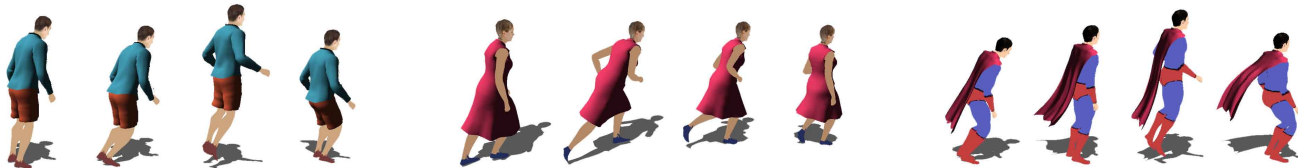# Stable Spaces for Real-time Clothing

Edilson de Aguiar[1,*]   Leonid Sigal[1,†]   Adrien Treuille[2,‡]   Jessica K. Hodgins[1,2,§]

[1]Disney Research, Pittsburgh   [2]Carnegie Mellon University

Figure 1: *Our method enables the fast animation of detailed garments for human characters. We demonstrate that by using a simple, efficient technique, the motion of the clothing (skirts, dresses, shirts) for a thousand or more characters can be realistically computed in real-time.*

Abstract

We present a technique for learning clothing models that enables the simultaneous animation of thousands of detailed garments in real-time. This surprisingly simple conditional model learns and preserves the key dynamic properties of a cloth motion along with folding details. Our approach requires no *a priori* physical model, but rather treats training data as a "black box." We show that the models learned with our method are stable over large time-steps and can approximately resolve cloth-body collisions. We also show that within a class of methods, no simpler model covers the full range of cloth dynamics captured by ours. Our method bridges the current gap between skinning and physical simulation, combining benefits of speed from the former with dynamic effects from the latter. We demonstrate our approach on a variety of apparel worn by male and female human characters performing a varied set of motions typically used in video games (*e.g.*, walking, running, jumping, *etc.*).

CR Categories: I.3.7 [Three-Dimensional Graphics and Realism]: Animation—; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

Keywords: cloth animation, character animation, virtual reality, cloth simulation, video games

1   Introduction

Clothing simulation can now produce stunningly realistic examples of detailed folding and rich knit textures. However, these approaches require high resolution meshes to represent fine detail, complex time-stepping methods to avoid instability, and expensive nonlinear solvers to resolve collisions. Therefore, real-time applications, such as animation prototyping, training simulations, and computer games, have relied on fast, low-dimensional, coarse models. Ideally, we would like to combine the benefits of low-dimensional representations with the expressive power of detailed cloth discretization. Interactive applications also require stability

---

*e-mail: edilson@disneyresearch.com

†e-mail: lsigal@disneyresearch.com

‡e-mail: treuille@cs.cmu.edu

§e-mail: jkh@disneyresearch.com

over long timesteps while maintaining collision constraints. Unfortunately, no existing cloth models satisfy these characteristics.

In this paper, we show that existing cloth simulations can be leveraged to create extremely fast *proxy* models of clothing with all of these properties. To build these models, we exploit two basic properties of interactive spaces. First, digital characters exist in low-dimensional, sometimes finite, configuration spaces such as linear skinning models or motion graphs. This property allows us to distill the output of complex simulations into compact linear models of the pose and clothing. Second, clothing collisions are dominated by body-cloth contact, not self-penetration, particularly for dynamic motions. This property allows us to build a surprisingly simple conditional model of the clothing configuration that preserves folding details. Unlike surface skinning approaches [James and Twigg 2005; Kavan et al. 2010], our method distinguishes between the clothing and the body state, enabling the cloth to move not only based on character pose, but also due to internal dynamics. Unlike model reduction approaches [Barbič and James 2005; Treuille et al. 2006], we assume no *a priori* physical model; instead, we learn a linear cloth update based on the pose, cloth history, and a small number of meta-parameters. We show that our method behaves surprisingly well even for highly non-linear phenomena such as sliding contact that would be difficult for pure model reduction to handle.

Our method has several advantages over previous simulation and learning approaches. First, it is *fast*: timesteps require three matrix multiplies, enabling the generation of over one-thousand detailed garments in real-time. Second, it is *stable*: we present a fundamental stability criterion that our method satisfies. Third, it *approximates collision handling* well. Fourth, it requires *no knowledge of physics*, instead treating the training simulator as a "black box." Finally, our model is *simple to implement* and achieves results superior to simpler ones while avoiding the overfitting of more complex models. We illustrate our model on a variety of garments and motions typically used in video games. In addition, we analyze, both quantitatively and qualitatively, the performance of our method as a function of parameter and modeling choices.

The ability to realistically animate clothing for a large number of characters in real-time has many potential applications. Virtual worlds increasingly rely on physical simulation, and our approach offers the opportunity to incorporate clothing. Because clothing greatly influences character appearance, our technique could assist in the accurate preview of keyframe animation or motion capture. In fact, with a sufficiently rich clothing database, this approach might offer a routine alternative to the limited range of "skintight" clothing styles allowed by traditional skinning models.

2 Related Work

The last decade has seen major advances in cloth simulation for computer graphics, and we provide only a brief introduction to the state of the art. A major trend has been the development of increasingly sophisticated models of cloth energy [Baraff and Witkin 1998; Grinspun et al. 2003] as well as understanding the geometric limits of these approaches [English and Bridson 2008; Goldenthal et al. 2007]. Maintaining stability has been addressed by developing special timestepping schemes, such as implicit integration [Baraff and Witkin 1998; Volino and Magnenat-Thalmann 2005], and a new approach to asynchronous timestepping [Harmon et al. 2009]. Another vital issue is cloth collisions, both with external objects [Cordier and Thalmann 2002; Bridson 2005; Vassilev et al. 2001], and with itself [Bridson et al. 2003; Baraff et al. 2003]. Our work builds upon this research by using a state-of-the-art cloth simulator [Stam 2009] as training data for our learning approach.

As off-line cloth simulation moves to millions of triangles [Selle et al. 2009], researchers have explored real-time approaches to cloth simulation [Rudomin and Castillo 2002], by, for example, coupling coarse simulation with geometric wrinkle models [Kang et al. 2001; Kang and Cho 2002] to add fine-scale details. Other research has accelerated cloth animation by mapping simulation methods to the graphics processing unit (GPU) to achieve a large constant factor speedup over CPU algorithms [Nguyen and Donnelly 2005; Vassilev et al. 2001]. In contrast to our approach, these GPU methods do not alter the simulation time complexity of their CPU counterparts.

Subspace methods represent an emerging alternative approach to animating complex phenomena by drastically reducing the number of parameters through linear dimension reduction. For example, meshes can be represented by a linear combination of basis vectors [James and Twigg 2005; Kavan et al. 2010], effectively decoupling the state size from the underlying geometric and dynamic complexity of the mesh.

Surface skinning methods (*e.g.*, [Kim and Vendrovsky 2008; Shi et al. 2008]) have also been used as an alternative to simulation and are amenable to fast real-time implementation in hardware [Kry et al. 2002]. However, these approaches are only able to model passive dynamic effects by either learning the parameters of the skinning from examples [Shi et al. 2008], or by applying secondary skinning operations based on local visco-elastic elements that control the behavior of the soft tissue [Larboulette et al. 2005]. Hence, unlike our approach, they are unsuitable for skirts and other garments whose state is determined by factors other than the pose of the character.

Model reduction can handle dynamical effects in linear low-dimensional models, but requires that the dynamics be known *a priori* [An et al. 2008; Barbič and James 2005; Treuille et al. 2006]. In our approach, by contrast, we do not assume that the dynamics are known, instead treating the entire dynamical system as a "black box" and learning a quasi-linear proxy to the dynamical system.

Some research has also looked into learning deformation models for data-driven simulations. Cordier and Magnenat-Thalmann [2005] augment a coarse cloth simulation with fine wrinkles generated through linear interpolation of precomputed, full-resolution cloth simulations. For tighter garments, an artist-directed animation of wrinkles was introduced by interpolating wrinkles at reference poses [Cutler et al. 2005].

James and Fatahalian [2003] use a subspace method and learn the dynamics of the system as a black box, as we do. While their method can handle arbitrary nonlinearities, they do so by tabulating a discrete transition table over a subspace of the configuration space. By contrast, we learn a continuous linear model, which is more generalizable and vastly more compact. Continuous autoregressive models have been proposed by Reissell and Pai [2001] to model interactive dynamic effects (*e.g.*, candle flame in the wind). While conceptually similar, their method relied on hand-specified low-dimensional input and output control signals. In contrast, we formulate our model in a learned linear subspace.

The learning and control theory communities have also extensively looked into learning dynamics, a problem often called *system identification* [Soderstrom and Stoica 1989]. For linear systems, various approaches to parameter learning have been proposed, including expectation maximization [Ghahramani and Hinton 1996] and gradient descent [Ljung 1986]. *Subspace methods* reduce the dimensionality of the system by learning a lower-dimensional linear subspace for the dynamics, often called the *hidden state* [Viberg 1995], which better conditions the learning process. We use the widely adopted least-squares approach [van Overschee and de Moor 1996] which reduces the learning step to a single matrix inversion.

When the dynamics are unknown, a learned model can be unstable, even if the underlying system is stable [Chui and Maciejowski 1996]. Various methods have been proposed to address this problem. Gestel and colleagues [2001] add a regularization term to the least-squares objective, Lacy and Bernstein [2003] constrain the largest eigenvalue using semi-definite programming, and recently Siddiqi and colleagues [2007] propose a convex optimization method that iteratively constructs a set of constraints on matrix stability. We have found that the models built with our technique are stable; we derive a measure of stability in Section 6.1 and show that all our learned models satisfy this criterion.

3 Overview

In general, the appearance of the clothing on a body is a function of (1) the body pose and the shape at a given moment in time, (2) the dynamics (*e.g.*, velocity, acceleration) of the cloth at that instant, and (3) the cloth geometry and material parameters (*e.g.*, whether it is silk, leather, *etc.*). This paper shows that by observing cloth behavior under physical simulation, we can learn a compact *proxy* model of the cloth's behavior that captures key dynamic properties and interactions with the body. We can then use this model in lieu of physical simulation to efficiently produce realistic dynamic cloth animations for virtual characters.

We start by simulating cloth on a skinned character animated by motion capture data. Based on simulation data, we learn a low-dimensional representation for both the cloth and the outer surface of the body by constructing two low-dimensional linear subspace models. The resulting low-dimensional latent spaces encapsulate the possible deformations of the cloth and the body respectively, while accounting for fine detail such as cloth folds. We then learn a conditional dynamical model of the cloth in the low-dimensional linear *cloth* space. The learned conditional dynamical model provides an efficient means of estimating the latent state of the cloth over time, based on the current latent state of the body, the history of past cloth states, and meta-parameters encoding the character's root motion in space. We show that the learned models are stable and are capable of generalizing across different motions and temporal executions at test time, distinct from the original training set. We also show that *simpler* models are inherently incapable of modeling important aspects of cloth motion, while more *complex* models lead to overfitting and unwarranted (in terms of performance) time complexity.

4 Low-dimensional Representation

Given the constrained topology of a particular piece of clothing and the way that piece of clothing fits on and moves with the body, the motion of individual facets of the mesh representing the clothing are not independent. This observation implies that the number of underlying degrees of freedom in the clothing is much lower than the number of vertices, and, we argue, largely independent of the resolution. Hence, we can capture most of the variation in the geometric appearance of the clothing with relatively few degrees of freedom. A typical approach to this problem is to learn a subspace or manifold model using dimensionality reduction.

There is a large body of literature for learning both parametric and non-parametric models for dimensionality reduction. While non-parametric models (*e.g.*, GPLVM [Lawrence 2005], Locally Linear Embedding [Roweis and Saul 2000], Isomap [Tenenbaum et al. 2000] and Stochastic Neighbor Embeddings [Hinton and Roweis 2002]) often tend to produce better performance, their memory representation requirements (a function of the training data size) and speed make them unattractive for our application. Instead, we opt for a linear parametric model in the form of Principal Component Analysis (PCA) [Hotelling 1933]. The parametric nature of PCA ensures that our representation is independent of the training set size; the linear nature of the model directly translates to computational efficiency. Furthermore, we tested non-linear dimensionality reduction methods, namely GPLVM and Kernel Regression, but saw no significant increase in the quality of the results.

The variations learned using the first few principal components for a dress are illustrated in Figure 2. We apply PCA on the vertices of the cloth mesh directly; however, one practical issue is the choice of the space in which these meshes are represented. A representation in the global space hinders generalization[1]. With this in mind, we model the cloth (and the body surface) in a canonical space defined with respect to the skeleton of the underlying character. To do so, we first transform all cloth (and body) meshes into a canonical space by: (1) subtracting the global position of the skeleton's root, $\mathbf{p}_t \in \mathbb{R}^3$ and (2) orienting the meshes so that they are always facing in one direction (aligning the hips of the skeleton with the x-axis, by rotating around the vertical axis by $r_t \in \mathbb{R}^1$).

Formally, the cloth mesh at time t, $m_t^{(c)} \in \mathbb{R}^{3N_c}$, represented by a series of N_c vertices $\in \mathbb{R}^3$, can be encoded by coefficients, $\mathbf{y}_t$, in the learned *cloth* PCA space spanned by M_c linear bases:

$$m_t^{(c)} = R_z(r_t)\left[\Lambda^{(c)}\mathbf{y}_t + \mu^{(c)}\right] + \mathbf{p}_t, \qquad (1)$$

where $R_z(r)$ is the global 3×3 rotation matrix that transforms the mesh from the canonical coordinate frame to the world, $\Lambda^{(c)} \in \mathbb{R}^{3N_c \times M_c}$ are the learned bases obtained by singular value decomposition (SVD), and $\mu^{(c)}$ is the mean cloth computed over the entire simulated dataset.

Similarly, the outer surface of the body at time t, $m_t^{(b)} \in \mathbb{R}^{3N_b}$, represented by a series of N_b vertices, can be encoded by coefficients, $\mathbf{x}_t$, in the learned *body* PCA space spanned by M_b linear bases:

$$m_t^{(b)} = R_z(r_t)\left[\Lambda^{(b)}\mathbf{x}_t + \mu^{(b)}\right] + \mathbf{p}_t, \qquad (2)$$

---

[1]For example, if we only observe the behavior of the dress as the person moves along the x-axis, we will not be able to handle the same motion in the y-direction at test time.
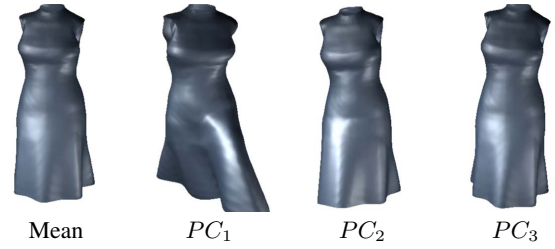
Figure 2: *PCA model for the dress: we show the cloth corresponding to the mean and the $+\sigma$ (where σ is the standard deviation) along each of the first three principal components.*

where $\Lambda^{(b)} \in \mathbb{R}^{3N_b \times M_b}$, and $\mu^{(b)}$ is the mean body computed over the dataset of skinned meshes in the canonical space.

5 Modeling Motion

Now that we have a low-dimensional representation for the cloth, we address the issue of estimating the state, $\mathbf{y}_t$, of this model over time such that the reconstructed high-dimensional cloth motion exhibits appropriate dynamics for a given motion of the body. The observation we make here is that the state of the cloth should be driven by two processes, (1) a conditional kinematic process that serves as *control* signal and (2) the internal dynamics of the cloth. Intuitively, (1) can be thought of as a steady-state appearance of the cloth once it settles on the body and (2) is the dynamic component that accounts, for example, for the residual swing of the cloth in a walk followed by an abrupt stop.

We start by introducing a pure conditional kinematic model (Model $\mathcal{A}$) that is only capable of accounting for motions of the cloth that do not exhibit a significant dynamic component (*e.g.*, very slow motions with heavy fabric). We then extend this model by introducing our latent linear dynamical system (Model $\mathcal{B}$) to model dynamic behavior of cloth in more realistic dynamic scenarios. Finally, we augment model $\mathcal{B}$ with additional conditional dynamic terms that result from un-modeled *global* motion of the character in the world (encoded by meta-parameters). This last addition gives the final form of our model that we refer to as Model $\mathcal{C}$.

5.1 Conditional Kinematics ($\mathcal{A}$)

The conditional kinematic process, in general, can be formulated as a regression problem, where given a state of the body, $\mathbf{x}_t$, we are interested in a function, $\mathbf{y}_t = f(\mathbf{x}_t)$, that would map that state to the corresponding state of cloth, $\mathbf{y}_t$. In the simplest form, this function can take the form of Nearest Neighbor regression, where given a query latent representation for the body, the closest corresponding body mesh (*e.g.*, in terms of average vertex distance) from the database can be found and the corresponding cloth state returned. In our experiments, however, this approach proved both noisy – due to the intractable size of the database required for fine resolution retrieval, and expensive – because the NN method needs to compare the query to every training exemplar in the database. The earlier issue can be avoided by Kernel Regression techniques that, in essence, average resulting cloth state over the set of nearest neighbors. The latter can be addressed using approximate nearest neighbor methods. However, this approach would still require storage that is proportional to the number of exemplars in the database. Such a solution would be impractical for models that span multiple motions and activities.

To facilitate faster real-time performance and limited storage requirements (making the model amenable to future GPU implemen-

tations) we opt for an efficient linear (parametric) regression model. We have also tested low-dimensional non-linear models (*e.g.*, variants of Shared GPLVM) but found them to perform similarly, requiring higher processing times and limiting the amount of training data.

The resulting conditional model can be expressed compactly using a single matrix multiplication:

$$\mathbf{y}_t = A\mathbf{x}_t, \tag{3}$$

where $A \in \mathbb{R}^{M_c \times M_b}$ is the matrix of learned regression coefficients obtained by minimizing a least-squares objective, and M_c and M_b are dimensions of the two latent spaces representing the cloth and the surface of the body respectively.

5.2 Conditional Latent Linear Dynamical Systems ($\mathcal{B}$)

Given a conditional model of the cloth, we now address the modeling of the residual dynamics that cannot be accounted for by the conditional kinematic model. For example, consider a case of a woman abruptly stopping in mid-stride, as depicted in Figure 3. While the pose and the surface of her body stops, her garment will continue to move. These are precisely the effects that the current skinned models cannot handle.

We express such stationary dynamics of the cloth in the latent space using a 2-nd order linear dynamical system (LDS) [Kalman 1960]. The model of this form facilitates smoothness and continuity in the motion of the cloth and can account for the dynamic effects such as the one discussed above. Formally, 2-nd order LDS assumes that the state of cloth at time t can be expressed as a linear combination of the states at time $t-1$ and $t-2$. The conditional kinematics can be interpreted, in this context, as a time-varying bias, resulting in:

$$\mathbf{y}_t = A\mathbf{x}_t + B_1\mathbf{y}_{t-1} + B_2\mathbf{y}_{t-2}, \tag{4}$$

where $B_i \in \mathbb{R}^{M_c \times M_c}$ are matrices of coefficients to be learned.

This formulation can be trivially extended to the N-th order LDS, but in our experiments, we demonstrate that a 2-nd order model performs best; with 0-th (Model $\mathcal{A}$) and 1-st order models providing inferior performance, and 3-rd, 4-th and 5-th order models adding complexity without noticeably improving performance (and in fact suffering from overfitting).

The formulation in Eq. 4 can further be interpreted probabilistically as a Kalman filter [Kalman 1960] designed to infer the state of the cloth over time, under Gaussian assumptions on the noise of the emission and transition. This interpretation relates our model to a rich literature on state-estimation in control theory [Soderstrom and Stoica 1989], and allows for a variety of learning procedures.

5.3 Residual Dynamics ($\mathcal{C}$)

Because we learn to model cloth and dynamics in the canonical space, some of the dynamics that are due to the change in the global position and heading of the body are left un-modeled. Consider a person turning left or right, for example. In this case the Model $\mathcal{B}$ described above would be able to model the forward swing of the dress, but not be able to model the twist in the appropriate direction (precisely because we normalize our representation with respect to the heading of the body). To account for this omission, we further condition the dynamics on the relative motion of the root (along the degrees of freedom that can not be accounted for in the canonical space). For the 2-nd order model, we condition on the history over

Model $\mathcal{A}$ **Model $\mathcal{C}$**

Figure 3: *A simple conditional kinematics model (left) is not able to account for the residual motion of the dress when the character abruptly stops; a proposed* 2*-nd order LDS model can, on the other hand, reproduces the desired residual motion.*

the past two frames. This change adds two conditional terms to Eq. 4 above, obtaining the final model:

$$\mathbf{y}_t = A\mathbf{x}_t + B_1\mathbf{y}_{t-1} + B_2\mathbf{y}_{t-2} + C_1\mathbf{z}_{t,t-2} + C_2\mathbf{z}_{t-1,t-2} \tag{5}$$

where $C_i \in \mathbb{R}^{M_c \times 5}$ are matrices of coefficients to be learned, and $\mathbf{z}_{t,j}$ are the meta-parameters encoding the relative position and heading of the root at frame t with respect to frame j:

$$\mathbf{z}_{t,j} = \begin{bmatrix} R_z(-r_j)\Delta\mathbf{p}_t \\ sin(\Delta r_t) \\ cos(\Delta r_t) \end{bmatrix} \tag{6}$$

where $\Delta\mathbf{p}_t \equiv \mathbf{p}_t - \mathbf{p}_j$, $\Delta r_t \equiv r_t - r_j$.

Notice that the representation of the cloth remains in the canonical space, but the dynamics are now conditioned on the meta-parameters (corresponding to the relative position and heading of the root).

6 Learning

Given the additive form of our model, the simultaneous learning of all parameters $[A, B_1, B_2, C_1, C_2]^T$ (the same is true for models $\mathcal{A}$ and $\mathcal{B}$ introduced earlier, and models with lower or higher order dynamics) can be formulated simply by minimizing the squared error between the observed and predicted value for $\mathbf{y}_t$ for a dataset $\mathcal{D}$ consisting of temporal sequences of length 3 (or length N, for a N-th order dynamical model):

$$\min_{A, B_1, B_2, C_1, C_2} \sum_{\mathcal{D}} \left\| \mathbf{y}_t - \begin{bmatrix} A^T \\ B_1^T \\ B_2^T \\ C_1^T \\ C_2^T \end{bmatrix}^T \begin{pmatrix} \mathbf{x}_t \\ \mathbf{y}_{t-1} \\ \mathbf{y}_{t-2} \\ \mathbf{z}_{t,t-2} \\ \mathbf{z}_{t-1,t-2} \end{pmatrix} \right\|_2^2.$$

This formulation is a least-squares problem and can be solved by standard techniques.

Once the parameters are learned, the state of the cloth can be predicted simply by conditioning on the latent parameterization of the body, previous cloth states, and the relative position and heading of the skeleton's root. The predictions in the canonical space can then be transformed into the world space for visualization. For initialization, because we typically do not have estimates for $\mathbf{y}_t$ and $\mathbf{y}_{t-1}$, we use a 0-th order model to bootstrap the process for the first two frames, at which point we switch to the 2-nd order model.

6.1 Stability

Interactive simulation requires stability guarantees. Because our model is linear, its stability can be verified using eigen-analysis [Siddiqi et al. 2007] which shows that all models learned by our system are stable in that there is an energy threshold $\bar{E}$ above which the energy will always be driven downwards: $E_t < E_{t-1}$. To show this property, we rewrite equation (5) as

$$\left[\begin{array}{c} \mathbf{y}_t \\ \mathbf{y}_{t-1} \end{array} \right] = M \left[\begin{array}{c} \mathbf{y}_{t-1} \\ \mathbf{y}_{t-2} \end{array} \right] + \mathbf{k}_{t-1} \qquad (7)$$

where

$$M = \left[\begin{array}{cc} B_1 & B_2 \\ I & 0 \end{array} \right]$$

encodes the linear dynamics and $\mathbf{k}_{t-1}$ accounts for all the additional terms in (5). Letting $Q\Lambda Q^{-1} = M$ be the eigendecomposition of M, we can substitute $\mathbf{q}_t = Q^{-1} \left[\mathbf{y}_t^T, \mathbf{y}_{t-1}^T \right]^T$ and rewrite (7) as:

$$\mathbf{q}_t = \Lambda \mathbf{q}_{t-1} + Q^{-1}\mathbf{k}_{t-1}. \qquad (8)$$

Given this form, it is natural to define the energy in terms of $\mathbf{q}$ as $E_t \equiv ||\mathbf{q}_t||$. We can now use the triangle inequality to rewrite (8) as

$$E_t \leq |\lambda| E_{t-1} + k_{\max}, \qquad (9)$$

where λ is the largest eigenvalue of Λ, and $k_{\max}$ is a bound on $||Q^{-1}\mathbf{k}||$ imposed by the limits on rotation, translation, and valid human pose. Straightforward algebraic manipulation of (9) shows that as long as $\lambda < 1$, our model is stable (in the above sense) with energy threshold $\bar{E} = k_{\max}/(1 - \lambda)$. This analysis shows that every model learnt with our system is stable (Table 1) and no further stability enforcement has been necessary. However in principle, stability could be explicitly enforced by limiting the dimensionality of the latent space, or through explicit constraints (§2).

6.2 Depth Consistency

Our method is able to construct models that approximately maintain depth consistency. However, in some situations the cloth may penetrate the body (or vice-versa) for small periods of time due to abrupt changes in acceleration or severe articulations. Such effects could be reduced by increasing and/or balancing the training set. However, we ues a simple rendering solution that alleviates such problems and ensures pixel-level depth consistency in most cases.

Our solution consists of two steps. First, we pre-compute which regions of the human body will always be occluded by the garment. This step enables us to easily discard the fragments of the underlying human body that are always occluded for each character–clothing pair. In addition, to alleviating depth inconsistencies, this strategy is more efficient for rendering. Second, to render each frame, we first render the human model(s). After all underlying models are rendered, we save the depth map in a texture and use it inside a simple fragment shader to compare the model's stored depth value d_{body} to the incoming depth value for the cloth fragment d_{cloth}. Instead of discarding the fragment of the cloth if $d_{cloth} > d_{body}$, we use a small threshold ϵ_{depth} and discard the incoming cloth fragment only if $d_{cloth} + \epsilon_{depth} > d_{body}$. As seen in the accompanying video and in the results section, this rendering strategy enables us to not only properly render the resulting cloth animations, but also generalizes well to characters with multiple garments (*e.g.*, shirt and pants, see Figure 1 left).

| | Mesh Resolution | | Stability | Runtime ($sec/frame$) | | |
| | | | | Our Model | | |
	#Vert	#Tri	λ	Simul	Total	Maya
dress	766	1,456	0.9798	4.8e-08	0.00049	0.37
dress$_{3,410}$	3,410	6,664	0.9798	4.8e-08	0.00246	1.20
dress$_{5,178}$	5,178	10,172	0.9757	4.8e-08	0.00377	1.90
dress$_{20,530}$	20,530	40,688	0.9778	4.8e-08	0.01497	13.20
skirt1	820	1,572	0.8950	4.8e-08	0.00040	0.25
skirt2	820	1,572	0.9563	4.8e-08	0.00041	0.33
long skirt	766	1,456	0.9736	4.8e-08	0.00041	0.40
Oscar dress	820	1,582	0.9654	4.8e-08	0.00039	0.77
cape	2,242	4,296	0.9915	4.8e-08	0.00157	0.57
shirt	3,329	6,468	0.9983	4.8e-08	0.00245	0.76
pants	560	1,074	0.9735	4.8e-08	0.00020	2.36
poncho	2,450	4,692	0.9995	4.8e-08	0.00169	0.80

Table 1: *The list of garments tested and their resolution, stability criterion for all models, and comparison of run-time between our method and Maya cloth (i.e., cloth simulator). Notice that all the models are stable based on the $\lambda < 1$ criterion derived in Section 6.1. Our models are also three orders of magnitude faster than standard Maya cloth simulator. For our method we report the simulation time (Eq. 5) and the total time (which includes reconstruction of the mesh from low dimensional representation); the simulation time is constant because all our models live in 64-dimensional subspace. The total time can further be improved by implementing the reconstruction on a GPU. All timings were obtained on a desktop machine with Intel® Core™ 2 Quad 3GHz CPU with 8Gb of memory and NVIDIA Quadro FX 3800 graphics card.*

7 Experiments and Results

We evaluated the proposed method both quantitatively and qualitatively. We explore how the performance of this model is affected by choices in parameters and structure. We show that the proposed model is effective in modeling the dynamic behavior of clothing for a variety of motion capture sequences; we further show that *simpler* models inherently do not capture some important aspects of the clothing motion.

Datasets. We collected a training set of 33 atomic motion sequences (performed by a single male subject), using a Vicon motion capture system, consisting of: *walking* and *running* at three different speeds; *turns* at $45°$, $90°$, $180°$ and $360°$; *locomotion with stops*; *ducking*; *jumping/leaping* to avoid obstacles and some *transitions* between these motions. We down-sampled all motion capture data from 120 Hz to 30 Hz. We can simulate with a much longer timestep then most cloth methods, yet our system remains stable. The set of motions collected was designed to be appropriate for constructing a motion graph consistent with the control of a typical character in a game.

As test data, we collected a separate set of 15 motion capture sequences (up to 45 seconds each) from the same subject. The test set consisted of arbitrary combinations of the motions from the training set performed freely by the subject and at different speeds. The test set contains a total of 9,881 frames of motion capture data. Our training and test sets are completely disjoint. We also collected two additional sequences consisting of atomic motions (walk and walk with a turn) to illustrate the ability of our method to handle a rotational motion properly. Additionally a set of test sequences was collected containing motions not in our training set to illustrate the ability of our method to generalize, as well as to verify the limits of the performance of the proposed method.

Training. We train one model across all motions in the training set for each garment. We model four different garments for a male

character and four for a female character. The list of garments and their resolutions are given in Table 1. To test the dependence of our method on mesh resolution, we construct four variants of the same *dress* by sub-dividing the original model. We also construct two models for the skirt that have the same geometry but different cloth parameters. We train our model in two steps. First, PCA latent space models for clothes and body are trained with $6,609$ meshes for the body (obtained by skinning a character in Maya) and the corresponding clothing (produced by simulating the clothing in Maya). Second, we train the motion model with an extended set that we produce by stopping each training motion five times at different locations uniformly (and simulating once again); resulting in a semi redundant dataset of $5 \times 6,609 \approx 33,045$ sequences of length 2 (or N for N-th order model). This two-step training takes on average 1.5 hours for our models on a single quad-core machine and is dominated by the SVD computation in PCA.

We utilize average Euclidean vertex distance as the metric of performance in our quantitative experiments. While this measure is not perceptually faithful, lower error typically corresponds to visually better performance, and allows us to quantitatively measure the performance of our approach. We report all errors by averaging across the entire test or training set respectively. All errors are in cm.

7.1 Quantitative Experiments

We first quantitatively explore the behavior of our model as a function of the parameter and the modeling choices. For convenience and brevity, we focus on the *dress* for all quantitative experiments, but the same trends can be observed on all of the garments that we tested.

We first explore how the dimensionality of the PCA space affects the performance (Table 2). While performance on the training set improves monotonically with dimensionality (as one would expect), the performance on the test set clearly indicates that the method starts to overfit in higher dimensions for all models. It also appears that model $\mathcal{B}$ is more prone to overfitting than either $\mathcal{A}$ or $\mathcal{C}$. In lower dimensional spaces, modeling of dynamics leads to inferior performance for models $\mathcal{B}$ and $\mathcal{C}$, but clearly improves the performance once the latent-space dimension is appropriate for the exhibited class of motions. Qualitatively, models in the lower dimensional space damp out the motion of the cloth, making it look "stiff", as demonstrated in the accompanying video.

Second, we look at the ability of our model to represent rotational motions. We do so by measuring the error on two additional atomic motion sequences (not part of our test set): a *walk* and a *walk with a turn*[2]. We find that the error for the latter is actually marginally lower than for the first in all cases (2.24 versus 2.07 using a 64 dimensional model). This result suggests that we are able to model the rotational swing of the dress. Furthermore, for such simple (walking) motions the best performance is actually observed with a 32 dimensional model, suggesting that more compact models may be appropriate for datasets that contain simpler or fewer motions.

Third, we look at the importance of the dynamics and the order of the dynamical model. As with the dimensions, from Table 3, we can see that the performance on the training set is monotonically increasing, but shows signs of overfitting for the dynamics of orders three or more. The best performance is achieved with the 2nd order model $\mathcal{C}$ (with about 5% improvement over the model that does not contain any dynamics – 0th order, Model $\mathcal{A}$). This result is perhaps not surprising, given that actual cloth dynamics in the high dimensional space can be expressed as the 2-nd order differential

---

[2]Subject walks, stops and then turns the upper body without lifting the feet.

		Dimension of the Latent PCA Space						
		10	16	32	50	**64**	80	128
Train	Model $\mathcal{A}$	2.26	1.90	1.46	1.32	1.26	1.21	1.14
	Model $\mathcal{B}$	2.89	2.24	1.50	1.26	1.17	1.10	1.00
	Model $\mathcal{C}$	2.86	2.20	1.36	1.09	1.01	0.96	0.87
Test	Model $\mathcal{A}$	2.37	2.04	1.51	1.40	1.37	1.32	1.44
	Model $\mathcal{B}$	2.99	2.48	1.62	1.43	1.36	1.37	1.56
	Model $\mathcal{C}$	3.33	2.58	1.59	1.38	**1.32**	1.34	1.49
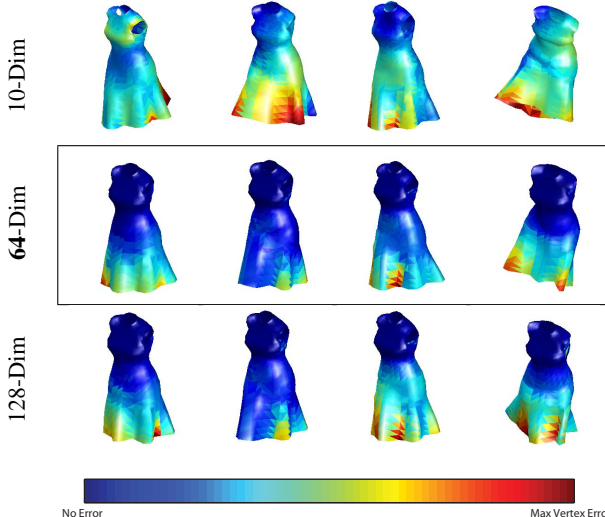
Table 2: *Performance of the 2-nd order model (Model $\mathcal{C}$), the simpler 2-nd order variant (Model $\mathcal{B}$) and the pure conditional model ($\mathcal{A}$) as a function of the dimensionality of the PCA space representing the cloth and the body. The best performance on the test set, corresponding to the proposed model, is in bold. Bottom rows visually illustrate per vertex error (as compared to the Maya cloth simulator) for four representative meshes obtained using $\mathcal{C}$ with 10, 64 and 128 dimensions; larger errors are illustrated with hot colors (red), and lower error with colder colors (blue).*

equation and the linear projection should not alter this 2-nd order relationship. Also note the importance of the meta-parameters in our formulation. The second column clearly shows that these parameters are very important in achieving the desired performance.

Fourth, we explore how the resolution of the clothing impacts the performance of the learned model. Table 4 reports the performance with the same dress but under 4 different mesh resolutions (obtained by sub-division). In all cases, we are able to produce realistic motions; however, quantitatively the results degrade marginally with resolution as we are not able to capture additional local structure (details) with the same dimensionality of the model. This comparison is not entirely fair, however, with respect to our model, because the simulation parameters in Maya are not dimension-less, and hence the dynamics is slightly different at different resolutions.

7.2 Qualitative Performance

We also qualitatively evaluate the performance of our method by animating and rendering virtual characters in real-time. We are able to animate characters performing a variety of complex motions with each of the garments, or combinations of them (*e.g.*, shirt and a skirt). Representative results from our method are illustrated in Figures 1, 5 and 6, as well as in the accompanying video.

Figure 4 (left) illustrates the ability of our method to produce fine detail, such as folds in the shirt that appear as the character starts to turn. We also explored the ability of our model to capture dynamics

Model	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$					
Dynamics	0th	2nd	0th	1st	**2nd**	3rd	4th	5th
Train	1.26	1.17	1.26	1.25	1.01	0.99	0.97	0.96
Test	1.37	1.36	1.37	1.53	**1.32**	1.39	1.37	1.40

Model $\mathcal{A}$ Model $\mathcal{B}$ Model $\mathcal{C}$
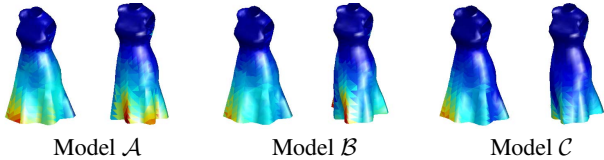
Table 3: *We illustrate the effect of the order in the dynamical model on the* dress *dataset. While performance on the training data monotonically increases, the performance on the test set asymptotes at the 2nd order model for Model $\mathcal{C}$. As in Table 2, the bottom row illustrates per vertex error for two meshes with models $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ (for $\mathcal{B}$ and $\mathcal{C}$ with 2-nd order dynamics).*

Model	$\mathcal{C}$ (2nd order dynamics, 64-dim)			
Garment	dress	$\text{dress}_{3,410}$	$\text{dress}_{5,178}$	$\text{dress}_{20,530}$
Test	1.32	1.58	1.70	1.66

Table 4: *We illustrate the effect of the mesh resolution on the* dress *dataset. The error increases marginally with increased resolutions, but then asymptotes.*

specific to a given fabric. To do so, we produced two datasets by simulating the same cloth geometry with different cloth parameters in Maya (see skirt1 and skirt2 in Table 1) and trained two separate models. The results are illustrated in Figure 4 (right). As it is visible in the figure and accompanying video the two models do capture the subtle differences in the original dynamics.

In the video, we also compare the results obtained by our model to the standard bone skinning produced by a skilled artist in Maya. Our Model $\mathcal{A}$ (*not* Model $\mathcal{C}$) can actually be interpreted as a generalized form of skinning [James and Twigg 2005; Kry et al. 2002]. Unlike traditional bone skinning methods, however, it does not assume that vertex positions are affected only by neighboring bones, but rather allows a vertex to be affected by any bone in the body; furthermore, our model also encodes an explicit prior over the plausible resulting *skinned* meshes. These differences can account for the better performance of our Model $\mathcal{A}$ with respect to Maya skinning; Model $\mathcal{C}$ introduces an additional dynamical component and further improves the performance.

We also test the ability of our method to generalize to motions outside of the training set. In the video, we illustrate how the learned model can generalize to different execution speeds and styles (*e.g.*, clothing for a stylized walk or a kick can be realistically rendered, while neither were part of the training dataset); our method is also able to deal with transitions between the motions in the training set (*e.g.*, walking to running). However, more severe deviations from the training set (*e.g.*, dancing), that fall well outside the realm of articulations and dynamics observed, could lead to artifacts.

Runtime Performance.   Because our method at runtime amounts to three[3] compact matrix multiplies, it can easily be implemented on parallel hardware architectures, such as a GPU. In particular, because the state of the body and the cloth are represented with few parameters in the latent space, the communication to the GPU can be minimized, reducing overhead. Our current implementation, however, does not take advantage of this fact and computes

---
[3]One to project the skinned body into the latent space, one to infer the state of the cloth from the latent body representation, and one to re-project the low-dimensional cloth state back to the high dimensional mesh.
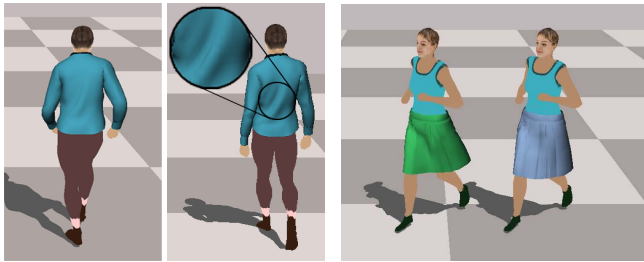
Figure 4: *The ability of our model to reproduce wrinkles and folds in the cloth is illustrated on the left. We are also able to train separate models to reconstruct different materials as illustrated on the right; notice the higher amplitude in the swing of the green skirt, relative to the blue skirt that had higher damping and thickness.*

**Figure 5:** *Types of clothes handled by our method.*

the models entirely on the CPU; subsequently passing the produced geometries to the GPU for rendering. Yet, we are still able to achieve three orders of magnitude speedup over Maya cloth simulation (see Table 1), and even with this naive implementation can animate about 170 pieces of clothes in real-time (including rendering); we are able to synthesize over $1,000$ models in real-time (without rendering). Machine specifications are given in Table 1.

8   Discussion

We present a learning-based approach to model the dynamic behavior of clothing. Our conditional linear model is efficient, yet allows us to model realistic and detailed clothing behaviors including folding. We illustrate the performance of our method on a variety of garments and show that the learned models produce realistic clothing behaviors for a diverse set of motions. We further analyze the stability of our models and illustrate that they are indeed stable. Finally, we show that within a linear class of methods, no simpler model can handle the full range of cloth dynamics which we capture, by comparing the performance of our method to a number of alternatives.

While our model is powerful, it does have a number of limitations. First, we treat physical constraints as a black box, and learn the physical interactions and phenomena only from the data. While this formulation allows us to have a simple and efficient model, it does hinder generalization. For example, our method may not generalize to physical phenomena not observed in the training set (*e.g.*, external forces like wind, changes in gravitational constant, *etc.*). Second, our approach does not model (at least explicitly) self collisions of the cloth, making it hard to apply it in scenarios where the body is significantly contorted or the clothing is highly dynamic. We also found abrupt changes in accelerations, particularly at joint limits, to be challenging for our model.

Our method approximately preserves depth consistency, but does not guarantee it. We add a rendering method for fixing the depth consistency in situations where there is a "near miss" and a slight
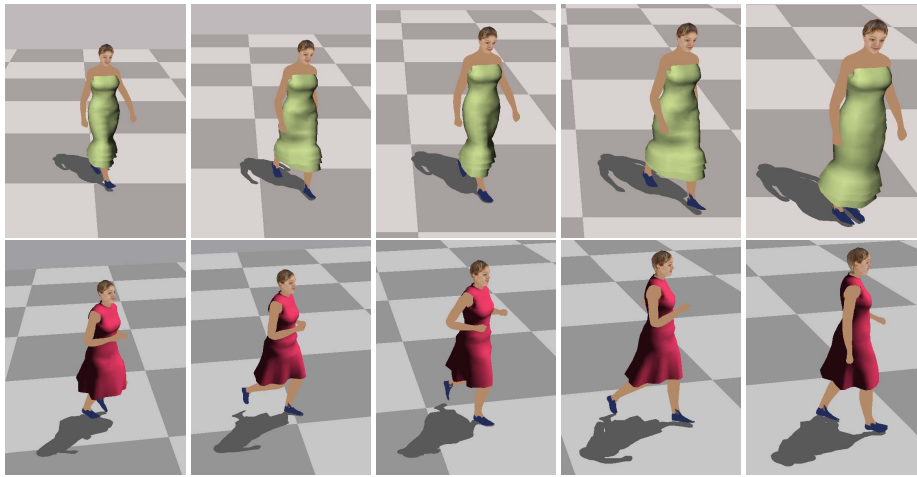
Figure 6: *Dynamic cloth motions recovered by our approach, for the Oscar dress (top) and for the normal dress (bottom).*

inter-penetration occurs. In our experience, this works well in most situations within the space of human motions and clothing styles where our key assumptions hold, but it will fail when the motions are highly dynamic or the cloth is essentially unconstrained by the motion of the body.

We are able to deal effectively with a variety of fairly loose clothing styles (untucked shirt, A-line skirt, board shorts). While this approach does not allow us to satisfactory handle very loose clothing, like a cape or a poncho, in all cases, it does allow us to deal with much of the clothing worn by normal people on a daily basis as well as much of that seen in current video games. We believe that there is a significant and important space of dynamic clothing motions between the almost skin tight clothing, that can be handled by skinning algorithms, and the unrestricted motion of very loose clothing, that must be directly simulated.

There is no guarantee that our learning technique will produce a stable model, although it has for all the examples we tested. If the learned model turns out to be unstable, we have several potential solutions. One would be to iteratively decrease the number of dimensions in the latent space until stability is achieved. Another alternative would be to use a method for learning inherently stable LDS [Lacy and Bernstein 2003; Siddiqi et al. 2007; Van Gestel et al. 2001]. It is our belief that a relatively large amount of data and the least-squares form of simultaneous learning for all the parameters are the factors that contribute to the stability of the learned models in practice.

Despite the aforementioned limitations, our approach is ideal for fast prototyping and games where real-time performance takes priority over time-consuming physical simulations. As it stands, it can also be applied to dynamically controlled characters where, for example, the speed or execution style for the motion of the virtual character may be altered by the user within some tolerable bounds. Our method is also amenable to other problems in computer graphics. For example, it could be extended to modeling non-rigid dynamic deformations of the outer body surface based on skeletal motion (*e.g.*, jiggling of the muscles or fat tissue). We also believe it may be applicable for fast hair animation. Finally, our model could accelerate high-resolution accurate cloth solvers by seeding each step of an implicit method with an approximate first guess.

We plan to explore extensions to our method that would allow us to parameterize the learned models with additional parameters such as material properties. While we have shown that we can learn models for a discrete set of materials, the current formulation is not able to interpolate among materials to achieve novel appearances; to do so, we are interested in looking at multi-linear models. We also want to look at explicit collision maintenance with low-dimensional models.

References

AN, S., KIM, T., AND JAMES, D. 2008. Optimizing cubature for efficient integration of subspace deformations. *ACM Transactions on Graphics 27*, 5.

BARAFF, D., AND WITKIN, A. P. 1998. Large steps in cloth simulation. *ACM Transactions on Graphics*, 43–54.

BARAFF, D., WITKIN, A., AND KASS, M. 2003. Untangling cloth. *ACM Transactions on Graphics 22*, 3, 862–870.

BARBIČ, J., AND JAMES, D. L. 2005. Real-time subspace integration for st. venant-kirchhoff deformable models. *ACM Transactions on Graphics 24*, 3, 982–990.

BRIDSON, R., MARINO, S., AND FEDKIW, R. 2003. Simulation of clothing with folds and wrinkles. In *ACM/Eurographics Symposium on Computer Animation*, 28–36.

BRIDSON, R. 2005. Cloth collisions and contact. *ACM SIGGRAPH 2005 Course Notes*.

CHUI, N. L. C., AND MACIEJOWSKI, J. M. 1996. Realization of stable models with subspace methods. *Automatica 32*, 11, 1587–1595.

CORDIER, F., AND MAGNENAT-THALMANN, N. 2005. A data-driven approach for real-time clothes simulation. In *Pacific Conference on Computer Graphics and Applications*, vol. 24, 257–266.

CORDIER, F., AND THALMANN, N. M. 2002. Real-time animation of dressed virtual humans. In *Computer Graphics Forum*, vol. 21, 327–335.

CUTLER, L., GERSHBEIN, R., WANG, X., CURTIS, C., MAI-GRET, E., AND PRASSO, L. 2005. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*.

ENGLISH, E., AND BRIDSON, R. 2008. Animating developable surfaces using nonconforming elements. *ACM Transactions on Graphics 27*, 3, 66:1–66:5.

GHAHRAMANI, Z., AND HINTON, G. E. 1996. Parameter estimation for linear dynamical systems. Tech. Rep. CRG-TR-96-2, University of Toronto.

GOLDENTHAL, R., HARMON, D., FATTAL, R., BERCOVIER, M., AND GRINSPUN, E. 2007. Efficient simulation of inextensible cloth. *ACM Transactions on Graphics 26*, 3, 49.

GRINSPUN, E., HIRANI, A. N., DESBRUN, M., AND SCHRĆDER, P. 2003. Discrete shells. In *2003 ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 62–67.

HARMON, D., VOUGA, E., SMITH, B., TAMSTORF, R., AND GRINSPUN, E. 2009. Asynchronous contact mechanics. *ACM Transactions on Graphics 28*, 3.

HINTON, G., AND ROWEIS, S. 2002. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems*, vol. 15, 833–840.

HOTELLING, H. 1933. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology 24*, 417–441.

JAMES, D. L., AND FATAHALIAN, K. 2003. Precomputing interactive dynamic deformable scenes. *ACM Transactions on Graphics 22*, 3, 879–887.

JAMES, D. L., AND TWIGG, C. D. 2005. Skinning mesh animations. *ACM Transactions on Graphics 24*, 3, 399–407.

KALMAN, R. 1960. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering 82*, Series D, 35–45.

KANG, Y.-M., AND CHO, H.-G. 2002. Bilayered approximate integration for rapid and plausible animation of virtual cloth with realistic wrinkles. In *Proceedings of Computer Animation*, 203–211.

KANG, Y.-M., CHOI, J.-H., CHO, H.-G., AND LEE, D.-H. 2001. An efficient animation of wrinkled cloth with approximate implicit integration. *The Visual Computer 17*, 3, 147–157.

KAVAN, L., SLOAN, P.-P., AND O'SULLIVAN, C. 2010. Fast and efficient skinning of animated meshes. *Computer Graphics Forum 29*, 2.

KIM, T.-Y., AND VENDROVSKY, E. 2008. Drivenshape - a data-driven approach for shape deformation. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*.

KRY, P. G., JAMES, D. L., AND PAI, D. K. 2002. Eigenskin: real time large deformation character skinning in hardware. In *ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, 153–159.

LACY, S., AND BERNSTEIN, D. 2003. Subspace identification with guaranteed stability using constrained optimization. *IEEE Transactions on Automatic Control 48*, 7, 1259–1263.

LARBOULETTE, C., CANI, M.-P., AND ARNALDI, B. 2005. Dynamic skinning: Adding real-time dynamic effects to an existing character animation. In *Spring Conference on Computer Graphics (SCCG)*.

LAWRENCE, N. 2005. Probabilistic non-linear principal component analysis with gaussian process latent variable models. *Journal of Machine Learning Research 6*, 1783–1816.

LJUNG, L. 1986. *System identification: theory for the user*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

NGUYEN, H., AND DONNELLY, W. 2005. Hair animation and rendering in the nalu demo. In *GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation (Gpu Gems)*, M. Pharr and R. Fernando, Eds. Addison-Wesley Professional, ch. 23.

REISSELL, L., AND PAI, D. 2001. Modeling stochastic dynamical systems for interactive simulation. In *Computer Graphics Forum*, vol. 20, 339–348.

ROWEIS, S. T., AND SAUL, L. K. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 5500, 2323–2326.

RUDOMIN, I., AND CASTILLO, J. L. 2002. Real-time clothing: Geometry and physics. *WSCG 2002 Posters*, 45–48.

SELLE, A., SU, J., IRVING, G., AND FEDKIW, R. 2009. Robust high-resolution cloth using parallelism, history-based collisions, and accurate friction. *IEEE Transactions on Visualization and Computer Graphics 15*, 2, 339–350.

SHI, X., ZHOU, K., TONG, Y., DESBRUN, M., BAO, H., AND GUO, B. 2008. Example-based dynamic skinning in real time. *ACM Transactions on Graphics 27*, 3, 1–8.

SIDDIQI, S., BOOTS, B., AND GORDON, G. 2007. A constraint generation approach to learning stable linear dynamical systems. In *Advances in Neural Information Processing Systems*.

SODERSTROM, T., AND STOICA, P. 1989. *System Identification*. Prentice-Hall, Upper Saddle River, NJ.

STAM, J. 2009. Nucleus: Towards a unified dynamics solver for computer graphics. In *IEEE International Conference on Computer-Aided Design and Computer Graphics*, 1–11.

TENENBAUM, J., DE SILVA, V., AND LANGFORD, J. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 5500, 2319–2323.

TREUILLE, A., LEWIS, A., AND POPOVIĆ, Z. 2006. Model reduction for real-time fluids. *ACM Transactions on Graphics 25*, 826–834.

VAN GESTEL, T., SUYKENS, J., VAN DOOREN, P., AND DE MOOR, B. 2001. Identification of stable models in subspace identification by using regularization. *IEEE Transactions on Automatic Control 46*, 9, 1416–1420.

VAN OVERSCHEE, P., AND DE MOOR, B. L. R. 1996. *Subspace identification for linear systems: theory, implementation, applications*. Springer.

VASSILEV, T., SPANLANG, B., AND CHRYSANTHOU, Y. 2001. Fast cloth animation on walking avatars. *Computer Graphics Forum 20*, 3, 260–267.

VIBERG, M. 1995. Subspace-based methods for the identification of linear time-invariant systems. *Automatica 31*, 12, 1835–1851.

VOLINO, P., AND MAGNENAT-THALMANN, N. 2005. Implicit midpoint integration and adaptive damping for efficient cloth simulation: Collision detection and deformable objects. *Computer Animation and Virtual Worlds 16*, 3-4, 163–175.