Sensitive Couture for Interactive Garment Modeling and Editing

Nobuyuki Umetani
The University of Tokyo

Danny M. Kaufman
Columbia University

Takeo Igarashi
The University of Tokyo / JST ERATO

Eitan Grinspun
Columbia University

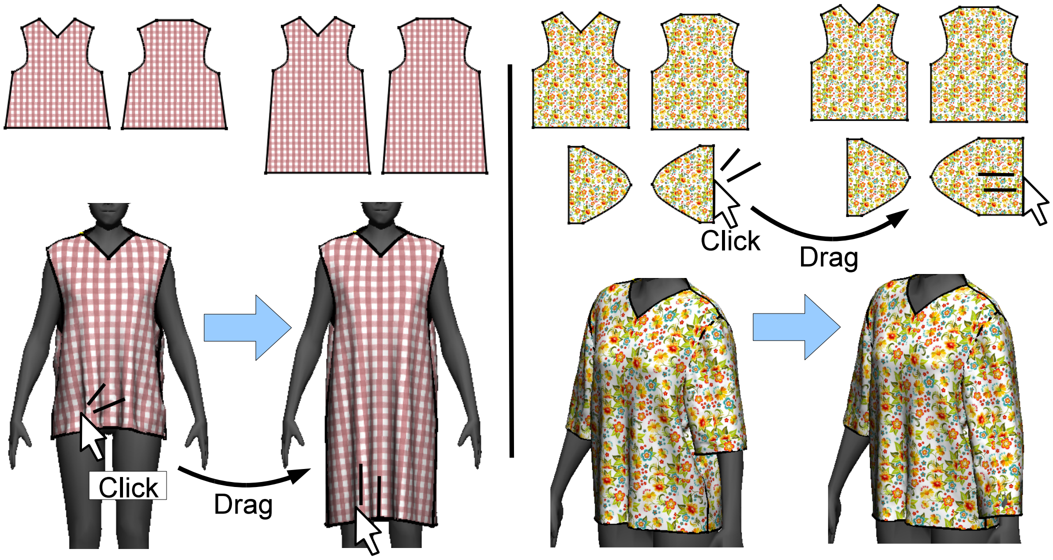


Figure 1: ***"2D or not 2D?"*** *This timeless question is rendered moot by* Sensitive Couture*, our tool for simultaneous, synchronized modeling and editing of both a 2D garment pattern* (top) *and its corresponding 3D drape* (bottom).

Abstract

We present a novel interactive tool for garment design that enables, for the first time, interactive bidirectional editing between 2D patterns and 3D high-fidelity simulated draped forms. This provides a continuous, interactive, and natural design modality in which 2D and 3D representations are simultaneously visible and seamlessly maintain correspondence. Artists can now interactively edit 2D pattern designs and immediately obtain stable accurate feedback online, thus enabling rapid prototyping and an intuitive understanding of complex drape form.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Physically based modeling;

Links: DL PDF WEB VIDEO CODE

1 Introduction

The multi-billion dollar fashion industry caters to every strata of consumption, from penny socks to haute-couture pieces costing thousands (even millions) of dollars [Sherman 2006]. Fashion's pervasion is rooted in clothing's role as a medium for personal expression. Garments that "suit us" invoke a sense of confidence and satisfaction.

Tailoring requires the insight to combine flat 2D panels of woven textiles that, stitched together into a garment, exhibit an expressive 3D shape when worn. In the language of differential geometry, a garment is simultaneously both a 2D and a 3D object: on the one hand, it may be viewed as the initial assembly of flat panels, each having holes and curved/kinked boundaries; on the other hand, it can be understood by its ultimate 3D form.

Classical iterative design The garment design process involves many iterations of drafting, synthesis, and revision that alternate between 2D and 3D perspectives: a tentative 2D panel design is created from which a corresponding garment is manufactured; the resulting garment reveals desired alterations to the 3D form that, in turn, induce revisions of the 2D design; and so forth. These many iterations consume raw materials, time, and energy.

Even veteran dressmaking teams go through many iterations where the *designer* conceptualizes 3D forms in sketches and the *pattern maker* drafts precise 2D outlines. The drape of a garment over a curved body is affected by frictional contact and the map from the 2D to 3D representation is complex and nonlinear. The challenge in sketching 3D forms is to stay true to the wrinkles and bulges that will be formed when a nearly-inextensible surface is draped over a body. Vice versa, revising 2D patterns often induces not only the expected alteration of the 3D forms but also unintended "side-effects" (pinching, buckling, tight spots) which are often only discovered after time- and resource-consuming assembly. It takes great effort and many attempts to bring the 2D and 3D views into correspondence, making design an inherently iterative, painstaking process.

While both the 2D ("intrinsic") and 3D ("extrinsic") perspectives are innate aspects of garment design, existing software tools force one view in lieu of the other. The situation is amplified in the computer animation industry. In general, artists either *sculpt* in 3D or *draft* in 2D but not both, and the choice is typically determined by the software used at the studio. Sculpting tools provide artists with a familiar interaction paradigm, but do not account for the special (low Gaussian curvature) structure of draped fabric, thus results appear less natural. Sometimes the 3D mesh is computationally flattened [Decaudin et al. 2006], a process that alters the 3D form and consumes time. On the other hand, drafting tools ensure that the final form is natural, but the artist must learn to draft 2D patterns, and wait for draping computations that produce the 3D form [Volino et al. 2005]. To sum, existing flattening (3D⟶2D) and draping (2D⟶3D) computations are akin in their effect to the time and material impediments of manual dressmaking: they divide design into discrete tasks and inhibit a free-flowing creative process.

***Sensitive Couture* (SC) for interactive bidirectional design** We present a novel approach to garment design, similar in spirit to the system proposed by Volino et al. [2005]. Our software tool provides a continuous, interactive, natural design modality. The 2D design and 3D draped form receive equal status, are simultaneously visible, and seamlessly maintain correspondence. The artist may interactively edit the 2D design immediately observing how these changes affect 3D form.

The impact on the user is more than merely a quantitative acceleration of the old design loop. Consider a small metallic "brain-teaser" puzzle, in which two linkages must be separated. Staring at the puzzle and thinking for a long time does not get one very far. But a little bit of manual tinkering quickly builds intuition about the "design space" and voilà! In this vein, an interactive, synchronized garment design process offers a qualitative jump in the way that design is carried out, where the discrete transitions between phases vanish: drafting, manufacturing, and critique of 3D form are fused into one seamless, natural process.

Figure 2: *Can you solve a brain teaser without interaction?*

There are several technical hurdles to incorporating online cloth simulation in the design process: existing high-resolution simulation codes are not yet fast enough for maintaining interactive-rate correspondence between the 2D and 3D views. The simulation must account for geometric nonlinearity and frictional contact while remaining stable even under rapid user input.

To address these challenges, our design tool employs a combination of techniques, including (i) fast prediction of 3D forms from cached shapes using sensitivity analysis and generalized moving least squares, (ii) fast invisible remeshing using positive mean value coordinates to accommodate arbitrary revisions of the pattern boundary, and (iii) stable and accurate cloth modeling using an isometric bending model, a modified St. Venant-Kirchhoff membrane element, and progressive refinement. The novel use of sensitivity analysis in enabling interactive-rate synchronization of the 2D and 3D perspectives is reflected in the tool's name, *Sensitive Couture*.

2 Related Work

Fashion being a huge field, computer-aided garment design has grown into an industry of its own, as evidenced by the myriad companies producing garment CAD products; spanning the five inhabited continents, companies such as AGE Technologies, AMSSystems, Browzwear, CADTERNSCustom Clothing Inc., Fashion CAD, Fashion Matters, Gemini CAD, GRAFIS, Investronica Systems, Marvelous Designer, Mechanix, OptiTex, Pad Systems, Pattern Works Int'l, and others provide a variety of tools focusing on 2D pattern creation and/or 3D draping for garment design.

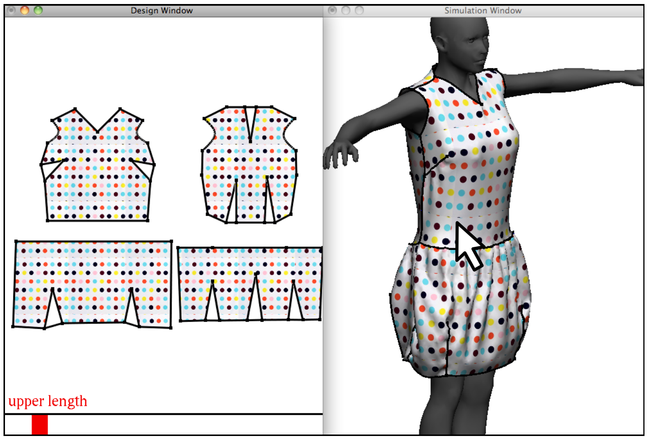


Figure 3: *A* Sensitive Couture *screen snapshot.*

Sketch-Based Design Sketch-based approaches to garment design have produced promising 2D⟶3D design processes that enable the mapping of fashion sketches to procedurally generated three-dimensional garments [Decaudin et al. 2006; Igarashi and Hughes 2003; Turquin et al. 2007]. These methods leverage a designer's honed intuition about garment shape and flow. Advances in flattening sketch space to developable patterns [Decaudin et al. 2006], placement and sliding of generated garments [Igarashi and Hughes 2003], and position-based garment generation [Turquin et al. 2007] have all enabled improved one-way work flow. However, such approaches are still hampered by the limitations of procedural, rather than physically based, garment generation and have yet to be extended to enable bidirectional editing.

Cloth Simulation The simulation of cloth, and more generally shells, is a widely treated area [Bridson et al. 2003; Baraff and Witkin 1998; Goldenthal et al. 2007]. Mass-spring models generally enable fast and simple computational models for rapid cloth simulation [Meyer et al. 2001; Choi and Ko 2002] providing responsiveness at the cost of accuracy and stability. Recent methods have also generated real-time wrinkle synthesis [Rohmer et al. 2010], as well as more generally, real-time, data-driven reduced cloth modeling [Feng et al. 2010; Wang et al. 2010].

Interactive Cloth Design and Simulation Physically based interactive design of clothing has long been a grand challenge for simulation and graphics [Volino et al. 2005]. Recently, promising frameworks have been proposed that suggest that interactive design processes combining 2D and 3D perspectives may be in reach for textile design [Volino et al. 2005]. Complementary developments in direct interactive editing on garment meshes [Keckeisen et al. 2004] and the real-time coupling between edits and simulation for simple mass-spring systems [Mori and Igarashi 2007], have likewise offered tantalizing forward progress towards seamless bidirectional garment authoring.

Design Sensitivity Analysis In the context of fashion, the preferences of a designer are hard to quantify and their choices difficult to predict. Design sequences can be expected to evolve *throughout* the creative process, potentially requiring reliable computation over broad ranges of design space. Thus we put offline optimization aside in favor of a sensitivity based strategy that enables a truly interactive exploration of design.

Similar in spirit to *differential manipulation* [Gleicher and Witkin 1991], which uses sensitivity maps to enforce geometric constraints at interactive rates, we propose to use design sensitivity to maintain the static equilibrium of drape pose.

In the engineering literature, this formulation is central to *design sensitivity analysis*, which attempts to provide (first-order) insight into the relationships between design and performance in either one-off calculations or offline optimizations; Van Keulen [2005] provides a recent survey, and Derouet-Jourdan et al. [2010] provide a recent application to rod design.

3 Interaction Tools

As depicted in Fig. 3 and the accompanying movie, Sensitive Couture (SC) presents the 2D and 3D design perspectives simultaneously. All selection operations are naturally at home in either the 2D or 3D window.

A typical session begins with layout of a "blank canvas." While SC affords from-scratch creation of patterns, designers typically begin with a *sloper* (or *block*), a basic pattern drafted to standard measurements intended as a generic starting point [Digest 2002] (see Fig. 4). Slopers are typically *parameterized* by height, girth, sleeve length, and so forth.

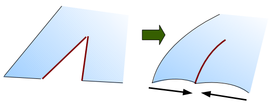
Whereas typical CAD tools might navigate the parametric space using sliders, SC further allows the designer a "tangible" navigation of parameter space via direct manipulation of the 2D pattern and the 3D draped shape, reinforcing the idea that all components are synchronized and can be accessed in any order.

Indeed, every aspect of the SC user experience emphasizes a free-flowing (as opposed to sequential or "linear") design process. For instance, as the designer makes detailed alterations ("strokes") to patterns by inserting darts, modifying boundary curves, etc., these strokes "ride" over the sloper, in the sense that the designer may revisit the parameters of the sloper at any time without undoing the creative, style-defining strokes.

If a sloper serves as blank canvas, then the rich diversity of final designs comes from the various creative strokes applied by designers. Our research goal was to understand the under-the-hood infrastructure needed for interactive design in a real-application (practical) context, thus we focused on a small but useful set of front-end tools intended to stress-test the underlying simulation infrastructure:

Curve edits The designer may alter the shape and position of the pattern boundary, which is defined by the control degrees of freedom (DOFs) of a spline. Positions are stored relative to the sloper, and are therefore naturally maintained over adjustment to the underlying sloper's dimensions.

Darts The designer may add or modify *darts,* triangular folds or excisions that induce intrinsic curvature (so-called *cone singularities*) thus "filling out" the 3D form. SC understands darts as "first-order primitives," so they have dart-specific DOFs to control their position, shape, and size. The designer adds a dart by drawing a line using the dart pen. If the line intersects a boundary, SC creates a triangular dart that "rides" the boundary, i.e., the designer may later freely slide the dart along the boundary.



Sewing/pleating The designer can specify that two boundary segments should be sewn. When the two segments differ in length, a *pleat* is formed, i.e., a sequence of attractive doubled-back folds that gather a longer piece of fabric into a shorter length. In all cases, SC automatically selects the boundary orientations that avoid cloth inversion.

Symmetry The designer may mark boundary pieces as symmetric about an axis, and subsequently SC enforces these symmetries.

Sloper parameters We emphasize that the designer can adjust sloper parameters (e.g., sleeve length, waist width) at any time. These adjustments can be made by manipulating a slider, or by direct manipulation in the 3D view. For example, the user can "tug" on a hemline to make a skirt longer (see Fig. 1, left and accompanying movie).

4 Sensitive Interaction

As the user revises the design, SC must keep the 2D and 3D perspectives synchronized. The 2D and 3D configurations, each realized by a mesh of n vertices, are related by the static equilibrium equation

$$\mathbf{R}(\mathbf{X},\mathbf{x}) = \mathbf{F}(\mathbf{X},\mathbf{x}) - \mathbf{Q}(\mathbf{X},\mathbf{x}) = 0\,, \qquad (1)$$

where $\mathbf{X} \in \mathbb{R}^{2n}$ is the undeformed configuration given by the 2D perspective, $\mathbf{x} \in \mathbb{R}^{3n}$ is the deformed configuration given by the 3D perspective, and $\mathbf{F}, \mathbf{Q} \in \mathbb{R}^{3n}$ are the external (e.g., gravitational) and internal (e.g., elastic) forces, respectively. The 2D and 3D pictures are in correspondence when the residual $\mathbf{R}(\mathbf{X},\mathbf{x}) \in \mathbb{R}^{3n}$ vanishes.

We solve (1) from scratch once for the initial drape (§7), an operation too costly to repeat at interactive rates due to the nonlinearity of (1). Fortunately, in design processes, most user input effects *incremental revisions* to the existing design.

Design sensitivity analysis Suppose that the designer alters the 2D configuration $\mathbf{X}$ to a nearby configuration $\mathbf{X} + \Delta\mathbf{X}$. Expanding (1) to first order yields the *incremental update* equation

$$(\nabla_{\mathbf{X}}\mathbf{R}(\mathbf{X},\mathbf{x}))\,\Delta\mathbf{X} + (\nabla_{\mathbf{x}}\mathbf{R}(\mathbf{X},\mathbf{x}))\,\Delta\mathbf{x} = 0$$

relating the change $\Delta\mathbf{X}$ in the 2D to the change $\Delta\mathbf{x}$ in the 3D picture, by a linear map $\mathrm{S} : \Delta\mathbf{X} \mapsto \Delta\mathbf{x}$

$$\Delta\mathbf{x} = (\nabla_{\mathbf{x}}\mathbf{R})^{-1}(\nabla_{\mathbf{X}}\mathbf{R})\Delta\mathbf{X} = \mathrm{S}\Delta\mathbf{X}.$$

where S is encoded by a $3n \times 2n$ *design sensitivity matrix*. In practice one applies the map by solving the incremental update equation using a linear solver.

Note that $\nabla_{\mathbf{x}}\mathbf{R}$ is the usual *stiffness matrix* associated with a nonlinear statics problem: if the static problem has a nonsingular stiffness matrix, then $\nabla_{\mathbf{x}}\mathbf{R}$ is full rank.

Linear sensitivity is insufficient Buckling, wrinkling, and static friction all make for a nonlinear relation between the 2D design and the 3D draped form. While linear sensitivity is a good first step, we found that for the larger edits typical at the start of the design process, or for radical redesigns, the local model produced by sensitivity analysis (i.e., linearization about the pre-revised configuration) does not remain valid for the full extent of the editing operation. In short, a linear sensitivity response centered about the original design is insufficient when the change in design is large. Therefore, we extend the basic sensitivity-based approach by accounting for nonlinearity via *coupled simulation* and *progressive nonlinear modeling.*

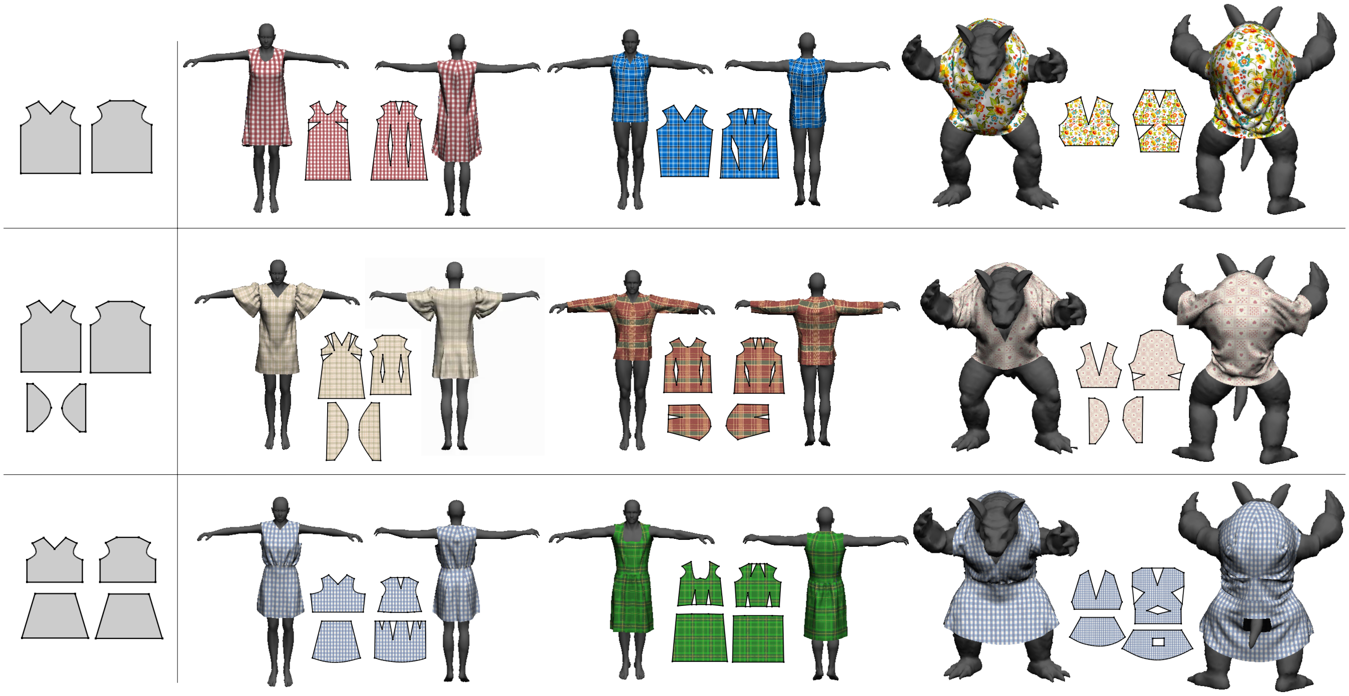


Figure 4: *Starting with just a few standard sloper templates artists can easily generate a wide variety of garments.*

Coupled simulation During idle times, such as pauses in mouse movement, we integrate the system in time, starting from the configuration produced via sensitivity-based increments. If this were a dynamic simulation, the interruption of time integration with sensitivity-based positional updates would likely cause great instabilities or other artifacts. Fortunately, our interest in static draping rather than inertial response allows us to use kinetic damping (§7) and hence to be well insulated from any "spikes" that might result from interleaving time integration with direct positional updates.

Progressive nonlinear modeling In the most crucial interactive editing operations—those where the designer hesitates in selecting between multiple design alternatives—the editing operation may endure over a longer duration (a few dozen seconds), and pause at various points in design space as options are being considered. In these cases, SC takes advantage of the available time to evaluate and cache additional linearizations (sensitivity matrices), thus building up a nonlinear model of drapes in the local design space. Compared to an up-front precomputation, this *progressive* enrichment of the local model allocates computation proportionally to the interest in a given region of design space, i.e., it is most accurate near designs that interest the user.

How do we aggregate the linearizations into a local model of simulation response in design space? In the context of offline optimization, some works have considered *moving least squares* (MLS) interpolation [Breitkopf et al. 2005]. MLS requires dense, well-spaced samples, does not make use of derivative information, and breaks down when the sampling pattern degenerates (i.e., there are too few samples along one dimension). To alleviate these difficulties, Martin et al. [2010] use *generalized* moving least squares (GMLS), which extend MLS interpolation to incorporate derivative information. Indeed, GMLS can make direct use of the sensitivity matrices that we compute [Atluri et al. 1999; Fries and Matthies 2003].

When the pointer moves to position $\mathbf{d} = (d_1, d_2)^T$, SC can draw upon two kinds of readily reusable data: the draped configuration $\mathbf{x}^0 \in \mathbb{R}^{3n}$ (i.e., zeroth-order data) at the previous pointer position $\mathbf{d}^0 \in \mathbb{R}^2$, and the sensitivities $\mathbf{s}_1^m = \partial\mathbf{x}/\partial d_1, \mathbf{s}_2^m = \partial\mathbf{x}/\partial d_2$ of the draped configurations $\mathbf{x}^m \in \mathbb{R}^{3n}$ (i.e., first-order data) at the cached previous configurations $\mathbf{d}^m \in \mathbb{R}^2$ $(m = 1, \ldots, M)$. Since GMLS is a direct extension of MLS, it is possible to extend the usual derivations of MLS and GMLS interpolation to now account for a combination of zeroth- and first-order samples. In particular, following the notation of Martin et al. [2010], the interpolated configuration field is given by $\mathbf{x}(\mathbf{d}) = \mathbf{a}(\mathbf{d})\mathbf{p}(\mathbf{d}) \in \mathbb{R}^{3n}$, where $\mathbf{a} \in \mathbb{R}^{3n\times 3}$ is a coefficient matrix applied to the monomial vector $\mathbf{p} = (1, d_1, d_2)^T$. At a given pointer position $\mathbf{d}$, the coefficients $\mathbf{a}$ are the minimizers of the least-squares error metric

$$J(\mathbf{a}) = \sum_{m=0}^{M} w(\mathbf{d}-\mathbf{d}^m)\,\|\mathbf{a}\mathbf{p}(\mathbf{d}^m) - \mathbf{x}^m\|^2 + \sum_{m=1}^{M} w(\mathbf{d}-\mathbf{d}^m) \sum_{j=1}^{2} \left\| \mathbf{a}\frac{\partial \mathbf{p}}{\partial d_j} - \mathbf{s}_j^m \right\|^2 ,$$

where w is a suitably chosen weighting function. We use $w(\mathbf{d} - \mathbf{d}^m) = 1/(\|\mathbf{d}^m - \mathbf{d}\|^2 + \epsilon^2)$, with ϵ a small constant guaranteeing finite weight (we use $\epsilon^2 = 10^{-3}$). As in the usual (G)MLS derivation, we analytically minimize J with respect to $\mathbf{a}$, obtaining

$$\mathbf{x}(\mathbf{d}) = \sum_{m=0}^{M} \mathbf{x}^m N^m(\mathbf{d}) + \sum_{m=1}^{M}\sum_{j=1}^{2} \mathbf{s}_j^m N_j^m(\mathbf{d}),$$

$$N^m(\mathbf{d}) = \mathbf{p}(\mathbf{d})^T \mathbf{G}(\mathbf{d})^{-1} \mathbf{p}(\mathbf{d}^m) w(\mathbf{d} - \mathbf{d}^m) \ ,$$

$$N_j^m(\mathbf{d}) = \mathbf{p}(\mathbf{d})^T \mathbf{G}(\mathbf{d})^{-1} \frac{\partial \mathbf{p}}{\partial d_j} w(\mathbf{d} - \mathbf{d}^m) \ ,$$

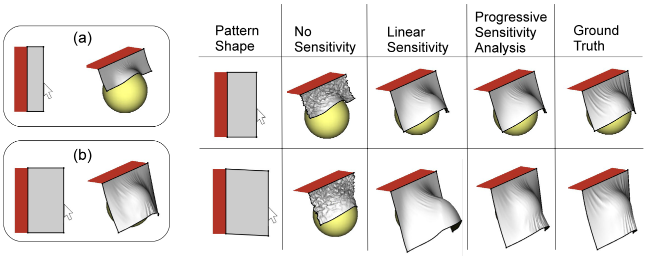


Figure 5: Comparison of sensitivity strategies. *Cloth is draped on a sphere with respectively, from left to right, no sensitivity, linear sensitivity, progressive sensitivity analysis with GMLS, and nonlinear ("ground truth") static equilibrium.*

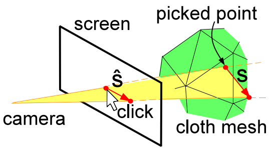
and

$$\mathbf{G}(\mathbf{d}) = \sum_{m=0}^{M} w(\mathbf{d}-\mathbf{d}^m)\mathbf{p}(\mathbf{d}^m)\mathbf{p}(\mathbf{d}^m)^T + \sum_{m=1}^{M} w(\mathbf{d}-\mathbf{d}^m) \sum_{j=1}^{2} \frac{\partial \mathbf{p}}{\partial d_j}\frac{\partial \mathbf{p}}{\partial d_j}^T .$$

Comparing different strategies Consider as a canonical example a hanging cloth partly draped on a sphere, and an editing operation in which the (undeformed, 2D material-space) length of the strip of cloth is increased. Fig. 5 and the accompanying movie compare the result of an editing operation conducted using only a dynamic, kinetically-damped simulation, augmented with linear sensitivity analysis, or augmented with progressive GMLS modeling. Here Fig. 5 (a) and (b) give the cached solutions employed by sensitivity. Observe (left to right in Fig. 5) that the dynamic simulation lags behind the user's edit; the formation of distracting wrinkling artifacts can be understood by interpreting the dynamic simulation as a process that approaches the final equilibrium state in a fine-to-coarse manner. Linear sensitivity, which can be viewed as a "global" or "implicit" approach, eliminates these artifacts, but, in using a linear model, does not approximate the overall draped shape well. The GMLS interpolation exhibits stable results that better agree with ground truth (Fig. 5, far right).

4.1 Bidirectional Sensitivity Enables Direct 3D Editing

Sensitivity information also enables SC to interpret editing operations applied directly to the 3D form.



Consider editing a sloper parameter $g \in \mathbb{R}$ (e.g., sleeve length), as depicted in Fig. 1, left. When the pointer is depressed over the 3D view, SC identifies the corresponding material point $(u, v) \in \mathbb{R}^2$ on the cloth, and computes the sensitivity vector $\mathbf{s} = \nabla_g \mathbf{x}(u, v) \in \mathbb{R}^3$, i.e., the first-order 3D motion of the cloth at the picked point with respect to sloper parameter g.

The 3-vector $\mathbf{s}$ is projected to the screen space vector $\hat{\mathbf{s}} \in \mathbb{R}^2$, which gives the first-order motion of the picked screen point with respect to g. As the user drags the pointer from $\mathbf{d} \in \mathbb{R}^2$ to $\mathbf{d} + \Delta\mathbf{d} \in \mathbb{R}^2$, SC updates the sloper parameter via the incremental relation

$$\Delta g = \hat{\mathbf{s}} \cdot \Delta\mathbf{d} / ||\hat{\mathbf{s}}||^2 .$$

Observe that when $||\hat{\mathbf{s}}||$ is small, the selected screen point is insensitive to g; for example, the position of a shirt collar may be independent of sleeve length. Thus, for small $||\hat{\mathbf{s}}||$ we neglect the drag. A simple and useful extension would be to visualize, before the mouse is depressed, the degree of sensitivity of a screen point, as a guide for the user.

5 Cloth Model

In selecting a cloth model for an interactive tool, our primary desiderata were stability and fast computation. Typical to most cloth models, we treat bending and stretching models separately [Baraff and Witkin 1998; Bridson et al. 2003]. We chose to work with triangle meshes, but the overall framework does not depend on this choice.

Bending Our final implementation uses the *isometric bending model* (IBM) of Bergou et al. [2006], which has a constant energy Hessian (force Jacobian), therefore (a) eliminating the cost of the force Jacobian computation for implicit time integration, (b) providing a simple matrix-vector multiplication for bending force computation (which would be easy to port to the GPU), and (c) ensuring that the Hessian remains positive semi-definite for all configurations and thereby stabilizing numerics. Before adopting IBM, we implemented a co-rotational treatment of the *Discrete Kirchhoff Triangle* [Batoz et al. 1980; Khosravi et al. 2007], but found that we needed a stronger guarantee of stability in the context of our interactive tool. We then adopted nonlinear hinges [Bridson et al. 2003; Grinspun et al. 2003], which lose some of the meshing-independence of DKT [Batoz et al. 1980] but increase stability. Ultimately, the cost of force Jacobian computation in the nonlinear models drove us to use IBM, which drops the generality of an arbitrary stress-strain map in favor of increased stability and speedier computation.

Membrane Our final implementation uses a *stabilized* St. Venant-Kirchhoff (StVK) *constant strain triangle* (CST). The usual StVK CST element, when it is compressed, can induce instability [Volino et al. 2009], because its force Jacobian becomes indefinite. Similar to Teran et al. [2005], when the element is in a compressed configuration, we adjust the Jacobian entries to eliminate negative eigenvalues. This stabilization affects only the *trajectory* toward the draped configuration, but it does not alter the set of solutions of the static equilibrium equations. To sum, this stabilization assures stability without affecting the draped shape.

We first tried a geometric model governed by changes in edge length and triangle area [Grinspun et al. 2003]. We found that the area term induced instabilities – the length term alone was stable, but lacked similitude to fabric [Delingette 2008]. We therefore transitioned to the CST model, first using a corotational treatment [Bittnar and Sejnoha 1996], which was stable, but was relatively slow in converging to the equilibrium draped shape StVK membrane, compared to our final (quartic) CST element. In hindsight, we might have also applied the stabilization to the area term of the geometric model, but since the stabilized StVK CST worked well, and is a convergent model, we were satisfied.

Robustness evaluation We have found the combination of the guaranteed positive semi-definite IBM bending and stabilized StVK CST membrane models to be more stable than other combinations in ongoing usage of SC. Since, to our knowledge, this is the first report of combining these approaches, we include for reference the results of two stability benchmarks (see Fig. 6) in which the static equilibrium (1) is solved with kinetic damping (see §7) starting from a random initial guess. Alternatives, discussed above, that we tried earlier were either unstable or were slower to converge.

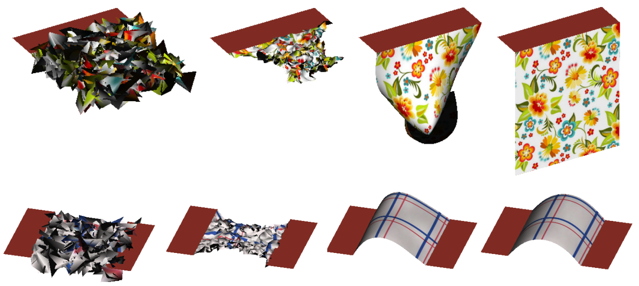


Figure 6: ***Stability Benchmark:*** ***(top)*** *Snapshots at initial, third, 50th, and 200th iterations for the draping of a rectangular (membrane-dominated) textile (3K triangles, 12s total computation).* ***(bottom)*** *Snapshots at initial, third, 50th, and 100th iterations for static equilibrium of a rectangular (bending-dominated) thin plate with bending stiffness 100× that of textile (2K triangles, 4s total computation).*

6 Mesh Updates

When the designer uses the pointer to drag 2D pattern elements (e.g., boundary vertices, darts, boundary spline tangents), SC updates the (unseen) positions of the internal vertices (see Fig. 8) thus maintaining uniform, well-shaped elements throughout the material domain. The concatenated change of all internal and boundary vertex positions makes up the change ΔX in the 2D configuration fed to the sensitivity-based 3D update.

While such a mesh update might be achieved by solving equilibrium of the mass spring system, the resulting nonlinear motion with respect to pattern element alteration makes first-order sensitivity analysis incorrect. Instead, we adopt a coordinate-based mesh manipulation that is *linear* in pattern element alteration.

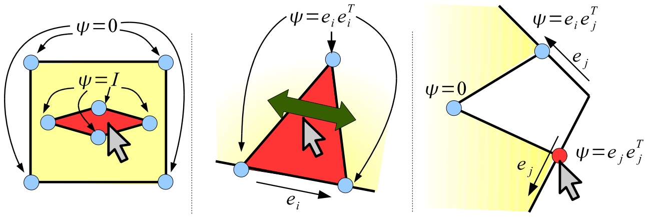


Figure 7: ***Action of*** $\boldsymbol{\Psi}$ *for* (left) *translation of an interior (diamond) dart,* (center) *translation of a boundary dart, and* (right) *adjustment of boundary dart opening.*

Mapping pointer movement to alterations For each kind of pattern element, we first define how pointer motion affects the control vertices (CVs) of a 2D pattern element; e.g., for a boundary dart, a drag centered inside the dart moves the three CVs identically, whereas a drag of the dart's interior CV leaves the boundary vertices stationary. Let $\boldsymbol{\Psi}_i$ be the 2×2 tensor relating pointer movement $\Delta\mathbf{d}$ to the movement $\boldsymbol{\Psi}_i\Delta\mathbf{d}$ of the ith CV. Typically, our tools use: (Fig. 7-*left*) $\boldsymbol{\Psi} = \mathbf{I}$ or $\boldsymbol{\Psi} = 0$, in which the CV follows the pointer or remains stationary, respectively; (Fig. 7-*center*) $\boldsymbol{\Psi} = \mathbf{e}\mathbf{e}^T$, in which the CV shadows the pointer movement only in the direction $\mathbf{e}$; or (Fig. 7-*right*) $\boldsymbol{\Psi} = \mathbf{e}_i\mathbf{e}_j^T$ in which the CV moves along $\mathbf{e}_i$ when the pointer moves along $\mathbf{e}_j$.

Interpolating the alteration Having defined $\boldsymbol{\Psi}$ at the CVs, it remains to interpolate the motion in the remainder of the domain. Since $\boldsymbol{\Psi}$ is known at all boundary CVs, we linearly interpolate along the boundary, and use *positive mean value coordinates* (PMVC) [Lipman et al. 2007] to efficiently determine the tensor field $\boldsymbol{\Psi}$ throughout the domain. PMVC builds on *mean value coordinates* (MVC) [Floater 2003] by incorporating a notion of *visibility*, which we found enhances the coordinate's interpolation capability in the non-convex higher genus shapes typical of 2D design patterns. Fig. 8 compares the two *linear* mesh deformation models we implemented based on MVC and PMVC, where we observed that PMVC produces a homogenous deformation that avoids the distortions and inversions.

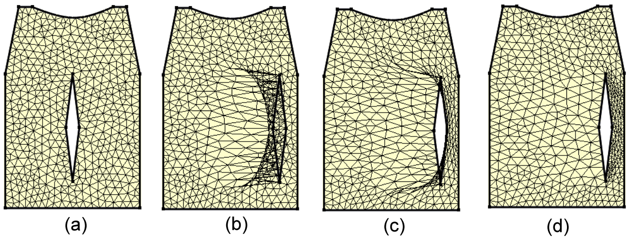


Figure 8: *Our 2D pattern manipulation employs* ***positive mean value coordinates (PMVC) with Delaunay smoothing****. This allows for linear interpolation over a domain, while enabling a more satisfying interpolation in nonconvex higher-genus domains. Here we illustrate the advantages: consider (a) an undeformed mesh, (b) mesh manipulation with MVC, (c) mesh manipulation with PMVC, and finally (d) PMVC with Delaunay smoothing.*

Applying the sensitivity map With the tensor field $\boldsymbol{\Psi} = [\boldsymbol{\Psi}_1\ \boldsymbol{\Psi}_2\ \ldots]^T$ in hand, sensitivity is calculated by mapping pointer movement to 2D mesh movement $\Delta\mathbf{X} = \boldsymbol{\Psi}\Delta\mathbf{d}$. We note that the typical sensitivity analysis considered in engineering design does not involve this extra degree of mapping, since sensitivity is typically conducted with respect to a specific boundary mesh element [Van Keulen et al. 2005].

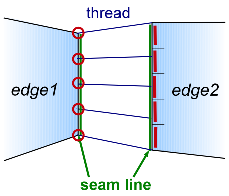
Fast and slow remeshing Over the course of multiple large manipulations, the mesh may distort necessitating remeshing. We first attempt *Delaunay smoothing*, updating mesh connectivity retaining nodal positions. With our finite element discretization, displacement and sensitivity are stored at vertices and need not be recomputed, making this an inexpensive way to improve the mesh. If mesh quality (ratio of diameter of incircle against the maximum edge length) remains poor, SC rebuilds the mesh from scratch and interpolates the simulation state to the new mesh using barycentric coordinates.

7 Numerics of Time Integration

SC intends to display the static drape of the cloth, consequently any computation invested into capturing dynamics is wasted effort. When a previous static solution is not available for an incremental sensitive update, SC must find the static equilibrium from scratch, and do so quickly. A from-scratch solve is needed not only for the initial drape, but also when new 2D pattern elements are added midway through design.

One of the fastest ways we found to solve the static equilibirum (1) from scratch is to employ *kinetic damping* [Barnes 1988; Volino and Magnenat-Thalmann 2007], which integrates the (undamped) equations of motion while monitoring total kinetic energy at each time step. When the kinetic energy reaches a local maximum (a condition evaluated by considering three consecutive time steps), the kinetic damping approach zeros the velocity (the kinetic energy). The intuition behind this is that in a conservative oscillatory system, when kinetic energy is at a maximum, potential energy is at a minimum. When the system is far from the minimal potential, damping would slow down the approach to the minimum; when the system is close to the minimal potential, reducing the momentum helps to avoid overshooting the minimum. We found that, in practice, kinetic damping is simple to implement, and reaches the draped configuration faster than a dynamic simulation with Rayleigh damping, an application of gradient descent, or the stabilized Newton's method.

We apply kinetic damping to the semi-implicit time integration scheme proposed by Baraff and Witkin [1998]. Since the coefficient matrix of the dynamic simulation and sensitivity analysis are both positive-definite, we solve the system using conjugate gradients preconditioned with ILU(0) [Saad 2003]. While the IBM bending model has a constant Hessian, the StVK CST membrane model does not, and we must numerically factorize the matrix at every time step.



The performance of ILU(0) preconditioning is (surprisingly) significantly influenced by choices in the treatment of *seams*. SC sews boundaries of corresponding panels using Hookean springs. In general, the boundaries do not correspond in length—an important feature in dressmaking used to effect pleats and ruffles—nor in connectivity, therefore, SC connects the ("emitting") vertices of the one panel with springs anchored at ("receiving") boundary edges of the other panel (see Fig. 7). To avoid gaps at the seams, the seam springs are much stiffer than textile tensile stiffness. The consequent linear system has stiff and weak components, thus the success of ILU(0) preconditioning depends on the permutation of matrix entries [Saad 2003]. In a nutshell, ILU(0) favors permutations where large entries appear earlier (toward top-left of matrix) and small entries appear later (bottom-right). Entries associated to emitting vertices dominate entries of receiving vertices, which in turn dominate all other vertices.

Before considering penalty-based seams, we used Lagrange multipliers, which allow for exact constraint enforcement without preselection of a spring stiffness. We were not satisfied with the increase in the linear system size and the indefiniteness of the resulting system matrix, which combined slowed convergence. We then observed that the usual problems plaguing penalty methods are less relevant to our context: since our problem setting is fixed (the spatial scale of the cloth designs do not vary by orders of magnitude), estimating the penalty stiffness once up-front is sufficient to obtain a good seal at the seams. The penalty method maintains the positive-definiteness of the system, and since the set of dominating matrix entries is given by a simple tallying of seam vertices (a set which remains constant except during exceptional stitching events), the permutation for ILU(0) is of negligible implementation and computational cost.

Contact model SC models contact between the garment and the body, but does not consider garment self-contact. Contact and friction between body and cloth are essential for cloth drape simulation. We opted for speed and robustness at some expense to accuracy. As with seaming we enforce contact constraints at mesh nodes using penalty springs. For contact we place normal springs at collision sites detected by a one-time, precomputed adaptive signed distance field [Frisken et al. 2000]. We model friction using a moving anchor spring method [Hasegawa et al. 2003] that enables both static and dynamic friction modes. Contacting nodes are connected by springs to seeded anchors vertices placed on the contacting surface. We then update (or release) anchor positions, with respect to nodal movement, to ensure Coulomb's Law is satisfied by these anchor spring forces. This allowed us to obtain stable draping and frictional wrinkling.

Full-rank of sensitivity matrix In general, for a nonlinear cloth statics problem, there exist configurations for which the stiffness matrix is close to singular. Consider a horizontal cantilevered rectangle of sufficiently thin fabric: intuitively, it can't hold itself up. Quantitatively, gravity acts against the bending mode. The ratio of bending membrane stiffness of a thin plate is $O(h^2)$ where h is the thickness measured relative to the object's characteristic length [Malvern 1969]. In principle one could select h so small that the configuration is near singular, with kernel corresponding to normal displacements. In practice, we did not encounter this problem while using SC. However, for completeness, we did experiment with the cantelevered rectangle to induce this problem. Since the kernel (must) correspond to normal displacements, the remedy is direct: if the solver fails to converge, we recompute S with bending stiffness corresponding to $h = 1/100$, which is sufficiently thick to avoid a singular matrix. The modification of the bending stiffness is local to the sensitivity computation, and does not affect the bending stiffness used in determining the ultimate static drape. We found that this stabilization was not invoked during cloth design, including all the examples shown in the video.

7.1 Progressive Refinement

We find that as users explore design space with SC, detailed cloth behavior is unnecessary during the sketch-like exploratory phases where rapid-fire design choices are made. On the other hand, during the slower, refinement-oriented phases of garment design, fine-level details are critical. We therefore focus on *progressive refinement* a technique long-used in graphical applications to provide a sense of immediacy [Hoppe 1996]. We employ the so-called *Cascading Multigrid* approach [Bornemann and Deuflhard 1996]. As users provide input, we initially solve the drape using a coarsened mesh. If we reach convergence at the coarse-level *prior* to the initiation of new design edits, we then warm start our fine mesh solve with the coarse solution. For preview we always display the fine mesh representation. Fine-level nodes are updated progressively: first via barycentric coordinates from the coarsened solution and later, when not interrupted, via direct update from the fine-level solve.

7.2 Sensitive Couture Workflow

Our approach is then best illustrated by considering a typical editing step (see also Fig. 11):

(1) Mouse click *Design sensitivity computation.* The mapping between mesh updates and XY mouse movement is computed. Sensitivity analysis then calculates the corresponding bi-modal linear response, at the clicked point, with respect to these maps (§6).

(2) Drag *Sensitive response.* While edits are performed with the mouse button down, SC provides instantaneous linear response. As the editing process continues progressive nonlinear modeling enriches the local model and the corresponding response (§4).

(3) Release *Time Integration.* When the mouse button is released the current sensitivity configuration warm starts the next time-integration cycle (§7).

examples	coarse nodes	fine nodes	bending stiffness (Nm)	stretching stiffness (N/m)	area density (kg/m^2)	drag operations (fps)			coarse solve (fps)			fine solve (fps)			remesh (ms)	sensitivity (ms)			GMLS interp. (ms)
						avg	max	min	avg	max	min	avg	max	min	avg	avg	max	min	avg
Blue Armadillo (Fig. 4)	1682	10168	10^{-8}	10	0.20	17.0	19.9	10.1	17.4	20.4	15.3	2.4	2.7	1.9	233	145	178	82	0.38
Man (Fig. 9)	1634	10149	10^{-8}	10	0.20	15.0	17.2	11.1	20.4	22.7	18.8	2.4	3.2	2.0	227	316	492	170	0.36
Woman (Fig. 9)	1580	10040	10^{-8}	10	0.20	19.5	23.0	16.0	25.1	28.5	22.9	2.9	3.8	2.6	144	310	440	155	0.38
Armadillo (Fig. 9)	1600	10073	5×10^{-6}	200	0.15	19.3	22.0	17.0	25.1	26.3	23.8	3.1	3.2	3.1	146	228	513	116	0.40
Heart (Fig. 10)	1543	5032	3×10^{-4}	2000	0.08	20.7	22.2	19.3	24.2	25.5	22.7	7.2	10.0	5.1	123	639	419	767	0.39

Table 1: ***SC session parameters, statistics, and timings:*** *all timings were measured on an Intel Core 2 Duo 2.66 GHz laptop, 4Gb memory.*

8 Results and Discussion

Figure 1 illustrates the bidirectional SC workflow for designing garments on 3D models. SC enables designer edits of both the 2D pattern (Fig. 1, right) *and* the 3D draped model (Fig. 1, left). Material parameters for fabrics used are estimated both by measurement and experiment. We measure the areal density of fabrics and then adjust stiffnesses to match simulations with on-sphere draping tests. A video accompanying this paper demonstrates the interactive, stable workflow provided by SC during the various phases of the garment design process.

Building high-fidelity interactive rate simulation and editing tools stressed a range of tried and true methods. In many cases we discovered that techniques that performed well in standard settings did not satisfy our needs for seamless, interactive rate updates. Often obtaining working solutions required unexpected and complex combinations of methods. Throughout our discussion we have documented both *what did not work*, as well as what did (and thus made SC successful). Correspondingly we have provided comparisons throughout our discussion to explain these issues. These comparisons and validations can be found in Figures 5, 6, and 8, while in Table 1 we present timing and performance statistics for a range of SC sessions.

8.1 Evaluation and Garment Manufacture

Here we wish to focus on how these parts sum to a whole in SC. As such, we have documented and captured (see our accompanying movies) the editing process and work flow of a variety of skilled and amateur clothing designers as they use SC to design a range of garments (see Fig. 9).

Garments designed in our SC sessions were then manufactured from the generated patterns. See Figure 9 for snapshots of the editing process, the completed 2D designs, simulated final drapes, and corresponding manufactured garments.

9 Conclusion

We have presented a novel interactive garment design tool, Sensitive Couture (SC), that, for the first time, offers seamless bidirectional design and editing capabilities for the generation of 2D patterns and the online simulation of drape. SC provides a continuous, interactive, natural design modality in which the 2D design and 3D draped form receive equal status, are simultaneously visible, and seamlessly maintain correspondence. As such, artists are enabled to interactively edit and explore 2D designs and immediately observe how these changes affect 3D form.

9.1 Future extensions to Sensitive Couture

While contact remains well-resolved for body-garment interaction, complex folding patterns will additionally require self-contact resolution. We are interested in exploring local methods of identifying and robustly treating self-contact and intersection in the SC framework. Similarly, as in the traditional couture setting, we currently consider the drape of SC garments on static 3D models. Clearly a desirable extension is for a dynamic preview capability allowing the simulation of 3D garment behavior subject to 3D model motion.

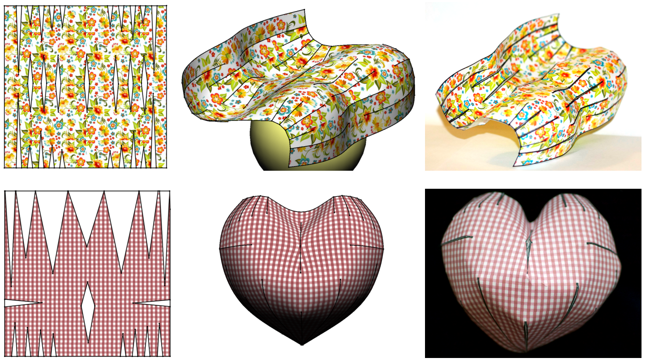


Figure 10: *In principle, SC could be extended to many other flexible materials, enabling the design of upholstery, aluminum sculptures, or these paper sculptures.* (Left to right): *an artist produced 2D pattern of darts, the corresponding 3D paper simulation, and the manufactured papercraft objects. We have yet to extend SC to incorporate the constitutive model of materials such as paper.*

We look forward to extending SC to a user-evaluated, fully-featured tool targeting design professionals in both animation and fashion. However, the design of shell-based objects also has exciting applications beyond garment design (see Fig. 10). We expect that SC-type tools can be extended to address interactive design needs in architecture, industrial design, and engineering, e.g., tensile structures, metal folding processes, upholstery, balloons.

In principle, the 3D editing of more than two design degrees of freedom (DOFs) could follow the same line of thought: so long as no more than two of the DOFs have non-negligible screen space vectors at the drag point, the editing operation is unambiguous and straightforward to implement. However, the likelihood that the edit operation remains unambiguous decreases as the number of available DOFs increases, raising interesting directions for future work: (a) SC should intuitively convey to the user which DOFs are affected by the edit (information that varies over screen space), (b) is there an intuitive (unambiguous and deterministic) way of mapping the 2D drag to the simultaneous revision of more than two DOFs?

9.2 Future work in integrated design, simulation, and interaction

Integrated design, simulation, and interaction (IDSI) is an outstanding challenging open problem in graphics and engineering. SC grew out of our interest in tackling this longstanding problem. Many application domains do not admit interactive, physics-in-the-loop design solutions for problems of practically relevant complexity. In this work we identified garment design as a domain for which IDSI was (barely) within reach for problems of real-world complexity on current hardware.

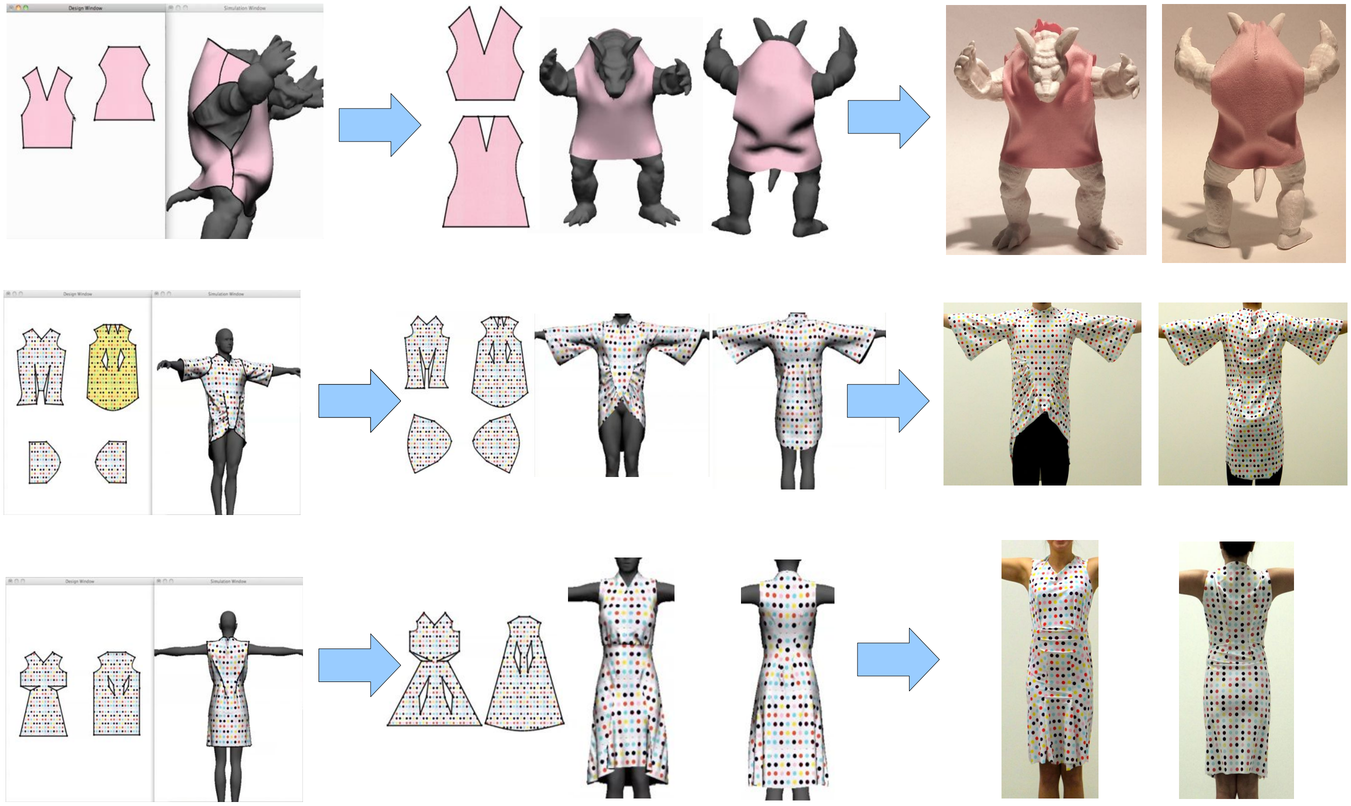


Figure 9: The SC workflow: *an interactive design session leads to a final 2D pattern and corresponding 3D drape pose which is then realized from the pattern.*

To reach interactive rates in SC required leveraging observations shared by many potential IDSI problems. Among these are (a) faster, more stable computation of equilibrium, at the expense of dynamics, for design of static structures; (b) synchronized editing and bidirectionality for a seamless design process; (c) sensitivity analysis for interactive exploration of nearby designs; and (d) adaptive enrichment of sensitivity information using GMLS to leverage users' natural pauses and inclination to explore multiple design variants. We hope that insights such as these, originally garnered for SC, will prove portable to a broad range of domains, and that SC itself will serve as a motivating example spurring research into fusing design, simulation, and interaction for a broad range of exciting domains.

Acknowledgements

We would like to thank Charles Han for dress design, garment manufacture, and modeling; Susana Kim for dress modeling; and Brandon Arrington for SC testing, and papercraft design and manufacture. Nobuyuki Umetani was supported in part by the Funai Overseas Scholarship and the JSPS. Danny Kaufman and Eitan Grinspun were supported in part by the Sloan Foundation, the NSF (CAREER Award CCF-0643268, IIS-0916129), Adobe, ATI, Autodesk, Intel, mental images, Microsoft Research, NVIDIA, Side Effects Software, the Walt Disney Company, and Weta Digital.

References

ATLURI, S., CHO, J., AND KIM, H. 1999. Analysis of thin beams, using the meshless local Petrov–Galerkin method, with generalized moving least squares interpolations. *Computational Mechanics 24*, 5, 334–347.

BARAFF, D., AND WITKIN, A. 1998. Large steps in cloth simulation. In *SIGGRAPH '98*, ACM, 43–54.

BARNES, M. 1988. Form-finding and analysis of prestressed nets and membranes. *Computers & Structures 30*, 3, 685–695.

BATOZ, J.-L., BATHE, K.-J., AND HO, L.-W. 1980. A study of three-node triangular plate bending elements. *Int'l Journal for Numerical Methods in Engineering 15*, 12, 1771–1812.

BERGOU, M., WARDETZKY, M., HARMON, D., ZORIN, D., AND GRINSPUN, E. 2006. A Quadratic Bending Model for Inextensible Surfaces. In *Fourth Eurographics Symposium on Geometry Processing*, 227–230.

BITTNAR, Z., AND SEJNOHA, J. 1996. *Numerical Methods in Structural Mechanics*. American Society of Civil Engineers.

BONET, J., AND WOOD, R. D. 2008. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press.

BORNEMANN, F., AND DEUFLHARD, P. 1996. The cascadic multigrid method for elliptic problems. *Numerische Mathematik 75*, 2, 135–152.

BREEN, D. E., HOUSE, D. H., AND WOZNY, M. J. 1994. Predicting the drape of woven cloth using interacting particles. In *SIGGRAPH '94*, ACM, 365–372.

BREITKOPF, P., NACEUR, H., RASSINEUX, A., AND VILLON, P. 2005. Moving least squares response surface approxima-

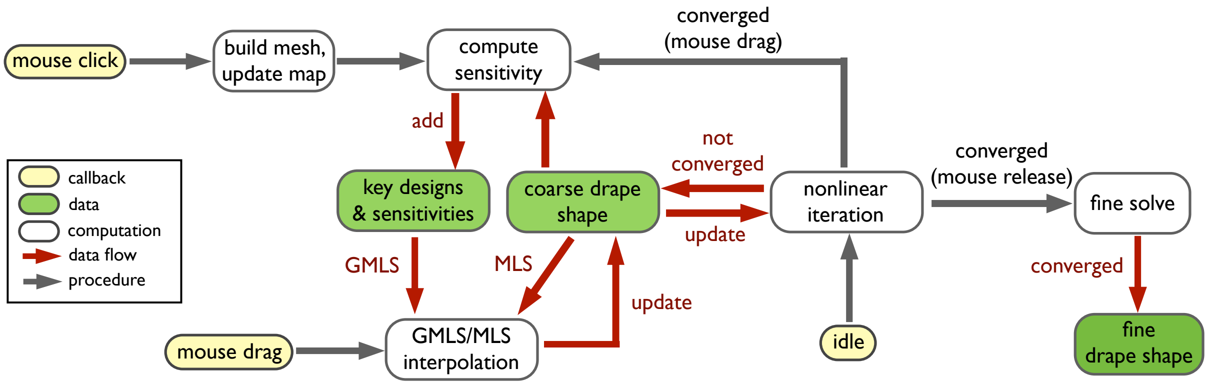


Figure 11: ***SC workflow:*** *an overview of SC steps.*

tion: formulation and metal forming applications. *Computers & Structures 83*, 17-18, 1411–1428.

Bridson, R., Marino, S., and Fedkiw, R. 2003. Simulation of clothing with folds and wrinkles. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 28–36.

Choi, K.-J., and Ko, H.-S. 2002. Stable but responsive cloth. *ACM Trans. Graph. (SIGGRAPH 02) 21*, 3, 604–611.

Daz 3D. http://www.daz3d.com/.

Decaudin, P., Julius, D., Wither, J., Boissieux, L., Sheffer, A., and Cani, M. 2006. Virtual garments: A fully geometric approach for clothing design. *Computer Graphics Forum 25*, 3, 625–634.

Delingette, H. 2008. Triangular springs for modeling nonlinear membranes. *IEEE Transactions on Visualization and Computer Graphics 14*, 2, 329–341.

Derouet-Jourdan, A., Bertails-Descoubes, F., and Thollot, J. 2010. Stable inverse dynamic curves. *ACM Trans. Graph. (SIGGRAPH Asia) 29*, 137:1–137:10.

Digest, R. 2002. *New Complete Guide to Sewing (Readers Digest)*. Readers Digest, 10.

Erleben, K., Sporring, J., Henriksen, K., and Dohlmann, H. 2005. Charles River Media.

Feng, W.-W., Yu, Y., and Kim, B.-U. 2010. A deformation transformer for real-time cloth animation. *ACM Trans. Graph. (SIGGRAPH 10) 29*, 4, 108:1–108:9.

Floater, M. 2003. Mean value coordinates. *Computer Aided Geometric Design 20*, 1, 19–27.

Fries, T., and Matthies, H. 2003. Classification and overview of meshfree methods. *Tech Report., TU Braunschweig, Germany*.

Frisken, S. F., Perry, R. N., Rockwood, A. P., and Jones, T. R. 2000. Adaptively sampled distance fields: a general representation of shape for computer graphics. In *Proc. ACM SIGGRAPH 00*, 249–254.

Gleicher, M., and Witkin, A. 1991. Differential manipulation. In *Graphics Interface*, 61–67.

Goldenthal, R., Harmon, D., Fattal, R., Bercovier, M., and Grinspun, E. 2007. Efficient simulation of inextensible cloth. *ACM Trans. Graph. (SIGGRAPH 07) 26*, 3.

Grinspun, E., Hirani, A., Desbrun, M., and Schröder, P. 2003. Discrete Shells. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 62–67.

Hasegawa, S., Fujii, N., Koike, Y., and Makoto, S. 2003. Real-time rigid body simulation based on volumetric penalty method. In *Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*.

Hoppe, H. 1996. Progressive meshes. In *SIGGRAPH '96*, ACM, 99–108.

Igarashi, T., and Hughes, J. F. 2003. Clothing manipulation. *ACM Trans. Graph. 22*, 3, 697–697.

Igarashi, Y., Igarashi, T., and Suzuki, H. 2009. Interactive cover design considering physical constraints. *Computer Graphics Forum 28*, 7, 1965–1973.

James, D., and Fatahalian, K. 2003. Precomputing interactive dynamic deformable scenes. *ACM Trans. on Graph. (SIGGRAPH 03) 22*, 3, 879–887.

Keckeisen, M., Feurer, M., and Wacker, M. 2004. Tailor tools for interactive design of clothing in virtual environments. In *Proceedings of the ACM symposium on Virtual reality software and technology*, 182–185.

Khosravi, P., Ganesan, R., and Sedaghati, R. 2007. Corotational non-linear analysis of thin plates and shells using a new shell element. *Int'l Journal for Numerical Methods in Engineering 69*, 859–885.

Lipman, Y., Kopf, J., Cohen-Or, D., and Levin, D. 2007. GPU-assisted positive mean value coordinates for mesh deformations. In *Proceedings of the fifth Eurographics Symposium on Geometry Processing*, Eurographics, 117–123.

Malvern, L. E. 1969. *Introduction to the mechanics of a continuous medium*. Prentice-Hall.

Martin, S., Kaufmann, P., Botsch, M., Grinspun, E., and Gross, M. 2010. Unified simulation of elastic rods, shells, and solids. *ACM Trans. Graph. (SIGGRAPH 10) 29*, 4, 1–10.

McNamara, A., Treuille, A., Popović, Z., and Stam, J. 2004. Fluid control using the adjoint method. *ACM Trans. Graph. (SIGGRAPH 04) 23*, 3, 449–456.

Meyer, M., Debunne, G., Desbrun, M., and Barr, A. H. 2001. Interactive animation of cloth-like objects in virtual reality. *J. Vis. Comput. Animat. 12*, 1 (2), 1–12.

MORI, Y., AND IGARASHI, T. 2007. Plushie: An interactive pattern design for plush toys. *ACM Trans. Graph. (SIGGRAPH 07) 23*, 3.

MÜLLER, M. 2008. Hierarchical position based dynamics. *Proc. of Virtual Reality Interactions and Physical Simulations*.

OPTITEX. http://www.optitex.com/.

ROHMER, D., POPA, T., CANI, M.-P., HAHMANN, S., AND SHEFFER, A. 2010. Animation wrinkling: augmenting coarse cloth simulations with realistic-looking wrinkles. *ACM Trans. Graph. (SIGGRAPH Asia) 29*, 6, 157:1–157:8.

SAAD, Y. 2003. *Iterative methods for sparse linear systems*. SIAM, Philadelphia.

SCHMIDT, R., GRIMM, C., AND WYVILL, B. 2006. Interactive decal compositing with discrete exponential maps. *ACM Trans. Graph. (SIGGRAPH 06) 25*, 3, 605–613.

SHERMAN, L. 2006. The cult of couture. *Forbes*.

SMITH, J., HODGINS, J., OPPENHEIM, I., AND WITKIN, A. 2002. Creating models of truss structures with optimization. *ACM Trans. Graph. (SIGGRAPH 02) 21*, 3, 295–301.

TERAN, J., SIFAKIS, E., IRVING, G., AND FEDKIW, R. 2005. Robust quasistatic finite elements and flesh simulation. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 181–190.

TURQUIN, E., WITHER, J., BOISSIEUX, L., CANI, M.-P., AND HUGHES, J. 2007. A sketch-based interface for clothing virtual characters. *IEEE Computer Graphics & Applications*.

VAN KEULEN, F., HAFTKA, R., AND KIM, N. 2005. Review of options for structural design sensitivity analysis. Part 1: Linear systems. *Computer Methods in Applied Mechanics and Engineering 194*, 30-33, 3213–3243.

VOLINO, P., AND MAGNENAT-THALMANN, N. 2007. Stop-and-go cloth draping. *The Visual Computer 23*, 9-11, 669–677.

VOLINO, P., CORDIER, F., AND MAGNENAT-THALMANN, N. 2005. From early virtual garment simulation to interactive fashion design. *Computer-Aided Design 37*, 6, 593–608.

VOLINO, P., MAGNENAT-THALMANN, N., AND FAURE, F. 2009. A simple approach to nonlinear tensile stiffness for accurate cloth simulation. *ACM Trans. Graph. (TOG) 28*, 4, 1–16.

WAGGONER, C., AND BARAFF, D. 2007. Virtual tailoring for Ratatouille: clothing the fattest man in the world. In *ACM SIGGRAPH 2007 sketches*.

WANG, H., HECHT, F., RAMAMOORTHI, R., AND O'BRIEN, J. F. 2010. Example-based wrinkle synthesis for clothing animation. *ACM Trans. Graph. (SIGGRAPH 10) 29*, 4, 107:1–107:8.

YONGJOON LEE, SUNG-EUI YOON, S. O. D. K., AND CHOI, S. 2010. Multi-resolution cloth simulation. *Computer Graphics Forum (Pacific Graphics) 29*, 7.

ZHU, Y., SIFAKIS, E., TERAN, J., AND BRANDT, A. 2009. An efficient multigrid method for the simulation of high-resolution elastic solids. *ACM Trans. Graph. (TOG) 28*, 2, 16:1–16:18.