Capturing Real Folds in Cloth

Ryan White
David Forsyth
Jai Vasanth

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2006-10

http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-10.html

February 2, 2006

Copyright © 2006, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Capturing Real Folds in Cloth

Ryan White, D.A. Forsyth
{ryanw,daf}@cs.berkeley.edu

February 2, 2006

Abstract

Cloth is difficult to simulate because it forms small, complex folds. These folds make cloth configuration difficult to measure. Particular problems include fast motion (ruling out laser ranging methods), the necessity for high resolution measurement, the fact that no viewing direction can see into the folds, and the fact that many points are visible with either small baseline or in only one view.

We describe a method that can recover high resolution measurements of the shape of real cloth. Our method uses multiple cameras, a special pattern printed on the cloth, and high shutter speeds to capture fast motions. Cameras are calibrated directly from the cloth pattern. Folds result in local occlusion effects that can make identifying feature correspondences very difficult. We build correspondences between image features and material coordinates using novel techniques that apply approximate inference to exploit both local neighborhood information and global strain information. These correspondences yield an initial reconstruction that is polished using a combination of bundle adjustment with a strain minimization to get very good 3D reconstructions of points seen in multiple views. Finally, we use a combination of volume occupance cues derived from silhouettes and strain cues to get very good 3D reconstructions of points seen in just one view. Our observations provide both a 3D mesh and a parameterization of that mesh in material (or, equivalently, texture) coordinates.

We demonstrate that our method can capture fast cloth motions and complicated configurations using a variety of natural cloth configurations, including: a view of a bent arm with extensive, complex folds at the elbow; a pair of pants moving very fast as the wearer jumps; and cloth shuddering when it is hit by dropped coins.

1 Overview

Cloth modelling is an important technical problem, because people are interesting to look at and most people wear clothing. Traditionally, cloth models have been built using simulation, but recent work has started to focus on capturing cloth in the real world [13, 9, 16, 4]. This body of work views the task as one of structure-from-motion — using multiple views of a 3D objects to reconstruct its shape. By printing a color-coded pattern on the cloth, the reconstruction process becomes easier, and these papers show that systems capable of reconstructing large garments are reasonable for the task. We provide methods to deal with the technical challenges that limit current approaches: recovering folds in considerable geometric detail and handling the complex geometry of real garments that moves quickly. Like previous methods, we handle reconstruction frame by frame and focus our attention on single frame reconstruction. Viewing the problem from the single frame perspective, capturing fast motion is easy: large amounts of light and fast shutter speeds create crisp images that can be processed using the same techniques as slower movement.

There are four reasons that good single frame reconstruction is hard. Reconstructions need to be at **high resolution** to capture the relatively small folds that are so characteristic of cloth. To obtain this high spatial resolution, we print a pattern of thousands to tens of thousands of fine colored triangles. Each triangle in this pattern is a different color, though some color differences are subtle enough that color alone is insufficient to identify the triangle.

This very large number of triangles means determining **correspondence** between triangles in a given image and triangles on the pattern (and so, by extension, between images) is difficult. There is a second, very important, difficulty. We cannot simply use the neighbors of a given triangle to identify it (as earlier methods have done), because small folds in the cloth cause triangles to disappear in an image (see figure 2, which shows a fold about the scale of a single triangle across). Instead, we rely on two novel strategies: First, we use careful color processing to identify triangles with the minimum of ambiguity (section 3). Second, we use a probabilistic inference method to enforce two constraints: no triangle on the cloth appears twice in any single image, and triangles that are close together on the cloth cannot be far apart in the image.

This is because cloth admits only limited strain. Cloth tends to resist even small strains quite strongly, and at larger strains the fibers in the weave lock and further strain is resisted extremely strongly; this property is an important source of stiffness in cloth simulation. It provides a powerful cue that has no analogy in conventional structure from motion.

Measurement noise is inescapable, not least because pattern elements that can be localized to very small fractions of a pixel are small and tend to be easy to miss. Because many points are seen with a relatively small baseline (figure 4 explains why), small errors in localization can lead to large errors in 3D and so to a strained reconstruction. Remarkably, as figure 5 and section 2 show, adding a strain term to bundle adjustment results in a 3D reconstruction with a plausible strain distribution at almost no increase in reprojection error.

Finally, there are **many points seen in only one view**. This cannot be resolved by simply adding large numbers of cameras, because most such points would be viewed from so short a baseline as to yield no depth measurement (figure 6). This typically occurs in garments such as pants and shirts where one appendage occludes another in most views. In practical systems, roughly one third of the points are observed in one view (figure 12). We integrate silhouette cues with single view and strain constraints to reconstruct points seen in a single image (section 4). Experimentally, we have found the accuracy of these points to be very good: with a median error of 3.3 mm on a pair of pants (roughly 0.3% of the object size).

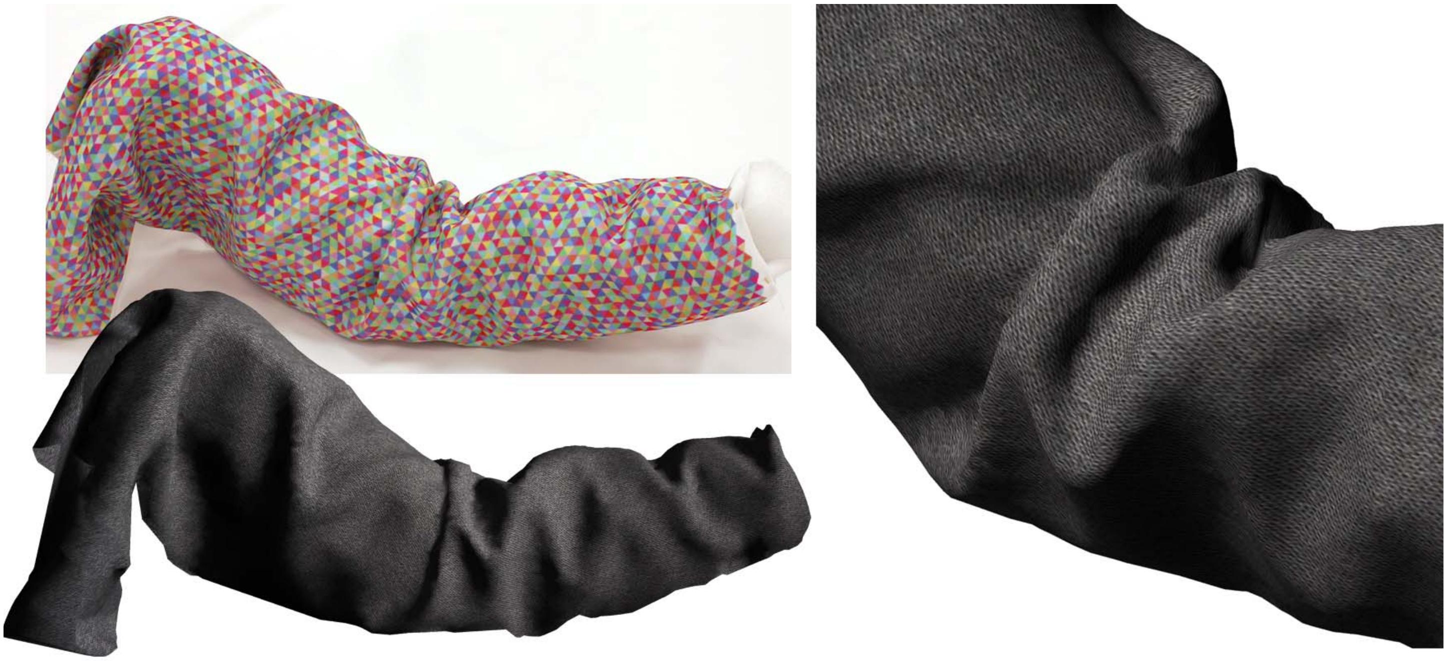

Figure 1: High resolution meshes of cloth can be captured from multiple images of an actual piece of cloth. We print a pattern on the cloth that uses an unlimited number of colors to build a set of features that can be easily recognized in challenging conditions, such as the folds in the elbow of this figure. The folds in the crease of the elbow on the right are the shape of actual folds in real cloth at a level of detail that is not easily simulated. This reconstructed mesh is made of 7,557 vertices comprising 15,036 triangles and has a reprojection error (a measure of the alignment between the model and image data) of 0.23 mm (230 μm) or 0.05% of the object size.

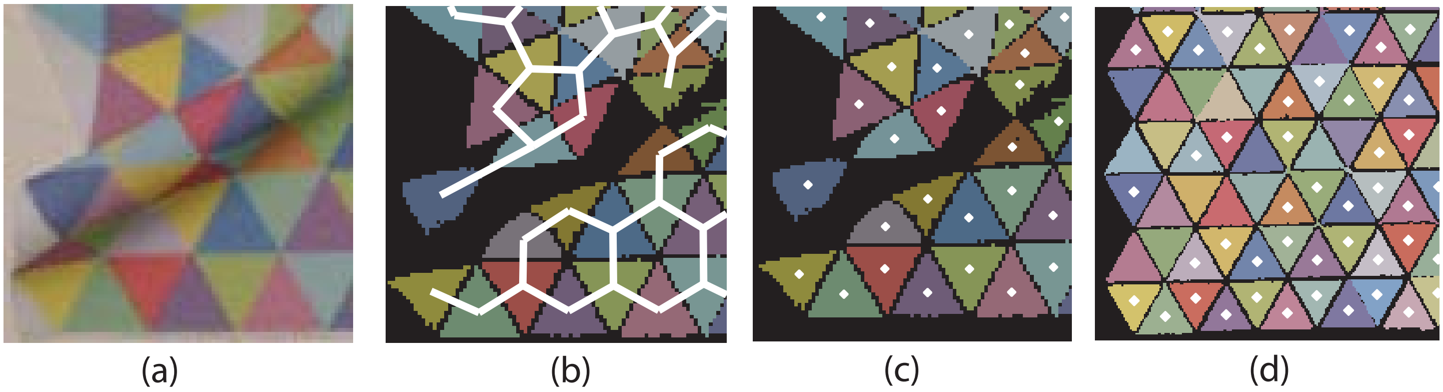

Figure 2: The biggest difficulty in capturing cloth is folds – where context makes it difficult to determine what is going on. We build reconstructions from correspondences between views by identifying the unique identity of each triangle viewed in an image. We use the context of each triangle to decode its identity: in flat regions, the local structure is visible and the process is simple. In folded regions, however, errors are inevitable and we must use strain cues to determine the identity. The original image in this example **(a)**, has folds that occlude many triangles. Our link structure **(b)** labels triangles that are neighbors in the image. While neighbors in flat regions of the image are neighbors on the surface of the cloth, on the fold the links are incorrect. Using belief propagation (section 3), we produce the accurate labeling depicted in **(c)** and **(d)**. Each point drawn on the image **(c)** corresponds to a point drawn in the domain of the cloth **(d)**. We have correctly identified triangles where the fold has disrupted the neighborhood structure and can identify what portions of cloth are missing from view.

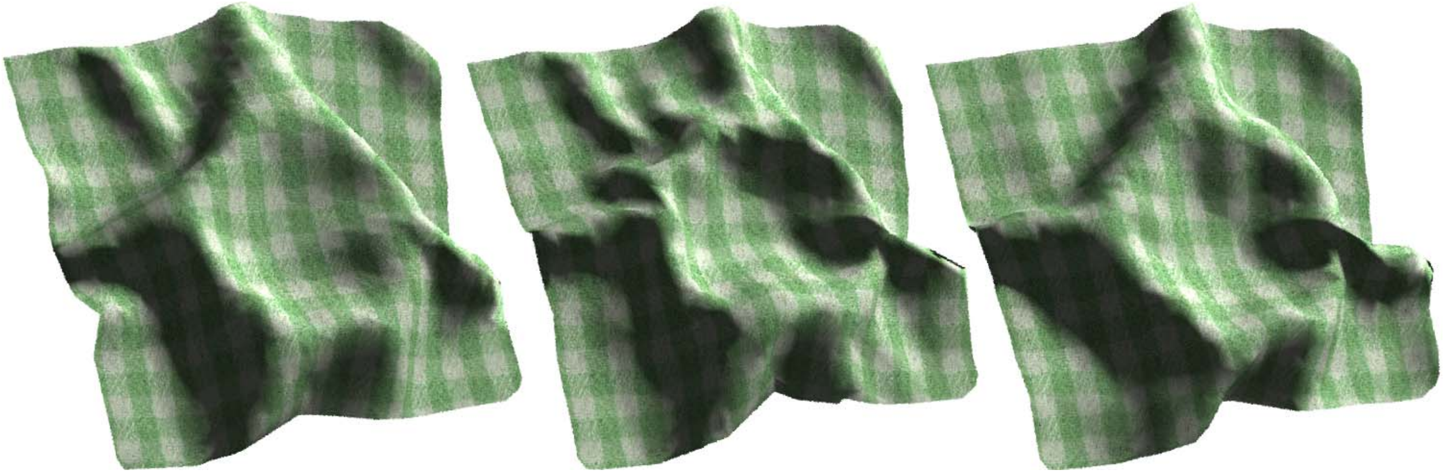

Figure 3: Fast motions, such as the ripple of cloth after a sudden impact with a coin, are easily captured using bright lights and a fast shutter speed. The cloth is draped over a shallow bowl and the coin lands in the second frame. The frames shown are taken 42 ms apart (or 1/24 of a second) — just how quickly the folds from impact appear, ripple through the cloth and then disappear. However, the result of the impact has a lasting change: the folds along the top start as two separate folds but merge into one on impact. The dynamics of this motion are too fast to be captured by a laser range scanner.

1.1 Background

There is a substantial literature on cloth modelling; only a superficial introduction is possible in space available. Cloth is difficult to model for a variety of reasons. It is much more resistant to stretch than to bend: this means that dynamical models result in stiff differential equations (for example, see [2, 18]; the currently most sophisticated integration strategy is [6]) and that it buckles in fine scale, complex folds (for example, see [5]). Stiff differential equations result in either relatively small time steps — making the simulation slow — or in relatively heavy damping — making the cloth slow-moving and "dead" in appearance. Cloth has complex interactions: it collides with itself and rigid objects; it is driven by forces that are hard to model, including human motion and aerodynamics. Collisions create difficulties because the fine scale structure tends to require large, complex meshes, and resolving collisions can be tricky; for example, careless resolution of collisions can introduce small stretches (equivalently, large increments in potential energy) and so make a simulation unstable (for example, see [3]). A summary of the recent state of the art appears in [11]. While each of these issues can be controlled sufficiently to produce plausible looking simulations of cloth, the process remains extremely tricky, particularly for light, strong cloth (e.g. woven silk), where the difficulties are most pronounced.

Attempts to motion capture cloth probably date to [8], who mark surfaces with a grid and track the deformation of elements of this grid in the image. This work does not report a 3D reconstruction, because the pattern of elements is periodic, meaning that one would have to solve a difficult correspondence problem to obtain a 3D reconstruction. Guskov, Klibanov and Bryant give views of a 3D reconstruction in real time, obtained by printing square elements with internal patterns on the surface, estimating local homographies at each element, then linking these estimates into a surface reconstruction [9]. The resulting surfaces — for a hand, an elbow, and a T-shirt — are fair but noisy. These internal patterns limit the size of the cloth that can be recovered and don't allow measurements of folds with complicated occlusion relationships. They do not use strain constraints.

[12, 13] use a calibrated stereo pair and SIFT feature matches to build a 3D model. They observe that one can obtain a parameterization of this model — which is essential for retexturing — by matching to a flat view of the cloth. Because they use features with structure at fine spatial scales, there are difficulties caused by motion blur, which reduce the accuracy of the match. Their reconstructions are limited by the region viewed by two cameras and they do not handle small folds or significant occlusions. Use of strain constraints is limited.

[15] obtains better surfaces by using optical flow predicted from a deformable model, with matches constrained to produce the correct silhouette. Again, occlusions and complicated folds are missing. [16] use a pseudo-random pattern of colored circles on custom-made garments to reconstruct both a parameterization and geometry. There method is useful for large sections of relatively flat cloth (they separately show a skirt and a shirt on a subject holding up their arms to prevent occlusion), but they choose a cloth that doesn't reveal fine scale folds and constrain their subjects to prevent significant occlusions. Again, their method does not use strain constraints. We generalize their color coded detection method to enhance detection of folds and add volume cues and strain constraints to make the system useful on realistic cloth. Instead of five colors, our pattern uses an unlimited number and contains structure useful for auto-calibration. Second, our approximate loopy belief propagation is a more general form of their seed and growth method – allowing much higher detection rates in complicated geometry.

2 Multi-View Reconstruction

Motion capturing cloth is fairly clearly a structure from motion problem. The area is now very well understood, with comprehensive reviews in two excellent books [7, 10]. Though often combined, structure from motion can be broken into two subproblems: determining the location of the cameras (calibration) and reconstructing the 3D locations of the points from multiple views. We calibrate with a method that uses points and normals to build an orthographic calibration, then enhance this model to include perspective effects [1]. While there are many choices of calibration methods, this method has the advantage that it does

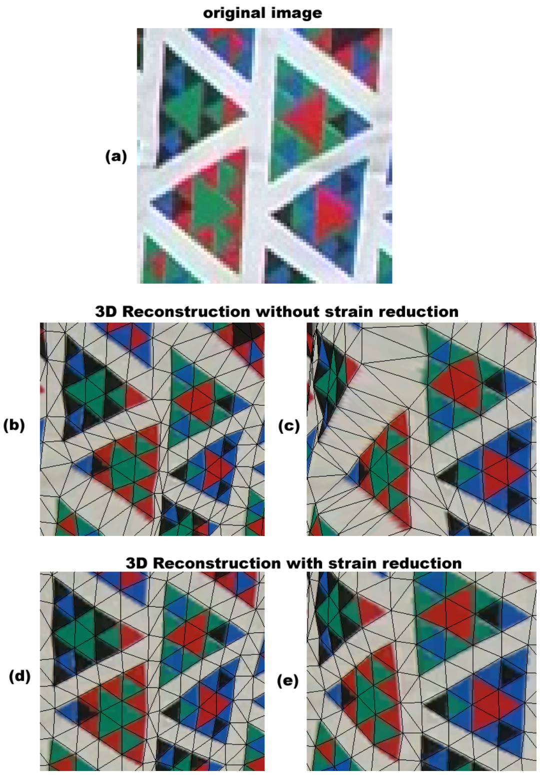
original image
(a)
3D Reconstruction without strain reduction
(b)
(c)
3D Reconstruction with strain reduction
(d)
(e)

Figure 4: Because we have a small number of views of each triangle, minimizing the reprojection error alone only produces an accurate mesh when viewed from similar viewpoints. Images **(b)** and **(d)** are **rendered** views of the **reconstructed** mesh (textured with a frontal view of the flattened cloth) taken from viewpoints similar to the original image **(a)**. However, without strain reduction, novel views do not exhibit cloth-like structure. The reconstructed mesh in image **(c)**, produced by minimizing reprojection error alone, is rendered from a view significantly different from all the original cameras. Note that this results in significant variance in the mesh — indicated by large variations in edge length. Image **(e)** shows a similar rendered view of a reconstructed mesh produced by **simultaneously** minimizing reprojection error and strain (section 2.1). Now, the structure of the mesh is more realistic and true to the original image data.

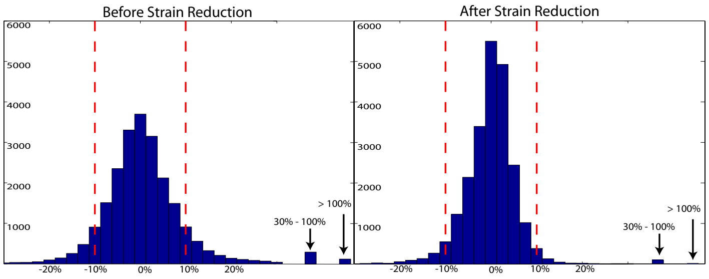

Figure 5: Strain reduction dramatically cuts down on strain in the mesh while preserving agreement with the image data. Using the data from figure 1, the reprojection error (the agreement between the model and the image data) increases a very small amount (10 μm) but creates a substantial reduction in strain. The red lines indicate a strain of 10% – an amount used in simulation as a typical maximum. We physically measured a maximum strain of 15% at 45° to the grain of this particular polyester cloth by stretching it to the point that there was a concern that the cloth would tear.

not require additional equipment (a calibration object) and that it tends to produce accurate calibration for cloth. Once the cameras have been calibrated, 3D reconstruction requires correspondence between frames. Like previous methods, we print a pattern on the cloth that allows easy identification, then use these correspondences to determine 3D locations.

However, recovering the shape of moving cloth is not like the traditional model of structure from motion: the number of views of any one configuration is small and the number of points is very large. One expects to have significantly fewer cameras than a typical structure from motion setup (we use six cameras in most cases; tens of cameras would be practical, but hundreds of cameras would offer no improvement because baselines would be too short). Many of the points seen in multiple views may be seen with a short baseline (figure 6), and so depth measurements may be inaccurate (we discuss points seen in only one view in section 4).

2.1 Combined Strain Reduction and Reconstruction

Now assume we know correspondence between views of points seen in two or more views (section 3 describes how we obtain correspondence). Reconstruction is not straightforward, because many points are seen with short baselines. However, we can exploit cloth's resistance to strain to improve the reconstruction.

Standard bundle adjustment proceeds by minimizing the reprojection error to polish both 3D locations of reconstructed points and camera parameters. We improve upon this by penalizing large strains, after an idea due to Provot [14]. Small strains in cloth result in relatively small forces, but slightly larger strains can produce very large forces. Because we have recovered a parameterization, we can observe strains in the recovered cloth model. We create a global cost function that combines the reconstruction error in each camera with the strain on the mesh (defined by a Delaunay triangulation of the points in the cloth domain). Using $\|\mathrm{e}\|$ as the edge length, $\|\mathrm{e}_r\|$ as the rest length, $E_r(\mathbf{p})$ as the reconstruction error and k_s as the weight of strain relative to reconstruction error; our cost function is defined as:

$$\begin{aligned} \mathrm{strain}(\mathrm{e}) &= \begin{cases} (\|\mathrm{e}\| - \|\mathrm{e}_r\|)^2 & \text{if } \|\mathrm{e}\| > \|\mathrm{e}_r\| \\ 0 & \text{otherwise} \end{cases} \\ \mathrm{cost} &= k_s \sum_{\mathrm{e} \in \mathrm{edges}} \mathrm{strain}(\mathrm{e}) + \sum_{\mathbf{p} \in \mathrm{points}} E_r(\mathbf{p}) \end{aligned}$$

Because optimizing this objective function involves simultaneously solving for thousands of variables, we adopt a multi-stage approach to reconstructing the 3D points. First, the points are reconstructed without any strain information because each 3D location can be computed independently. Because many observational errors occur at the scale of the large triangles, we minimize a coarse scale version of the global objective function to produce a startpoint for the final optimization problem.

Even with a good starting point, this large optimization problem is intractable without careful attention to detail. First, we reduce computation in numerically computing the gradient by exploiting conditional independence between points on the surface that are significantly far apart. Second, by exploiting the connectivity structure of the surface, we constrain numerical estimates of the Hessian to a sparse matrix form (c.f. [19]).

The combined strain reduction, point reconstruction creates reconstruction results that are significantly better, yet has little effect

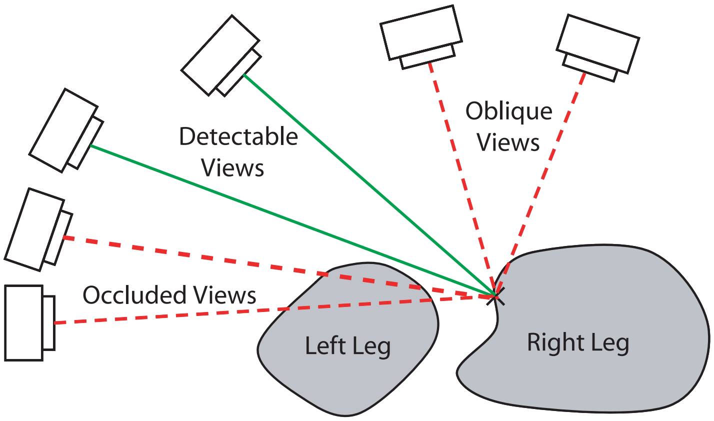

Figure 6: In scenes with multiple cloth surfaces (such as a pair of pants, viewed as a horizontal slice above), occlusion can make multi-view reconstruction difficult. In this case, the surface point can only be seen from a narrow range of views. To use multi-view reconstruction techniques, cameras must be placed very close together — implying large numbers of cameras. Even worse, nearby cameras exhibit the baseline problem: the closer together the cameras, the worse the estimate of the distance from the cameras. To account for this, we use integrate volume cues with single view reconstruction in figure 8.

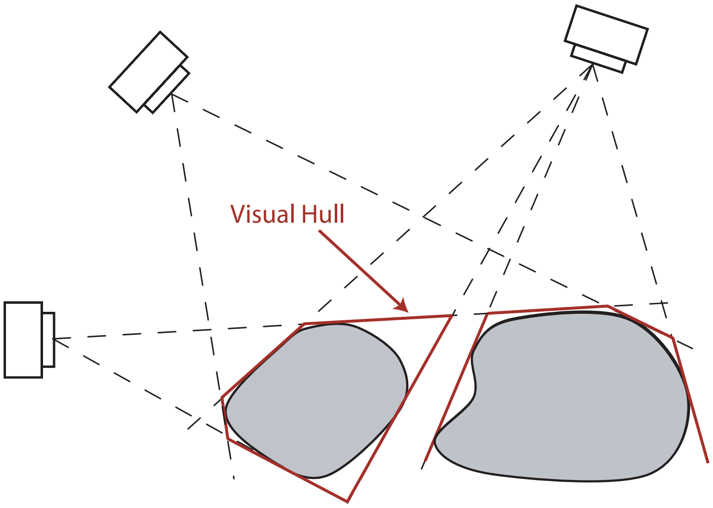

Figure 7: Visual hulls can be computed from silhouette edges in images and provide volume constraints on the resulting cloth reconstruction. The visual hull is an outer bound on the location of the cloth.

on the reprojection errors: typically an increase of less than a fraction of a pixel (in figure 5, this reprojection error increase is 0.03 pixels) Because the most accurate views of a triangle are typically separated by a small baseline, small errors in localization become large errors in depth estimation. The strain reduction only needs to have small effects on the reprojected location of the point to dramatically increase the quality of the reconstructed mesh, as shown in Figure 4.

3 Correspondence

Multi-view reconstruction uses correspondence between views to reconstruct the 3D location of a point on the surface. Instead of computing correspondence between views, we compute correspondence between each image and the parametric domain. This has several advantages: (1) it automatically provides a parametric representation of the resulting surface for use in rendering, (2) it allows us to compute strain on the mesh, and (3) it is a simple way to encode the number of views available for each point in each view. Our parametric domain is constructed as an image (or multiple images) of flattened pieces of cloth.

We print a pattern on the cloth that makes this correspondence easier and use combine cues using an approximation to loopy belief propagation on a graphical model to resolve the identity (parametric location) of each triangle in the image. Our model incorporates two cues: the local neighborhood structure of the cloth, approximated by the neighbors of each triangle; and the limited strain on the cloth. Our efforts are complicated by the unavoidable problems in image processing — some triangles are not detected and neighboring triangles in the image are not necessarily neighboring triangles in the parametric domain. These problems are observable in figure 2.

3.1 Printing a Pattern on the Cloth

We print a color-coded pattern on cloth to simplify correspondence. This general idea is not new [9, 16], but we generalize previous methods by using an un-limited number of colors. Our pattern is composed of triangles with a randomly chosen colors that are dissimilar from their neighbors. (we choose colors in hue and saturation space, disallowing colors with low saturation and forcing the value to be 1) Triangles allow us to easily compute surface normals for automatic calibration [1].

If the color response of the printer and camera had perfect signal to noise ratios, the identity of each triangle could be measured by recording its unique color. Since the printing and recording process are not perfect, the colors are somewhat ambiguous. We use a gaussian emission model to cover the ambiguity inherent in each measurement.

Careful color calibration can significantly improve the accuracy with which colors identify triangles. As in previous approaches [13, 16], we use a frontal photo of the cloth to determine the colors and layout. Inconsistencies between cameras, white balance settings and lights yield colors that are dramatically different in each image (see figure 9). We perform a two stage color calibration. Using some known correspondences (generated by running the code in the next sections with de-tuned parameters), we compute a 3*x*3 color transformation matrix to maximize similarity. Then, we compare the hue of the resulting images to derive a smoothed hue conversion function that maps hue in one view to hue in another view. This color calibration procedure typically reduces the ambiguity between colors by a factor of 2 or 3 (one to two bits).

3.2 Determining Triangle Identity

Triangle identity cannot be determined uniquely from color, however, despite the improved color calibration. Furthermore, the neighbors of a triangle in an image are a helpful, but not reliable guide to its identity, because a fold may mean that the neighbors come from a very different region of the cloth (figure 2). There are two reliable cues: First, any given triangle in the cloth can appear only once in the image (**exclusion**), and second, triangles that are close in the cloth cannot be far apart in the image (**strain**). All this information must be incorporated to identify the triangle.

For each triangle, we make a list of possible identities and the probability that triangle has that identity. Initially, these identities

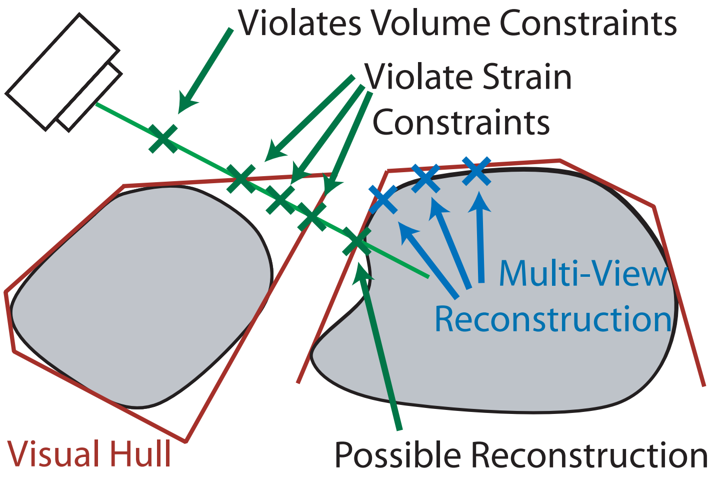
Violates Volume Constraints
Violate Strain Constraints
Multi-View Reconstruction
Visual Hull
Possible Reconstruction

Figure 8: 3D locations can be computed from a single view of the cloth by combining other cues. Using the visual hull and the ray extending from the camera, the range of possible reconstructions is limited to several line segments. Because the amount of strain in the cloth is limited, the number of choices is cut down even more. The final 3D point is chosen to be the closest point to the camera that satisfies all constraints.

are assigned using a Gaussian emission model (the difference between image color and cloth color is a normal random variable). For each triangle, we make a list of possible identities and corresponding probabilities. Initial probabilities are assigned using the gaussian color model mentioned in the previous section. A graphical model is built on this structure, where the nodes are triangles observed in the image. We establish two forms of edge: edges between image neighbors (for the flat regions) and edges between all pairs of triangles that encode strain limits.

Straightforward loopy belief propagation (e.g. [20, 21]) on this structure would not impose the exclusion constraint, and the large number of edges would make it run slowly. As a result, we modify belief propagation. First, after an iteration of belief propagation we check for triangles that are essentially unambiguous and fix their identity (we threshold when the most likely identity has ten times the probability of the second most likely). The probability that any other triangle has that particular identity is now zeroed out for every triangle in the rest of the graph. The strain between a pair of triangles is most likely to constrain their identities when one of the two is known, so we use strain constraints only for those pairs of triangles where one of the two is known. Furthermore, it is our experience that triangles can be identified without looking at all possible such pairs — we pick twenty, at random, per triangle per iteration. Finally, because image neighbors are not necessarily neighbors on the cloth, in late iterations we search over the possibility that some edges are bad.

For edges that encode strain relations, we apply a simple test: if the strain is too large (greater than 40%), then we force the probability for that identity on the unassigned triangle to zero. This technique assumes that we treat the image as a scaled orthographic view, and that we know the scale of the image. As part of the calibration process, scales are estimated for each view [1]. The assumption of scaled orthography is common in many vision settings. We have detuned the strain constraint (typically, strain is limited to 10 or 15%) to account for the possibility of some perspective effects and, in practice, have not observed any failures. This constraint is particularly effective for images that contain a large portion of the domain — because large distances tend to rule out more possibilities.

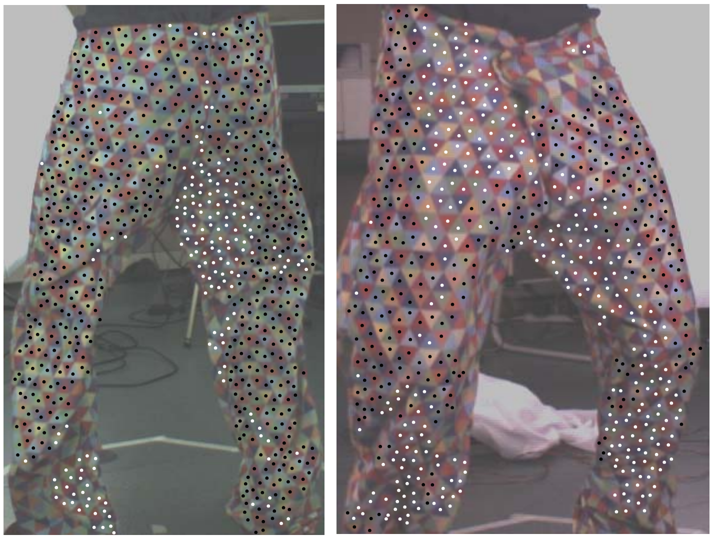

Figure 9: With only six cameras, it is common for points to be seen in only one view. We can still use these points for reconstruction by confining the point to a ray and using volume and strain constraints (section 4). In the images above, **black points** are detected in at least two cameras while **white points** are viewed in only one camera. Notice that the colors in these two images are significantly different: they were taken using the same model cameras with identical settings at the same moment in time, yet the internal processing of the cameras yields very different colors.

3.3 Image Processing

We start by converting each image to HSV, disregarding the value and using polar coordinates to compute distances in hue and saturation. To detect triangles, our code looks for uniformly colored blobs in two stages: first regions are built by growing neighborhoods based on similarity between pixels. This method is sensitive to image noise and can produce oversized regions when the color boundaries are smoothed. The second stage takes the center of mass of each blob from the first stage, computes the mean color and grows a region based on distance to the mean color (it is computationally intractable to use this as the first stage of the blob detection). The process is iterated for increasing thresholds on the affinity value in the first stage, using the portions of the image where detection failed in previous stages. Finally, blobs are thresholded based on size.

Next, we need to determine the neighborhood relationships. For each triangle, we construct a covariance neighborhood and vote for links to the three closest triangles with similar covariance neighborhoods. This measures distances appropriately in parts of the scene where the cloth is receding from view and discourages links between triangles with wildly different tilts. All links that receive two votes (one from either side) are kept while the rest are discarded.

4 Points in One View

We use silhouettes to build volume constraints that confine reconstruction for points that are only viewed in a single camera. When combined with strain, the reconstruction of these points is still not

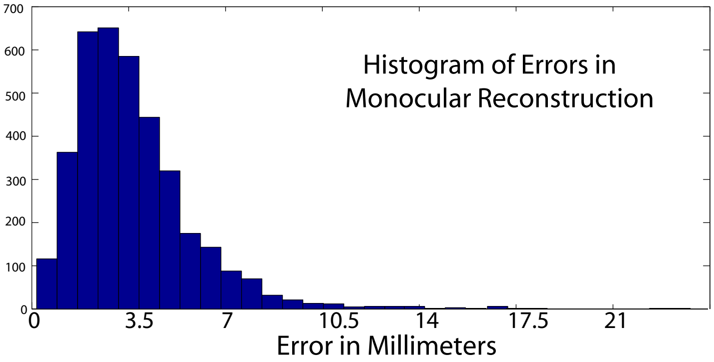

Figure 10: Many points must be reconstructed from only one view. Our method does this using a combination of cues, described in section 4. We evaluate the quality when reconstructing these points by running the same method on points seen in more than one view and measure the distance between the two reconstructions. The histogram of distances above shows that these errors are still small with respect to the 1m+ scale of the overall object. The median error is 3.3 mm or roughly 0.3%. For comparison, the median reprojection error for mutli-view reconstructions is 0.9 mm.

as accurate as multi-view reconstructions, but the errors are typically small with respect to the scale of the cloth (figure 8).

We motivate this section with the observation that real clothing is difficult to view in its entirety — typically an arm or a leg gets in the way. As figure 6 shows, it not uncommon for a portion of the cloth to be visible in a small range of views. As a result, using conventional structure from motion, many of these features will either be discarded (if only seen in one camera) or poorly reconstructed due to a narrow baseline (if seen by two cameras that are very close together).

To capture silhouette constraints, we start with a grid of voxels covering a volume that is slightly larger than the range of points reconstructed using multi-view techniques (this is reasonable, since exterior points tend to be easily recognized). Using a segmentation of each image into foreground and background, we remove all voxels that project onto the background of each image. The resulting object, known as the visual hull, is a very rough approximation of the shape of the object (see figure 7 for a schematic example).

Points viewed in a single image are constrained to a ray in 3D that extends from the focal point of the camera through the observation of the point on the image plane into the rest of the scene. We reconstruct these points by projecting the point onto the closest surface of the volume that obeys strain constraints (figure 8). Because the visual hull is a coarse approximation to the geometry, it isn't uncommon that the closest surface to the camera is actually on a different portion of the object. In most cases, strain constraints remove this possibility. Then, just like points reconstructed using multi-view techniques, points viewed in only one camera are fed into a combined reprojection error strain reduction optimization. In this case, each point has reprojection error in only one image.

Our representation of the visual hull is common to other reconstruction techniques and could be expanded. For example, we carved out voxels of the scene that appeared between the cameras and their multi-view reconstructions, but found that in general this had little effect on the monocular reconstruction. In the future, we plan to run space carving on the remaining voxels to obtain better surfaces for projecting the points. Second, we currently hand mask the silhouette edges. In video, this would not work and an automated method would be required.

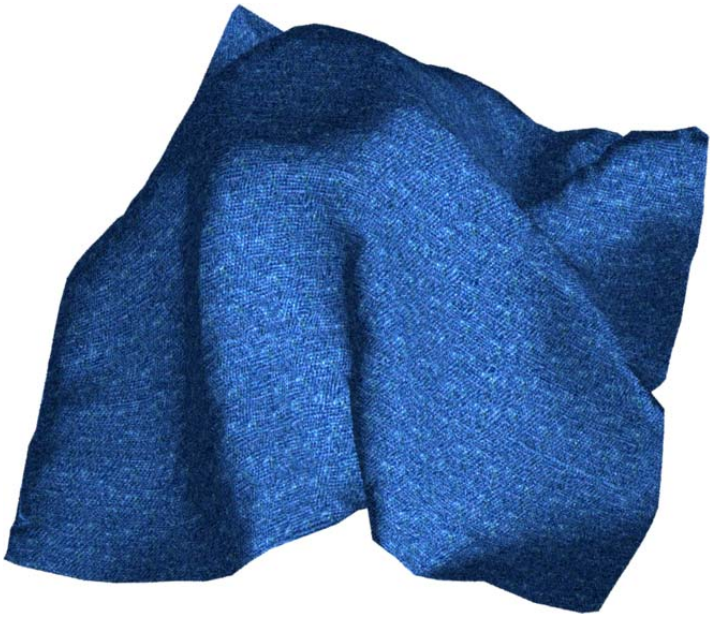

Figure 11: A reconstruction of a piece of cloth draped over a coffee cup reconstructed from 7 views. The folds in this model present challenges for existing methods that require large patches of surface to reconstruct points.

5 Results

Our primary contribution is one of measurement — we capture the state of cloth in real situations and report 3D locations for each of the points. To evaluate the quality of our results, we triangulate the points add a new texture and render from novel viewpoints.

5.1 Mesh Construction and Rendering: Sheets

To render a sheet of cloth, we need to triangulate the points to produce a mesh and establish texture coordinates. Our texture coordinates come directly from the observations: each 3D point corresponds to a 2D point on the image of the frontal cloth (used previously for establishing correspondence). To compute the mesh itself, we take all of the *observed* points and place them in the parametric coordinate system. We use Delaunay triangulation on this flattened piece of cloth and lift the mesh up to 3D.

However, the Delaunay triangulation is not perfect for our use. It produces triangles along the edge of the mesh that are very long and thin in 2D. When lifted to 3D, points that were almost collinear on the plane create very large triangles with significant texture warpage. We remove triangles on the boundary of the mesh that have very small angles.

5.2 Mesh Construction and Rendering: Garments

While the same strategy applies to rendering garments, there are two major problems: (1) there is no single frontal view of the cloth; (2) ideally seams in the cloth will include triangles (alternatively, we could produce a separate mesh for each piece of fabric used to make the garment). To triangulate the surface, we proceed in two steps: First, we triangulate each piece of cloth separately. This produces a mesh that has gaps around all of the seams. Second, we take photos of each of the seams and triangulate the seams in a second pass. To make sure that the triangles in the first pass line

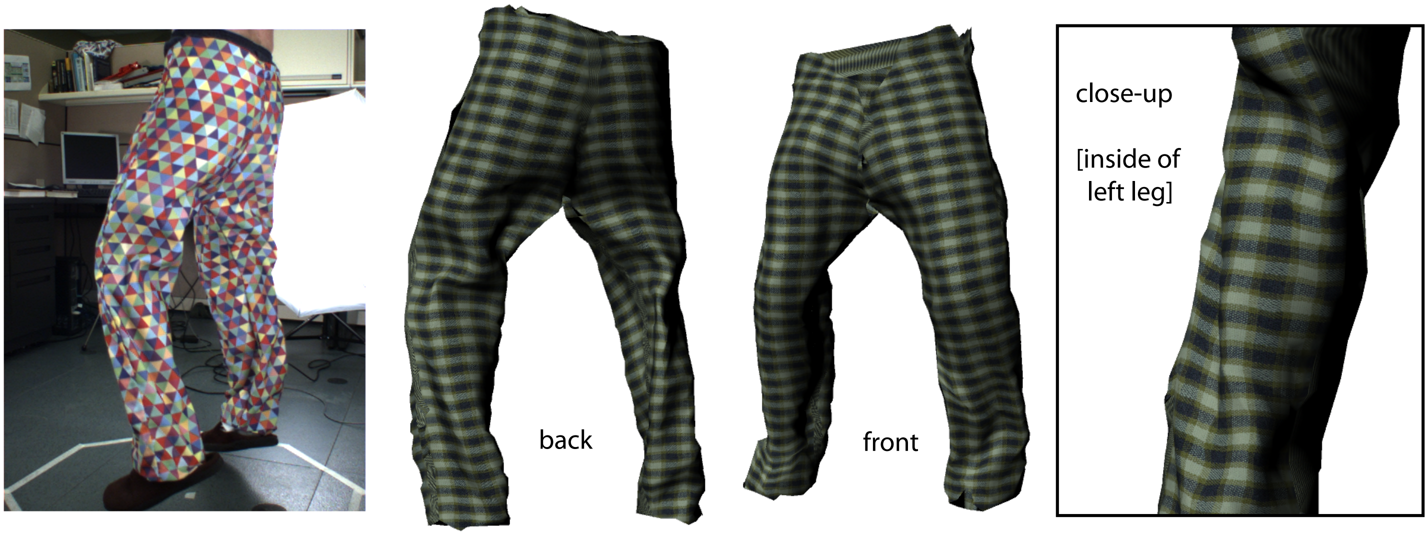

Figure 12: When people jump about, the cloth can move about in dramatic ways. Above, the pants on a human subject in the process of landing after a jump have folds that are unlikely to appear in static scenes (like the fin shaped fold on the back of the right leg). The interaction between applied and inertial forces would be difficult to model without real data. This scene was captured in six cameras and includes significant self-occlusion (one leg blocks views of the other): 1844 triangles are detected in multiple views while another 1079 are detected in only one view. Our cue combination approach to reconstructing points seen in only one view is effective: regions that are difficult to recover (such as the inside of the leg on the **right**) appear realistic when rendered at several times the resolution of the original input data.

up with triangles in the second pass, we project the points and triangles from the two pieces of cloth that have been sewn together into the image of the seam. Then, using the edges from the established triangles, we run constrained Delaunay triangulation [17] to produce additional triangles along the seams. These additional triangles do not contain texture coordinates. It's plausible that there is a better strategy — this method leaves some holes in the mesh and could be improved.

5.3 Examples

We demonstrate the quality of our reconstructions on two static scenes (the sleeve in figure 1 and the cloth over a cup in figure 11) and two dynamic scenes (the coins being tossed on a piece of cloth in figure 3 and the pants in figure 12). Static scenes allow us to use higher resolution cameras and take more images, while dynamic scenes show cloth in positions that are unlikely in a static scene. In both cases, the quality is very high and the results would be very challenging to get from a laser scanner that does not produce a parameterization (meaning no strain constraints) and cannot act quickly enough to capture fast motion.

In figure 1, we used 10 images taken at 1500x1000 pixels to reconstruct 7557 points on the surface of the sleeve. The distance between points on the cloth was 7.5 millimeters in a scene that was approximately 500 millimeters in the longest direction. The average reprojection error was 0.643 pixels — which is approximately 0.23 millimeters (230 micrometers) or 0.05% of the scene size. The resulting mesh had 15,036 triangles.

In figure 3, we show a cloth with high frequency folds due to an impact with a coin. We used six synchronized 640x480 firewire cameras to capture frames of the cloth at 42 millisecond spacing. Roughly 1100 points were reconstructed per frame making a total of roughly 2000 triangles in the resulting mesh. Reprojection error varied per frame, but was typically about 0.5 pixels or 0.35 mm. The cloth is 300 millimeters on a side and the distance between points on the surface is 7.5 mm.

Figure 11 shows a cloth draped over a cup reconstructed using 7 640x480 images to obtain 2066 points. The calibration error is 0.46 pixels or 0.37 mm on a cloth that is 400 mm across.

Finally, figure 12 shows a pair of pants captured in the middle of a jump using the same recording equipment as the coin sequence (6 synchronized 640x480 cameras). There are 2923 points in 5768 triangles spaced 34 mm apart on the surface of the cloth. The average reprojection error is 3.3 mm in a scene that is roughly 1 meter from top to bottom.

References

[1] Anonymous. Deforming objects provide better camera calibration, 2005. Technical Report.

[2] David Baraff and Andrew Witkin. Large steps in cloth simulation. *Computer Graphics*, 32(Annual Conference Series):43–54, 1998.

[3] David Baraff, Andrew Witkin, and Michael Kass. Untangling cloth. *ACM Trans. Graph.*, 22(3):862–870, 2003.

[4] K. Bhat, C. D. Twigg, J. K. Hodgins, P. K. Khosla, Z. Popovic, and S. M. Seitz. Estimating cloth simulation parameters from video. In *Proc. Symposium on Computer Animation*, 2003.

[5] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. *Computer Graphics*, (Annual Conference Series):594–603, 2002.

[6] R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on*

Computer animation, pages 28–36. Eurographics Association, 2003.

[7] Olivier Faugeras and Quang-Tuan Luong. *Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene and Some of the Applications*. MIT press, 2004.

[8] I. Guskov and L. Zhukov. Direct pattern tracking on flexible geometry. In *The 10-th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2002 (WSCG 2002)*, 2002.

[9] Igor Guskov, Sergey Klibanov, and Benjamin Bryant. Trackable surfaces. In *Eurographics/SIGGRAPH Symposium on Computer Animation (2003)*, 2003.

[10] R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2000.

[11] D.H. House, D. Breen, and D. Breen, editors. *Cloth Modelling and Animation*. A.K. Peters, 2000.

[12] D. Pritchard. Cloth parameters and motion capture. Master's thesis, University of British Columbia, 2003.

[13] D. Pritchard and W. Heidrich. Cloth motion capture. *Computer Graphics Forum (Eurographics 2003)*, 2003.

[14] Xavier Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Graphics Interface 95*, 1995.

[15] V. Scholz and Marcus A. Magnor. Cloth motion from optical flow. In *Proceedings of 9th International Fall Workshop on Vision, Modeling and Visualization (VMV 2004)*, 2004.

[16] V. Scholz, T. Stich, M. Keckeisen, M. Wacker, and M. Magnor. Garment motion capture using color-coded patterns. In *Proc. Eurographics*, volume 24, 2005.

[17] Jonathan Richard Shewchuk. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In Ming C. Lin and Dinesh Manocha, editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag, May 1996. From the First ACM Workshop on Applied Computational Geometry.

[18] D. Terzopolous, J. Platt, A. Barr, and K. Fleischer. Elastically deformable models. *Computer Graphics (SIGGRAPH 87 Proceedings)*, pages 205–214, 1987.

[19] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *ICCV '99: Proceedings of the International Workshop on Vision Algorithms*, pages 298–372. Springer-Verlag, 2000.

[20] Y. Weiss. Belief propagation and revision in networks with loops. Technical report, Massachusetts Institute of Technology, 1997.

[21] Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. In *Exploring artificial intelligence in the new millennium*. 2003.