Cloth Capture

Ryan White, Anthony Lobay, D.A. Forsyth

{ryanw,lobay,daf}@cs.berkeley.edu

Report No. UCB/CSD-5-1387

May 2005

Computer Science Division (EECS)
University of California
Berkeley, California 94720

Abstract

We present a method for capturing the geometry and parameterization of fast-moving cloth using multiple video cameras, without requiring camera calibration. Our cloth is printed with a multiscale pattern that allows capture at both high speed and high spatial resolution even though self-occlusion might block any individual camera from seeing the majority of the cloth. We show how to incorporate knowledge of this pattern into conventional structure from motion approaches, and use a novel scheme for camera calibration using the pattern, derived from the shape from texture literature. By combining strain minimization with the point reconstruction we produce visually appealing cloth sequences. We demonstrate our algorithm by capturing, retexturing and displaying several sequences of fast moving cloth.

1 Overview

Cloth modelling is an important technical problem, because people are interesting to look at and most people wear clothing. As a result, 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 [1, 18]; the currently most sophisticated integration strategy is [5]) and that it buckles in fine scale, complex folds (for example, see [4]). 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 [2]). 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.

[3] show that useful settings of simulation parameters can be estimated by observing cloth. A natural extension of this strategy is to attempt to motion capture the cloth itself. This paper reports motion capture of cloth that can capture fast movements in high detail.

2 Previous work

Motion capturing cloth is fairly clearly a structure from motion problem, and we review that area briefly for useful terminology. The area is now very well understood, with comprehensive reviews in two excellent books [6, 10]. Multiple views of a rigid object can be used to obtain a reconstruction of both the geometry of the object, and the

extrinsic (configuration) and **intrinsic** (focal length, camera center, etc.) parameters of all cameras. The standard method involves: identifying **interest points**; using **appearance**, **epipolar** and **three view** constraints to build frame-frame correspondences between these points; obtaining a **projective reconstruction** — which yields geometry and cameras up to a 3D projective transformation — using one of several current factorization methods; and then using either **calibration objects**, **known geometry**, or **auto-calibration** — which applies where each view is from a camera with the same intrinsic parameters — to obtain an **upgrade** to a Euclidean reconstruction. The reconstruction and cameras are then cleaned up with a **bundle adjustment**, which minimizes reprojection error as a function of reconstruction and camera parameters.

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, 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 homographies tend to be noisy because perspective effects are weak or unobservable at the scale of an element, meaning that considerable work must be done to get a set of consistent estimates. The resulting surfaces — for a hand, an elbow, and a T-shirt — are fair but noisy and do not move fast. There is no bundle adjustment. [17] obtain better surfaces by using optical flow predicted from a deformable model, with matches constrained to produce the correct silhouette; again, there is no bundle adjustment. [14, 15] 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. Again, there is no bundle adjustment.

It is not possible to simply drop the successful, well-established recipe for structure from motion onto cloth motion capture. First, cloth is not rigid. This means that each frame will need to be reconstructed separately from multiple camera views, which in turn means that autocalibration from shared camera parameters is not available. Although cloth is not rigid, it is highly resistant to strain. Like rigidity in structure from motion, strain minimization allows more accurate reconstructions with fewer views and more noise. Second, while it is clearly useful to print a special pattern on the cloth, such patterns are too small to serve as useful calibration objects and it is impractical to insist a calibration object be present. These two points have pushed us to use a novel camera calibration method. However, we can adopt components of the recipe; first, it is extremely helpful to concentrate on the frame-frame correspondence problem; second, bundle adjustment makes major contributions to accurate reconstructions.

2.1 Overview

Our system adopts important components of each of the existing approaches, but differs by using a novel variant of conventional structure from motion, and by careful engineering of cloth motion capture as a structure from motion problem. First, we sim-

plify correspondence by printing cloth with a **distinctive pattern**, that allows simple identification of feature points at a relatively fine spatial scale that is robust to motion blur (section 3). Second, our pattern is chosen to reveal a parameterization of the cloth reconstruction immediately, without difficulties caused by motion blur. Third, our structure from motion algorithm involves: adjusting our cameras so that scaled orthography is a reasonable *local* approximation; reconstructing using this approximation — which allows autocalibration using a novel method involving surface normals (section 4); and then bundle adjusting to correct for perspective effects (section 5); there is no projective reconstruction, or upgrade. Finally, by combining reconstruction with strain reduction, we compute a mesh that is both consistent with the original image data and simple cloth properties (section 6).

3 Obtaining Features with a Cloth Pattern

We print a pattern on our cloth which is carefully chosen to allow robust observation. Our pattern is a set of large equilateral color coded triangles where the coloring of each triangle identifies the location and orientation on the cloth. Each large triangle consists of a number of small triangles — where the vertices of the small triangles form a fine grid like structure over the cloth (figure 1). First, elements are highly distinctive and there is little repetition over a large area of cloth. If the cloth is moving quickly, some cameras may see only a small fraction of the entire cloth, so that global correspondence reasoning is impractical. A distinctive element allows reconstruction even in this difficult case. For small sheets, we use entirely unique elements, but on larger sheets we distribute unique triangles over the cloth. Second, our pattern offers a high degree of spatial accuracy, while allowing robust observations during dynamic sequences. These two requirements are in tension because motion blur tends to obscure the high frequency information need for accurate localization. Our large triangles are relatively easy to identify despite motion blur; a deformable template approach then yields the interior structure (15 vertices of the smaller triangles), in a form of coarse-to-fine search.

Coarse step — finding large triangles and their normals: We assume that, over the scale of an individual triangle, perspective effects are negligible; that over the scale of the whole frame, the effects of perspective are small; and, for the purposes of normal estimation, that the surface curvature at the scale of a triangle is small. Our search runs through several steps: threshold the image at different values, perform rough fit of locations and normals, combine triangles at different thresholds, and finally perform a nonlinear optimization over triangle parameters. In each case, the current step either narrows the search space, or provides a better initialization for the next step.

Given variation in lighting and shadows, different regions have greatly varying intensities. A simple threshold for intensity on grayscale images is not enough to find all of the triangles in these different regions, so we threshold at multiple values. For each thresholded image, we use morphological operations to find all blobs of the appropriate size. Because perspective is negligible at the scale of an element, each triangle is imaged through an affine transformation that is a function of camera scale, the slant and tilt of the plane on which the triangle lies, and the in-plane rotation of the triangle. We obtain a rough estimate of camera scale by assuming some triangles in the sequence

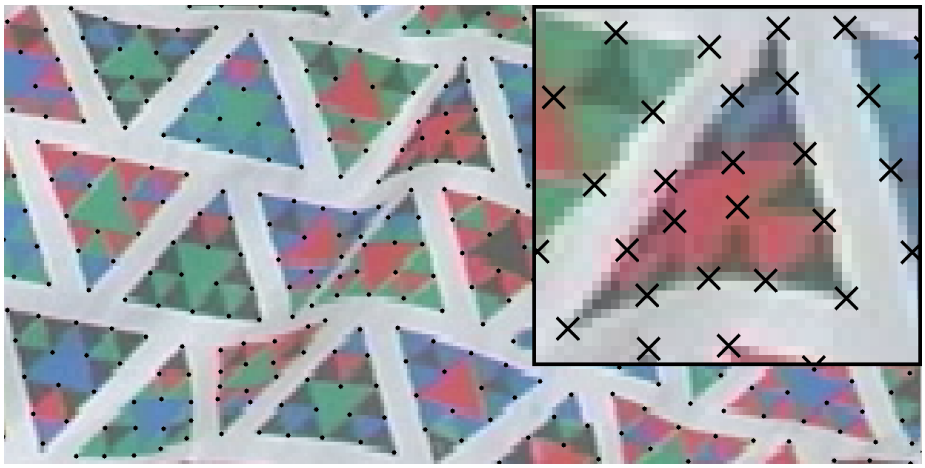

Figure 1: *Our pattern consists of repeating triangles at multiple scales. The large scale triangle has a unique pattern that can easily accommodate 216 unique elements. The vertices of the small scale triangles allow for fine sampling of point locations. The results of our localization code are shown as black dots. Notice how the patterns within the large triangles identify them, and so, through the small triangles, the vertices, leading to a mesh structure — and so a measurement of the cloth parameterization. Accurate localization drives the reconstruction process — important for both bundle adjustment (section 5) and strain reduction (section 6).*

will be viewed frontally by this camera, so that the largest triangles offer an estimate of camera scale. We now use the scale estimate to precompute views of each triangle at a set of different slants, tilts and in-plane rotations. The blobs are compared against this precomputed set — using the number of mismatched pixels as a cost for the quality of the fit.

To combine the triangles at different thresholds, we keep the blob in each area of the image that has the lowest cost. Using the precomputed triangle as a start point, we run a continuous optimization over scale, slant, tilt and in-plane rotation (we attempted to optimize over location as well, but found that typical variation was less than a 1/4 pixel and increased convergence time). At the end of this optimization, we have a model of the triangle location and normal without taking into account local curvature.

Fine step — localizing triangle vertices: For each large triangle, at the fine scale we extract two quantities: sub-triangle colors and sub-triangle vertex locations. Again, we work through several steps of processing, moving from high level information to lower level information, using higher level information to drive the lower level search. Continuing the refinement, we start with a course deforming model for each triangle, then assign colors and finally run a fine deforming mesh. Figure 3 contains an overview of this process.

The course scale deforming model is made up of 4 triangles with 6 unique vertices. Initializing with the planar triangle defined in the previous section, we allow each of the 6 vertices to move freely, penalizing errors with the thresholded image of the triangle

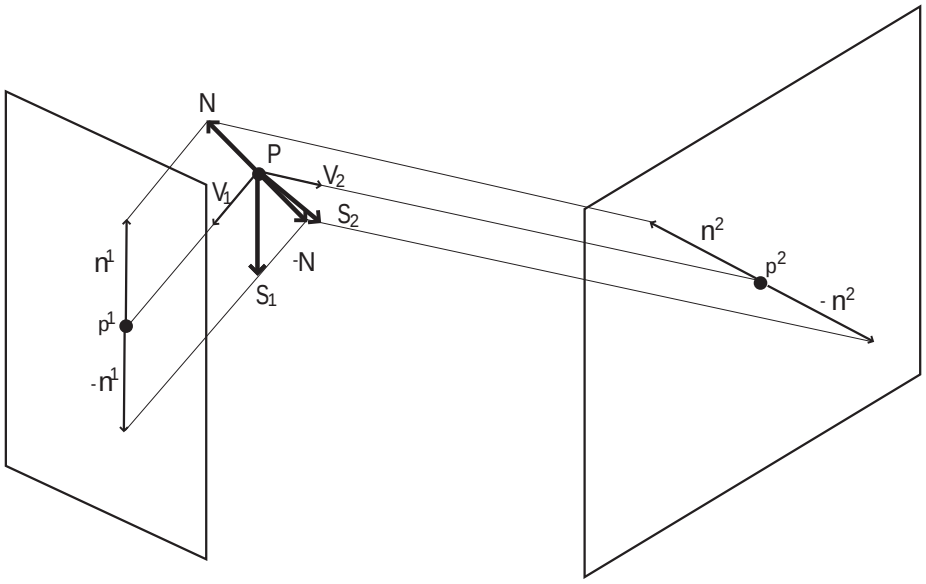

Figure 2: *Notation for normal ambiguity in two views. There are two simple orthographic views of the point* $\boldsymbol{P}$*, with normal* $\boldsymbol{N}$*; view directions are* $\boldsymbol{V}_1$ *and* $\boldsymbol{V}_2$*. The text shows that* $\boldsymbol{S}_1$ *and* $\boldsymbol{S}_2$ *— ambiguous normals in their respective views — have heads lying on the same epipolar plane and that an incorrect match of these leads to a reconstruction of* $-\boldsymbol{N}$*.*

taken from the corresponding threshold choice.

Dividing this course triangle mesh into 16 triangles, we have indices for determining the sub-triangle colors. Image color is a surprisingly poor guide to object color, as it is affected by shadows, variations in printing, lighting and camera sensitivities. Because we know the location of the large triangles, we can rectify the color with a simple strategy. We know that the color pixels are distributed uniformly amongst red, green, blue and black. We then allocate pixels to colors using a greedy strategy, rather like round-robin: assign the red most pixel the label red, the green most green, et cetera, then repeat until no pixels are left. Now, to determine colors for each of the sub-triangles, we group sub-triangles by known relationships (for instance, the four sub-triangles in the middle are always the same color), choose a color and work outward. The sub-triangles that are the farthest from the center of the large triangle are the most problematic. Triangles that deviate from known patterns are thrown out as poor matches.

We now localize each point using a deformable model. Starting with the course scale deforming model, we allow all 15 of the vertices of the smaller triangles to deform. In many cases, the colors assigned to each pixel in the previous stage are erroneous because large variations in lighting across the image create large deviations in color. Instead of making a hard assignment, for the final matching, we warp the color space in a heuristic way to estimate likelihood. On a triangle by triangle basis, we take the raw image values from the original image and warp the color-space to force the black most pixels to be black, red most pixels to be red, et cetera. In the optimization over the deformable model, we charge for deviations from expected colors. We take the final positions of these vertices to be the vertices used for reconstruction.

Implementation note: In several parts of this section, nonlinear optimization is used to match a model of the triangle to an observation of the triangle. To achieve reasonable results, it is important that the objective function be continuous. These objective functions are defined as a sum over pixel differences. To achieve continuity, the pixel values from the model must change continuously with the locations of the vertices. For pixels at the edge of the triangle, we approximate the area of the pixel covered by the triangle as linear in the distance from the side of the triangle.

Results are shown in figure 1.

Manual cleanup: In the interest of time, in some of the shorter sequences, we manually deleted a number of erroneous matches between large triangles (less than thirty per sequence). In the sequence of the skirt, this process was automated.

4 Euclidean from Scaled Orthography

Reconstruction from scaled orthographic views is now a standard algorithm (originating in [19]; many important variants appear in [10]). One builds a **data matrix** containing the coordinates of each view of each point, observes that this matrix is radically rank-deficient, and estimates factors to obtain an affine reconstruction. If there are more than two cameras, a metric reconstruction is available by enforcing scale and angle properties of the camera basis. However, this approach ignores the fact that we know — at least, on the scale of the large triangles — estimates of the geometry (because we know what a large triangle looks like, and so know a surface normal). A

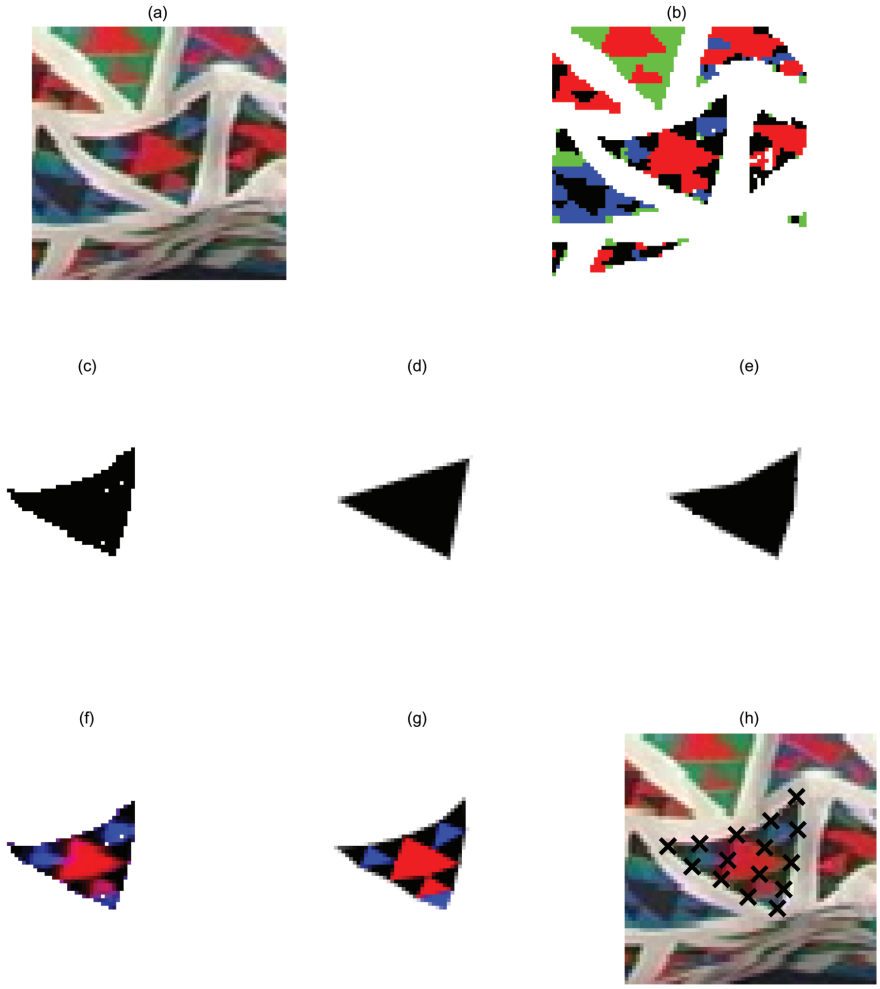

Figure 3: *The triangle matching procedure slowly works from a course model to a fine model, starting with the original image data. Assuming that the appropriate threshold for the image has been picked, we start by cutting out the original image* ***(a)****. Using information from the entire frame, we can assign a color to each pixel using a round robin scheme* ***(b)****. Looking at this figure, one should note that the color assignment problem is the biggest bottleneck — erroneous assignments are common. The process of matching the shape of the triangle to our internal model involves* ***(c)*** *segmenting and thresholding the image,* ***(d)*** *fitting a planar triangle model to the image and* ***(e)*** *using a course deformable model to account for some amount of curvature. Using this deformable model* ***(e)*** *with the color assignment* ***(b)****, we can record the colors of each sub-triangle, and use these colors to warp the original image colorspace* ***(f)****. Finally, we use a fully deformable template to find the vertex positions* ***(g)*** *and mark them on the original image* ***(h)****.*

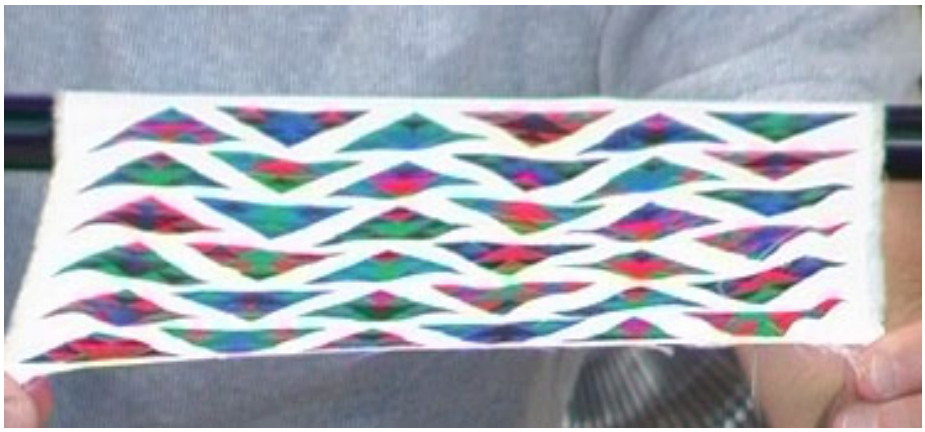

Figure 4: *Cloth can show significant perspective effects in some views, typically when the plane of the cloth is at about* 90° *to the plane of the camera.*

metric reconstruction isn't possible from two views in simple orthographic cameras without calibration of camera extrinsics or some known length or angle [21, 12]. Since the cloth moves fast and we may be stuck with only two views, and to incorporate our normal information, we adopt a method that exploits surface normals to obtain a metric reconstruction.

In a single scaled orthographic view, we know the normal of the plane on which the pattern element lies *up to a two-fold ambiguity* (e.g. [7, 13]). This ambiguity occurs because we can identify the cosine of the slant angle — usually written as $\cos\sigma$ — but not its value from a single view. For example, a scaled-orthographic view of a circle looks like an ellipse; we know the extent of the slant (and so the length of the normal) but the circle could have been slanted either forward or backward to yield this ellipse. As a result, we know the projected normal up to an ambiguity of π radians.

The most natural way to incorporate this information into existing multiple view results is to think of the normal as an arrow of known length protruding from the surface at the point in question. The base of the arrow is the point in question, and projects as usual. The results above mean we know (up to a two-fold ambiguity) to what point in the image the head of the vector projects — in turn, having a normal from texture repetition is equivalent to having a second point *and* having some metric information *because we know the length of the normal vector*. For convenience, in what follows we refer to an isolated point as a **point**, and a point with the normal information described as a **patch**.

4.1 The 3D Ambiguity of Normals

Assume that we are dealing with a pair of simple orthographic cameras. Furthermore, assume that the scale of the cameras is the same (we can obtain the relative scale from the size estimates for triangles, above), and that the extrinsics are calibrated. We know that, in a single view, the projected normal is known up to an ambiguity of π radians. What ambiguity is there in 3D reconstruction of the normal?

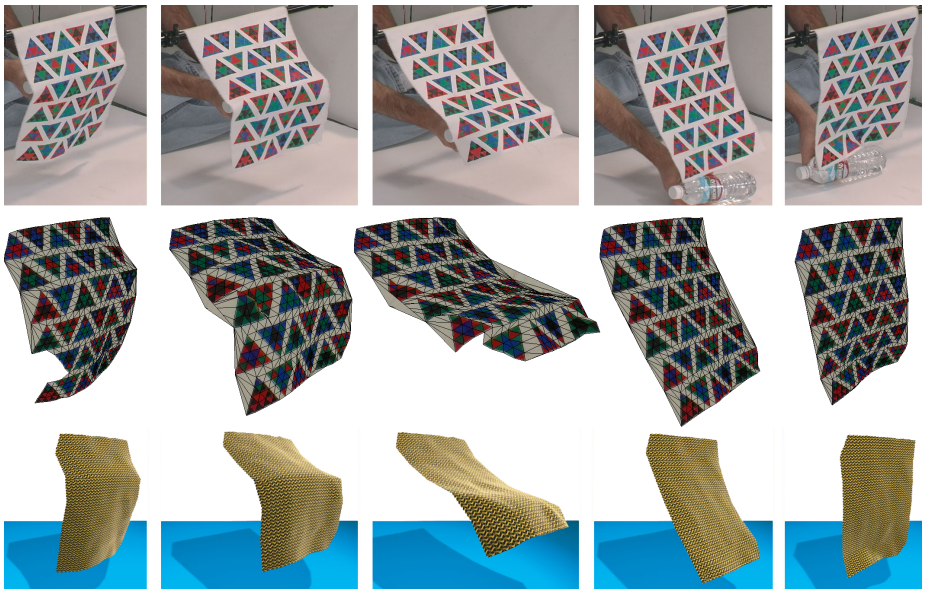

Figure 5: *On the top row, camera frames showing the cloth being pushed by a bottle. The middle row shows reconstruction without interpolation or strain reduction. On the bottom row, the sequence has been strain reduced as a post-processing step and is rendered with a new cloth texture. This suggests that combining the reconstruction and strain reduction will produce a sequence that is both visually convincing and true to the image data.*

Write the normal as $\mathbf{N}$ and the i'th view vector pointing toward the camera (figure 2) as $\mathbf{V}_i$. In the i'th view, there are two possible 3D normals, $\mathbf{N}$ and $\mathbf{S}_i$ (the ambiguous normal in the i'th view). Because the image ambiguity is π radians, $\mathbf{N}$, $\mathbf{V}_i$ and $\mathbf{S}_i$ must be coplanar. Because the projected length of $\mathbf{S}_i$ is the same as the projected length of $\mathbf{N}$, $\mathbf{V}_i \cdot \mathbf{N} = \mathbf{V}_i \cdot \mathbf{S}_i$. This means that we have $\mathbf{S}_i = 2(\mathbf{N} \cdot \mathbf{V}_i)\mathbf{V}_i - \mathbf{N}$ The epipolar planes consist of every plane whose normal is $\mathbf{E} = \mathbf{V}_1 \times \mathbf{V}_2$. The "heads" of $\mathbf{S}_1$ and $\mathbf{S}_2$ lie on the same epipolar plane, because $\mathbf{E} \cdot \mathbf{S}_1 = \mathbf{E} \cdot \mathbf{S}_2 = -\mathbf{E} \cdot \mathbf{N}$. In the circumstances described, there are two possible matches for the "head" of the normal. First, the correct matches are available, resulting in a reconstruction of $\mathbf{N}$; second, one can match the image of the "head" of $\mathbf{S}_1$ with the image of the "head" of $\mathbf{S}_2$. The second case results in a reconstruction of $-\mathbf{N}$ (figure 2); this is easily dealt with, because visibility constraints mean that $-\mathbf{N} \cdot \mathbf{V}_i < 0$ for both i.

All this yields **Lemma:** *A metric reconstruction from two simple orthographic views is available from two patch correspondences. There is a maximum of sixteen ambiguous cases, yielding no more than four camera reconstructions.* **Proof:** (see appendix)

There is an obvious **corollary:** *A fundamental matrix is available from two patch correspondences, up to at worst a four-fold ambiguity*.

4.2 Obtaining a Euclidean Camera Solution

To obtain a camera solution, we pick two patches at random, determine a solution for those points, compute a reconstruction using that solution, then repeat this process until we obtain small reprojection error. To obtain a solution, we move the center of gravity to the origin, then use singular value decomposition to get a factorization of the best rank three approximating matrix. We obtain a factorization in the form $\mathcal{KP}$, where $\mathcal{K}$ has the form $[\mathbf{i}^T, \mathbf{j}^T, \mathbf{r}_1^T, \mathbf{r}_2^T]$, where **i**, **j** are in the coordinate axis directions as usual and $\mathbf{r}_1^T$ and $\mathbf{r}_2^T$ are the first two rows of the camera rotation matrix $\mathcal{R}$. There is a one parameter family of such factorizations (appendix). Write $\mathbf{N}_i^1$ for a reconstruction of normal i in camera one's frame. We use gradient descent to obtain a factorization that minimizes

$$\text{reprojection error} + \sum (1 - \mathbf{N}_i^1 \cdot \mathcal{R}(t) \mathbf{N}_i^2)$$

where t is the parameter. We deal with the four ambiguities described in the appendix by search.

5 Bundle Adjustment

At this point, we have a structure estimate and an estimate of camera extrinsics and scale, both assuming scaled orthography. Our cameras may not, in fact, be scaled orthographic cameras, and some lateral views of cloth display mild perspective effects (figure 4). This results in potentially large reprojection errors and poor reconstructions. We correct for perspective by using our structure and extrinsic parameter estimates to start a bundle adjustment procedure.

Bundle adjustment (see reviews in [10, 20]) involves minimizing reprojection error as a function of camera parameters and point positions. The cost function is the reconstruction error of the points in each image. We have been able to successfully ignore camera centers, to assume that camera axes are at right angles and that pixels are square. Using (x_i^c, y_i^c) as the observed points in camera c, $\mathcal{C}$ as the set of cameras that observe the point, **p** as the point in 3D in affine coordinates, $(\mathbf{r}_{1c}^T, \mathbf{r}_{2c}^T, \mathbf{r}_{3c}^T)$ as the rows of the camera rotation matrix $\mathcal{R}_c$ for the c'th camera, (t_{xc}, t_{yc}, t_{zc}) as the translation vector for the c'th camera, f_c as the focal length of the c'th camera, we compute the reconstruction error as:

$$E_r = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} ((x_i^c - f_c \frac{(\mathbf{r}_{1c}^T \mathbf{p} + t_{xc})}{\mathbf{r}_{3c}^T \mathbf{p} + t_{zc}})^2 + (y_i^c - f_c \frac{(\mathbf{r}_{2c}^T \mathbf{p} + t_{yc})}{\mathbf{r}_{3c}^T \mathbf{p} + t_{zc}})^2)$$

In an inner loop, we optimize over the 3d locations **p**. With the optimal locations of all of the points in the mesh, we can compute an average reconstruction cost associated with a set of parameters – allowing a larger optimization in these parameters. This search is slow and non-linear, but the initialization with an orthographic camera is good enough to yield good results.

It is well-known that this reconstruction error has many local minima; much of the current structure from motion literature treats obtaining estimates of parameters that are good enough to start a successful minimization attempt. We have obvious good start points for most parameters from our orthographic reconstruction (i.e. the rotation parameters, t_x and t_y), but must find start points for t_{zc} and f_c. A natural approach is to

Figure 6: *In a typical cloth simulation environment, combining a fluid simulator with a cloth modeler presents many challenges. However, complicated aerodynamic effects are easy to produce by cloth capture.*

set f_c/t_{zc} to the camera scale, and t_{zc} to a value obtained based on physical estimates of our recording setup. With these initial estimates, we have been able to perform a successful bundle adjustment. Our per feature error is on the order of 1.5 pixels throughout the sequences. We believe most of this error is due to the lack of time sync between cameras – noting that the error is less than 0.5 pixels in slower portions of the sequences.

6 3D Reconstruction

6.1 Matching Triangles

In cloth segments that are large enough for real world applications, our pattern does not include enough distinctive elements to cover the entire cloth. We compensate for this by minimizing repetition and distributing a small number of unique elements over the larger surface. (in the skirt example, there are roughly 35 unique triangles on a cloth with 432 total triangles)

The unique triangles are used for camera calibration, and provide a starting point for matching. Non unique triangles face two problems: correspondence and parameterization. Correspondence can be easily found through epipolar constraints, but parameterization is harder. We phrase the problem as follows: Given a number of triangles in the 2D cloth domain with similar coloring, which triangle in the 3D reconstruction corresponds with which 2D triangle? We solve this with a simple heuristic — local neighborhoods should be similar. While our heuristic is fallible, in practice we have observed no failures.

6.2 Combined Optimization

Our reconstruction method takes on an unusual form of structure from motion. Because cloth changes shape in every frame, the number of views of any one configuration is small. We have only four cameras — significantly fewer than a typical structure from motion setup. As a result, heavily foreshorted triangles are problematic, and can

easily be missed in some views. It is not uncommon for these triangles to be viewed exclusively by cameras with a small baseline, causing minor errors in observation to result in large errors in depth estimation. Figures 8 and 9 emphasize these problems.

We build on the standard approach by using cloth specific knowledge to drive reconstruction. The conventional argument prescribes minimizing the reprojection error to reconstruct the 3D locations of points from image correspondence. We improve upon this by penalizing large strains, after an idea due to Provot [16]. Small strains in cloth result in relatively small forces, but 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 defined in the previous section and k_s as the weight of strain relative to reconstruction error; our cost function is defined as:

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

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 course scale version of the global objective function to produce a start-point for the final optimization problem.

Even with a good starting point, the 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. [20]).

The combined strain reduction, point reconstruction has little effect on the actual reprojection errors: typically an increase of less than 0.2 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 8.

7 Post Processing

In general, we wish to post process reconstructions as little as possible, because we have been careful with our measurements. However, some steps produce a worthwhile improvement. We start with a slightly noisy mesh with holes for each time frame.

Interpolation works by identifying a neighbourhood of the points — ideally, in space and time; in time, if spatial neighbours are missing; in space, if temporal neighbours are missing — and re-estimating the configuration of the missing point using a

multilinear interpolate. This estimation is done *before* the final minimization of strain and reprojection error — so that large strains are removed. Because the interpolated points are not observed, there is no corresponding reprojection cost.

Smoothing: There is still a high-frequency component of temporal noise in the reconstruction; we polish the 3D points with a Gaussian low-pass filter, with σ of one inter-frame interval.

8 Results

We printed our four-color pattern using a screen printing kit onto rayon, using care to deposit a minimum of ink to reduce the effect on the cloth dynamics; we used markers to touch up some areas. To film the dynamics of the cloth, we used 4 digital video cameras, each sampling at 30 frames per second with a shutter speed of 1/250. Faster shutter speeds adversely affected the color quality of the recording while slower shutter speeds had unacceptable motion blur. Scenes are lit with four lights to minimize shadows. Cameras were not genlocked. Results are best assessed by looking at the accompanying video, which shows the extent to which we have been able to capture fast cloth movement.

Appendix: Metric Upgrade Proof

A metric reconstruction isn't possible from two views in simple orthographic cameras without calibration of camera extrinsics or some known length or angle [21, 12]. The reconstruction ambiguity is instructive to study further. Write $\mathcal{D}$ for a view by point data matrix and $\mathcal{P}$ for a 3xpoint geometry matrix; there must be a minimum of four points. Define a **canonical two-camera matrix** to be a matrix of the form

$$\mathcal{C} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \end{bmatrix}$$

and $\mathbf{e}_1$, $\mathbf{e}_2$ are arbitrary orthonormal 3 vectors. We move the origin to the center of gravity, absorb scale into the points, and place the first camera in canonical position to obtain $\mathcal{D} = \mathcal{C}\mathcal{P}$ where $\mathcal{C}$ is a canonical two-camera matrix. Now, if $\mathcal{L}$ is a matrix such that $\mathcal{C}' = \mathcal{C}\mathcal{L}$ is also a canonical two-camera matrix, the reconstructions $\mathcal{P}$ and $\mathcal{L}^{(-1)}\mathcal{P}$ are both available. Note that $\mathcal{L}$ is a matrix of the form $[[1,0,0];[0,1,0];[a,b,c]]$. A one parameter family of such $\mathcal{L}$ exists, and they are not Euclidean transformations. Now assume that we are working with patches.

Lemma: *A metric reconstruction from two simple orthographic views is available from two patch correspondences. There is a maximum of sixteen ambiguous cases, yielding no more than four camera reconstructions.*

Proof: We must first deal with scale, as the two cameras may have pixels of different sizes. Scale commutes with reconstruction, meaning that a camera with small pixels produces a larger frontal view of the texture elements. The ratio of camera scales is then found by scaling a frontal view of an element in the first camera to be the same

Figure 7: *Large scale folds and wrinkles can be captured with high precision and detail. On the mid and lower portions of the skirt, the folds are faithfully recovered. However, fine scale folds can be missed when triangles are heavily fore-shortened, occluded or curved. In this figure, some level of detail is lost in the upper left hand corner. However, without viewing the original image, the resulting mesh is still convincingly cloth-like.*

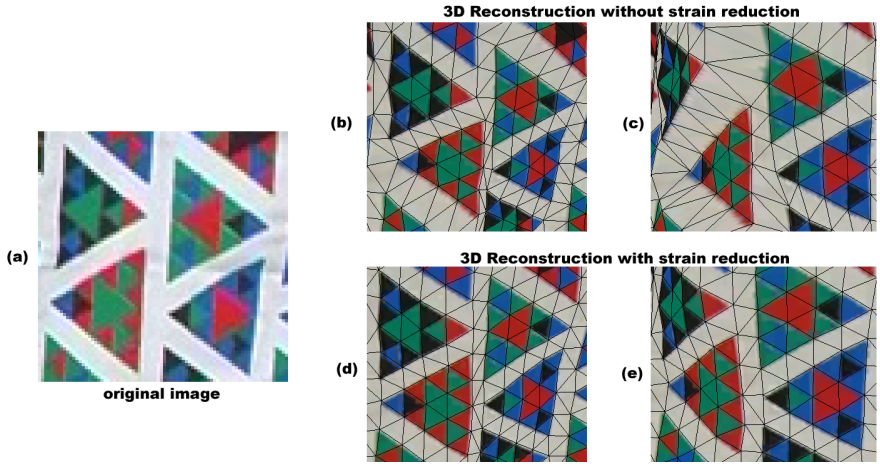

Figure 8: *Because we have a small number of views of each triangle, minimizing the reprojection error alone only produces an accurate mesh when viewed from similar viewpoints. Images **(b)** and **(d)** are **rendered** views of the **reconstructed** mesh (textured with a frontal view of the flattened cloth) taken from viewpoints similar to the original image **(a)**. However, without strain reduction, novel views do not exhibit cloth-like structure. The reconstucted 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 6). Now, the structure of the reconstructed mesh is significantly more realistic — seen as uniform edge lengths — while still true to the original image data.*

size as a frontal view of an element in the second camera. Note that correspondences between element *instances* are not necessary to do this. Each patch consists of a point and a projected normal vector. Write the j'th point as $\mathbf{P}_j$ and the i'th view of the j'th point as $\mathbf{p}_j^i$. Write the j'th normal as $\mathbf{N}_j$ and i'th view of the j'th projected normal vector as $\mathbf{n}_j^i$. What we have referred to as the "head" of the i'th view of the j'th projected normal vector is then $\mathbf{p}_j^i + \mathbf{n}_j^i$; it is easier here to work with the vector directly. First, a metric reconstruction is available because the normal vectors are unit vectors in 3D; we can obtain the metric reconstruction by choosing the element of the one parameter family $\mathcal{L}$ that makes the first normal a unit vector. Ambiguity is more interesting. Our ambiguity in the projected normal vector is a sign ambiguity, yielding a total of sixteen ambiguous cases (two per view per patch). However, these ambiguities have an important internal structure. Write $\mathcal{D}_{(kl)}$ for

$$\begin{bmatrix} \mathbf{p}_1^1 & \mathbf{p}_2^1 & \mathbf{n}_1^1 & \mathbf{n}_2^1 \\ \mathbf{p}_1^2 & \mathbf{p}_2^2 & (-1)^k\mathbf{n}_1^2 & (-1)^l\mathbf{n}_2^2 \end{bmatrix}$$

and $\mathcal{I}^{(ij)}$ for $\mathrm{diag}(1,1,-1^i,-1^j)$. We then have that the ambiguous cases are $\mathcal{D}_{(kl)}\mathcal{I}^{(ij)}$ for $(i,j,k,l) \in [0,1]^4$. Now if $\mathcal{D}_{(kl)} = \mathcal{C}_{(kl)}\mathcal{P}_{(kl)}$, we have that $\mathcal{D}_{(kl)}\mathcal{I}^{(ij)} = \mathcal{C}_{(kl)}\mathcal{P}_{(kl)}\mathcal{I}^{(ij)}$.

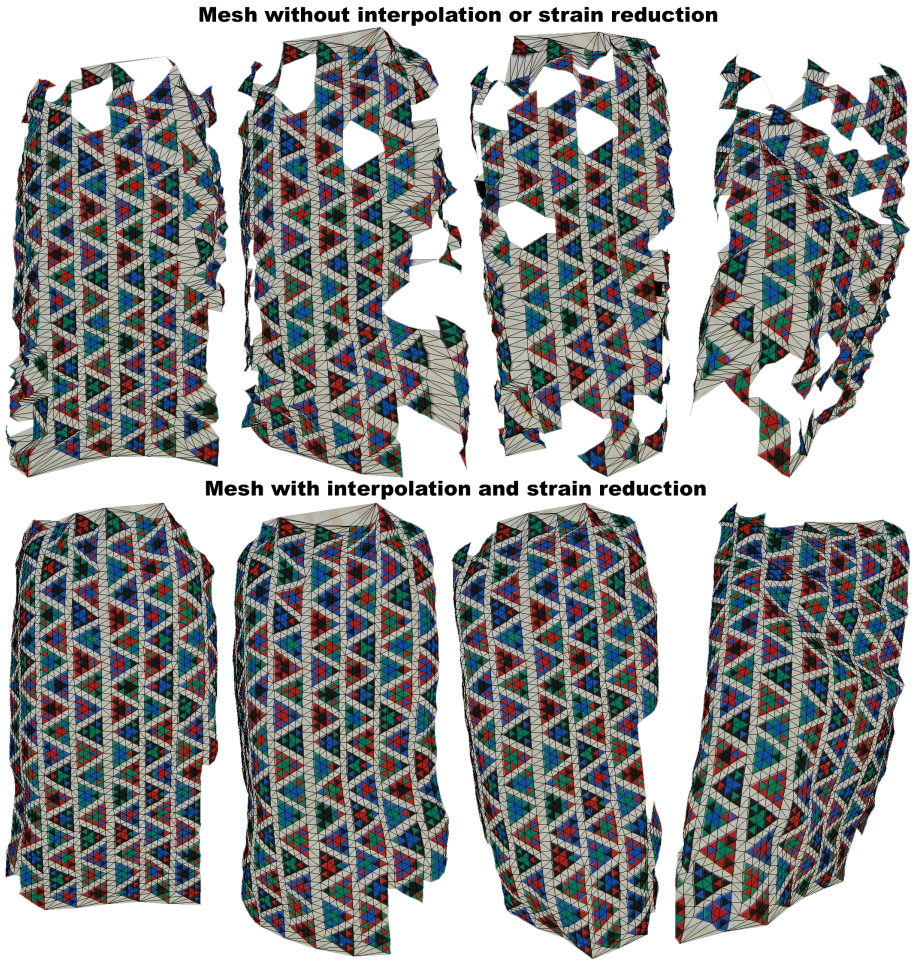

Figure 9: *We demonstrate the strength of our pattern, the accuracy of localization and the importance of combining the minimization of strain and reprojection error. This sequence shows the front of a skirt, viewed from a direction not seen by any camera, retextured with the original pattern for clarity. Each frame contains roughly 2000 vertices comprising almost 4000 triangles. In the* ***top row****, the surface has been reconstructed by minimizing reprojection error alone — gaps appear in the mesh when fewer than two cameras view any large triangle. Furthermore, when relatively few cameras observe a camera, the reconstruction can be inaccurate in some directions — seen as triangles that deviate significantly from the rest of the mesh. This can be corrected by taking strain into account. The* ***bottom row*** *uses interpolation to fill in missing points before running a simultaneous minimization of both reprojection error and strain. An actual image of the skirt taken from one of the cameras can be found in figure 7*

This means that there are only four cases for the camera matrix. Furthermore, our

Figure 10: *Re-texturing a sequence with any pattern or texture is easy because the coordinates in the parameter space are kept at every stage of the sequence. The* ***top row*** *shows a sequence of the front of a skirt rendered with a new texture from a new view. The* ***bottom row*** *shows the same sequence of from one of the camera viewpoints.* ***Important:*** *There is roughly a* 45° *change in angle between the camera viewpoint and the cloth to emphasize the folds in the captured images.*

ambiguities do not interfere with metric reconstruction. Note that

$$\mathcal{P}_{00}\mathcal{I}_{(kl)} = \left[\mathbf{P}_1\mathbf{P}_2(-1)^k\mathbf{N}_1(-1)^l\mathbf{N}_2\right]$$

so that for any of four cases $\mathcal{D}_{00}\mathcal{I}^{(ij)}$ we will obtain the correct camera by insisting that the third column of $\mathcal{P}$ is a unit vector. Furthermore, in any of these four cases the fourth column will be a unit vector, too. We do not expect this to be the case for any of the other twelve cases in general — though specific geometries may make it possible — so that the correct camera is generally easily identified. □

References

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

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] D.A. Forsyth. Shape from texture without boundaries. In *Proc. ECCV*, volume 3, pages 225–239, 2002.

[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] T.S. Huang and C.H. Lee. Motion and structure from orthographic projections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):536 – 540, 1989.

[13] Anthony Lobay and D.A. Forsyth. Recovering shape and irradiance maps from rich dense texton fields. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2004.

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

[15] D. Pritchard and W. Heidrich. Cloth motion capture. *Computer Graphics Forum (Eurographics 2003)*, 22(3):263–271, September 2003.

[16] Xavier Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Graphics Interface 95*, pages 147–154, 1995.

[17] 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.

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

[19] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. Comput. Vision*, 9(2):137–154, 1992.

[20] 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.

[21] S. Ullman. *The Interpretation of Visual Motion*. MIT press, 1979.