Non-Rigid Structure from Locally-Rigid Motion

Jonathan Taylor Allan D. Jepson Kiriakos N. Kutulakos
Department of Computer Science
University of Toronto
{jtaylor,jepson,kyros}@cs.toronto.edu

Abstract

*We introduce locally-rigid motion, a general framework for solving the M-point, N-view structure-from-motion problem for unknown bodies deforming under orthography. The key idea is to first solve many local 3-point, N-view rigid problems independently, providing a "soup" of specific, plausibly rigid, 3D triangles. The main advantage here is that the extraction of 3D triangles requires only very weak assumptions: (1) deformations can be locally approximated by near-rigid motion of three points (*i.e.*, stretching not dominant) and (2) local motions involve some generic rotation in depth. Triangles from this soup are then grouped into bodies, and their depth flips and instantaneous relative depths are determined. Results on several sequences, both our own and from related work, suggest these conditions apply in diverse settings—including very challenging ones (*e.g.*, multiple deforming bodies). Our starting point is a novel linear solution to 3-point structure from motion, a problem for which no general algorithms currently exist.*

1. Introduction

The last 30 years have seen tremendous progress on the structure-from-motion problem. Already, early work on minimal point configurations (*e.g.*, four points in three views [1], eight points in two views [2], *etc.*) has turned into systems for city-scale reconstruction, with millions of points and hundreds of thousands of views [3]. A key ingredient in this success is *global rigidity*, *i.e.*, the assumption that the entire set of points can be thought of as moving rigidly from one view to the next. Global rigidity makes "global" approaches to the structure-from-motion problem highly effective. These approaches, of which factorization [4] is a prime example, take full advantage of this assumption by using all points and all views simultaneously in a single, 3D shape-and-motion estimation step.

Far less is known about how to solve the structure-from-motion (SFM) problem when the scene is not rigid. Non-rigidity is ubiquitous in images and video and covers a broad spectrum—deforming surfaces, articulated structures, groups of rigidly-moving bodies, and any combination thereof, are just a few examples (Figures 1 and 4).

Clearly, if M points move independently across N views, the SFM problem is under-constrained. This has generated a lot of interest in reducing the problem's dimensionality so that global SFM algorithms can be extended to the non-rigid case. Although a wide spectrum of algorithms now follow this approach, they all rest on assumptions about the scene's *global* spatio-temporal behavior. These include (1) deformations that span a low-dimensional shape space [5–10]; (2) trajectories that span a low-dimensional motion space [11]; (3) textured meshes with a regularized shape and low-order deformation [12–16], or a known template shape [17]; and (4) scenes composed of rigid bodies moving independently [18–20] or in articulated configurations [21,22]. Unfortunately, global behavior is hard to predict in all but highly-constrained settings (*e.g.*, faces, articulated bodies, *etc.*). Even then, pre-segmentation, training data, or information about the solution manifold may be necessary.

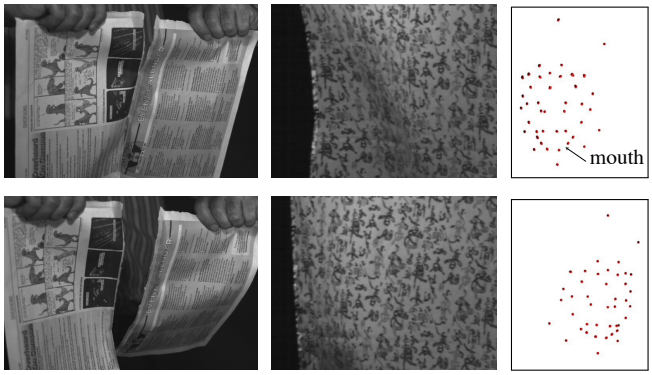

Figure 1: Example sequences. *Left to right:* newspaper being torn apart; silk scarf deforming freely; points on a face (from [5]).

As a first step in overcoming these limitations, we consider an alternative paradigm where the global non-rigid SFM problem is decomposed into many local rigid ones. Our approach is based on a simple intuition: many non-rigid motions, even very complex ones, can be approximated locally by a rigid transformation involving three points.

We make this intuition concrete by solving non-rigid SFM bottom-up, in four steps: (1) given M points in N views, generate a large collection of triplets of nearby points; (2) solve an orthographic, 3-point-N-view rigid-SFM problem independently for each triplet and identify 3D triangles consistent with near-rigid motion; (3) assign the 3D triangles to one or more deforming bodies; and (4) reconstruct the instantaneous shape of each body by recovering the flips and the relative depths of the triangles in each view. The key advantage of this formulation is that the estimation of rigid 3D triangles in the first two steps requires only weak assumptions on the nature of the local deformations and the view changes. As such, they can be solved by optimizing image re-projection error, without shape or motion priors.

Although not used for SFM in recent decades, the general

paradigm we follow is very old. Indeed, in his original work on the problem, Ullman [1,23] suggests grouping points in quadruplets, testing for rigidity, solving 4-point SFM, and combining the results. Our work can therefore be thought of as a modern re-interpretation of Ullman's original scheme, applied to general non-rigid SFM and made even more local—with three points instead of four. Our generate-and-test procedure also suggests a similarity to robust parametric estimation methods like RANSAC [24]. This similarity, however, is superficial: our generation procedure is not random, we do not seek consensus among 3D triangles, and never combine them into larger, parametric structures. In effect, each 3D triangle approximates a distinct piece of the scene and exists separately from all the rest.

At the heart of our approach lies a solution to the problem of computing structure from $N \geq 4$ views of just three rigidly-moving points. This is a hard problem that briefly attracted SFM theorists for the case of $N = 4$ [25–27] but was abandoned without algorithmic solutions for large N. In this respect, our solution represents a new result in rigid SFM and is one of the key theoretical contributions of this paper. This solution is particularly important in the context of non-rigid SFM for four reasons. First, the small number of points makes it much more likely that local rigidity and orthography hold through a sequence. Second, since all geometric computations involve three points and thus are very local, handling many views offers substantial protection against noise. Third, it naturally handles sequences with densely-tracked points, since 3-point configurations are not degenerate even in small surface neighborhoods (unlike the case of four non-coplanar points). Fourth, it makes it very efficient to identify locally-rigid motions among large point sets.

The output of our non-rigid SFM algorithm is neither a deforming surface nor a moving 3D point set; it is a "soup" of independently-moving, rigid triangles whose apparent coherence arises exclusively from satisfying point-wise re-projection error constraints. As such, it is related to non-parametric, sample-based representations of geometry [28, 29] and shares many of their features: without built-in smoothness or connectivity constraints, the representation is very flexible; it can fit diverse global shapes and motions; and enables automatic segmentation and reconstruction of independently-deforming bodies.

2. Three-Point Structure from Motion

What can we infer from an orthographic image sequence of just three moving points? With such a limited point set, affine structure [30] is under-constrained and factorization-based methods [4] do not apply.

The definitive answer was given by Bennett and Hoffman [27]. They proved that four views of three points are necessary and sufficient to decide whether the points' motion is rigid. Their work included a non-constructive proof showing that shape and motion are highly ambiguous in this case: up to 32 interpretations exist when points do move rigidly in four views. This work, along with earlier studies of the 3-point, 3-view problem [25,26], analyzes the algebraic structure of over-constrained systems of polynomial equations and is mainly of theoretical interest. We are not aware of algorithmic implementations of these ideas and it is unclear how to incorporate noise, approximate rigidity [23], and more than four views. Here we develop a formulation that applies to any number of views of a moving 3D triangle; can measure the triangle's degree of non-rigidity; can recover rigid shape approximations for motions that are only approximately rigid; and can deal with image noise within a fairly standard least-squares setting.

Our main observation is that computing the length of edges on a 3D triangle is much easier than computing the triangle's pose or 3D coordinates. We exploit this observation by deriving a novel, coordinate-free relation between lengths on a 3D triangle and lengths in its projection. This relation, which we call the Projected-Length Equation, leads to a linear method for recovering 3D lengths that uses all views simultaneously and enforces all available metric constraints.

Using this as a starting point, we solve 3-point SFM in three steps: (1) estimate 3D lengths by applying the linear method to N images of a 3D triangle, (2) use these lengths to estimate the triangle's pose independently for each image, and (3) jointly refine lengths and poses with a non-linear algorithm that minimizes re-projection error over all images.

2.1. The Projected-Length Equation

The foreshortening of a triangle viewed under orthography depends on the relative depth of its vertices (Figure 2) :

$$\|\mathbf{p}_i - \mathbf{p}_j\|^2 \; - \; \|\mathbf{q}_i - \mathbf{q}_j\|^2 \; = \; (z_i - z_j)^2 \;\; , \quad (1)$$

where $\mathbf{q}_i$ is the projection of vertex $\mathbf{p}_i$ and z_i is its depth (*i.e.*, distance from the image plane).

The sum of pairwise relative depths is always equal to zero:

$$(z_2 - z_1) + (z_3 - z_2) + (z_1 - z_3) \; = \; 0 \;\; . \quad (2)$$

Combining Eqs. (1) and (2) and using the notation

$$L_{ij} = \|\mathbf{p}_i - \mathbf{p}_j\|^2 \quad (3)$$

$$l_{ij} = \|\mathbf{q}_i - \mathbf{q}_j\|^2 \;\; , \quad (4)$$

we obtain an expression that eliminates dependence on depth, is entirely coordinate free, and relates actual and projected (squared) lengths:

$$\sqrt{L_{21} - l_{21}} \; \pm \; \sqrt{L_{32} - l_{32}} \; = \; \mp \sqrt{L_{13} - l_{13}} \;\; . \quad (5)$$

Although seemingly quite complex, Eq. (5) can be simplified substantially. Observe that we can eliminate the square root and sign ambiguity on the equation's right-hand side by squaring both sides of the equation. Applying this observation twice, along with some algebraic manipulation, we arrive at a very simple quadratic constraint linking lengths in 2D and 3D (see supplementary material for a derivation):

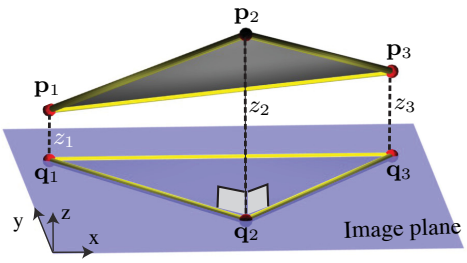

Figure 2: Viewing geometry. We assume that the viewing direction is along the z-axis of the camera-centered coordinate system.

Projected-Length Equation

$$\boxed{\mathbf{L}^T \mathbb{A}\, \mathbf{L} \;-\; 2\mathbf{L}^T \mathbb{A}\, \mathbf{l} \;+\; \mathbf{l}^T \mathbb{A}\, \mathbf{l} \;=\; 0}$$

where

$$\mathbf{L} \;=\; \begin{bmatrix} L_{21} \\ L_{32} \\ L_{13} \end{bmatrix}, \;\; \mathbf{l} \;=\; \begin{bmatrix} l_{21} \\ l_{32} \\ l_{13} \end{bmatrix}, \;\; \mathbb{A} \;=\; \begin{bmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix} \quad .$$

2.2. Linear Intrinsic Shape from N Views

Now suppose that the triangle moves rigidly across N images. Since its motion is rigid, the length of each edge, and hence the vector $\mathbf{L}$, is constant. Applying the Projected-Length Equation for each image, we obtain a system of N quadratic equations:

$$\mathbf{L}^T \mathbb{A}\, \mathbf{L} \;-\; 2\mathbf{L}^T \mathbb{A}\, \mathbf{l}_1 \;+\; \mathbf{l}_1{}^T \mathbb{A}\, \mathbf{l}_1 \;=\; 0 \qquad (6)$$

$$\cdots$$

$$\mathbf{L}^T \mathbb{A}\, \mathbf{L} \;-\; 2\mathbf{L}^T \mathbb{A}\, \mathbf{l}_N \;+\; \mathbf{l}_N{}^T \mathbb{A}\, \mathbf{l}_N \;=\; 0 \qquad (7)$$

where the vectors $\mathbf{l}_1, \ldots, \mathbf{l}_N$ collect projected lengths for images 1 through N, respectively.

Since the only quadratic term in the system is constant for all equations, we eliminate it by subtracting Eq. (6) from the rest. This yields a linear system of $N-1$ equations and three unknowns, *i.e.*, the squared 3D lengths in vector $\mathbf{L}$:

Linear Length Recovery Equation

$$2 \begin{bmatrix} \mathbf{l}_1{}^T - \mathbf{l}_2{}^T \\ \cdots \\ \mathbf{l}_1{}^T - \mathbf{l}_N{}^T \end{bmatrix} \mathbb{A}\, \mathbf{L} \;=\; \begin{bmatrix} \mathbf{l}_1{}^T \mathbb{A} \mathbf{l}_1 \;-\; \mathbf{l}_2{}^T \mathbb{A} \mathbf{l}_2 \\ \cdots \\ \mathbf{l}_1{}^T \mathbb{A} \mathbf{l}_1 \;-\; \mathbf{l}_N{}^T \mathbb{A} \mathbf{l}_N \end{bmatrix} \quad . \qquad (8)$$

To compute these lengths, we simply solve Eq. (8) for $\mathbf{L}$.

Equation (8) implies that although the structure-*and*-motion problem has many discrete ambiguities and is hard to analyze even for four views, the *structure* problem generically has a unique, easily-computable solution for any $N \geq 4$. Moreover, the equation decouples the problem of estimating intrinsic structure from the problem of estimating extrinsic properties such as pose or 3D coordinates. On the practical side, we can incorporate all measurements into a single estimation step, for improved accuracy over long sequences.

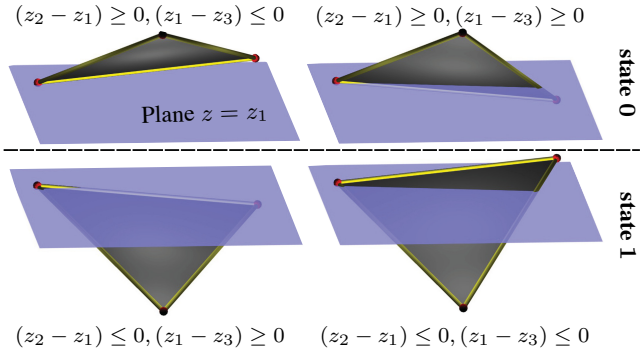

Figure 3: Reflection ambiguity. The top left triangle is as in Figure 2. Reflecting vertices $\mathbf{p}_2$ and/or $\mathbf{p}_3$ about plane $z = z_1$ produces three more triangles that differ in the sign of relative vertex depths. Of these, only triangles related by a mirror reflection (*i.e.*, same column) have the same edge lengths, and thus are ambiguous. We use 0/1 to denote these "reflection states," as indicated.

2.3. Pose Ambiguities

A unique solution to the structure problem implies that we can use Eq. (1) to determine the relative depth of any pair of vertices up to a sign flip [31]. Geometrically, this makes each image consistent with two triangles, related by a reflection about the image plane (Figure 3).

Neither this reflection ambiguity nor the triangle's absolute depth, always lost under orthography, can be resolved when the triangle is viewed in isolation. Rather than make an arbitrary and potentially-erroneous choice, we do not resolve them during local, three-point SFM computations. Below we focus on pose estimation modulo these ambiguities.

2.4. Non-Linear Pose Estimation from One View

In theory, once we know a triangle's edge lengths, we can recover its pose in a given image by solving Eq. (1) in terms of (unsigned) relative depths. In practice, however, solving this equation directly can lead to inaccurate results because the edge lengths, both in 3D and in 2D, may not be known with high accuracy.

To get more reliable pose estimates, we solve the well-known problem of three-point exterior orientation [32] under orthography: given the length vector $\mathbf{L}$ from Eq. (8) and an image n, we estimate 2D position and 3D orientation by minimizing average squared re-projection error:

$$\mathcal{E}_n(\boldsymbol{\theta}, \mathbf{t}, \mathbf{L}) \;=\; \frac{1}{3} \sum_{i=1}^{3} \left\| \left[\begin{smallmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{smallmatrix} \right] \mathbb{R}(\boldsymbol{\theta}) \hat{\mathbf{p}}_i(\mathbf{L}) + \mathbf{t} \;-\; \mathbf{q}_{in} \right\|^2$$

$$\boldsymbol{\theta}_n, \mathbf{t}_n \;=\; \arg\min_{\boldsymbol{\theta}, \mathbf{t}} \; \mathcal{E}_n(\boldsymbol{\theta}, \mathbf{t}, \mathbf{L}) \;, \qquad (9)$$

where $\mathbf{q}_{in}$ is the projection of vertex i in image n; $\boldsymbol{\theta}$ is a vector that represents three rotational degrees of freedom which define the 3×3 rotation matrix, $\mathbb{R}(\boldsymbol{\theta})$; $\mathbf{t}$ is a 2D translation vector; and $\hat{\mathbf{p}}_i(\mathbf{L})$ are 3D vertex coordinates when the

triangle is in an *a priori*-specified "reference" pose.[1]

We rely on Levenberg-Marquardt minimization in Eq. (9), using the exponential map to represent orientations and $\boldsymbol{\theta} = [0.1\ 0.1\ 0.1]$ for the initial rotation estimate.

2.5. Multi-View Shape & Pose Refinement

Although the linear system in Eq. (8) gives a very simple and efficient way to estimate 3D lengths from a potentially large number of images, it relies on minimizing an algebraic error functional. This functional measures consistency with a system of Projected-Length Equations and produces sub-optimal length estimates in the presence of localization noise (*e.g.*, from feature tracking) or modeling errors (*e.g.*, from slightly non-rigid motions). To ensure the best possible estimates of both pose and length, we use a final refinement stage that optimizes them jointly by minimizing average squared re-projection error across all images:

$$\mathcal{E}(\mathbf{L}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N, \mathbf{t}_1, \ldots, \mathbf{t}_N) = \sum_{n=1}^{N} \frac{\mathcal{E}_n(\boldsymbol{\theta}_n, \mathbf{t}_n, \mathbf{L})}{N} \ . \quad (10)$$

We optimize this error iteratively with conjugate gradients,[2] from initial lengths and per-image poses (Algorithm 1).

3. 3-SFM for Non-Rigid Scenes

Suppose we are given a sequence of M features tracked over N images. We assume that these features may be distributed over several independently-moving bodies, each of which undergoes unknown and possibly non-rigid motion. To handle sequences of this generality, we treat each body as a collection of "loosely-coupled" rigid triangles and use 3-SFM as our basic computational step.

[1] This pose can be arbitrary; here we place the triangle on the xy-plane, with a vertex at the origin and one edge along the x-axis. In particular, we define $\hat{\mathbf{p}}_1(\mathbf{L}) = [0\ 0\ 0]$, $\hat{\mathbf{p}}_2(\mathbf{L}) = [0\ \sqrt{L_{21}}\ 0]$ and set $\hat{\mathbf{p}}_3(\mathbf{L})$ to be the positive-y intersection of two circles on the xy-plane—one with radius $\sqrt{L_{13}}$ centered at $\hat{\mathbf{p}}_1(\mathbf{L})$ and one with radius $\sqrt{L_{32}}$ centered at $\hat{\mathbf{p}}_2(\mathbf{L})$.

[2] Since the optimization occurs in a space of $6N + 3$ dimensions, with $N > 1000$ for long sequences, methods that do not maintain an explicit representation of the Hessian, such as conjugate gradients, are preferable.

Algorithm 1: 3-Point Structure from Motion (3-SFM)

Input: feature positions $\mathbf{q}_{1n}, \mathbf{q}_{2n}, \mathbf{q}_{3n}, n = 1, \ldots, N$
Output: squared pairwise distance vector $\mathbf{L}$; per-frame poses $\boldsymbol{\theta}_n, \mathbf{t}_n$ & per-frame unsigned relative depths, $|z_{2n} - z_{1n}|, |z_{1n} - z_{3n}|$; root-mean-squared re-projection error ϵ

1. for each frame n, compute the squared pairwise distances of $\mathbf{q}_{1n}, \mathbf{q}_{2n}, \mathbf{q}_{3n}$ & place them in vector $\mathbf{l}_n$;
2. solve Eq. (8) in terms of $\mathbf{L}$ to get initial estimate of pairwise squared distances in 3D;
3. for each frame n, conduct the minimization in Eq. (9) to obtain initial estimates of $\boldsymbol{\theta}_n$ and $\mathbf{t}_n$;
4. minimize the functional in Eq. (10) to obtain final estimates of $\boldsymbol{\theta}_n, \mathbf{t}_n$ for all n, and of $\mathbf{L}$;
5. set $\epsilon = \sqrt{\mathcal{E}(\mathbf{L}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N, \mathbf{t}_1, \ldots, \mathbf{t}_N)}$;
6. for each frame n, compute unsigned relative depths from Eq. (1), with $\mathbf{q}_i = \mathbf{q}_{in}$ and $\mathbf{p}_i = \mathbb{R}(\boldsymbol{\theta}_n)\hat{\mathbf{p}}_i(\mathbf{L})$.

3.1. Discovering Near-Rigid Triangles

An exhaustive search for near-rigid triangles would require running 3-SFM on all possible three-feature combinations among M features. Not all combinations, however, are equally likely to yield near-rigid motion. For instance, three points that lie far apart on a deforming body (or on separate bodies), are unlikely to move rigidly. By ignoring feature combinations that involve such points, it is possible reduce the size of the search space from $O(M^3)$ to roughly $O(M)$.

To take this heuristic into account, we apply 3-SFM only to triplets of nearby features [1]. We use a topological criterion to choose them in a scene- and image-independent way (Figure 4, left): the features must belong to a triangle in the 2D Delaunay triangulation of one of the input images. Although this criterion can select $O(NM)$ triangles in the worst case, triangulations of nearby frames are usually similar and do not contribute many new ones.[3]

Testing for near-rigidity If a 3D triangle undergoes generic non-rigid motion across four images, its projection will have no rigid interpretation for all but a measure-zero set of motions [27]. Therefore 3-SFM can be expected to return a non-negligible re-projection error. We consider errors above a fixed tolerance ϵ^* to be from non-rigid triangles.[4]

Re-projection errors below ϵ^* are either due to near-rigid triangles, or triangles that *appear* to be near-rigid but are not (*i.e.*, non-generic deformation). Specifically, if (1) two features in a triplet have constant distance in all images and (2) the third feature always falls on an ellipse that is axis-aligned with the other two, the triplet always has a rigid interpretation. For example, a 2D feature point translating perpendicular to the line between two other points can be interpreted as the foreshortening of a rigid triangle. Although this is unlikely to happen in long sequences of real-world surface deformation, in practice it does occur for triplets with an "outlier" feature, *i.e.*, matched erroneously across images or lying on a different object (Figure 4(left)). Nevertheless, triangles reconstructed from such triplets have a characteristic geometry: they are typically very oblique in 3D and have large 3D lengths in order to account for the relative motion of the outlier feature. Here, we discard as potentially non-rigid any 3D triangle that passes one of two tests: (1) the angle between two 3D edges is less than 10°, and (2) the 3D length of an edge is at least $2.5\times$ the median length across all reconstructed triangles.

[3] It is possible to expand the search space by selecting features that lie δ edges apart on the Delaunay triangulation graph. We found, however, that such an expansion (*e.g.*, with $\delta = 1$) did not improve results and added a significant computational overhead.

[4] In practice, localization noise also contributes to re-projection error and ϵ^* needs to allow for this as well.

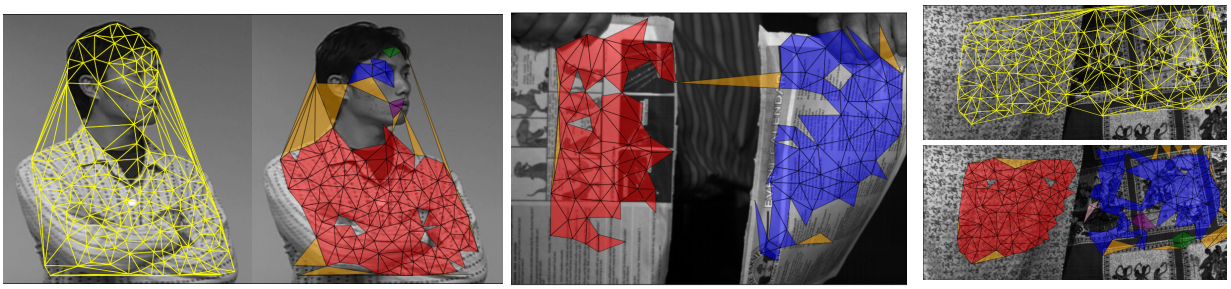

Figure 4: Near-rigid triangle discovery and segmentation results for three sequences—the person sequence from [22], the *tear* sequence from Figure 1, and a *two-cloth* sequence, with two independently-deforming bodies (a thick tablecloth and a thin scarf). In all cases, unfilled triangles denote the Delaunay triangulation of tracked feature points. Filled triangles are the subset identified as near-rigid; orange triangles denote triangles identified as non-generic and discarded. Other colors indicate object membership. In the leftmost sequence, most features tracked on the head were lost or occluded due to head rotation.

3.2. Flexible Triangle Pairs

When two feature triplets have two features in common, the rigid triangles produced by 3-SFM are highly constrained (Figure 5a). We call these triangles a *flexible triangle pair*. Intuitively, triangles in a flexible pair behave like a "loose hinge:" even though each is rigid, their relative pose can change freely from image to image along an implicit hinge axis, to account for deformations (*e.g.*, bending). Additional degrees of freedom come from the tolerance ϵ^* on re-projection error: by allowing independent minor adjustments to the triangles' other pose parameters, this tolerance makes them fit small shape distortions that do not have a simple parametric form. In this respect, a flexible pair can be thought of as an implicit, non-parametric model of local deformation. We make extensive use of this model below, to analyze non-rigid deformation at a global level.

3.3. Grouping Triangles into Non-Rigid Bodies

Flexible triangle pairs occur only when four scene points preserve at least five of their pairwise distances across the entire sequence. As such, the existence of a flexible pair is a strong cue for local connectivity, *i.e.*, that all four scene points belong to the same rigid or non-rigid body. We use this cue in a simple three-step algorithm that groups reconstructed triangles into objects: (1) define a graph that has a node for every reconstructed triangle and an edge for every flexible pair; (2) find the graph's connected components; and (3) treat each component as a separate, independently-moving body. Figure 4 shows some examples.

4. Full 3D Reconstruction

Sections 2 and 3 suggest that one can go a long way with strictly local geometric processing: measuring distances between nearby scene points, estimating instantaneous local surface orientation (up to reflection), and motion-based grouping are all possible without reasoning about global 3D geometry. Local processing, however, hits its limit for tasks where the outstanding triangle ambiguities—reflection state and depth translation—must be resolved. To do this, we consider all near-rigid triangles on a body simultaneously.

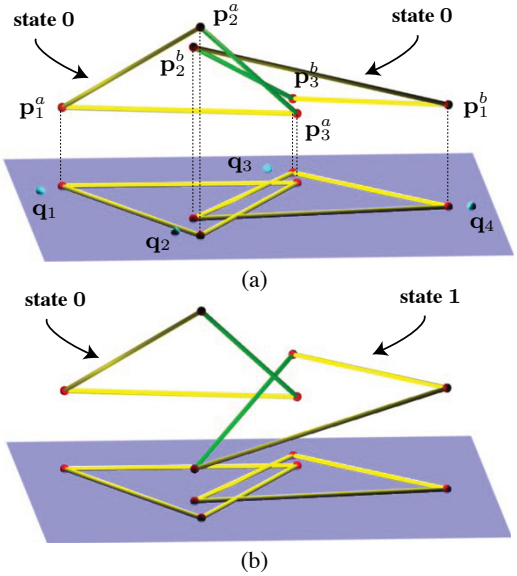

Figure 5: (a) A flexible triangle pair computed by running 3-SFM independently on features $\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3$ and $\mathbf{q}_2, \mathbf{q}_3, \mathbf{q}_4$. Since $\mathbf{q}_2$ and $\mathbf{q}_3$ participate in both computations, they act as a "hinge" constraint on the flexible pair's triangles, $\mathbf{p}_1^a, \mathbf{p}_2^a, \mathbf{p}_3^a$ and $\mathbf{p}_1^b, \mathbf{p}_2^b, \mathbf{p}_3^b$: the triangles' pose in any given image must keep two vertices and the edge between them (shown in green) aligned. This alignment will be approximate for re-projection tolerances $\epsilon^* > 0$. (b) Flipping the reflection state of the rightmost 3D triangle in (a) causes a misalignment of the flexible pair's hinge edges (green).

4.1. Recovering Instantaneous Depths

Recovering the depth of a connected component of triangles is easy once we know their reflection states. We assume below that all these states are known and revisit their computation in Section 4.2.

If two feature triplets on the same body have a feature in common, the triangles produced by 3-SFM must agree on the depth of the common feature in every image (Figure 5a). For a specific image n, this gives an equality constraint that links the absolute depth of a vertex i on one triangle and a

vertex j on the other:

$$z^a_{1n} + (z^a_{in} - z^a_{1n}) = z^b_{1n} + (z^b_{jn} - z^b_{1n}) \Leftrightarrow \quad (11)$$
$$z^a_{1n} - z^b_{1n} = s^b_{nj} |z^b_{jn} - z^b_{1n}| - s^a_{ni} |z^a_{in} - z^a_{1n}| ,$$

where a, b denote the two triangles; $|z^a_{in} - z^a_{1n}|, |z^b_{jn} - z^b_{1n}|$ are the unsigned relative depths returned by 3-SFM; and s^a_{ni}, s^b_{nj} are their signs. Since reflection states are known, these signs are known as well. Equation (11) is therefore linear in the unknown absolute depths, z^a_{1n} and z^b_{1n}.

By applying Eq. (11) to all triplets with common features on the same body, we get a linear system of equations that typically contains many more equations than unknowns. We solve the system independently for each image and each body to get the instantaneous depth of every triangle up to a global depth translation.[5]

4.2. Determining Reflection States

A set of T triangles defines $T \times N$ binary variables—one reflection state per triangle per image. We seek an assignment to these variables that conforms to two basic geometric constraints: (1) the angle between hinge edges in a flexible pair should be the smallest possible in every image (Figure 5b) and (2) the pose of each triangle should change as little as possible from one video frame to the next. These constraints are sufficient to constrain reflection state assignments in all but two special cases involving fronto-parallelism.[6]

Both constraints can be encoded in a constraint graph whose nodes are the $T \times N$ reflection state variables and whose edges represent the geometric constraints between them. Since triangle poses are noisy and constraints vary with orientation, this assignment is most appropriately expressed as an energy minimization problem over a binary Markov random field. Unfortunately, the energies involved are not sub-modular [33], making optimization difficult. Moreover, current methods [34] either provide just a partial solution (QPBO) or improve an existing one (QPBOI). As optimization is not our focus, we use a simpler (but inferior) approach: we greedily assign values to all reflection state variables in the graph using a constraint propagation scheme.

More specifically, each edge in the graph couples the reflection states of two triangles. This coupling is "strong" when a flip in one triangle's reflection state causes a significant misalignment (e.g., Figure 5) and is "weak" when all combinations of reflection states for the two triangles yield nearly-identical alignment (e.g., due to fronto-parallelism). We model this by assigning to each edge a weight that describes how strongly two triangles mutually constrain their reflection state. We then propagate reflection states in four steps: (1) compute a minimum spanning forest; (2) choose a node, make it the root and assign it an arbitrary reflection state; (3) traverse the spanning tree starting from the root, assigning to each child the reflection state that maximizes geometric alignment with its parent; and (4) repeat these steps until all nodes are visited. Our optimization is implemented exactly as stated (*i.e.*, no missing steps) and determined by two functionals—edge weights and pairwise alignment costs. See the supplementary materials and [35] for details.

The result of this procedure is a complete assignment of reflection states to all triangles in all images. It is important to note, however, that this assignment will be ambiguous if the graph is not connected. The most common example involves different bodies, as computed in Section 3.3. A more subtle case involves connectivity breaks *within* a body. For example, if a surface deforms into a fronto-parallel plane somewhere in the sequence, it is impossible to tell whether the surface crosses the image plane after that event or stays on one side of it. Here we follow a principle of *least commitment*: we recover triangle depths independently for each component (Section 4.1) and never attempt to combine the resulting triangle "soups"—even if they belong to the same body. We believe that this is most appropriately handled at a higher-level of processing, not within the realm of purely geometric, prior-free SFM computations.

5. Experimental Results

We applied locally-rigid SFM to a variety of challenging video and motion-capture (mocap) sequences, ranging from 37 to 1000 images. We discuss some of them below; see supplementary materials and [35] for videos, code and more results. To get feature trajectories from video, we initialized a standard tracker [36] at 150 randomly-chosen corners in one frame of each sequence. For mocap, we simply projected 3D feature trajectories along the z-axis to obtain an orthographic sequence. We applied our algorithm to raw trajectory data in all cases, completely automatically, and with *identical* parameters except for the tolerance ϵ^* (it took one of three values as noted below). The result is a collection of triangle "soups" (Section 4.2), with body labels attached (Section 3.3).

Baseline accuracy of 3-SFM We synthetically generated N random orthographic views of an equilateral triangle with 3D edge length L and added Gaussian noise of standard deviation σ. The top table in Figure 6 shows that 3D error varies linearly with σ: with 20% noise added to each point in each frame, the RMS error is just 19% of L.

Ground-truth evaluation Sequences corresponding to random 3D orientations represent the best-case scenario for 3D reconstruction; to assess the accuracy of reconstructed triangle soups under more realistic conditions, we used two mocap-derived sequences: a sheet of paper held in front of a blowing fan, captured with a Vicon system (*Wind*); and a face sequence from [5] (*Jacky*). Following the approach in [8], we added Gaussian noise of standard deviation $\sigma = \frac{\rho}{100}\mu$ to every point in every frame, where μ is the maximum distance of image points from their centroid across all frames (*i.e.*, ρ represents % image error). We measure accuracy by computing the absolute RMS 3D error

[5] Since this final depth ambiguity cannot be resolved under orthography, we arbitrarily set to zero the average depth of each object.

[6] Specifically, if a hinge edge in a flexible pair becomes fronto-parallel in some image, all combinations of reflection states produce identical angles for that flexible pair in that image. Similarly, if a moving triangle becomes fronto-parallel, its pose relative to the next frame will be identical for both reflection states.

between the reconstruction and the mocap "ground truth" over all points and all frames. Since 2D points corresponding to more than one triangle in the reconstructed soup are assigned one 3D coordinate per triangle, we average these coordinates before computing RMS error.

The second table in Figure 6 shows accuracy comparisons for four methods (ours, ours+QPBOI, and [5, 8]). Our approach has comparable performance on the *Jacky* sequence and significantly outperforms the state-of-the-art [5, 8] on the *Wind* sequence. *Jacky*, one of the few ground-truth evaluation datasets for non-rigid SFM, is in fact nearly rigid: singular value analysis on the centered measurement matrix shows that 96% of the variance is explained by *rigid* motion. In contrast, only 84% of the variance is explained this way on *Wind*—*i.e.*, locally-rigid SFM can handle more significant non-rigidity. Taken together, these results suggest that locally-rigid SFM is effective in reconstructing accurate 3D shape even under noisy conditions and strong deformation; that local minima are generally not a problem for 3-SFM; and that QPBOI, although arguably more principled, offers minor improvement.

Video sequences We applied our algorithm to a variety of 30Hz video sequences from the literature [20, 22, 37]; Figure 6 shows results from one of them, *Paper*. Unlike previous attempts to reconstruct this scene, ours assumes orthography and does not need a user-defined mesh, surface texture, manual alignment [16], or a reference pose [37]. We also used our algorithm to reconstruct a set of very challenging deformations (tearing, flapping), never before reconstructed monocularly. The motions themselves very brief—lasting about $0.2s$ each—and could only be captured with a high-speed video camera. We used a MEMRECAM to do this at 500Hz. The captured sequences were characterized by rapid deformation, changes in surface topology, lost or mismatched features, and occlusion.

Discussion Four observations can be made about our results. First, we are aware of no methods for non-rigid SFM that have demonstrated 3D reconstructions on a similar range of sequences (*e.g.*, raw tracks, significant deformations, more than one body, *etc*.). Second, existing methods assume that features belong to one deforming object and thus implicitly assume pre-segmentation. In contrast, since locally-rigid motion computations are local, they have no trouble handling multiple bodies and topological changes. Third, while many methods can handle missing features, they implicitly assume that there no outliers or bad tracks (*e.g.*, [5,8]). Here we identify and remove them, via 3-SFM. The price we pay for this flexibility is reliance on tracking individual points: if a track gets lost, so will some triangles. Fourth, despite the difficulty of the sequences we tested, the recovered soups are plausible and the deformations are highly detailed (see videos). This suggests that local rigidity is applicable to a broad spectrum of deformations.

6. Concluding Remarks

We consider locally-rigid SFM to be a first—but not last—word on bottom-up structure from motion for general non-rigid settings. Although we relied on fairly unsophisticated methods for processing triangle soups, more advanced techniques are surely possible. We are currently investigating several such directions, including (1) combining 3-SFM with RANSAC for increased robustness, (2) using spectral methods for triangle grouping, (3) building spatiotemporally-coherent surface models from triangle soups, and (4) taking surface texture into account.

Acknowledgements The authors gratefully acknowledge the support of NSERC Canada under the Discovery, RGPIN and PGS-D programs.

References

[1] S. Ullman, "The interpretation of structure from motion," *Proc. R. Soc. Lon. Ser. B*, vol. 203, no. 1153, pp. 405—426, 1979.
[2] H. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, pp. 133—135, 1981.
[3] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, "Buliding rome in a day," *Proc. ICCV*, 2009.
[4] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *IJCV*, vol. 9, no. 2, pp. 137—154, 1992.
[5] L. Torresani, A. Hertzmann, and C. Bregler, "Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors," *IEEE T-PAMI*, vol. 30, no. 5, pp. 878–892, 2008.
[6] L. Torresani, D. Yang, E. Alexander, and C. Bregler, "Tracking and modeling non-rigid objects with rank constraints," *Proc. CVPR*, vol. 1, pp. 493—499, 2001.
[7] V. Rabaud and S. Belongie, "Re-thinking non-rigid structure from motion," *Proc. CVPR*, 2008.
[8] M. Paladini, A. D. Bue, M. Stosic, M. Dodig, J. Xavier, and L. Agapito, "Factorization for non-rigid and articulated structure using metric projections," *Proc. CVPR*, pp. 2898–2905, 2009.
[9] S. I. Olsen and A. Bartoli, "Implicit non-rigid structure-from-motion with priors," *JMIV*, vol. 31, no. 2-3, pp. 233–244, 2008.
[10] A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S. Olsen, and P. Sayd, "Coarse-to-fine low-rank structure-from-motion," *Proc. CVPR*, 2008.
[11] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, "Nonrigid structure from motion in trajectory space," *Proc. NIPS*, 2008.
[12] M. Salzmann, J. Pilet, S. Ilic, and P. Fua, "Surface deformation models for nonrigid 3d shape recovery," *IEEE T-PAMI*, vol. 29, no. 8, pp. 1481–1487, 2007.
[13] M. Salzmann, R. Urtasun, and P. Fua, "Local deformation models for monocular 3d shape recovery," *Proc. CVPR*, 2008.
[14] F. Moreno-Noguer, M. Salzmann, V. Lepetit, and P. Fua, "Capturing 3d stretchable surfaces from single images in closed form," *Proc. CVPR*, pp. 1842–1849, 2009.
[15] M. Salzmann and P. Fua, "Reconstructing sharply folding surfaces: A convex formulation," *Proc. CVPR*, pp. 1054–1061, 2009.
[16] A. Varol, M. Salzmann, E. Tola, and P. Fua, "Template-free monocular reconstruction of deformable surfaces," *Proc. ICCV*, pp. 1811–1818, 2009.
[17] M. Perriollat, R. Hartley, and A. Bartoli, "Monocular template-based reconstruction of inextensible surfaces," *Proc. BMVC*, 2008.
[18] J. Costeira and T. Kanade, "A multi-body factorization method for motion analysis," *Proc. ICCV*, pp. 1071—1076, 1995.
[19] R. Tron and R. Vidal, "A benchmark for the comparison of 3-d motion segmentation algorithms," *Proc. CVPR*, 2007.
[20] R. Vidal and R. Hartley, "Three-view multibody structure from motion," *IEEE T-PAMI*, vol. 30, no. 2, pp. 214–227, 2008.
[21] J. Yan and M. Pollefeys, "A factorization-based approach to articulated motion recovery," *Proc. CVPR*, vol. 2, p. 1203, 2005.
[22] J. Yan and M. Pollefeys, "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate," *Proc. ECCV*, pp. 94–106, 2006.
[23] S. Ullman, "Maximizing rigidity: The incremental recovery of 3-d structure from rigid and nonrigid motion," *Perception*, vol. 13, no. 3, pp. 255—274, 1984.
[24] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *CACM*, vol. 24, no. 6, pp. 381—395, 1981.
[25] T. Huang and C. Lee, "Motion and structure from orthographic projections," *IEEE T-PAMI*, vol. 11, no. 5, pp. 536—540, 1989.
[26] D. Hoffman and B. Bennett, "The computation of structure from fixed-axis motion: Rigid structures," *Biol. Cybern.*, vol. 54, no. 2, pp. 71—83, 1986.
[27] B. Bennett and D. Hoffman, "Inferring 3d structure from three points in rigid motion," *JMIV*, vol. 4, no. 4, pp. 401—406, 1994.

3-SFM Accuracy vs. Noise					
σ/L	0	.06	.11	.15	.2
RMSE/L	0	.07	.11	.15	.19

RMSE for Ground-Truth Datasets						
	Wind			*Jacky*		
ρ	0	0.5	1	0	0.5	1
greedy	1.8	2.6	3.4	3.8	5.4	4.8
QPBOI	1.7	2.5	3.4	3.8	4.4	4.7
[5]	8.3	8.2	8	3.2	3.3	3.5
[8]	38.9	38.9	39.8	3.1	3.1	3.4

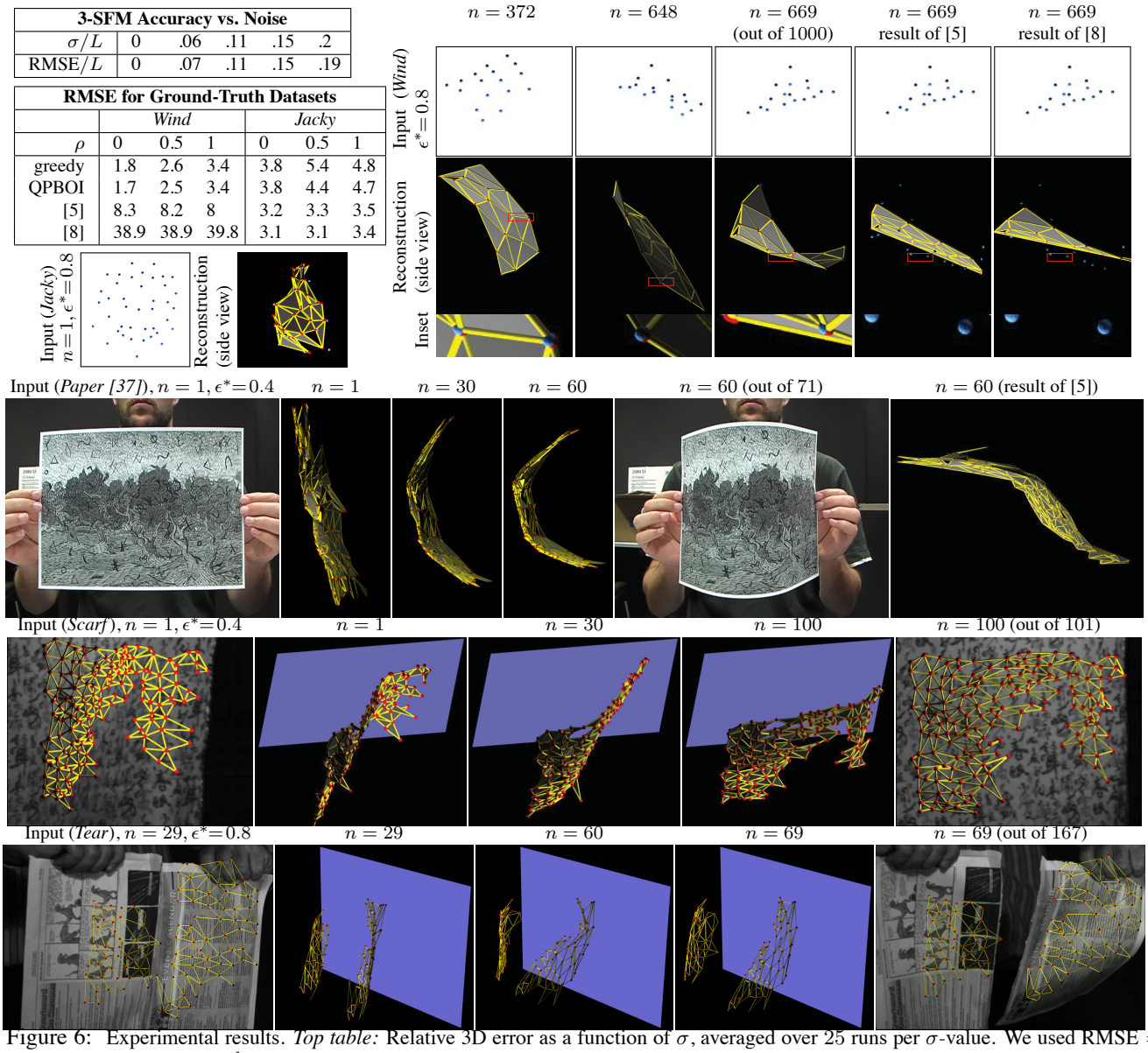

Figure 6: Experimental results. *Top table:* Relative 3D error as a function of σ, averaged over 25 runs per σ-value. We used RMSE $= (\frac{1}{3N}\sum_{n,i}\|\mathbf{p}_{in} - \mathbf{p}^*_{in}\|^2)^{\frac{1}{2}}$ where $\mathbf{p}_{in}$ and $\mathbf{p}^*_{in}$ are reconstructed and ground-truth vertices, respectively, in frame n. *Bottom table:* Ground-truth accuracy results for (1) locally-rigid SFM using the greedy approach of Section 4.2; (2) after refining reflection states with QPBOI; and (3) using the methods of [5] and [8]. For these, we used author-supplied code and report RMSE for the number of basis shapes minimizing it. *Reconstruction results:* Please zoom in to the electronic images for a detailed view of each reconstruction. All renderings in each sequence are from the same viewpoint. For *Wind* and *Jacky*, blue dots represent ground-truth 3D points while red dots are reconstructed triangle vertices. These vertices align very well with the ground-truth—in contrast, [5] and [8] yield shapes that are clearly incorrect for *Wind*. This also occurs in the *Paper* sequence, where the method of [5] fails to recover the paper's bent shape.

[28] H. Pfister, M. Zwicker, J. V. Baar, and M. Gross, "Surfels: Surface elements as rendering primitives," *Proc. ACM SIGGRAPH*, pp. 335—342, 2000.

[29] R. Carceroni and K. Kutulakos, "Multi-view scene capture by surfel sampling: From video streams to non-rigid 3d motion, shape and reflectance," *IJCV*, vol. 49, no. 2, pp. 175—214, 2002.

[30] J. Koenderink and A. V. Doorn, "Affine structure from motion," *JOSA-A*, vol. 8, no. 2, pp. 377—385, 1991.

[31] C. Taylor, "Reconstruction of articulated objects from point correspondences in a single uncalibrated image," *CVIU*, vol. 80, no. 3, pp. 349—363, 2000.

[32] B. K. P. Horn, "Projective geometry considered harmful," *Unpublished Memo*, 1999.

[33] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?," *IEEE T-PAMI*, vol. 26, no. 2, pp. 147—159, 2004.

[34] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer, "Optimizing binary mrfs via extended roof duality," *Proc. CVPR*, 2007.

[35] `www.cs.toronto.edu/~kyros/research/LRSFM`.

[36] A. Jepson, D. Fleet, and T. El-Maraghi, "Robust online appearance models for visual tracking," *IEEE T-PAMI*, vol. 25, no. 10, pp. 1296—1311, 2003.

[37] M. Salzmann, R. Hartley, and P. Fua, "Convex optimization for deformable surface 3-d tracking," *Proc. ICCV*, 2007.