Subspace Clothing Simulation Using Adaptive Bases

Fabian Hahn[1,2] Bernhard Thomaszewski[2] Stelian Coros[2] Robert W. Sumner[2] Forrester Cole[3]
Mark Meyer[3] Tony DeRose[3] Markus Gross[1,2]

[1]ETH Zurich [2]Disney Research Zurich [3]Pixar Animation Studios

Pixar Technical Memo #14-03

Figure 1: *Example result of our method: A tight-fitting sweater exhibits wrinkles and torsoidal folds under the effects of gravity and as the underlying torso is twisting. This example used only 12 adaptively chosen basis vectors and ran 18 times faster than a full simulation.*

Abstract

We present a new approach to clothing simulation using low-dimensional linear subspaces with temporally adaptive bases. Our method exploits full-space simulation training data in order to construct a pool of low-dimensional bases distributed across pose space. For this purpose, we interpret the simulation data as offsets from a kinematic deformation model that captures the global shape of clothing due to body pose. During subspace simulation, we select low-dimensional sets of basis vectors according to the current pose of the character and the state of its clothing. Thanks to this adaptive basis selection scheme, our method is able to reproduce diverse and detailed folding patterns with only a few basis vectors. Our experiments demonstrate the feasibility of subspace clothing simulation and indicate its potential in terms of quality and computational efficiency.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

Keywords: physically-based simulation, subspace simulation

1 Introduction

Clothing plays a central role in compelling animation by contributing to the style and personality of animated characters while evoking the impression of realism and complexity that comes from detailed folding patterns. On a practical level, a great deal of clothing used in animated films consists of close-fitting garments that move along with the body. For example, a sweater may not show noticeable dynamics under normal body motion, but it will exhibit noticeable buckling patterns at the arm bends and oblique torsional folds rising up from the waist. Frictional contacts and the overall nonlinear nature of cloth mean that these quasi-static folds depend not only on the current pose but also on the path taken in pose space to arrive at the present body configuration. This property gives clothing an infinite source of diversity and detail—and also shows that cloth simulation is an indispensable tool in feature animation, even for close-fitting garments.

Although simulation systems can compute the deformation of cloth at a remarkable level of realism and detail, they incur an extremely high computational cost. Subspace methods have proven very effective at reducing computational cost for other applications such as finite-element based solid simulation. These methods are most effective when deformations are small or predictable. In such cases, one can construct a low-dimensional linear subspace whose basis remains constant over time, thus delivering high computational efficiency. Unfortunately, due to hysteresis and its inherent nonlinear nature, cloth deformations are generally neither small nor predictable, which is seemingly at odds with subspace simulation and probably explains why so few attempts have been made so far in this direction.

Our work explores subspace methods in the context of clothing simulation and demonstrates one approach to overcome the challenges described above. Our method relies on two key insights. First, we note that subspace simulation is effective only when it can take advantage of structure in the simulation. And, although the movement of free-flowing *cloth* is largely arbitrary and unstructured, the movement of *clothing*—especially close-fitting clothing—does indeed contain a great deal of structure. We thus employ a kinematic deformation model as reference state that takes advantage of the structure of clothing simulation by capturing rotations that would otherwise prevent a linear-subspace, ill-suited for modelling rotational motions, from succeeding. Our second insight is that the rich deformations seen in clothing cannot be reproduced adequately with a global, low-dimensional basis. However, around a particular pose, the local space of deformations is much lower-dimensional. This observations motivates an adaptive, pose-dependent basis and allows our system to represent a broad range of complex wrinkles and folds while maintaining a small set of active basis vectors at any point during the simulation.

With these core insights in hand, we present a subspace approach to clothing simulation that uses a dynamically updated subspace basis in order to best reflect the deformations around the current pose. Our prototype implementation improves the performance of state-of-the-art cloth simulation codes by a factor of up to 22 while still reproducing the rich set of of wrinkles and folds evident in the full-space solution.

2 Related Work

Cloth Simulation is a well-explored field and existing works are far too numerous to be listed here. The work of Baraff and Witkin [1998] was a major breakthrough in terms of computational efficiency and even though the following 15 years have seen many improvements, high-resolution cloth simulation is still very time-consuming.

Nevertheless, there are many methods that aim for faster cloth simulation. One line of work combines simulation on a coarse base mesh with a fast method for adding geometric details. The method of Rohmer et al. [2010] adds geometrically generated wrinkles based on the strain field of the coarse simulation. Mueller and Chentanez [2010] attach a high-resolution mesh to a coarse simulation, whose deformation is determined using fast statics solves.

Another stream of work exploits precomputed data to avoid run-time simulations altogether. De Aguiar et al. [2010] present a technique for learning a linear conditional cloth model that can be trained with data from physics-based simulations. The method achieves very fast computation times, but it primarily targets low-complexity cloth with little folding. The method of Guan et. [2012] factors clothing deformations into components due to body shape and pose. A linear model is learned in order to quickly dress different body shapes and pose without run-time simulations. A different way of exploiting precomputed data was suggested by Kim et al. [2013], who create an exhaustive set of secondary motion to accompany a given primary motion graph. Since no run-time simulation is required, this method is very fast. However, the character's range of motion has to be small enough to fit a motion graph, which is not the case for production-level character animations.

Yet another class of methods combines coarse simulations and precomputed data. Feng et al. [2010] describe an approach which decomposes a high-res simulation into mid- and fine-scale deformations. For the mid-scale deformations, the mesh is decomposed into a set of bone clusters for which skinning weights are fit in a way similar to [James and Twigg 2005], while fine-scale details are added based on a PCA-analysis of residual vectors as in [Kry et al. 2002]. Both mid- and fine-scale details are then driven by a coarse scale simulation, which is fast enough to yield real-time rates. Focussing on fitted clothing, Wang et al. [2010] present an example-based approach that augments coarse simulations with pose-dependent detail meshes obtained from a wrinkle database. The wrinkle database stores per-joint wrinkle meshes that are precomputed from high-res simulations and merged together at run time. Targeting the more general case of free-flowing cloth, Kavan et al. [2011] describe a method for learning linear upsampling operators from high-res simulations. With similar goals, Zurdo et al. [2013] combine multi-resolution and pose space deformation (PSD) techniques in order to augment coarse simulations with example-based wrinkles.

Similar to these works, our method uses data from high-res simulations, but rather than augmenting a coarse simulation, we construct a low-dimensional subspace that allows for fast simulation of detailed clothing deformations.

Finally, another option for performance improvements is to leverage the processing power of parallel architectures [Selle et al. 2009]. However, while significant acceleration factors have been reported for large data (e.g. 2 million triangles [Selle et al. 2009]), the improvements for typical problem sizes are rather modest.

Subspace Simulation is generally most attractive when high-resolution models undergo low-rank deformations. The problem of subspace intergration and model reduction for simulation of elastics was originally formulated in the field of engineering [Krysl et al. 2001], but we focus on works from graphics for the sake of conciseness. In this context, the method of Barbič and James [2005] was the first to demonstrate, and unleash, the potential of model reduction for accelerating elastic deformations on high-resolution meshes. Subsequent work by An et al. [2008] showed that a selective evaluation of elemental contributions (also known as cubature) can improve the asymptotic complexity of subspace methods. While subspace methods can be very efficient for cases with small or predictable deformations, the generalization is made difficult by the discrepancy between a low-dimensional basis and a large range of deformations. Kim et al. [2009] address this problem with a hybrid solution that combines subspace and fullspace simulation and, for each step, decides which one to use. One technically interesting aspect of this work is the use of an adaptive basis that is dynamically updated with results from the full-space solver. Our method also relies on a dynamically updated basis, but our updates are much more frequent (once per Newton iteration) and do not require online full-space simulation.While most subspace methods rely on a linear basis, the work of Hahn et al. [2012; 2013] simulates the deformation of a character's fat and muscles in the nonlinear subspace induced by its rig.

The problem of detecting and resolving collisions in the context of subspace simulation has recently gained attention. Barbič and James [2010] showed how bounding volume hierarchies can be enhanced by certificates that allow aggressive culling over overlapping tests, and Zheng and James [2012] extended this approach to also consider deformation energy. Wong et al. [2013] propose a method for efficient self collision culling for skeleton-driven deforming meshes. While we do not address subspace self-collision culling in this work and simply resort to full-space collision resolution, we note that many of these ideas could also be applied to our setting of subspace simulation using adaptive bases.

Harmon et al. [2013] dynamically augment the subspace basis with analytical functions that model deformations due to individual contact points. Since these augmentation vectors only add very localized displacements, they are able to project their current subspace coordinate vector into the new basis whenever it changes to ensure temporal coherence. In our case, the basis regularly incurs drastic changes which motivates the use of a more sophisticated approach to obtain smooth transitions.

Pose-Space Deformation The concept of making shape depend on positions in a *pose space* was originally proposed by Lewis et al. [Lewis et al. 2000] and further extended in the context of example-based deformations [Sloan et al. 2001], medical imaging [Kurihara and Miyata 2004] and real-time applications [Kry et al. 2002]. Meyer and Anderson [2007] proposed Key Point Subspace Acceleration and soft caching, which accelerate pose-dependent deformation queries. Zurdo et al. [2013] use PSDs to enhance a low-res simulation with example-based wrinkle details. The quality of pose-space deformation methods heavily depends on the way the scattered-data interpolation problem in pose space is resolved. To this end, Lee [2009] explored the space of basis functions, while Bengio and Goldenthal [2013] propose a simplicial interpolation scheme to make the interpolation space more controllable.

Similar to these works, our method is also based on the assumption that deformations are inherently pose-dependent. However, instead of interpolating deformations in pose space, we select them automatically from nearby locations and let our solver handle the transitions between them.

3 Overview

Our method consists of several stages, each of which we briefly outline in this section.

Input We expect as input a linear blend skinning (LBS) rig for the character, i.e., a rigid skeleton, an undeformed surface mesh, and a set of skinning weights that determine how the surface mesh deforms according to the pose of the character. Furthermore, we assume that there are animation sequences provided that are representative of the typical motion that is expected during animation. In a production environment, these animation clips could correspond to the calisthenics sequences that are typically set up for testing the rig. Finally, we expect geometry for the clothing to be provided and pre-positioned in a way that fits the neutral pose of the character. Using the input rig and the pre-positioned clothing for the neutral pose, we construct a kinematic deformation model for the clothing that will serve as the reference state during subspace simulation.

Training Stage For each of the input animations, we perform a full-space cloth simulation. The resulting cloth configurations are associated with the corresponding poses of the character. Typically, these animations will lead to multiple cloth configurations for the same point in pose-space. For example, this will always be the case when the same animation is run at different speeds. But also simple motion like bending and straightening an arm will generally lead to different cloth configurations due to collisions, friction, and the overall nonlinear nature of clothing.

Pose-Space Database Construction The training stage provides us with a data structure that holds all simulation results associated with their corresponding points in pose space. However, we eventually need a data structure that provides us with a subspace basis for any point in pose-space. To this end, we cluster the simulation data in pose space and perform PCA on each cluster. We keep the most important modes and associate them with the pose-space point corresponding to the center of the cluster. We refer to this location as a *site*.

Reduced-Space Simulation For each step of the simulation, we want to retrieve a subspace basis according to the current pose of the character and, potentially, the current state of the clothing. Since the character's pose will generally not coincide with any of the sites, we need a way to construct the set of m basis vectors from its surrounding sites. Our approach is to select basis vectors from neighboring sites considering their distance in pose-space and how well they match the current dynamic state of the clothing.

4 Pose-Space Database

As one of the key concepts of our approach, the pose-space database (PSDB) is a data structure that holds bases (sites) distributed across pose space. During subspace simulation, the PSDB is responsible for providing a low-dimensional basis that describes the behavior of the clothing around a given current pose.

There are a number of questions that we must answer in order to design such a PSDB. We must find an adequate pose-space parameterization, we have to determine how to associate data with pose, and we have to decide which information to extract and store from the large amount of training data.

Relation between pose and clothing deformation A central question that influences subsequent design decisions is whether we should assume a locality relation between the character's pose and the deformation of its clothing. Clearly, the deformations induced by moving the shoulder are most significant around the shoulder—but how far do they extend? Wang et al. [2010] segment the clothing mesh into parts according to the body joints and assume that each joint influences only the two clothing segments adjacent to it. While this approach has the advantage that localized data can be stored per joint, a disdavantage is that far-reaching deformation effects such as the torsional folds at the waist that arise when moving the shoulder — which can be seen in Figure 1 or by self-experimentation — are not captured. We consider such effects to be essential for our target application and therefore assume that each change in pose can potentially induce deformations everywhere in the clothing. Consequently, we associate deformation data with character poses represented by points in pose space.

Pose-space parameterization We assume an LBS rig as input, which is parameterized by joint angles. The natural way of representing the pose of the character is thus by a vector $\mathbf{j}$ holding the angle values for each joint. However, this approach is not without problems: first, the number of angles required to describe the pose of a character is typically quite large (already 57 for our *torso* model). Sampling in such a high-dimensional space is impractable since, unless there are truly redundant dimensions, the likelyhood that two samples are far away frome each other is very high — a phenomenon known as the *curse of dimensionality*. For another, many rigs exhibit redundant controls around the clavicle and the shoulder that can be convenient for an artist but problematic in our context when associating deformations with poses.

Another approach is to measure the distance between two poses by measuring the *induced difference in geometry* of the corresponding meshes. This approach allows for more intuition as to what distance means, and it avoids the problem of redundancies. We perform a Principal Component Analysis (PCA) on all the posed meshes in the training data, truncating the basis acoording to significance in singular values and normalizing the dimensions to have equal variance. This results in a transformation to a coordinate space where the $L2$-distance is such a desired measure.

Data generation and model reduction We perform simulation runs for each of the input animations and associate the clothing shape of each time step with its corresponding character pose. Storing such a massive amount of data is neither practical nor useful and we would prefer a form which captures the diversity of deformations in a concise way. PCA with truncation is a natural candidate for this purpose as it provides a principled way to balance between the captured variability of the data and the dimensionality of the approximation. Rather than constructing a global basis, our approach relies on many localized bases distributed across pose-space. In this way, we can exploit the fact that, locally, deformations can be approximated with a low-dimensional basis but still account for the large variability of deformations across pose-space.

Since PCA on the training data produces global deformation modes, another option would be to use localized basis vectors. However, while it would be possible to use localized basis vectors that only affect single regions, it is unclear how to partition a given set of global deformations into localized ones.

Basis creation In order to construct a set of distributed bases, we must decide how many sites with corresponding local basis to use, where to put them and what data to use for each site. We note that, once the number of sites is determined, the question of where to place the sites and which data points to associate with them is answered in a natural way using Voronoi partitioning. We found that two to three sites per input animation work well when using calisthenics-like input sequences that cycle through one specific motion. We then run the k-means algorithm in order to compute the desired number of clusters, obtaining the pose-space locations $\mathbf{s}_i$ of the sites and the sets of data $\mathcal{D}_i$ belonging to them.

Before we can analyse the clusters, there is one more transformation that needs to be applied to the data. If we were to perform PCA directly on the geometry of the clothing as returned by the full-space simulator, the variability would be dominated by the motion of the body, i.e., rotations of its joints. However, were are not interested in changes that are captured by the kinematic model, we want to analyze how the clothing changes its shape relative to the kinematic model. For this reason, we first substract the state of the kinematic model $\mathbf{X}(\mathbf{p})$ evaluated at the current pose $\mathbf{p}$ from the clothing configuration $\mathbf{x}$ to obtain a world-space displacement vector $\mathbf{u}(\mathbf{p}) = \mathbf{x} - \mathbf{X}(\mathbf{p})$. This displacement vector is then transformed back to the neutral pose $\bar{\mathbf{p}}$ of the character by applying the inverse LBS transformations as $\bar{\mathbf{u}} = \mathrm{LBS}^{-1}(\mathbf{u})$.

Once we have applied this transformation to all data points in all clusters, we are all set for analysing the data. For each cluster, we perform PCA on $\mathcal{D}_i$ and truncate the basis either after the d-th vector or when the ratio between the corresponding singular value and the largest singular value drops below a given threshold value $\varepsilon_{\mathrm{PCA}}$ (we use $\varepsilon_{\mathrm{PCA}} = 0.01$). The basis vectors $\mathbf{b}_i^j$ of all sites are then stored in the PSDB, indexed by the corresponding pose-space location $\mathbf{s}_i$ of the sites.

Data Retrieval Once the PSDB is populated, the question is how to retrieve data at run time: Given a pose for the character, what data should be returned? Obviously, since the pose will generally not coincide with one of the sites, we cannot just return one of the bases but need a retrieval scheme for returning adequate inbetween data that reflects the influence of neighboring sites.

One possibility for a such a retrieval scheme would be to interpolate between the bases of different sites. This bears the promise that a smooth basis interpolation would translate into temporal smoothness for the simulation. Unfortunately, constructing a principled scheme for bases interpolation appears to be difficult. While interpolating pairs of vectors can be done easily, extending this concept to interpolating between sets of vectors (i.e., bases) is difficult because it is unclear what the correspondence between vectors in different sets should be. But even without this problem, it remains questionable whether an interpolated basis is meaningful to begin with.

For these reasons, we settled for an approach that only uses basis vectors from the original sites. More concretely, when the database is queried with a given pose $\mathbf{p}_i$, we create a pool of candidate basis vectors that includes the vectors of all sites that are with in the support of a kernel function $\phi(\mathbf{p}_i)$ centered at the current pose. The subspace simulator then selects a set of basis vectors from this candidate pool as explained in Section 5.2.

5 Subspace Cloth Simulation

5.1 Equations of Motion

Equations of Motion in Full Space The basis for our subspace simulation method is a variational formulation of the implicit Euler scheme similar to Martin et al. [2011]. We define the corresponding objective function as

$$H(\mathbf{x}_{t+1}) = \frac{1}{2}\mathbf{a}_{t+1}^T \mathbf{M}\mathbf{a}_{t+1} + hW(\mathbf{x}_{t+1}) \tag{1}$$

where $\mathbf{x}_{t+1}$ denotes end-of-time-step positions, h is the time step, $\mathbf{M}$ is the diagonal mass matrix, and $\mathbf{a}_{t+1} = \frac{1}{h^2}(\mathbf{x}_{t+1} - 2\mathbf{x}_t + \mathbf{x}_{t-1})$ are nodal accelerations. The potential energy $W = W_{\mathrm{el}} + W_{\mathrm{ext}}$ consists of an elastic energy W_{el} due to cloth deformations and an external energy W_{ext} due to gravity. Our cloth model uses constant strain triangles for stretching [Thomaszewski et al. 2008] and hinge elements for bending [Grinspun et al. 2003]. Furthermore, we use axis-aligned bounding boxes for collision detection and a combination of spring-like penalty forces and velocity filtering [Bridson et al. 2002] for collision response.

Equation (1) provides a consistent formulation to perform simulations in both full and reduced space: when doing full-space simulation for generating training data, we directly minimize (1) for each time step. In order to do subspace simulation, we restrict the end-of-time-step positions $\mathbf{x}_{t+1} = \mathbf{x}(\mathbf{q}_{t+1})$ to a subspace described by a basis matrix $\mathbf{A}$ and reduced coordinates $\mathbf{q}$.

Kinematic Cloth Model The basis of our subspace simulator is a kinematic cloth model, which we construct by extending the skinning transformation of the LBS rig onto the clothing. In the neutral pose $\bar{\mathbf{p}}$ of the character, we find for each vertex $\mathbf{X}_i^0$ of the prepositioned clothing the closest point on the character's surface. We then determine skinning weights ω_i^j corresponding to the bones of the LBS rig using barycentric interpolation. The kinematic model for any given pose $\mathbf{p}$ is then given by the usual LBS transformation

$$\mathbf{X}_i(\mathbf{p}) = \varphi_{\mathrm{LBS}}(\mathbf{p}, \bar{\mathbf{X}})_i = \sum_j \omega_i^j \mathbf{T}_j^T(\mathbf{p})\bar{\mathbf{X}}_i, \tag{2}$$

where $\mathbf{T}_j(\mathbf{p})$ is the transformation matrix of bone j corresponding to the current pose. While (2) is a nonlinear function of body pose, it is linear in positions and its Jacobian $\mathbf{B} = \frac{\partial \varphi_{\mathrm{LBS}}}{\partial \bar{\mathbf{X}}_i}$ with respect to position is a block-diagonal matrix with 3×3-blocks given as

$$\mathbf{B}_i^t = \sum_j \omega_i^j \cdot \mathbf{T}_j^T \,. \tag{3}$$

Subspace Cloth Model The basic approach of our subspace model is to extend the kinematic model with displacements computed from a subspace. One approach would be to compute displacements $\mathbf{u}$ directly in world-space as

$$\mathbf{x}(\mathbf{p}, \mathbf{q}) = \mathbf{X}(\mathbf{p}) + \mathbf{u}(\mathbf{q}) \,, \tag{4}$$

where $\mathbf{q}$ are the reduced coordinates. However, a world-space displacement is only meaningful for the pose at which it was computed—rotations induced by the rig by would immediately invalidate the displacement. We therefore choose to compute displacements $\bar{\mathbf{u}}$ with respect to the untransformed setting,

$$\mathbf{x}(\mathbf{p}, \mathbf{q}) = \varphi_{\mathrm{LBS}}(\mathbf{p}, \bar{\mathbf{X}} + \bar{\mathbf{u}}) = \varphi_{\mathrm{LBS}}(\mathbf{p}, \bar{\mathbf{x}}) \,, \tag{5}$$

where $\bar{\mathbf{x}} = \bar{\mathbf{X}} + \bar{\mathbf{u}}$ denotes the untransformed clothing state. This formulation has the advantage that a displacement vector for a given pose will look plausible throughout a certain region in pose-space.

Our approach uses a linear subspace, such that displacements $\bar{\mathbf{u}}(\mathbf{q}) = \mathbf{A}\mathbf{q}$ are defined by a basis matrix $\mathbf{A}$, whose columns correspond to the basis vectors, and reduced coordinates $\mathbf{q}$. For a fixed $\mathbf{A}$ and $\mathbf{q}$, the deformed configuration of the cloth is thus given as

$$\mathbf{x}(\mathbf{p}, \mathbf{q}) = \varphi_{\mathrm{LBS}}(\mathbf{p}, \bar{\mathbf{X}} + \mathbf{A}\mathbf{q}) \,. \tag{6}$$

Subspace Simulation Algorithm Algorithm 1 describes our subspace integration method, which uses Newton's method to minimize (1) with respect to the reduced coordinates $\mathbf{q}$, and where $\mathbf{K}$ and $\mathbf{r}$ are the reduced space Hessian and gradient. During subspace simulation, we always preserve the invariant that the current cloth deformation is the LBS-transform of the untransformed $\bar{\mathbf{x}}$.

Algorithm 1 Subspace integration with adaptive basis.

1: **for** $i := 1$ **to** n_{iter} **do**
2: update full-space state $\mathbf{x} = \varphi_{\mathrm{LBS}}(\mathbf{p}, \bar{\mathbf{x}})$
3: compute full-space gradient $\mathbf{g}(\mathbf{x})$
4: selectBasisVectors$(\mathbf{p}, \mathbf{g}, \mathbf{A})$
5: computeReducedSystemMatrix$(\mathbf{A}, \mathbf{K})$
6: solve $\mathbf{K}\mathbf{q} = -\mathbf{r}$
7: update untransformed state $\bar{\mathbf{x}} = \bar{\mathbf{x}} + \mathbf{A}\mathbf{q}$
8: **end for**

System Assembly Algorithm 1 requires the assembly of the system $\mathbf{K}\mathbf{q} = -\mathbf{r}$ in each iteration. This involves computing the reduced system matrix $\mathbf{K}$ and the reduced gradient $\mathbf{g}$. The expressions for these components are obtained by using (6) in (1) and differentiating with respect to $\mathbf{q}$. We first note that, using (3) and the chain rule, the Jacobian $\mathbf{J}$ of (6) follows as

$$\mathbf{J} = \frac{\partial \mathbf{x}}{\partial \mathbf{q}} = \mathbf{B} \cdot \mathbf{A} \,. \tag{7}$$

The reduced gradient and Hessian of (1) are obtained as

$$\mathbf{r} = \mathbf{J}^T \left(\mathbf{M}\mathbf{a} + \frac{\partial W}{\partial \mathbf{x}} \right) \tag{8}$$

$$\mathbf{K} = \mathbf{J}^T \left(\frac{1}{h^2}\mathbf{M} + \frac{\partial^2 W}{\partial \mathbf{x}^2} \right) \mathbf{J} \,. \tag{9}$$

The resulting system is dense but only of dimension $r \times r$ instead of $n \times n$ for the full-space variant. For a typcial example like the sweater, the dimension of the reduced space was $r = 12$, while the full-space clothing had $n =$ 80k degress of freedom. Indeed, the cost of solving the reduced system is negligible, but the assembly can take up a significant fraction of the overall computation time. The main contributors are the computation of the full-space Hessian $\mathbf{H} = \frac{\partial^2 W}{\partial \mathbf{x}^2}$ and the multiplications with the projection matrix $\mathbf{J}$.

Solver Optimizations In our experiments, we found that simulations in subspace exhibit very good convergence and are generally much less susceptible to instabilities than in full-space. In particular, we never encountered indefinite systems—a major struggle for full-space simulation—removing the need for line-search and expensive regularization altogether. Similar to Hahn et al [2013], we also found that, even when reusing the same full-space Hessian over many times steps, stability was not affected and the visual impact on the simulation results was minimal. Our approach is therefore to keep the Hessian constant and only do one Newton step by default. Only if the norm of the current basis vectors projected onto the full-space gradient (see Section 5.2) is still above a threshold σ, we recompute the $\mathbf{H}$ and perform further steps. Choosing $\sigma = 0.1$ worked well for all our examples.

Figure 2: *Using a fixed basis of 100 vectors for the* Torso Twist *sequence produces large jumps between different cloth poses. The three frames shown here were captured whithin a short duration of only 0.12 seconds.*

Figure 3: *One frame of the* Torso Twist *sequence when simulated with only the gradient as basis (left), adaptive basis selection without adding the gradient to the basis (middle), and our method using the gradient in addition to the adaptive basis selection.*

5.2 Basis Construction

Our subspace integration algorithm selects a new set of basis vectors in every iteration of the Newton solver. We would like this basis to be low-dimensional, and we want it to capture the deformations that the clothing can undergo around the current character pose. Given the current pose, the pose-space database provides a pool of candidate vectors, typically much larger than the desired dimension of the subspace. We therefore select a subset of vectors according to how far the site is away from the current pose and how well a given vector fits the current configuration.

Each iteration of Algorithm 1 solves the full-space linear problem $\mathbf{H}\mathbf{x} = -\mathbf{g}$ projected into to current subspace. Clearly, if we can find a basis that spans the full-space solution $\mathbf{H}^{-1}\mathbf{g}$, we will be able to accurately solve the full-space linear problem in the subspace. How helpful is a given basis vector $\mathbf{v}$ for this purpose? Assuming that $\mathbf{H}$ is positive definite, we know that $\mathbf{v}$ and $\mathbf{g}$ must have a positive dot product for $\mathbf{v}$ to be a descent direction for (1). Put differently, if this dot product is zero, $\mathbf{v}$ cannot help in solving the full-space system. This observation motivates a selection scheme that gives preference to basis vectors that are well-aligned with the gradient at the current configuration. We therefore project all basis vectors in our pool onto $\mathbf{g}$ and reorder according to score. In order to give preference to vectors from sites close to the current pose, we additionally scale the score with the inverse distance before sorting. We note that there is no need to maintain an orthogonal basis since potential redundancies do not cause any adverse effect.

In order to analyze the efficiency of our selection scheme, we compared it to using a fixed basis created from extracting 100 PCA vectors from the training data for simulation. We noticed that even though the fixed basis is able to reproduce deformations for many individual poses correctly, it tends to jump between them, making the resulting animation not smooth as can be seen in Figure 2.

Adding the Gradient Using the same motivation as before, it seems natural to add the full-space gradient $\mathbf{g}$ as a basis vector. This ensures that the solution of the subspace problem will always reduce (1) and at the same time enlarges the range of possible deformations. We noticed that adding the gradient leads to significantly improves simulation results, as can be seen in Figure 3.

Incremental State Updates & Undo Vector A particular aspect of our approach is that we update the full-space configuration $\mathbf{x}$ with subspace displacements $\mathbf{Aq}$ in each iteration of Algorithm 1. As a notable difference to existing works, the full-space configuration is generally not in the span of the current subspace basis. Indeed, due to the large range of deformations observed in clothing, we find that it is impractical to restrict the full-space solution in this way while still obtaining the same degree of variability in clothing shape. However, when the basis changes in every iteration, it is not possible to change the component of the current state that is not in the span of the current basis. We solve this problem by always adding the difference vector $\bar{\mathbf{x}} - \bar{\mathbf{X}}$ to the basis, thus allowing the clothing to *undo* the solution of the previous step, if necessary.

6 Results

Setup & Validation For all results, we first ran a state-of-the-art cloth simulation code on the *Torso Shirt* mesh with 29,510 vertices and 58,660 triangles to generate around 10,000 frames of training data for our method. Some of the deformations in our training data can be seen in Figure 5. Using this training data, we automatically created a PSDB using 18 sites with 10 to 20 basis vectors each, resulting in a total number of 212 basis vectors. Four of the basis vectors are visualized in Figure 4 and can also be seen in the accompanying video. We first validated our subspace simulation method by running it on the same animations that were used to create the training data. One such example for the *Twist* animation can be seen in Figure 1. Using our fast subspace simulation method, we were able to obtain total runtime speedups of up to 22x over the full-space simulation. Comparing only the simulation time yields a speedup of up to 60x. The timings and speedups for all tested sequences are reported in Table 1.

We observed no visual differences between the naïve subspace solver and the optimized solver as described in Section 5.2. Increasing the size of our adaptive basis also only has small effects on the resulting deformations when using the same PSDB. Increasing the number of vectors per site in the PSDB by reducing ε_{PCA} also has no noticeable effect unless combined with an increased adaptive basis size. While we observed slightly more dynamics in the formation of the wrinkles in the clothing, this gain comes at a significant computational cost.

Generalization to Novel Poses Even though we use global basis to construct our subspaces, many of the basis vectors tend to be sparse and local even though we do not ask for this explicitly. For this reason, different basis vectors from different sites can combine into new configurations that were not in the training data set. Thus, one application of particular interest for our method is the ability to generalize to new motions. To this end, we created two new animations that are not part of our training set and explore new portions of pose space. *Yo* simultaneously combines both twisting and arm bending, while *Gym* is a longer sequence that comprises a variety of upper body motions. Figure 6 shows that our method is able to generalize to these unseen animations and produce compelling folds and wrinkles.

7 Limitations & Future Work

While we are able to faithfully reproduce wrinkles and folds for the examples that we tested, one limitation of our method is the limited ability to reproduce dynamics, which is best seen in the accompanying video. The observation that wrinkles tend to move semi-rigidly in the normal direction of their fold could motivate the use of more sophisticated basis vector extraction than PCA. On a related note, shaping the subspace vectors manually to reflect the desired deformation could enable stylization and artist control.

Previous work by An et al. [2008] and Kim et al. [2009] showed that the performance of subspace simulations can be significantly increased using *cubature*. The underlying idea is to evaluate the full-space energy and its derivatives approximately using a small numer of key elements. However, compared to typical deformations for volumetric solids, the folds observed in cloth are large and localized, making efficient cubature challenging and thus an interesting avenue for future research.

We currently handle collisions uniformly on a per-vertex basis for both our full-space and subspace simulation methods. Our initial tests suggest that culling-based subspace collision techniques could be adapted to our setting, promising an additional potential speedup for our subspace method.

Another limitation is that it is currently not possible to adapt the material properties of the clothing in the subspace. Consequentially, the look of the cloth is determined by the full-space simulation that was used to generate the training data. While changing stiffness and bending coefficients would be easily possible, the physical interpretation of this is unclear and the results would likely be difficult to predict. That said, enabling fast subspace resimulation to explore the effect of different material properties would be an interesting application.

References

AN, S. S., KIM, T., AND JAMES, D. L. 2008. Optimizing cubature for efficient integration of subspace deformations. In *Proc. of ACM SIGGRAPH '08*.

BARAFF, D., AND WITKIN, A. 1998. Large steps in cloth simulation. In *Proc. of ACM SIGGRAPH '98*.

BARBIČ, J., AND JAMES, D. L. 2005. Real-time subspace integration for st. venant-kirchhoff deformable models. In *Proc. of ACM SIGGRAPH '05*.

BARBIČ, J., AND JAMES, D. L. 2010. Subspace self-collision culling. In *Proc. of ACM SIGGRAPH '10*.

BENGIO, J. C., AND GOLDENTHAL, R. 2013. Simplicial interpolation for animating the hulk. In *ACM SIGGRAPH 2013 Talks*, SIGGRAPH '13, 7:1–7:1.

BRIDSON, R., FEDKIW, R., AND ANDERSON, J. 2002. Robust treatment of collisions, contact and friction for cloth animation. In *Proc. of ACM SIGGRAPH '02*.

DE AGUIAR, E., SIGAL, L., TREUILLE, A., AND HODGINS, J. K. 2010. Stable spaces for real-time clothing. In *Proc. of ACM SIGGRAPH '10*.

FENG, W.-W., YU, Y., AND KIM, B.-U. 2010. A deformation transformer for real-time cloth animation. In *Proc. of ACM SIGGRAPH '10*.

Sequence	n_{frames}	Collisions	Full space	Naïve Subspace			Optimized Subspace		
		t_{frame}	t_{frame}	t_{frame}	tsp	ssp	t_{frame}	tsp	ssp
Torso Twist	1701	0.47s	13.20s	1.11s	12x	20x	0.71s	18x	52x
Torso Arms Bend	1451	0.41s	14.09s	0.95s	15x	25x	0.63s	22x	60x
Torso Arms Up	1251	0.45s	16.32s	1.29s	13x	19x	0.92s	18x	34x
Torso Arms Down	1251	0.40s	12.95s	1.17s	11x	16x	0.89s	15x	25x
Torso Lean Back	601	0.48s	12.68s	1.40s	9x	13x	0.82s	16x	37x
Torso Lean Forward	601	0.46s	12.59s	1.35s	9x	14x	1.07s	12x	20x
Torso Plane	1501	0.47s	12.41s	1.23s	10x	16x	0.90s	14x	28x
Torso Yo	1176	0.39s	—	1.05s	—	—	0.86s	—	—
Torso Gym	4701	0.42s	—	1.07s	—	—	0.82s	—	—

Table 1: *Timings and speedups for the different sequences we simulated on an Intel Core i7-3930K 6 x 3.2Ghz where* t_{frame} *is computation time per frame in seconds,* tsp *is the total speedup over the full-space simulation and* ssp *is the simulation speedup (not counting collision time) over the full-space simulation.* Yo *and* Gym *are generalization examples that did not appear in the training data, and we did not need to run the full space solver for them.*

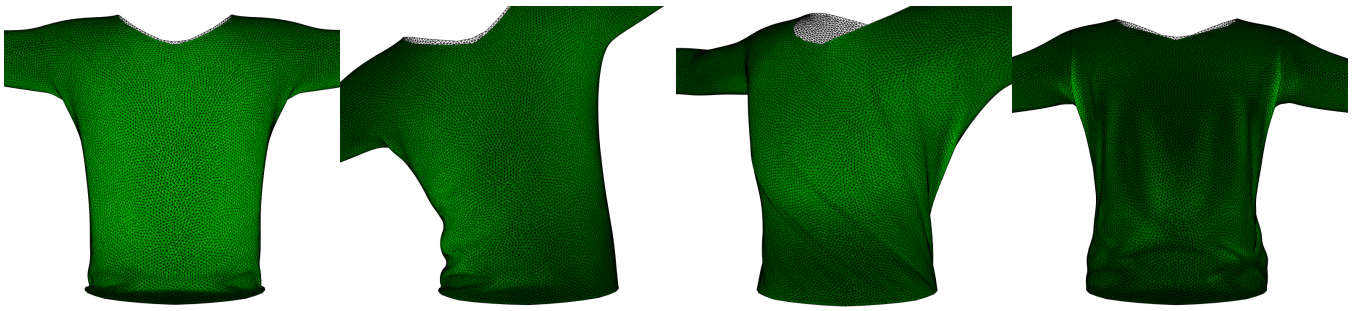

Figure 4: *Visualization of four basis vectors extracted from four different sites in our PSDB. The color intensity indicates the relative amount by which regions move when exciting the respective basis vector (dark: no motion, bright green: strong motion).*

Figure 5: *Sample frames of the seven training sequences we used as input for our method. From left to right, top to bottom:* Arms Bend, Lean Forward, Plane, Twist, Arms Down, Lean Back, Arms Up.

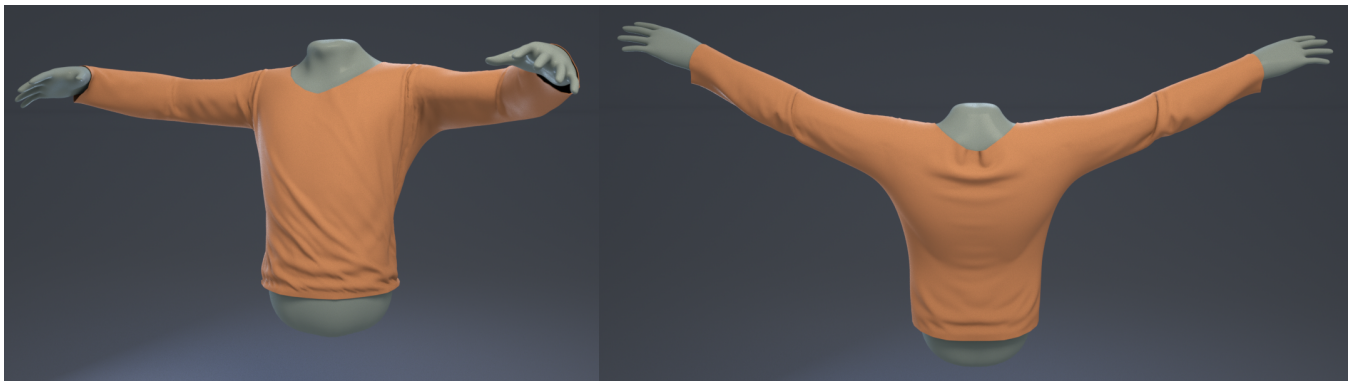

Figure 6: *Left: The* Yo *sequence exhibits both torsoidal folds and wrinkles around the elbows and armpits simultaneously, which was never seen in the training data. Right: One frame of the* Gym *sequence showing the combined effect of leaning back and stretching out the arms of the torso.*

GRINSPUN, E., HIRANI, A. N., DESBRUN, M., AND SCHRÖDER, P. 2003. Discrete shells. In *Proc. of Symp. on Computer Animation '03*.

GUAN, P., REISS, L., HIRSHBERG, D. A., WEISS, A., AND BLACK, M. J. 2012. Drape: Dressing any person. In *Proc. of ACM SIGGRAPH '12*.

HAHN, F., MARTIN, S., THOMASZEWSKI, B., SUMNER, R., COROS, S., AND GROSS, M. 2012. Rig-space physics. In *Proc. of ACM SIGGRAPH '12*.

HAHN, F., THOMASZEWSKI, B., COROS, S., SUMNER, R. W., AND GROSS, M. 2013. Efficient simulation of secondary motion in rig-space. In *Proc. of Symp. on Computer Animation '13*.

HARMON, D., AND ZORIN, D. 2013. Subspace integration with local deformations. In *Proc. of ACM SIGGRAPH '13*.

JAMES, D. L., AND TWIGG, C. D. 2005. Skinning mesh animations. In *Proc. of ACM SIGGRAPH '05*.

KAVAN, L., GERSZEWSKI, D., BARGTEIL, A. W., AND SLOAN, P.-P. 2011. Physics-inspired upsampling for cloth simulation in games. In *Proc. of ACM SIGGRAPH '11*.

KIM, T., AND JAMES, D. L. 2009. Skipping steps in deformable simulation with online model reduction. In *Proc. of ACM SIGGRAPH Asia '09*.

KIM, D., KOH, W., NARAIN, R., FATAHALIAN, K., TREUILLE, A., AND O'BRIEN, J. F. 2013. Near-exhaustive precomputation of secondary cloth effects. In *Proc. of ACM SIGGRAPH '13*.

KRY, P. G., JAMES, D. L., AND PAI, D. K. 2002. Eigenskin: real time large deformation character skinning in hardware. In *Proc. of Symp. on Computer Animation '02*.

KRYSL, P., LALL, S., AND MARSDEN, J. E. 2001. Dimensional model reduction in non-linear finite element dynamics of solids and structures. *International Journal for Numerical Methods in Engineering 51*, 479–504.

KURIHARA, T., AND MIYATA, N. 2004. Modeling deformable human hands from medical images. In *Proc. of Symp. on Computer Animation '04*.

LEE, G. S. 2009. Evaluation of the radial basis function space. In *ACM SIGGRAPH ASIA 2009 Sketches*, SIGGRAPH ASIA '09, 42:1–42:1.

LEWIS, J. P., CORDNER, M., AND FONG, N. 2000. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proc. of ACM SIGGRAPH '00*.

MARTIN, S., THOMASZEWSKI, B., GRINSPUN, E., AND GROSS, M. 2011. Example-based elastic materials. In *Proc. of ACM SIGGRAPH '11*.

MEYER, M., AND ANDERSON, J. 2007. Key point subspace acceleration and soft caching. In *Proc. of ACM SIGGRAPH '07*.

MÜLLER, M., AND CHENTANEZ, N. 2010. Wrinkle meshes. In *Proc. of Symp. on Computer Animation '10*.

ROHMER, D., POPA, T., CANI, M.-P., HAHMANN, S., AND SHEFFER, A. 2010. Animation wrinkling: Augmenting coarse cloth simulations with realistic-looking wrinkles. In *Proc. of ACM SIGGRAPH Asia '10*.

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 (Mar.), 339–350.

SLOAN, P.-P. J., ROSE, III, C. F., AND COHEN, M. F. 2001. Shape by example. In *Proc. of Symp. on Interactive 3D Graphics '01*.

THOMASZEWSKI, B., PABST, S., AND WOLFGANG, S. 2008. Asynchronous cloth simulation. In *Proc. of Computer Graphics International '08*.

WANG, H., HECHT, F., RAMAMOORTHI, R., AND O'BRIEN, J. 2010. Example-based wrinkle synthesis for clothing animation. In *Proc. of ACM SIGGRAPH '10*.

WONG, S.-K., LIN, W.-C., HUNG, C.-H., HUANG, Y.-J., AND LII, S.-Y. 2013. Radial view based culling for continuous self-collision detection of skeletal models. In *Proc. of ACM SIGGRAPH '13*.

ZHENG, C., AND JAMES, D. L. 2012. Energy-based self-collision culling for arbitrary mesh deformations. In *Proc. of ACM SIGGRAPH '12*.

ZURDO, J. S., BRITO, J. P., AND OTADUY, M. A. 2013. Animating wrinkles by example on non-skinned cloth. *IEEE Trans. Vis. Comput. Graph. 19*, 1, 149–158.