Wrinkling Captured Garments
Using Space-Time Data-Driven Deformation

T. Popa[1]   Q. Zhou[1]   D. Bradley[1]   V. Kraevoy[1]   H. Fu[1,2]   A. Sheffer[1]   W. Heidrich[1]

1) University of British Columbia,   2) Max-Planck-Institut für Informatik

Figure 1: *Reintroducing folds into captured garments: (left) input video-frames, (center) typical capture result [BPS*08], and (right) t-shirt wrinkled by our data-driven method.*

Abstract

The presence of characteristic fine folds is important for modeling realistic looking virtual garments. While recent garment capture techniques are quite successful at capturing the low-frequency garment shape and motion over time, they often fail to capture the numerous high-frequency folds, reducing the realism of the reconstructed space-time models. In our work we propose a method for reintroducing fine folds into the captured models using data-driven dynamic wrinkling. We first estimate the shape and position of folds based on the original video footage used for capture and then wrinkle the surface based on those estimates using space-time deformation. Both steps utilize the unique geometric characteristics of garments in general, and garment folds specifically, to facilitate the modeling of believable folds. We demonstrate the effectiveness of our wrinkling method on a variety of garments that have been captured using several recent techniques.

1. Introduction

Capturing the geometry of moving garments provides a data-driven alternative to cloth simulation in much the same way as motion capture provides an alternative to character animation. Recent garment capture techniques [BPS*08, VBMP08, dAST*08] are based on multi-view video capture and are quite successful at capturing the low-frequency garment shape and motion over time, and at establishing consistent correspondences across frames. However, as demonstrated in Figure 1 (center), they often fail to capture the numerous high-frequency folds characteristic of garments,

reducing the realism of the reconstructed dynamic models. Folds tend to be very shallow, making them hard to capture and separate from noise. However, due to large normal variation they are highly noticeable. Thus, garments that lack folds look unnatural.

Marker-based capture methods such as [WCF07] capture folds more accurately, but are restricted to custom made garments. Active lighting approaches, e.g. [HVB*07], do reasonably well in capturing fine details in single camera setups. However, in practice, due to interference of the light sources, such methods do not generalize well to 360° acqui-

sition of moving targets. The use of active lighting also adds considerably to the complexity, and in some cases, cost, of the setup.

In our work, we propose a simple, yet effective, method for reintroducing folds into models captured by any multi-view video technique using data-driven, dynamic wrinkling (Figure 1 (right)). To generate believable folds we utilize the unique geometric properties of garments, most notably the fact that most garments exhibit very low stretch under normal wear. This property imposes very strong constraints on the shape of garment folds, leading to the characteristic fold shapes we are all familiar with. To generate folds consistent with the captured garment motion we take advantage of the available video footage and use the video to guide fold modeling. Our method first analyzes the video capture to estimate the position and shape of folds, using their distinguishing shape characteristics (Figure 2 (center)). It then introduces those folds into the reconstructed models, using stretch-minimizing deformation, which naturally produces believable fold shapes (Figure 2 (right)). We use a novel space-time deformation framework to generate folds which are consistent across time.

Although we aim to place folds in places where they are observed in real video footage, we make no claim of reproducing the exact shape of each fold. That is, our goal is to generate believable wrinkling with an overall appearance similar to that of the original garment, instead of attempting to "measure" the true fold geometry.

2. Previous Work

Cloth & Garment Capture: In recent years several methods emerged specifically for capturing garment motion [BPS*08, WCF07, SSK*05, PH03], as well as for capture of moving humans dressed in loose clothing [VBMP08, dAST*08]. Marker-based methods such as the work by White et al. [WCF07] use a custom set of color markers printed on the surface of the cloth, and then use the markers combined with the assumption of low fabric stretch to detect and trace high-resolution folds. As this approach is restricted to custom made garments, a markerless alternative was recently proposed by Bradley et al. [BPS*08]. The method successfully captures off-the-shelf garments correctly reconstructing the low-frequency shape (see Figure 1, center), but fails to capture the high-frequency folds.

The template-based methods of Vlasic et al. [VBMP08] and de Aguiar et al. [dAST*08] focus on capturing the overall performance of actors rather than concentrating specifically on garments. While these methods also capture an approximation of the clothing the actor is wearing, the geometric details present in the reconstructions of these garments tend to be copied from a high-resolution template scan rather than represent actual high-frequency per-frame details.

Active lighting approaches such as photometric stereo are better suited for capturing fine geometric details. However, such approaches generalize poorly to the full 360° acquisition of deformable models. For example, Hernandez et al. [HVB*07] use red, green, and blue lights to estimate surface normals with a photometric stereo approach, making use of the three different color channels in a color camera. Generalizing this approach to a full ring of cameras and light would require at least a dozen light sources that emit light at different frequencies, as well as specialized multi-spectral camera hardware that can distinguish between those light sources. Ahmed et al. [ATD*08] use calibrated lighting and multi-view video to capture normal fields, and augment garment geometry templates to include wrinkles and folds. Calibrating the light sources adds complexity to the system. In addition, this technique requires knowledge of the reflectance model at each point on the surface.

Modeling Folds and Wrinkles: The presence of folds is important for garment realism and several approaches exist for generating folds on simulated virtual garments. To model the folds, most methods use a physically-based simulation approach (see [BMF03] and the references therein), which is time consuming, and requires accurate information about the motion of the character wearing the garment as well as numerous physical parameters for the simulation.

Several authors propose alternative, more efficient, geometric approaches for modeling folds, which work well on tight, high-stretch garments [HBVMT99, CGW*07, CMT05]. Hadap et al. [HBVMT99] utilize a user defined fold pattern to generate a bump map representation as a solution. Cutler et al. [CGW*07] and Cordier et al. [CMT05] use cloth examples with precomputed, simulated folds to predict new fold geometry. Decaudin et al. [DJW*06] suggest a procedural approach for adding folds to loose garments, based on an analysis of likely fold shapes. Their results tend to look very regular and thus somewhat artificial. Like the physics-based methods above, these geometric approaches require accurate information about the proportions and motion of the garment wearer, which might often be unavailable in a capture setup.

Garment folds share some shape similarities with skin wrinkles. To capture dynamic facial wrinkles Bickel et al. [BBA*07] define a small number of explicit locations on the surface where wrinkles can occur using face paint and track those over time. This approach is not feasible for tracking garment folds since folds can occur anywhere on the surface. Since no explicit tracking is possible in our inputs we utilize space-time deformation to ensure temporal coherence.

Adding High-Frequency Motion: Recently, we observed an emergence of several methods [KA08, SZT*08, PH08, SB08] that add realism to animation by introducing secondary high-frequency motion, which is not fully physically based, but which increases the realism of the models. Our method can be seen as a continuation of this trend, increasing the realism of garment capture by introducing believable high-frequency folds, without the additional costs of a full-blown physical simulation or a sophisticated capture setup.
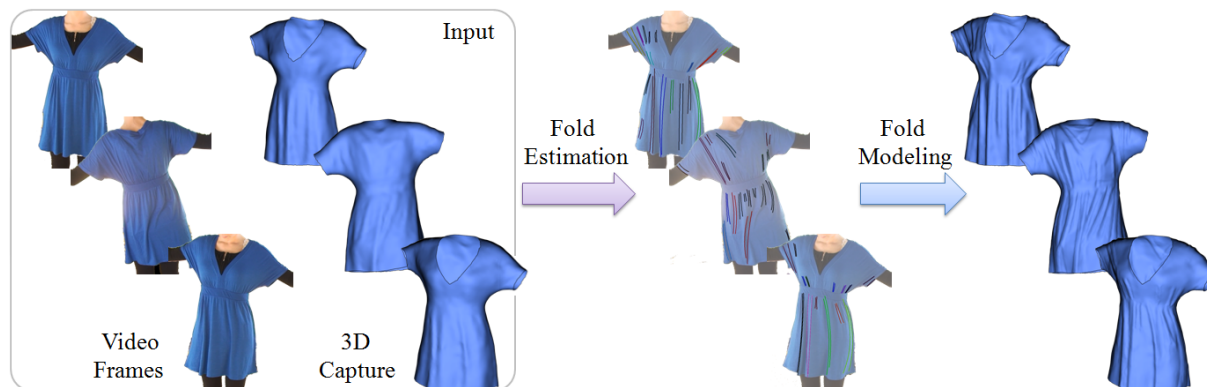
Figure 2: *Algorithm overview. From left to right: input video, captured smooth geometry, estimated folds, and wrinkled models generated by our space-time deformation approach.*

3. Background and Overview

To generate realistic looking folds we take advantage of the unique properties of garments that distinguish them from general deformable geometries. Most fabrics can undergo only very limited stretch [GHF*07]. Thus, from a geometry point of view, garments can be seen as piecewise quasi-developable surfaces, where the pieces correspond to the individual garment panels cut out of fabric. This observation imposes very strong constraints on the shapes that garments can form, and specifically on the shape of garment folds. We refer the reader to [DJW*06] for an overview of the common types of garment folds: sine wave, diamond, and twisted diamond (see Figure 3).

Based on the fold classification provided by Decaudin et al. [DJW*06], and on our own observations, we note that most folds are formed from generalized cylinders, with roughly sinusoidal cross-sections. This observation applies not only to the regular sine wave folds (Figure 3 (left)), but to the diamond/twisted diamond folds as well, as those can be separated into an interior valley region which is nearly flat, and the surrounding fold ridges formed by a union of generalized cylinders (Figure 3 (center) and (right)). These generalized cylinders tend to be fairly straight, with low-curvature axes and nearly constant cross-sections. We use these observations to estimate fold shape and location based on the input video (Figure 2 (center)) and to model the folds based on these estimates (Figure 2 (right)).

3.1. Video Based Fold Estimation

Based on the above discussion of fold shape, as well as some radiometric assumptions, we expect folds to show up in video frames as very specific edge structures robustly identifiable using standard edge detection filters. Specifically, we assume that the fabric of the garment is approximately Lambertian, and that the illumination is diffuse, i.e. uniform across all incident directions. For the purposes of this paper, we also assume garments without strong texture to avoid ambiguities between shading patterns and texture features.

Under the assumptions outlined above, the ridges of fold

Figure 3: *Types of garment folds: sine wave (left), diamond (center), and twisted diamond (right). The diamond and the twisted diamond are formed by four roughly cylindrical ridges each. Top: photos of real folds, bottom: modeling similar fold geometry.*

structures show up as bright areas, whereas the valleys are dark, since they receive light from a narrower solid angle. This model for shading of surface features is known as "dark-equals-deep" (e.g. [LB00, GWM*08]). As a result of this photometric model, one expects to observe image edges roughly corresponding to iso-lines (i.e. elevation contours) on the fold geometry. Moreover, due to the specific structure of garment folds, as outlined above, we expect each fold component (a generalized cylinder) to be observed as a pair of roughly parallel edges, flanking a ridge (bright) or a valley (dark) area (see Figures 2 (center) and 4). We refer to edges that fit these criteria as *fold edges*. Section 4 details the fold-edge extraction process.

Note that these properties are reasonably robust under moderate deviations from our idealized radiometric model (e.g. Figure 3 (top)). The assumption of an approximately Lambertian material holds well for many common fabrics including cotton and wool, although it may be violated by some synthetic fibers. Fortunately, the "dark-equals-deep" model is only violated for very shiny materials that are very rare in garments. Deviations from the ideal diffuse illumina-
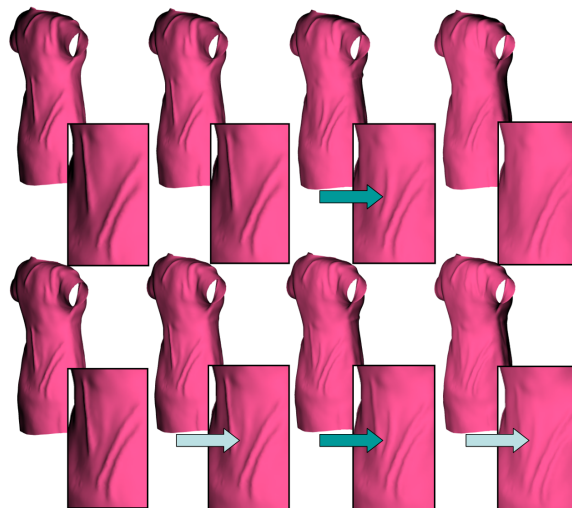
(a)   (b)

(c)   (d)
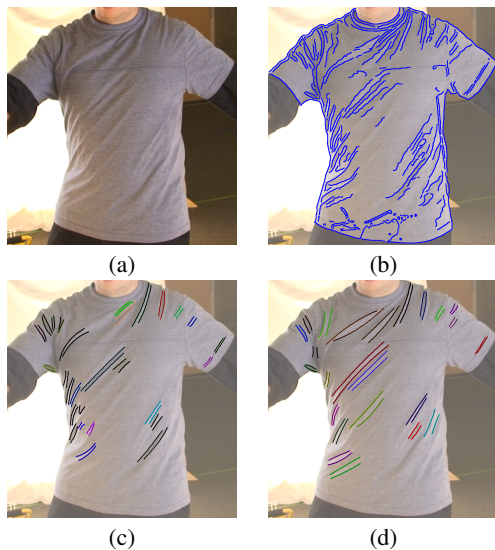
Figure 4: *Fold edge extraction: input image (a), raw edges (b), and paired fold edges (c) extracted from the image at a single resolution (edges not projected to the garment are discarded). (d) Fold edges after combining extraction results from several image resolutions and filter sizes.*

Figure 5: *In a frame-by-frame deformation setup (top) one fold pops up in a single frame (see arrow) while another vanishes abruptly. Using space-time deformation (bottom) folds both form and disappear gradually.*

tion scenario introduce a directional component that causes the brightest and darkest regions to be somewhat offset from the true ridges and valleys, respectively. In this situation, our approach will produce a fold that it slightly shifted from the true location, but has the correct size and orientation. As such, the overall appearance of the final garment should be very similar to the original. We also note that uniform diffuse illumination is the preferred lighting setup for movie shots and many capture setups, as it eliminates strong shadows, which would hinder reconstruction (e.g. [BPS*08]). So we expect deviations from the idealized setting to be minor.

The pairing requirement imposes a very strong constraint on the detection mechanism, which lets us effectively overcome the simplicity of the photometric model. Furthermore, the edge pairing requirement also helps eliminate detected edges that correspond to texture on the garment rather than to folds. As a result, we have observed very few false positives in our processing, where non-folds were classified as folds, or where the edge lines did not faithfully reproduce the fold geometry. We noticed that even in the case of a false positive, the resulting wrinkled models appear realistic as the placed folds satisfy the typical shape characteristics.

Leveraging Temporal Information: While folds typically persist across a sequence of frames, they can be more recognizable in some than in others. Since edge detection is by its nature discrete, it can potentially miss folds when they are less noticeable. Our method overcomes this problem, by utilizing the observation that garment motion is smooth across time. Our space-time deformation method, described next, explicitly enforces motion continuity. Thus, if a fold is detected in some, but not all frames of a sequence, the deformation effectively completes it in the frames where it is miss-

ing (Figure 5). This space-time leveraging allows us to use a conservative set of parameters that avoid false positives when extracting the fold edges.

3.2. Fold Modeling

We incorporate the detected image folds into the geometry of the garments using a novel dynamic deformation setup. We recall that fold edge pairs may correspond to parts of more complex fold structures. In order to capture such structures, we solve for a global deformation of a garment shape using all detected pairs of fold edges at once, rather than adding them one by one.

To generate the folds we bend the surface along the edges, using rotation driven deformation. We prefer to use a rotation-driven approach over the more common positional controls, since folds are characterized predominantly by changes in surface normals, and such changes are more naturally represented by rotations. Moreover, the rotation-based approach removes the need to detect actual depth and shape information from images, which could be quite challenging, as mentioned in our discussion of related work. Instead, we use the rotations as inputs to a stretch-minimizing deformation, which computes the positions of the mesh vertices in a way that conforms with the low-stretch property of garments, resulting in realistic looking folds.

Most recent deformation approaches minimize changes in mean curvature rather than stretch and are thus not suitable for our needs (e.g. [SLCO*04, LSLCO05]). Existing methods for stretch-minimizing deformation, such as the work by Kilian et al. [KMP07] are non-linear and computationally demanding. Converting them into a setup where we need to

solve for deformation across multiple frames at once could be prohibitively expensive.

Hence, to generate realistic dynamic wrinkles we introduce a new space-time quasi-isometric deformation method (Section 5). Key to our method is the observation that quasi-isometric deformation implies that as the mesh deforms the triangles shape is preserved. We therefore search for as-rigid-as-possible per-triangle transformations and shared vertex positions that satisfy these transformations. We use an iterative solution procedure which amounts to solving a number of very sparse linear systems, allowing us to efficiently wrinkle very large meshes (>100K triangles) simultaneously across numerous frames. The high mesh resolution is necessary to capture the fine fold geometry.

The final result of the algorithm is a sequence of realistic looking wrinkled meshes with the folds consistent across time (Figure 2 (right)).

4. Video-Based Fold Edge Extraction

To extract likely folds at each time step, we select the best front and back views of the garment. We then extract likely folds from each of the two still images separately using the photometric model described in Section 3.1, by first detecting edges present in the images and then filtering out edges that do not satisfy our fold edge criteria (Figure 4).

View Selection: For every frame in the given multiview video input, we automatically select the best front and back views. As a preprocess, the user selects two mesh vertices, one on the front and one on the back, in a single frame. Our method then tracks these vertices across all frames using the consistent mapping between the frame geometries provided by the original capture process. It computes per-frame stable normals at the vertices using a large (eight rings in our examples) neighborhood and selects the best, most orthogonal views, based on the dot product of the normals with the camera view vectors.

Edge Extraction: Our method uses the Canny edge detector [Can86] to extract raw image edges. Recall that fold components are expected to have low curvature in the direction of the fold axis. Since fold edges follow contours of the fold, they should inherit this property. Hence a fold edge can be approximated by a low degree polynomial with bounded curvature. We use quadrics to fit each raw edge, recursively splitting the edges if the fitting error or the curvature of the quadric along the edge are too high.

Edge Orientation: We expect fold edges to represent fold contours, and thus have a darker, i.e. deeper side (recall that we use the "dark-equals-deep" model). Thus, after extracting the smooth edges, we orient them based on the image gradient along the edge. We use the convention of selecting the orientation such that the darker side of the image is to the left of the edge. Edges with no clear orientation are discarded. We also discard edges which exhibit large difference in hue across the edge, as those typically capture texture rather than shape.

Merging: Edge detection often tends to break continuous fold contours into multiple disjoint edges. We recover continuous contours by merging edges if they are oriented consistently, their end vertices are close to one another, and the combined edge can still be fitted by a low curvature quadric.

Pairing: The most significant step of our processing is the pairing, which detects parallel edges with opposite orientation (Figure 4(c)). As noted earlier, garment folds are formed by a union of generalized cylinders, thus opposite contours on these cylinders are expected to be roughly parallel. Therefore, pairs of edges representing such contours should satisfy our criteria of parallelism and opposite orientation. We discard all edges that have no pairing as those are not likely to represent or form realistic folds. To test if two edges can be paired we check if they are parallel, by comparing tangents between closest points along the quadric curves. Since folds are expected to be relatively narrow, we also constrain the average and maximal distance between the paired edges.

The pairing process must address potential pairing ambiguities, as some edges may have multiple pairing choices. To resolve those, we prioritize pairings that are more likely to be real and significant folds. We observe that on real garments there are more ridge than valley folds. This observation leads us to prefer ridge pairs, namely ones where the area between the edges is brighter, i.e. the edges are to the right of one another. We also prioritize pairing of longer edges, as those indicate more influential folds.

Multi-Resolution Processing: To maximize the number of folds detected, the detection and filtering steps are done on several image resolutions and with different Gaussian filter sizes in the Canny edge detector. The results are then combined together, creating a union of edge pairs (Figure 4(d)). When pairs from different layers overlap, we select the best one using the same criteria as for pairing. The resulting set of pairs represents believable folds in the input images, and is used as input for wrinkling the garment surface.

5. Space-Time Deformation

We now discuss our space-time deformation method for smoothly wrinkling the garment surface over time, using the folds estimated from the video sequence.

5.1. Control Mechanism

The deformation is controlled by the fold edges extracted from the individual frames, which indicate where the sides of the newly formed folds should be. To wrinkle the surface we therefore bend it along the edges by rotating the corresponding, *anchor* triangles, counterclockwise around the edges. This rotation is consistent with the assumption that the left, darker, side of each fold is expected to be deeper than the right one, i.e. closer to the body of the wearer. To locate the anchor triangles we project the computed fold edges onto the corresponding garment frames, using the available camera calibration (Figure 6 (left)). We discard fold edge pairs where one of the edges projects outside the garment or
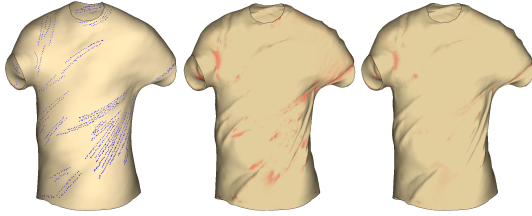
Figure 6: *Deformation: (left) anchor triangles (fold edges projected to input garment); (center) vertex positioning result; (right) garment after stretch reduction. The coloring (center, right) shows per-triangle stretch. The iterations reduce the maximal per-triangle stretch from 8% to 4%.*

onto its silhouette. We tested several models for setting the rotation angle, including using the gradient along the edge as a measure of the amount of rotation, but found no correlation between the two. In empirical testing we achieve the best results when setting the rotation angle to be inversely proportional to the width of the fold edge pairs with the angle varying from $60°$ for narrow folds to $30°$ for wide ones.

5.2. Iterative Space-Time Deformation

To obtain smooth garment motion across time, we solve for the new shape of the mesh across multiple time frames. Quasi-isometric deformation implies preserving the shape of the triangles during deformation, while allowing the dihedral angles between them to change. Solving simultaneously for as-rigid-as-possible per-triangle transformations and shared vertex positions that satisfy these transformations, leads to a non-linear formulation, which would be prohibitive to optimize in space-time. Therefore, similar to [LSLCO05, SP04], our formulation decouples the rotation computation from the vertex positioning. We first compute optimal triangle rotations for all triangles based on the input rotations across all frames. We then compute vertex positions consistent with the rotations, obtaining wrinkled meshes that satisfy most of our requirements, but which can exhibit non-negligible stretch near the newly formed folds (Figure 6 (center)). We therefore iteratively update the rotations and the positions to reduce the stretch to an acceptable level (Figure 6 (right)). We found that the stretch reduction can be accomplished using rotations computed per-triangle, without a need for solving yet another global system to compute them. In other words, the per-triangle update combined with the vertex repositioning satisfies the temporal and spatial requirements we impose on the folds, without explicitly enforcing them. We now describe the three steps in more detail.

5.3. Space-Time Rotation

Given the specified *anchor triangles* and the corresponding rotation angles for each time frame, we solve for the rotations of the mesh triangles in multiple frames simultaneously. Recall that since we operate on a space-time captured mesh, we have the same connectivity in all frames, thus we can explicitly compute the rotations for all frames at once.

The space-time solution is key to making the rotations and hence the final deformation consistent across time, enforcing smooth garment motion (Figure 5).

For the folds to look realistic, in addition to smooth change across time, rotations have to satisfy two additional requirements. First, we expect garments to be smooth, thus adjacent triangles should rotate in roughly the same way. Second, we expect folds to be fairly local, and thus triangles spatially or temporally distant from the anchors should rotate very little, if at all.

To propagate the rotations across the meshes we describe the per-triangle rotations as convex combinations of anchor rotations and the identity rotation. The inclusion of the identity mitigates the anchor influence away from the folds. Our challenge is to compute the per-triangle, convex combination blending weights $\mathbf{w}_i^t$, where i goes over the mesh triangles and t goes over the time steps. Each weight is a vector with entries corresponding to the anchors. The sum of the weights is less or equal to one, with the weight assigned to the identity rotation completing the sum to be exactly one, thus ensuring the convex combination property. We solve for the weights by minimizing the following quadratic functional,

$$\arg\min_{\mathbf{W}} \ c_s \sum_{t,e=(i,j)} \|\mathbf{w}_i^t - \mathbf{w}_j^t\|^2 + c_t \sum_{i,t} \|\mathbf{w}_i^t - \mathbf{w}_i^{t-1}\|^2 +$$
$$c_i \sum_{(i,t)\notin A} \|\mathbf{w}_i^t\|^2 + c_a \sum_{(i,t)\in A} \|\mathbf{w}_i^t - \mathbf{I}_i^t\|^2,$$

where $e = (i,j)$ are the edges shared by triangles i and j, A is the set of anchor triangles, and $\mathbf{I}_i^t$ is a vector with one in the entry that corresponds to the anchor (i,t) and zero everywhere else. The first two terms of the functional ensure rotation continuity across space and time. The third term provides for rotation mitigation away from the anchors, and the last term constrains the anchor rotations. The coefficients which control the influence of each individual term were set to $c_s = 0.9, c_t = 0.5, c_i = 0.1, c_a = 10$ in all our examples. Our input meshes are fairly uniform, thus our formulation does not weigh triangles or edges differently based on their area or length. Having uniform weights improves the conditioning of the linear system solved, and thus speeds up convergence. To perform the actual blending, given the weights, we follow Popa et al. [PJS06], and use the transformation algebra presented by Alexa [Ale02].

Numerical Solution: Minimizing the above functional amounts to solving a linear system with multiple right hand sides, corresponding to the individual entries in the weight vectors. Our matrix is positive definite and very sparse (approximately six non-zero entries per row), thus we chose to use the conjugate gradient solver to compute the solutions. Since we do not need very accurate weights, we use a fairly lax convergence tolerance of 10^{-3}, resulting in the solver converging in under ten iterations. The temporal influence of anchors reduces significantly over time, thus rather than solving for all time steps at once we use a staggered approach significantly reducing both memory footprint and

runtime. We solve for overlapping sequences of ten frames at a time, with a time shift of seven frames between them and blend the rotations obtained in the three overlapping frames, to maintain rotation smoothness. The rotations generated with this staggered approach are practically identical to those generated using a global solution.

5.4. Vertex Positioning

Given the new rotations, we solve for vertex positions using a least-squares solution, which aims to preserve the computed rotations as the per-triangle transformation gradients. We label the vertices of a given triangle as $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ and add a virtual vertex $\mathbf{v}_4$ computed by offsetting $\mathbf{v}_1$ by the triangle normal. We then define the local triangle frame $\mathbf{V}_i^t$ as $\mathbf{V}_i^t = (\mathbf{v}_4 - \mathbf{v}_1, \mathbf{v}_4 - \mathbf{v}_2, \mathbf{v}_4 - \mathbf{v}_3)$. To obtain the vertex positions, we minimize the following quadratic functional [SP04],

$$\arg\min_{\tilde{\mathbf{v}}} \sum_{i,t} \|\tilde{\mathbf{V}}_i^t (\mathbf{V}_i^t)^{-1} - \mathbf{R}_i^t\|_F^2, \tag{1}$$

where $\tilde{\mathbf{v}}$ are the new positions of the vertices, $\mathbf{V}_i^t$ and $\tilde{\mathbf{V}}_i^t$ the local frames before and after the deformation, and $\mathbf{R}_i^t$ are the previously calculated triangle rotations. Since this objective functional has no temporal component it can be minimized independently for each frame. We use a direct solver (SuperLU [DEG*99]) to solve the corresponding linear system, fixing one of the mesh vertices to remove the redundant translational degrees of freedom.

5.5. Stretch Reduction

After the positioning step, the deformed per-frame meshes satisfy the anchor rotations (in a least-squares sense) and are consistent across time. The overall stretch of the output meshes, compared to the inputs, is typically very low, as the rotation step bends the surface along feasible fold-lines while the vertex repositioning step attempts to preserve the rotation-only gradients. A typical example shown in Figure 6 (center) has L_2 stretch [SSGH01] of 1.00036 (the optimum is one). However near the folds the outputs often exhibit local stretch which is higher than what many fabrics can tolerate, and one that can interfere with the visual consistency of the folds across time.

To reduce stretch to an acceptable level, we search for updated rotation-only gradients, which are more consistent across adjacent triangles, and thus lead to less triangle stretch during vertex repositioning. One way to obtain new rotations would be to formulate and solve a new space-time rotation optimization problem with the set of spatial and temporal requirements listed earlier. However, we found that such a global solution is unnecessary in our setup. We note that the rotations implied by the difference between the triangle normals before the deformation $\mathbf{n}$ and after $\tilde{\mathbf{n}}$ are close enough to our input rotations to maintain the desired properties. At the same time, these rotations are more feasible to satisfy in Equation 1, leading to a smaller minimum and thus

less stretch. We therefore simply use them as the new per-triangle rotations $\mathbf{R}_i^t$ (rotation axis is $\mathbf{n} \times \tilde{\mathbf{n}}$ and the angle is $\arccos(\mathbf{n} \cdot \tilde{\mathbf{n}})$) and reapply the vertex positioning procedure. We found that one or two iterations of stretch reduction were sufficient to reduce the maximal per-triangle stretch to under 5%, at which point it becomes visually unnoticeable. In the example in Figure 6 (right) the maximal per-triangle stretch in the final model is 4%, and the L_2 stretch is 1.00017.

6. Results

We tested our method on a variety of garments generated by three state-of-the-art capture techniques [BPS*08, VBMP08, dAST*08]. Figures 1 through 8 demonstrate the method's behavior on five diverse garment models captured by Bradley et al. [BPS*08] using a sixteen camera setup: two T-shirts, two dresses and a jacket. In all the examples our method adds believable folds to the models. As demonstrated in the attached video, the generated folds change smoothly across time, emulating realistic fold behavior.

We also tested the method on data from Vlasic et al. [VBMP08] and de Aguiar et al. [dAST*08] (Figure 9). Since de Aguilar et al. preserve fine details from the template throughout the sequence (Figure 9 (left,middle)) we smooth those out, prior to applying our dynamic wrinkling algorithm. Notice how the new folds vary across time, while the original ones remain fixed (see video for more examples). The video footage used by these two methods is of lower resolution than that of Bradley et al., and therefore the number of folds visible in the footage and captured by our method is lower. Since both methods heavily rely on silhouettes, they pay less attention to lighting, thus providing conditions that are far from optimal for our approach. Despite these conditions, our method manages to extract sufficient fold information to generate believable dynamic folds. Both setups use only eight cameras, raising a concern of temporal consistency when the fold extraction switches between different camera views. Nevertheless, our technique reconstructs realistic folds even in these sub-optimal conditions with no noticeable artifacts. In situations where the camera views are too far apart, temporal inconsistencies can occur.

Runtimes: The most time consuming step of our method is the space-time rotation computation. For a 100K triangle mesh we typically have on the order of 2000 anchors per-frame, so computing the weights for a block of 10 frames requires solving a linear system of approximately a million variables 2000 times for each frame. This takes about one minute per frame on a 27 Intel Xeon 3GHz CPU cluster. The rest of the computation is done on a single CPU and takes about two minutes per frame which are split roughly equally between edge detection and vertex positioning (including the stretch reduction).

Parameters: Our estimation of fold edges requires a few parameters that depend on the garment fabric and cut. The parameters are maximal distance and angle between paired fold edges, and Canny edge detector radii. In our implementation those are set by the user once per garment. In all the

examples the angle was set to $45°$, the other parameters are listed in Table 1. In the future it might be possible to learn those from fabric parameters or example inputs.

	Pink Dress	Blue Dress	T-Shirts	Red Jacket	Dancer and capoeira kick
Maximal distance	∞	100	100	100	12
Canny radii	1.5, 2.5, 3.5	1, 2, 3	1, 2, 3	0.5, 2, 4	0.5, 1, 1.5

Table 1: *Fold edge parameters*

7. Conclusions

We have presented the first, to the best of our knowledge, method for augmenting the realism of captured garments by introducing believable dynamic wrinkles. As demonstrated, our method works robustly on numerous input examples, generated by different capture methodologies.

Our algorithm does not aim at exact reproduction of fold shapes observed in the video footage. It might be possible to increase fold capture accuracy by combining our deformation approach with a more sophisticated photometric model using information from multiple cameras simultaneously.

Our current method operates directly on meshes, and thus the size of the folds it can capture is bounded from below by the mesh resolution. Since we already use meshes with around 100K triangles, increasing the resolution further is not really feasible. One alternative could be to combine geometric deformation with normal space processing, developing a mixed geometry and bump map modeling approach.

References

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

[ATD*08] AHMED N., THEOBALT C., DOBREV P., SEIDEL H.-P., THRUN S.: Robust fusion of dynamic shape and normal capture for high-quality reconstruction of time-varying geometry. In *CVPR 2008* (2008).

[BBA*07] BICKEL B., BOTSCH M., ANGST R., MATUSIK W., OTADUY M., PFISTER H., GROSS M.: Multi-scale capture of facial geometry and motion. *ACM Trans. Graph. 26*, 3 (2007), 33.

[BMF03] BRIDSON R., MARINO S., FEDKIW R.: Simulation of clothing with folds and wrinkles. In *SCA '03* (2003), pp. 28–36.

[BPS*08] BRADLEY D., POPA T., SHEFFER A., HEIDRICH W., BOUBEKEUR T.: Markerless garment capture. *ACM Trans. Graph. 27*, 3 (2008), 99.

[Can86] CANNY J.: A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell. 8*, 6 (1986), 679–698.

[CGW*07] CUTLER L. D., GERSHBEIN R., WANG X. C., CURTIS C., MAIGRET E., PRASSO L., FARSON P.: An art-directed wrinkle system for CG character clothing and skin. *Graphical Models 69*, 5-6 (2007), 219–230.

[CMT05] CORDIER F., MAGNENAT-THALMANN N.: A data-driven approach for real-time clothes simulation. *Computer Graphics Forum 24*, 2 (2005), 173–183.

[dAST*08] DE AGUIAR E., STOLL C., THEOBALT C., AHMED N., SEIDEL H.-P., THRUN S.: Performance capture from sparse multi-view video. *ACM Trans. Graph. 27*, 3 (2008), 98.

[DEG*99] DEMMEL J. W., EISENSTAT S. C., GILBERT J. R., LI X. S., LIU J. W. H.: A supernodal approach to sparse partial pivoting. *SIAM J. Matrix Analysis and Applications 20*, 3 (1999), 720–755.

[DJW*06] DECAUDIN P., JULIUS D., WITHER J., BOISSIEUX L., SHEFFER A., CANI M.-P.: Virtual garments: a fully geometric approach for clothing design. *Computer Graphics Forum 25*, 3 (2006), 625–634.

[GHF*07] GOLDENTHAL R., HARMON D., FATTAL R., BERCOVIER M., GRINSPUN E.: Efficient simulation of inextensible cloth. *ACM Trans. Graph. 26*, 3 (2007), 49.

[GWM*08] GLENCROSS M., WARD G. J., MELENDEZ F., JAY C., LIU J., HUBBOLD R.: A perceptually validated model for surface depth hallucination. *ACM Trans. Graph. 27*, 3 (2008), 1–8.

[HBVMT99] HADAP S., BANGERTER E., VOLINO P., MAGNENAT-THALMANN N.: Animating wrinkles on clothes. In *VIS '99* (1999), pp. 175–182.

[HVB*07] HERNÁDEZ C., VOGIATZIS G., BROSTOW G., STENGER B., CIPOLLA R.: Non-rigid photometric stereo with colored lights. In *Proc. ICCV* (2007).

[KA08] KASS M., ANDERSON J.: Animating oscillatory motion with overlap: wiggly splines. *ACM Trans. Graph. 27*, 3 (2008), 28.

[KMP07] KILIAN M., MITRA N. J., POTTMANN H.: Geometric modeling in shape space. *ACM Trans. Graph. 26*, 3 (2007), 64.

[LB00] LANGER M. S., BÜLTHOFF H. H.: Depth discrimination from shading under diffuse lighting. *Perception 29* (2000), 649–660.

[LSLCO05] LIPMAN Y., SORKINE O., LEVIN D., COHEN-OR D.: Linear rotation-invariant coordinates for meshes. *ACM Trans. Graph. 24* (2005), 479–487.

[PH03] PRITCHARD D., HEIDRICH W.: Cloth motion capture. In *Proc. Eurographics* (2003), pp. 263–271.

[PH08] PARK S. I., HODGINS J.: Data-driven modeling of skin and muscle deformation. *ACM Trans. Graph. 27*, 3 (2008), 96.

[PJS06] POPA T., JULIUS D., SHEFFER A.: Material-aware mesh deformations. In *SMI '06* (2006).

[SB08] SCHECHTER H., BRIDSON R.: Evolving sub-grid turbulence for smoke animation. In *Prof. SCA* (2008).

[SLCO*04] SORKINE O., LIPMAN Y., COHEN-OR D., ALEXA M., RÖSSL C., SEIDEL H.-P.: Laplacian surface editing. In *Symposium on Geometry Processing* (2004), pp. 179–188.

[SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. *ACM Trans. Graph. 23*, 3 (2004), 399–405.

[SSGH01] SANDER P. V., SNYDER J., GORTLER S. J., HOPPE H.: Texture mapping progressive meshes. In *SIGGRAPH* (2001), pp. 409–416.

[SSK*05] SCHOLZ V., STICH T., KECKEISEN M., WACKER M., MAGNOR M.: Garment motion capture using color-coded patterns. In *Proc. Eurographics* (2005), pp. 439–448.

[SZT*08] SHI X., ZHOU K., TONG Y., DESBRUN M., BAO H., GUO B.: Example-based dynamic skinning in real time. *ACM Trans. Graph. 27*, 3 (2008), 29.

[VBMP08] VLASIC D., BARAN I., MATUSIK W., POPOVIĆ J.: Articulated mesh animation from multi-view silhouettes. *ACM Trans. Graph. 27*, 3 (2008), 97.

[WCF07] WHITE R., CRANE K., FORSYTH D.: Capturing and animating occluded cloth. *ACM Trans. Graph. 26*, 3 (2007), 34.
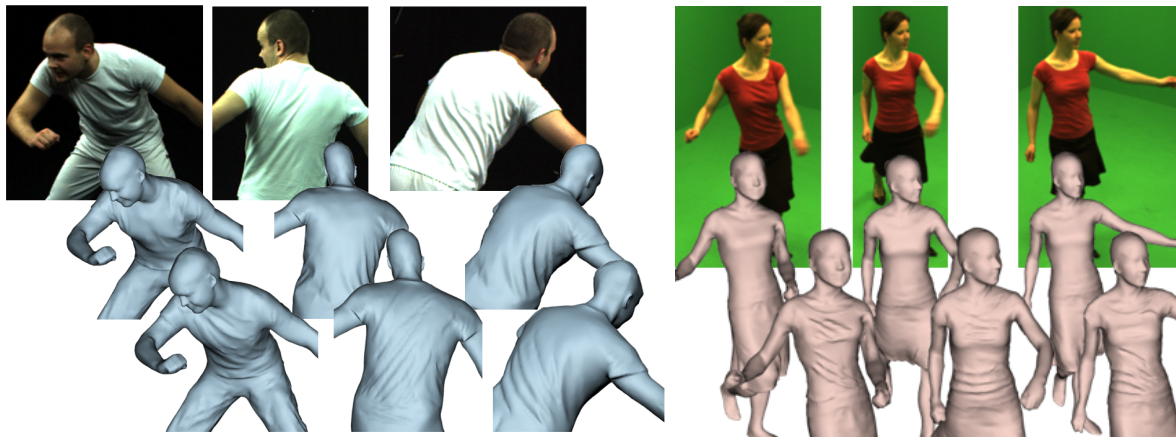
Figure 7: *Wrinkling dresses captured by [BPS*08], (top) video input, (middle) captured geometry, (bottom) our results.*

Figure 8: *Wrinkling a t-shirt and a jacket captured by [BPS*08], (top) video input, (middle) captured geometry, (bottom) our results.*

Figure 9: *Wrinkling outfits captured by [dAST*08] (left) and [VBMP08] (right): (top) video input, (middle) captured geometry, (bottom) our results. The captured geometry on the left contains template folds which remain constant throughout the sequence. In contrast, the folds in our outputs change dynamically following the character's motion.*