbol{f}$, we compute the directional derivative $\boldsymbol{g}\cdot\nabla$ of (2) along the vector $\boldsymbol{g}$, yielding

$$\begin{aligned} 0 &= \boldsymbol{g}\cdot\nabla\left[\mu\Delta\boldsymbol{u}_\varepsilon + \frac{\mu}{(1-2\nu)}\nabla(\nabla\cdot\boldsymbol{u}_\varepsilon) + \boldsymbol{b}\right] \\ &= \mu\Delta(\boldsymbol{g}\cdot\nabla\boldsymbol{u}_\varepsilon) + \frac{\mu}{(1-2\nu)}\nabla[\nabla\cdot(\boldsymbol{g}\cdot\nabla\boldsymbol{u}_\varepsilon)] + \boldsymbol{g}\cdot\nabla\boldsymbol{b} \\ &= \mu\Delta\,\widetilde{\boldsymbol{u}}_\varepsilon + \frac{\mu}{(1-2\nu)}\nabla[\nabla\cdot\widetilde{\boldsymbol{u}}_\varepsilon] + \widetilde{\boldsymbol{b}}. \end{aligned}$$

This implies that we can define a new body load $\widetilde{\boldsymbol{b}}(\boldsymbol{r}) = \boldsymbol{g}\cdot\nabla\boldsymbol{b}(\boldsymbol{r})$, and its associated elastic response corresponds to the displacement field $\widetilde{\boldsymbol{u}}_\varepsilon(\boldsymbol{r}) = \boldsymbol{g}\cdot\nabla\boldsymbol{u}_\varepsilon(\boldsymbol{r})$.

We now consider the nine possible deformations $\widetilde{\boldsymbol{u}}_\varepsilon^{ij}$ generated by setting $\boldsymbol{f} = \boldsymbol{e}_i$ and $\boldsymbol{g} = \boldsymbol{e}_j$ for every pair (i, j), where the vectors $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3\}$ form an orthonormal bases spanning $\mathbb{R}^3$. Due to superposition, we can linearly combine $\widetilde{\boldsymbol{u}}_\varepsilon^{ij}$ with scalar coefficients F_{ij}, and obtain a matrix-driven solution of (2) of the form

$$\widetilde{\boldsymbol{u}}_\varepsilon(\boldsymbol{r}) = \sum\nolimits_{ij} F_{ij}\,\boldsymbol{e}_j\cdot\nabla(\mathcal{K}_\varepsilon(\boldsymbol{r})\,\boldsymbol{e}_i) = \nabla\mathcal{K}_\varepsilon(\boldsymbol{r}) : \boldsymbol{F}, \tag{12}$$

where $\boldsymbol{F} = \left[F_{ij}\right]$ is a 3×3 force matrix, and the symbol : indicates the double contraction of $\boldsymbol{F}$ to the third-order tensor $\nabla\mathcal{K}_\varepsilon(\boldsymbol{r})$, thus returning a vector. Similarly, we can write the body load that generates the deformation $\widetilde{\boldsymbol{u}}_\varepsilon$ as

$$\widetilde{\boldsymbol{b}}(\boldsymbol{r}) = \sum\nolimits_{ij} F_{ij}\,\boldsymbol{e}_i\,\boldsymbol{e}_j^t\,\nabla\rho_\varepsilon(\boldsymbol{r}) = \boldsymbol{F}\,\nabla\rho_\varepsilon(\boldsymbol{r}). \tag{13}$$

By computing the spatial derivatives of $\boldsymbol{u}_\varepsilon$, we obtain the displacement field $\widetilde{\boldsymbol{u}}_\varepsilon(\boldsymbol{r})$ in terms of the force matrix $\boldsymbol{F}$:

$$\begin{aligned} \widetilde{\boldsymbol{u}}_\varepsilon(\boldsymbol{r}) = &-a\left(\frac{1}{r_\varepsilon^3} + \frac{3\varepsilon^2}{2r_\varepsilon^5}\right)\boldsymbol{F}\boldsymbol{r} \\ &+ b\left[\frac{1}{r_\varepsilon^3}\left(\boldsymbol{F}+\boldsymbol{F}^t+\mathrm{tr}(\boldsymbol{F})\boldsymbol{I}\right) - \frac{3}{r_\varepsilon^5}\left(\boldsymbol{r}^t\boldsymbol{F}\boldsymbol{r}\right)\boldsymbol{I}\right]\boldsymbol{r}. \end{aligned} \tag{14}$$

Note that the first term in (14) corresponds to an affine transformation $\boldsymbol{F}\boldsymbol{r}$ with a radial falloff similar to existing affine brushes, while the second term includes a symmetric affine transformation and controls volume compression through the Poisson ratio ν that b depends on. Therefore, we name this matrix-based extension of the fundamental solution of linear elasticity as a *locally affine regularized Kelvinlet*. For conciseness, we denote this displacement field by $\widetilde{\boldsymbol{u}}_\varepsilon(\boldsymbol{r}) \equiv \mathcal{A}_\varepsilon(\boldsymbol{r})\,\vec{\boldsymbol{F}}$, where $\vec{\boldsymbol{F}} \in \mathbb{R}^9$ is a vectorized form of $\boldsymbol{F}$ and $\mathcal{A}(\boldsymbol{r})$ is the 3×9 matrix that maps $\vec{\boldsymbol{F}}$ to a displacement at $\boldsymbol{r}$.

In contrast to (6), the deformation generated by (14) has zero displacement at the brush center, i.e., $\widetilde{\boldsymbol{u}}_\varepsilon(0) = 0$, but a non-trivial gradient in terms of $\boldsymbol{F}$ (the closed-form expression is provided in the supplemental material). As the radial scale ε approaches zero, our matrix-based solutions reproduce the definition of Kelvinlet doublets [Phan-Thien and Kim 1994]. These expressions also coincide with the regularized Stokeslet doublets [Ainley et al. 2008] in the incompressible limit ($\nu = 1/2$). Following the matrix decomposition in Section 3, we can further construct specialized versions of the locally affine regularized Kelvinlets by setting $\boldsymbol{F}$ to different types of matrices, as illustrated in Figure 5.

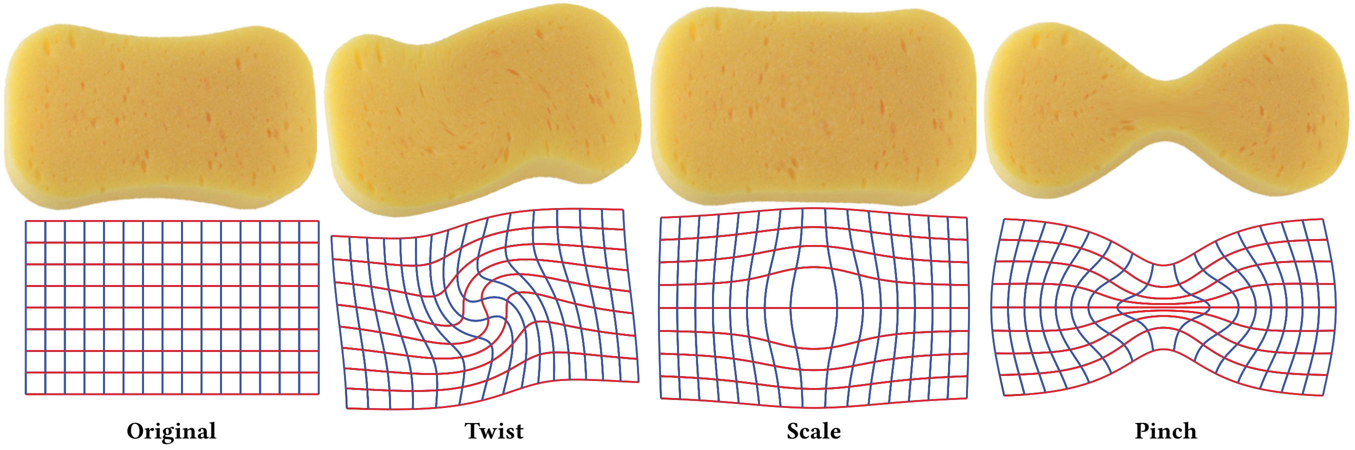


Fig. 5. **Elastic Locally Affine Brushes:** Given an input image (left), we compute a twisting (left-center), a scaling (center-right), and a pinching (right) deformation via a locally affine regularized Kelvinlet in 2D with Poisson ratio $\nu = 0.4$. Input image courtesy of Eftychios Sifakis.

Twisting: In the case of a skew-symmetric matrix, we can associate $\boldsymbol{F}$ to a vector $\boldsymbol{q}$ via the cross product matrix, i.e., $\boldsymbol{F} \equiv [\boldsymbol{q}]_\times$ where $[\boldsymbol{q}]_\times \boldsymbol{r} = \boldsymbol{q} \times \boldsymbol{r}$, then (14) simplifies to a twist deformation

$$\boxed{\boldsymbol{t}_\varepsilon(\boldsymbol{r}) = -\,a\left(\frac{1}{r_\varepsilon^3}+\frac{3\varepsilon^2}{2r_\varepsilon^5}\right)\boldsymbol{q}\times\boldsymbol{r}.} \qquad (15)$$

By analyzing the deformation gradient of (15) (see supplemental material), one can verify that its symmetric part (the strain tensor) is trivial for any $\boldsymbol{r}$. Consequently, this displacement field has zero divergence (i.e., $\nabla\cdot\boldsymbol{t}_\varepsilon(\boldsymbol{r})=0$) and defines a volume preserving deformation, independent of the Poisson ratio ν. Instead, its curl is non-zero and can be used to relate the vector $\boldsymbol{q}$ to the vorticity $\overline{\omega}$ at $\boldsymbol{x}_0$ via the linear constraint $\nabla\times\boldsymbol{t}_\varepsilon(0)=\overline{\omega}$.

Scaling: Another type of locally affine regularized Kelvinlet can be generated by a force matrix of the form $\boldsymbol{F} = s\,\boldsymbol{I}$, where s is a scalar. In this case, (14) reduces to a scaling deformation

$$\boxed{\boldsymbol{s}_\varepsilon(\boldsymbol{r}) = (2b-a)\left(\frac{1}{r_\varepsilon^3}+\frac{3\varepsilon^2}{2r_\varepsilon^5}\right)(s\,\boldsymbol{r}),} \qquad (16)$$

where positive values of s represent contractions, and negative values determine dilations. Notice that we have $\boldsymbol{s}_\varepsilon(\boldsymbol{r})=0$ for incompressible elastic materials, since $\nu = 1/2$ implies $a = 2b$.

Pinching: The last type of locally affine regularized Kelvinlet is constructed based on a symmetric force matrix $\boldsymbol{F}$ with zero trace (in a total of 5 DoFs), and it generates displacements of the form

$$\boxed{\boldsymbol{p}_\varepsilon(\boldsymbol{r}) = \frac{(2b-a)}{r_\varepsilon^3}\boldsymbol{F}\boldsymbol{r} - \frac{3}{2r_\varepsilon^5}\left[2\,b\,(\boldsymbol{r}^t\boldsymbol{F}\boldsymbol{r})\,\boldsymbol{I}+a\,\varepsilon^2\boldsymbol{F}\right]\boldsymbol{r}.} \qquad (17)$$

To geometrically characterize this deformation, we computed the displacement gradient $\nabla\boldsymbol{p}_\varepsilon$ at $\boldsymbol{x}_0$, and observed that the resulting matrix is also symmetric with zero trace. This indicates that the deformation generated by $\boldsymbol{p}_\varepsilon(\boldsymbol{r})$ compensates infinitesimal stretching in one direction by contractions in the other directions, thus resembling a physical pinching interaction.

Multi-scale extrapolation: Since our affine brushes are based on the derivative of (6), the extrapolation scheme described in Section 5 applies to these displacement fields with no modification. We can then combine locally affine regularized Kelvinlets linearly and generate new matrix-driven solutions of linear elasticity with arbitrarily fast far-field decay. In particular, we obtain bi-scale and tri-scale brushes with $O(1/r^4)$ and $O(1/r^6)$ falloffs, respectively, while the single-scale case has an $O(1/r^2)$ decay.

7 CONSTRAINED DEFORMATIONS

By exploiting the linearity of (2), regularized Kelvinlets and variants can be superposed to form new solutions of linear elasticity. These compound brushes are of particular interest for designing deformations with pointwise contraints. Since regularized Kelvinlets are linear in terms of force vectors $\boldsymbol{f}$ and matrices $\boldsymbol{F}$, these constrained deformations can be efficiently computed based on user-prescribed displacements and gradients. In this section, we describe some examples of constrained deformations.

Displacement Constraints: Previously we assigned the brush tip displacement $\overline{\boldsymbol{u}}$ and solved analytically for the necessary force vector $\boldsymbol{f}$ in (7). We can replicate this constraint numerically for a list of brushes. To this end, we consider n regularized Kelvinlets placed at $\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n\}$ and impose n displacement constraints $\{\overline{\boldsymbol{u}}_1,\ldots,\overline{\boldsymbol{u}}_n\}$, one for each brush center $\boldsymbol{x}_i$. By superposition, these brushes define a solution to (2) of the form

$$\boldsymbol{u}(\boldsymbol{x}) = \sum\nolimits_{i=1}^{n}\mathcal{K}_{\varepsilon_i}(\boldsymbol{x}-\boldsymbol{x}_i)\,\boldsymbol{f}_i, \qquad (18)$$

where ε_i and $\boldsymbol{f}_i$ indicate the radial scale and the force vector for the i-th brush, respectively. The displacement constraints $\overline{\boldsymbol{u}}$ and force vectors $\boldsymbol{f}$ are thus related by the linear system $\overline{\boldsymbol{u}} = \boldsymbol{K}\boldsymbol{f}$, where the $3n\times3n$ matrix $\boldsymbol{K}$ has the contribution $\mathcal{K}_{\varepsilon_i}(\boldsymbol{x}_j-\boldsymbol{x}_i)$ of every i-th brush to every j-th brush center.

An usual scenario for this constrained deformation is when we pin down zero displacement to $n-1$ locations, and keep the n-th brush "live" for interactive sculpting. Given a tip displacement $\overline{\boldsymbol{u}}_n$

updated every user edit, we can efficiently compute the force vectors $\boldsymbol{f}$ by exploiting the block structure of the linear system

$$\begin{bmatrix} \mathbf{0} \\ \bar{\boldsymbol{u}}_n \end{bmatrix} = \begin{bmatrix} K_{pp} & K_{pn} \\ K_{np} & K_{nn} \end{bmatrix} \begin{bmatrix} f_p \\ f_n \end{bmatrix}, \tag{19}$$

where the subscript p indicates values for the $n-1$ pinned brushes, and n denotes the active brush. Observe that the block $\boldsymbol{K}_{pp}$ is a nonsymmetric matrix that relates the $n-1$ pinned brushes to themselves, but it is independent of the n-th brush. Therefore, we can cache the LU factorization of $\boldsymbol{K}_{pp}$ and solve for $\boldsymbol{f}$ rapidly at every user interaction via the Schur complement [Cottle 1974], which then requires a simple 3×3 solve. Figure 6 (center-right) displays a result with displacement constraints used in a 3D editing session.

Gradient Constraints: We can also constrain the gradients of our deformations by employing locally affine regularized Kelvinlets in our compound displacement $\boldsymbol{u}(\boldsymbol{x})$. By doing so, the force matrix $\boldsymbol{F}$ can be used to precisely control the 9 DoFs of the deformation gradient via the linear constraint $\boldsymbol{G}(\boldsymbol{x}_i) = \boldsymbol{I} + \nabla \boldsymbol{u}(\boldsymbol{x}_i)$ at a brush center $\boldsymbol{x}_i$. This leads to the aggregated displacement field

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_{i=1}^{n} \mathcal{A}_{\varepsilon_i}(\boldsymbol{x} - \boldsymbol{x}_i)\, \vec{\boldsymbol{F}}_i. \tag{20}$$

More specialized gradient control can be further achieved via individual twist, scale, and pinch brushes (Section 6). For instance, a pointwise vorticity constraint $\nabla \times \boldsymbol{u}$ can be enforced based on the 3 DoFs of the vector $\boldsymbol{q}$ in a twist brush, while the single parameter s of a scale brush can determine a divergence constraint $\nabla \cdot \boldsymbol{u}$. We can also employ a pinch brush to control the infinitesimal stretches in two directions of a rotated frame, for a total of 5 DoFs. Note that the third infinitesimal strecth is induced by the first two values, since a pinch brush has a traceless gradient. Similar to the displacement case, these gradient constraints relate to their respective force parameters ($\boldsymbol{F}$, $\boldsymbol{q}$, or s) through a single linear system. As before, a Schur complement approach can be employed to speedup solve times when sculpting with fixed constraints. In Figure 6 (right), pointwise gradient constraints were used to ensure rigid deformations in selected regions of the eyeballs.

Symmetrized Deformations: A common feature in sculpting tools is the ability to symmetrize deformations by mirroring a brush operation across a user-specified plane. This is performed by replicating the brush at $\boldsymbol{x}_0$ to its image point $\boldsymbol{x}_1 = \boldsymbol{c} + \boldsymbol{M}(\boldsymbol{x}_0 - \boldsymbol{c})$, where $\boldsymbol{M} = \boldsymbol{I} - 2\boldsymbol{n}\boldsymbol{n}^t$ indicates the reflection matrix of a plane passing through a base point $\boldsymbol{c}$ and with a normal vector $\boldsymbol{n}$. The first case of symmetrized physically based deformation consists of copying a regularized Kelvinlet $\boldsymbol{u}_\varepsilon(\boldsymbol{r})$ of force vector $\boldsymbol{f}_0$ from $\boldsymbol{x}_0$ to $\boldsymbol{x}_1$, while keeping its radius ε but reflecting its force vector to $\boldsymbol{f}_1 = \boldsymbol{M}\boldsymbol{f}_0$. The resulting deformation is then written as

$$\boldsymbol{u}(\boldsymbol{x}) = \mathcal{K}_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_0)\boldsymbol{f}_0 + \mathcal{K}_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_1)\boldsymbol{f}_1. \tag{21}$$

We show in the supplemental material that (21) satisfies the identity $\boldsymbol{u}(\boldsymbol{M}\boldsymbol{x}) = \boldsymbol{M}\,\boldsymbol{u}(\boldsymbol{x})$ for any force vector $\boldsymbol{f}_0$ and at any evaluation point $\boldsymbol{x}$, i.e., the displacement generated by (21) at the image of $\boldsymbol{x}$ is the reflection of the displacement at $\boldsymbol{x}$. This also implies that the force vecor $\boldsymbol{f}_0$ can be computed from a given tip displacement $\bar{\boldsymbol{u}}$ via a single constraint at $\boldsymbol{x}_0$.

Similarly, the symmetrization of locally affine brushes can be computed by mirroring the force matrix $\boldsymbol{F}_0$ to $\boldsymbol{F}_1 = \boldsymbol{M}\boldsymbol{F}_0\boldsymbol{M}$. The symmetrized displacement field is then

$$\boldsymbol{u}(\boldsymbol{x}) = \mathcal{A}_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_0)\vec{\boldsymbol{F}}_0 + \mathcal{A}_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_1)\vec{\boldsymbol{F}}_1. \tag{22}$$

As before, the force matrix $\boldsymbol{F}_0$ can be obtained in terms of a gradient constraint, if necessary. At last, we point out that multi-scale extrapolation applies to these formulations with no modification. Figure 7 shows symmetrized edits done by grab and twist brushes.

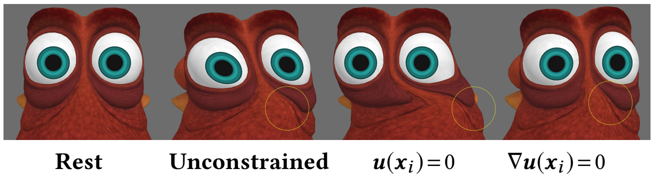

Rest
Unconstrained
$\boldsymbol{u}(\boldsymbol{x}_i) = 0$
$\nabla \boldsymbol{u}(\boldsymbol{x}_i) = 0$

Fig. 6. **Constraints:** The deformation of an unconstrained regularized Kelvinlet distorts the eyeballs (left-center). The roundness of the irises and pupils can be restored by adding pointwise constraints. This is achieved by either pinning down zero displacements at these selected points (center-right), or enforcing locally rigid edits via gradient constraints (right). This example used a single 3D brush with $\nu = 1/2$ and 160 constraints. ©Disney/Pixar

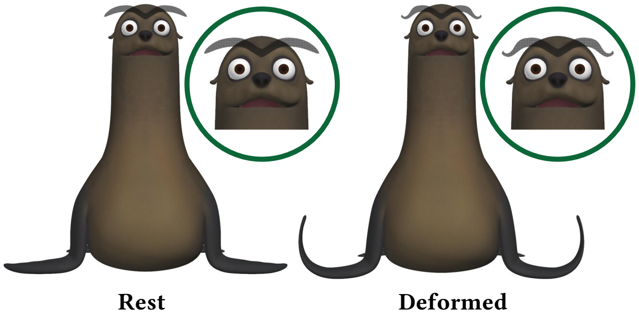


Fig. 7. **Symmetrized edits:** Regularized Kelvinlets also support symmetrized deformations. Here we show a symmetrized grab applied to the flippers, and a symmetrized twist sculpting the eyebrows of a sea lion. Both cases used a tri-scale brush in 3D with $\nu = 1/2$. ©Disney/Pixar

8 2D REGULARIZED KELVINLETS

We now discuss the construction of regularized Kelvinlets for an infinite elastic medium in $\mathbb{R}^2$. Besides image editing, these deformations are also suitable for volume sculpts restricted to the camera plane (Figure 8). In 2D, elastostatics is formulated using the same expressions for the energy (1) and for the equilibrium state (2) as in 3D, but now with differential operators (Δ, ∇, $\nabla\cdot$) including only (x, y)-derivatives [Slaughter 2002]. This restriction introduces a new log term to the singular fundamental solutions [Phan-Thien and Kim 1994], which makes our regularization scheme surpringsly more elaborated than the 3D case. As detailed next, we address the construction of 2D regularized Kelvinlets by considering a 2D body load as a 3D line force distribution passing through the brush center. This interpretation allows us to compute 2D solutions by integrating the associated 3D elastic responses through depth.

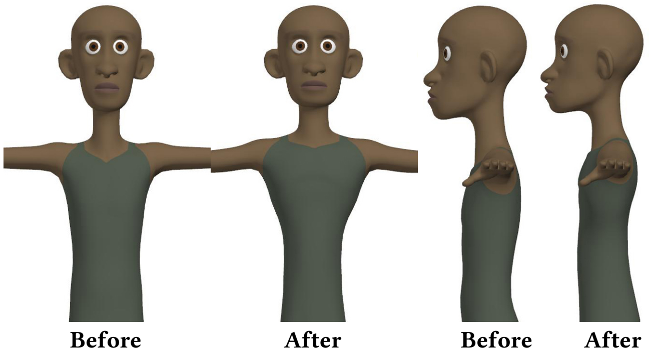


Fig. 8. **Depth-preserving edits:** A 2D regularized Kelvinlet with $\nu = 0.4$ was used to bulge the chest of a human character (left), while maintaing the body thickness (right). In this case, the deformation was applied tangent to the camera plane, thus keeping the depth information intact. ©Disney/Pixar

Derivation: Hereafter we denote depth by the z-axis and use $\mathfrak{r}$ to indicate the 2D relative position vector (i.e., $\boldsymbol{r} \equiv [\mathfrak{r}, z]$). We refer to the 2D norm as $\mathrm{r} = \|\mathfrak{r}\|$, and to the regularized 2D distance as r_ε (so that $r_\varepsilon^2 = \mathrm{r}_\varepsilon^2 + z^2$). We start by constructing a 2D force distribution that integrates the smoothed 3D density function $\rho_\varepsilon(\boldsymbol{r})$ in (5) along the z-axis, producing

$$\rho_\varepsilon(\mathfrak{r}) = \int_{-\infty}^{+\infty} \rho_\varepsilon(\boldsymbol{r})\, dz = \frac{2\varepsilon^2}{\pi} \frac{1}{\mathrm{r}_\varepsilon^6}. \tag{23}$$

With this density function, we can represent any 2D regularized Kelvinlet as the solution of (2) still in 3D, but now generated by an infinite line load $\boldsymbol{b}(\mathfrak{r}) = \boldsymbol{f} \rho_\varepsilon(\mathfrak{r})$, where $\boldsymbol{f}$ has its z-component set to zero. Due to symmetry, we can also show that the z-component of the 3D deformation generated by $\boldsymbol{b}(\mathfrak{r})$ is zero. Therefore, the equation of elastostatics in 3D with load $\boldsymbol{b}(\mathfrak{r})$ simplifies to its 2D counterpart with no z-derivatives. As a result, we can compute the 2D regularized Kelvinlet $\boldsymbol{u}_\varepsilon(\mathfrak{r})$ by carefully integrating the (x, y)-components of the 3D regularized Kelvinlet $\boldsymbol{u}_\varepsilon(\boldsymbol{r})$,

$$\begin{bmatrix} \boldsymbol{u}_\varepsilon^x(\mathfrak{r}) \\ \boldsymbol{u}_\varepsilon^y(\mathfrak{r}) \end{bmatrix} = \int_{-\infty}^{\infty} \begin{bmatrix} \boldsymbol{u}_\varepsilon^x(\boldsymbol{r}) \\ \boldsymbol{u}_\varepsilon^y(\boldsymbol{r}) \end{bmatrix} dz, \tag{24}$$

which leads to the closed-form expression (see Appendix B)

$$\boxed{\boldsymbol{u}_\varepsilon(\mathfrak{r}) = \left[2\,(b-a)\ln(\mathrm{r}_\varepsilon)\boldsymbol{I} + 2b\frac{\mathfrak{r}\mathfrak{r}^t}{\mathrm{r}_\varepsilon^2} + a\frac{\varepsilon^2}{\mathrm{r}_\varepsilon^2}\boldsymbol{I} \right] \boldsymbol{f}.} \tag{25}$$

We notice that (25) converges to the 2D singular Kelvinlets as ε approaches zero [Phan-Thien and Kim 1994] and, in the incompressible limit ($\nu = 1/2$), we obtain a 2D regularized Stokeslet. In contrast to [Cortez 2001], we employed a sharper 2D density function (given in (23)) that relates the regularization from 3D to 2D.

Multi-scale extrapolation: Compared to the 3D solutions, the regularized Kelvinlets in 2D introduce unbounded displacements as $\mathrm{r} \longrightarrow \infty$, due to the presence of the log term, thus making them inappropriate for localized edits. Fortunately, the multi-scale extrapolation presented in Section 5 can be used to produce spatial falloffs. In fact, since 2D regularized Kelvinlets are integrated versions of the 3D displacements, the same multi-scale extrapolation rules previously stated for the 3D case work in 2D, and the observed far-field decay for bi-scale and tri-scale 2D brushes are $O(1/\mathrm{r}^2)$ and $O(1/\mathrm{r}^4)$, respectively.

Extension to affine loads: Following the same derivation from Section 6, we can also construct 2D versions of the locally affine regularized Kelvinlets, resulting in:

$$\begin{aligned} \widetilde{\boldsymbol{u}}_\varepsilon(\mathfrak{r}) = & -2\,a\left(\frac{1}{\mathrm{r}_\varepsilon^2} + \frac{\varepsilon^2}{\mathrm{r}_\varepsilon^4}\right)\boldsymbol{F}\mathfrak{r} \\ & + 2\,b\left[\frac{1}{\mathrm{r}_\varepsilon^2}\left(\boldsymbol{F} + \boldsymbol{F}^t + \mathrm{tr}(\boldsymbol{F})\boldsymbol{I}\right) - \frac{2}{\mathrm{r}_\varepsilon^4}\left(\mathfrak{r}^t\boldsymbol{F}\mathfrak{r}\right)\boldsymbol{I}\right]\mathfrak{r}, \end{aligned} \tag{26}$$

where $\boldsymbol{F}$ is now a 2×2 force matrix. Expressions for twisting (1-DoF), scaling (1-DoF) and pinching (2-DoF) can also be derived. We provide the list of closed-form solutions for 2D locally affine regularized Kelvinlets in the supplemental material.

9 RESULTS

In this section, we detail our implementation and provide a series of experiments showcasing the performance and the visual quality of regularized Kelvinlets compared to existing sculpting tools. The accompanying video also includes several examples of editing sessions using our physically based brushes.

Implementation: We implemented regularized Kelvinlets and variants as a C++ plugin to the sculpting package of Pixar animation system (Presto). Once a brush type is selected, the brush center is assigned to the mouse click location, and the force parameters are set based on the mouse displacement. The user also has interactive control of the brush radius ε, the multi-scale mode (single, bi, or tri-scale), and the Poisson ratio ν. During a mouse event, we deform every selected point in the scene by moving them along the streamlines defined by the brush's displacement field. This point-based advection can be computed using any existing line integration method [Nielson et al. 1997]. In our experiments, we employed a fourth-order Runge-Kutta method similar to [von Funck et al. 2006], but other integrators or substepping schemes can be easily incorporated. Notice that our deformation is evaluated at every selected point, including those outside the brush radius, because they may

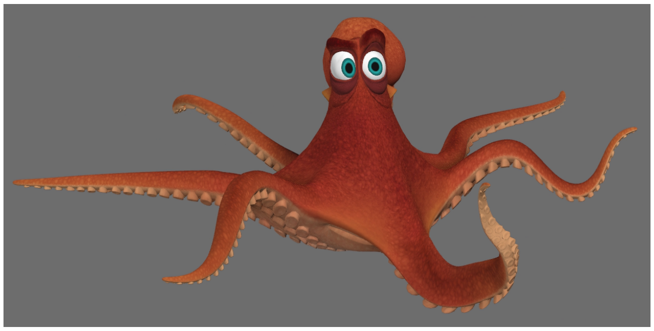


Fig. 9. **Posing via sculpts:** The final pose of a character sculpted by an artist using regularized Kelvinlet brushes exclusively. ©Disney/Pixar

still have a non-zero displacement depending on the decay chosen for the brush. Therefore, the computational cost of our sculpts is linear on the number of selected points, in contrast to existing brushes that restrict the edits within the brush radius. In our implementation, we use multi-threading via TBB [Reinders 2007], and observed performances averaging 57 frames per second on a 2.3 GHz Intel Xeon E5-2699 with 18 cores for scenes with 100k points. In the supplemental video, we also include a test with one million points, timing 18 ms per mouse drag. We expect even faster and more scalable running times by implementing regularized Kelvinlets in the GPU or as a geometry shader, which is feasible due to our closed-form analytical expressions.

Examples: Figures 2 and 5 illustrate the basic elastic deformations generated by displacement and affine loads, respectively. Figure 1 shows a gallery of facial expressions sculpted by an artist using combinations of our elastic brushes. Additionally, we include in Figure 9 a full-body deformed pose also sculpted by an artist via regularized Kelvinlets. We note that this model has multiple disconnected meshes, and our brush edits were applied to them simultaneously. In Figure 8, we used a 2D brush to bulge the chest of a character while preserving the body thicknesss. Figure 12 displays a sequence of volume edits with varying Poisson ratio. We demonstrate symmetrized grab and twist deformations in Figure 7. In Figure 6, we compare constrained deformations using displacement and gradient constraints. In this example, we employed the Schur complement approach described in Section 7 to precompute the constraint matrix only once during the mouse click, thus keeping the brush evaluation interactive. We point the reader to the supplemental video for additional examples and editing sessions.

Moving brushes: We can also modify the brush construction in order to support large deformations. So far our brush setup keeps its center stationary at the click location, and accumulates the force vector as the mouse moves. The resulting displacement may then introduce artifacts under large deformations, as expected from solutions of linear elasticity. Instead, by advecting the brush center throughout the course of the mouse edit, we can reset the force vector to the latest mouse shift and then reproduce within a single mouse click the deformation of a sequence of small clicks. Figure 10 shows an example that uses a moving brush to reduce distortion under large deformations. Note, however, that the deformation generated by a single click of a moving brush is path-dependent and may not restore the original shape as the mouse moves back to the click position, due to the accumulated displacements.

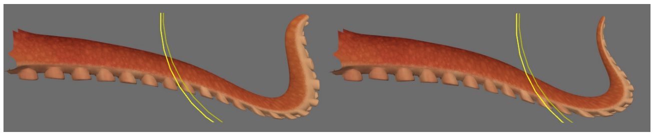


Fig. 10. **Moving brush:** Keeping the brush center stationary at the mouse click location leads to visual artifacts under large deformations (left). This distortion is alleviated by displacing the brush center during the mouse edit (right). ©Disney/Pixar

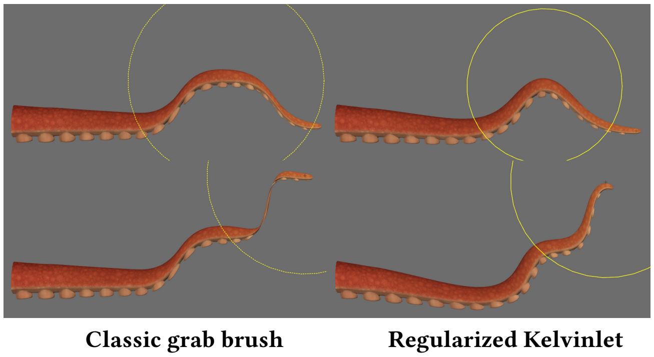


Fig. 11. **Comparison:** The left column shows results of two consecutive deformations using a grab brush with a Gaussian falloff. The right column shows the same two consecutive grabs now deformed by regularized Kelvinlets in 3D with Poisson ratio $\nu = 1/2$. Notice how our physically based sculpting tool improves the local volume preservation. ©Disney/Pixar

Discussion: Figure 11 exhibits the deformation generated by a few clicks on a tentacle mesh using a (Maya-like) grab brush with a Gaussian falloff versus our 3D regularized Kelvinlet. This example illustrates the ability of our tool to capture broad volume deformations even with no knowledge of the surface structure, in contrast to traditional brushes that tend to collapse the volume. This result also suggests that our deformations could be used as an initial guess for iterative mesh-based deformers [Sorkine and Alexa 2007] and simulators [McAdams et al. 2011], which can better capture mesh-dependent local features. While the physical values for the Poisson ratio lie within the range $[0, 1/2]$ [Slaughter 2002], our method supports a broader spectrum of values. In particular, we show in Figure 2 that we can reproduce existing brushes with radial falloffs by setting $\nu = \pm\infty$. We also notice that our twist deformation coincides to a swirling-sweeper brush with radial falloff [Angelidis et al. 2006].

Limitations: The simplicity and speed of our approach come at the cost of several limitations. Foremost, since regularized Kelvinlets are solutions to the elasticity equations in an infinite continuum, they know nothing about the geometry they are used to deform. Consequently, they can not be used for accurate elastic analysis of solid shapes. Furthermore, surfaces deformed using regularized Kelvinlets move as if embedded in an infinite elastic continuum, thus disregarding any form of surface traction. Embedded deformations can also produce unintended and undesirable nonlocal interactions. For example, grabbing one tentacle can cause a second nearby tentacle to deform, even though these parts are far away geodesically. These issues motivate the use in the future of post-processing steps, such as surface-based masking, combined to our sculpting tools. We also point out that, while regularized Kelvinlets can produce smooth elastoplastic deformations, distorting discrete shapes may require dynamic remeshing in order to sufficiently resolve the flow with no visual artifacts (see, e.g., [von Funck et al. 2006]). We finally note that, even though we show results for posing deformable characters (Figure 9), our approach is primarily useful for relatively modest shape edits, and is not intended to replace articulated character rigs.

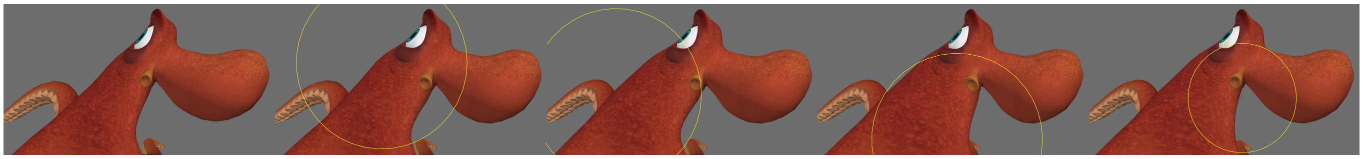


Fig. 12. **Volume adjustment:** A series of one-click sculpts used for local volume adjustments. These deformations were computed using a regularized Kelvinlet in 2D with various values for the Poisson ratio. ©Disney/Pixar

10 CONCLUSION

We have presented a family of deformation tools based on regularized fundamental solutions of linear elasticity for infinite media in 2D and 3D. Our method provides real-time feedback since it exploits closed-form analytical expressions and avoids the spatial discretization of elastic objects. This trade-off allows us to infuse "just enough" physical behavior into sculpting brushes to improve traditional grab, twist, pinch, and scale editing. We also addressed the problem of long-range interactions by introducing a multi-scale extrapolation scheme, and resolved various pointwise constraints using standard linear solves.

We believe this work opens up new possibilities for analytical methods to be used for real-time sculpting [Cani and Angelidis 2006] and contact interactions [Pauly et al. 2004]. We have introduced simple force primitives for brushes, but extensions with more general load distributions could be constructed for specialized purposes [von Funck et al. 2006]. Regularized fundamental solutions for a half-space elastic continuum, such as the Boussinesq's problem [Podio-Guidugli and Favata 2014], and the method of images [Ainley et al. 2008] are also of interest. Regularized Kelvinlets are also well-suited to kinesthetic haptic force and torque feedback, and to high-rate modeling applications in virtual and augmented reality.

ACKNOWLEDGMENTS

We are grateful to Alonso Martinez and Kevin Singleton for generating the artistic examples in this work, and to Andrew Butts and Venkateswaran Krishna for their assistance with the Presto sculpting package. We also thank Michael Comet, Jeremie Talbot, Evan Bonifacio, and Dirk Van Gelder for discussions, and Tony DeRose, Theodore Kim, and Mathieu Desbrun for feedback on the manuscript. The image examples are courtesy of Mirela Ben-Chen (Figure 2) and Eftychios Sifakis (Figure 5).

REFERENCES

J. Ainley, S. Durkin, R. Embid, P. Boindala, and R. Cortez. 2008. The method of images for regularized Stokeslets. *J. of Computational Physics* 227, 9 (2008), 4600–4616.

A. Angelidis, M. P. Cani, G. Wyvill, and S. King. 2006. Swirling-Sweepers: Constant volume modeling. *Graph. Models* 68, 4 (2006), 324âĂŞ332.

A. Angelidis, G. Wyvill, and M. P. Cani. 2004. Sweepers: Swept user-defined tools for modeling by deformation. In *Proc. of Shape Modeling Applications*. IEEE, 63–73.

Autodesk. 2016. Maya User Guide. (2016). https://autodesk.com/maya.

J. R. Barber. 2010. *Elasticity*. Solid Mechanics and Its Applications, Vol. 172. Springer.

J. Barbič and D. L. James. 2005. Real-Time Subspace Integration for St. Venant-Kirchhoff Deformable Models. *ACM Trans. Graph.* 24, 3 (2005), 982–990.

J. Barbič and Y. Zhao. 2011. Real-time Large-deformation Substructuring. *ACM Trans. Graph.* 30, 4, Article 91 (2011).

J. C. Bazin, C. Plüss (Kuster), G. Yu, T. Martin, A. Jacobson, and M. Gross. 2016. Physically Based Video Editing. *Computer Graphics Forum* (2016).

M. Ben-Chen, O. Weber, and C. Gotsman. 2009. Variational Harmonic Maps for Space Deformation. *ACM Trans. Graph.* 28, 3 (2009).

C. A. Brebbia, J. C. F. Telles, and L. Wrobel. 2012. *Boundary element techniques: theory and applications in engineering*. Springer Science & Business Media.

C. Brezinski and M. R. Zaglia. 2013. *Extrapolation methods: Theory and practice*. Studies in Computational Mathematics (Book 2), Vol. 2. Elsevier.

M. P. Cani and A. Angelidis. 2006. Towards Virtual Clay. In *ACM SIGGRAPH Courses*. 67–83.

A. T. Chwang and T. Y.-T. Wu. 1975. Hydromechanics of low-Reynolds-number flow. Part 2. Singularity method for Stokes flows. *J. of Fluid Mechanics* 67, 04 (1975), 787–815.

S. Coquillart. 1990. Extended Free-form Deformation: A Sculpturing Tool for 3D Geometric Modeling. In *ACM SIGGRAPH*. 187–196.

F. Cordier, Y. Gingold, E. Entem, M. P. Cani, and K. Singh. 2016. Sketch-based Modeling. In *Eurographics Tutorials*.

R. Cortez. 2001. The Method of Regularized Stokeslets. *SIAM J. on Scientific Computing* 23, 4 (2001), 1204–1225.

R. Cortez, L. Fauci, and A. Medovikov. 2005. The method of regularized Stokeslets in three dimensions: Analysis, validation, and application to helical swimming. *Physics of Fluids* 17 (2005).

R. W. Cottle. 1974. Manifestations of the Schur complement. *Linear Algebra and its Applications* 8, 3 (1974), 189 – 211.

G. Dewaele and M. P. Cani. 2004. Interactive Global and Local Deformations for Virtual Clay. *Graph. Models* 66, 6 (2004).

E. Ferley, M. P. Cani, and J.-D. Gascuel. 2001. Resolution Adaptive Volume Sculpting. *Graph. Models* 63, 6 (2001).

S. F. Frisken, R. N. Perry, A. P. Rockwood, and T. R. Jones. 2000. Adaptively sampled distance fields: a general representation of shape for computer graphics. In *ACM SIGGRAPH*. 249–254.

T. A. Galyean and J. F. Hughes. 1991. Sculpting: An Interactive Volumetric Modeling Technique. In *ACM SIGGRAPH*. 267–274.

F. Hahn, S. Martin, B. Thomaszewski, R. Sumner, S. Coros, and M. Gross. 2012. Rig-space Physics. *ACM Trans. Graph.* 31, 4 (2012).

G. J. Hancock. 1953. The Self-Propulsion of Microscopic Organisms through Liquids. *Proc. of the Royal Soc. of London A: Mathematical, Physical and Engineering Sciences* 217, 1128 (1953), 96–121.

A. Jacobson, I. Baran, L. Kavan, J. Popović, and O. Sorkine. 2012. Fast Automatic Skinning Transformations. *ACM Trans. Graph.* 31, 4 (2012).

A. Jacobson, I. Baran, J. Popović, and O. Sorkine. 2011. Bounded Biharmonic Weights for Real-time Deformation. *ACM Trans. Graph.* 30, 4, Article 78 (2011).

D. L. James and D. K. Pai. 1999. ArtDefo: Accurate Real Time Deformable Objects. In *ACM SIGGRAPH*. 65–72.

D. L. James and D. K. Pai. 2003. Multiresolution Green's Function Methods for Interactive Simulation of Large-scale Elastostatic Objects. *ACM Trans. Graph.* 22, 1 (2003), 47–82.

P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki. 2007. Harmonic Coordinates for Character Articulation. *ACM Trans. Graph.* 26, 3 (2007).

L. Kavan and O. Sorkine. 2012. Elasticity-Inspired Deformers for Character Articulation. *ACM Trans. on Graph.* 31, 6 (2012).

Lord Kelvin. 1848. Note on the integration of the equations of equilibrium of an elastic solid. *Cambridge and Dublin Mathematical Journal* 3 (1848), 87–89.

Y. Lipman, D. Cohen-Or, R. Gal, and D. Levin. 2007. Volume and Shape Preservation via Moving Frame Manipulation. *ACM Trans. Graph.* 26, 1, Article 5 (2007).

Y. Lipman, D. Levin, and D. Cohen-Or. 2008. Green Coordinates. *ACM Trans. Graph.* 27, 3, Article 78 (2008).

N. Magnenat-Thalmann, R. Laperrière, and D. Thalmann. 1988. Joint-dependent Local Deformations for Hand Animation and Object Grasping. In *Proc. on Graphics Interface*. 26–33.

A. McAdams, Y. Zhu, A. Selle, M. Empey, R. Tamstorf, J. Teran, and E. Sifakis. 2011. Efficient Elasticity for Character Skinning with Contact and Collisions. *ACM Trans. Graph.* 30, 4, Article 37 (2011).

G. Nielson, H. Hagen, and H. Müller. 1997. *Scientific Visualization*. IEEE Computer Society.

M. Pauly, D. K. Pai, and L. J. Guibas. 2004. Quasi-rigid Objects in Contact. In *Proc. of the 2004 ACM SIGGRAPH/Eurographics Symp. on Computer Animation*. Eurographics

Association, 109–119.

N. Phan-Thien and S. Kim. 1994. *Microstructures in elastic media : principles and computational methods.* Oxford U. Press.

P. Podio-Guidugli and A. Favata. 2014. *Elasticity for Geotechnicians.* Springer.

J. Reinders. 2007. *Intel Threading Building Blocks.* O'Reilly & Associates, Inc.

T. W. Sederberg and S. R. Parry. 1986. Free-form Deformation of Solid Geometric Models. In *ACM SIGGRAPH.* 151–160.

R. Setaluri, Y. Wang, N. Mitchell, L. Kavan, and E. Sifakis. 2014. Fast grid-based nonlinear elasticity for 2D deformations. In *Proc. of the ACM SIGGRAPH/Eurographics Symp. on Computer Animation.* 67–76.

K. Singh and E. Fiume. 1998. Wires: A Geometric Deformation Technique. In *ACM SIGGRAPH.* 405–414.

W. S. Slaughter. 2002. *The Linearized Theory of Elasticity.* Birkhäuser Basel.

O. Sorkine and M. Alexa. 2007. As-rigid-as-possible Surface Modeling. In *Proc. of the Fifth Eurographics Symposium on Geometry Processing.* 109–116.

W. von Funck, H. Theisel, and H. P. Seidel. 2006. Vector field based shape deformations. *ACM Trans. on Graph.* 25, 3 (2006), 1118–1125.

Wolfram Research Inc. 2016. *Mathematica Version 11.* Wolfram Research Inc.

A DERIVATION OF 3D REGULARIZED KELVINLETS

We derived the analytical expression (6) for the regularized Kelvinlet in 3D using the so-called *Papkovich-Neuber potentials* (or *PN potentials*); see, e.g., [Barber 2010]. Given a load distribution $\boldsymbol{b}(\boldsymbol{r})$, the PN potentials correspond to a pair of scalar-valued and vector-valued harmonic fields given by the integrals

$$\Psi(\boldsymbol{r}) = \int_{\mathbb{R}^3} \frac{\boldsymbol{b}(\boldsymbol{s})}{4\pi\|\boldsymbol{r}-\boldsymbol{s}\|} d\boldsymbol{s}, \quad \phi(\boldsymbol{r}) = \int_{\mathbb{R}^3} \frac{\boldsymbol{s}^t \boldsymbol{b}(\boldsymbol{s})}{4\pi\|\boldsymbol{r}-\boldsymbol{s}\|} d\boldsymbol{s}.$$

Once the PN potentials are known, the displacement solution for (2) can be computed by

$$\boldsymbol{u}(\boldsymbol{r}) = \frac{1}{\mu}\left[\Psi(\boldsymbol{r}) + \frac{1}{4(1-\nu)}\nabla\left(\phi(\boldsymbol{r}) - \boldsymbol{r}^t\Psi(\boldsymbol{r})\right)\right]. \tag{27}$$

For our specific load $\boldsymbol{b}(\boldsymbol{r}) = \boldsymbol{f}\rho_\varepsilon(\boldsymbol{r})$ with force vector $\boldsymbol{f}$ and normalized distribution $\rho_\varepsilon(\boldsymbol{r})$ given by (5), we have

$$\begin{cases} \Psi(\boldsymbol{r}) = \boldsymbol{f}\left[\displaystyle\int_{\mathbb{R}^3} \frac{\rho_\varepsilon(\boldsymbol{s})}{4\pi\|\boldsymbol{r}-\boldsymbol{s}\|} d\boldsymbol{s}\right] \equiv \boldsymbol{f}\;\psi_\varepsilon(\boldsymbol{r}), \\ \phi(\boldsymbol{r}) = \boldsymbol{f}^t\left[\displaystyle\int_{\mathbb{R}^3} \frac{\boldsymbol{s}\rho_\varepsilon(\boldsymbol{s})}{4\pi\|\boldsymbol{r}-\boldsymbol{s}\|} d\boldsymbol{s}\right] \equiv \boldsymbol{f}^t\Phi_\varepsilon(\boldsymbol{r}). \end{cases} \tag{28}$$

The scalar field ψ_ε can then be interpreted as the electric potential associated with the charge density $\rho_\varepsilon(\boldsymbol{s})$. Similarly, the i-th component of the vector field Φ_ε corresponds to the electric potential with charge density $s_i\rho_\varepsilon(\boldsymbol{s})$. The closed-form expression for ψ_ε can be found in [Cortez et al. 2005]:

$$\psi_\varepsilon(\boldsymbol{r}) = \frac{1}{8\pi}\left[\frac{2}{r_\varepsilon} + \frac{\varepsilon^2}{r_\varepsilon^3}\right].$$

The expression for Φ_ε is more complicated to derive: we used [Wolfram Research Inc. 2016] to compute an infinite Laplace series in spherical coordinates which, after a few algebraic manipulations, simplifies to

$$\Phi_\varepsilon(\boldsymbol{r}) = -\frac{3\varepsilon^2}{8\pi}\frac{\boldsymbol{r}}{r_\varepsilon^2}.$$

Finally, substituting Φ_ε and ψ_ε into (27) we obtain (6).

B DERIVATION OF 2D REGULARIZED KELVINLETS

We now detail the computations involved to obtain (25). Regularized Kelvinlets in 2D can be derived by integrating the 3D displacement field $\boldsymbol{u}_\varepsilon(\boldsymbol{r})$ given by (6) along the z-axis, as shown in (24). However, by naively substituting (6) to (24), we find

$$\int_{-\infty}^{\infty} \frac{1}{r_\varepsilon} dz = \lim_{L\to\infty}\left[2\ln\left(\frac{L + \sqrt{\mathrm{r}_\varepsilon^2 + L^2}}{\mathrm{r}_\varepsilon}\right)\right] \to \infty.$$

To overcome this issue, we exploit the fact that the equation of elastostatics is invariant to a global translation, i.e., we can translate a displacement satisfying (2) by a constant and it remains a solution. Based on this observation, we translate the 2D displacements in (24) by $-2\ln(L)$, which leads to

$$\lim_{L\to\infty}\left[2\ln\left(\frac{L + \sqrt{\mathrm{r}_\varepsilon^2 + L^2}}{\mathrm{r}_\varepsilon}\right) - 2\ln(L)\right] = -2\ln(\mathrm{r}_\varepsilon).$$

The remaining terms in (6) are easily integrated, resulting in (25).