Simultaneous Pose and Non-Rigid Shape with Particle Dynamics

Antonio Agudo[1] Francesc Moreno-Noguer[2]
[1]Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Spain
[2]Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Barcelona, Spain

Abstract

In this paper, we propose a sequential solution to simultaneously estimate camera pose and non-rigid 3D shape from a monocular video. In contrast to most existing approaches that rely on global representations of the shape, we model the object at a local level, as an ensemble of particles, each ruled by the linear equation of the Newton's second law of motion. This dynamic model is incorporated into a bundle adjustment framework, in combination with simple regularization components that ensure temporal and spatial consistency of the estimated shape and camera poses. The resulting approach is both efficient and robust to several artifacts such as noisy and missing data or sudden camera motions, while it does not require any training data at all. Validation is done in a variety of real video sequences, including articulated and non-rigid motion, both for continuous and discontinuous shapes. Our system is shown to perform comparable to competing batch, computationally expensive, methods and shows remarkable improvement with respect to the sequential ones.

1. Introduction

The problem of simultaneously recovering rigid shape and camera pose from a monocular sequence, known as Structure-*f*rom-Motion (S*f*M), has recently seen great progress [1, 22], even when dense reconstructions are required [23]. Yet, these methods cannot be applied to scenes undergoing non-rigid deformations. In these situations, the fact that many different 3D shapes can have very similar image projections produces severe ambiguities that can only be resolved by introducing prior knowledge about the camera trajectory and scene deformation.

Most Non-Rigid Structure *f*rom Motion (NRS*f*M) approaches solve this problem using statistical priors to model the global deformable structure as a linear combination of low-rank bases of shapes [9, 12, 21, 33] or 3D point trajectories [6, 16, 26]. This is typically used with additional smoothness constraints that further disambiguate the problem [8, 14, 25]. Yet, while low-rank methods can effectively encode global deformations, they cannot, in general, handle non-linear motion patterns and strong local deformations. Piecewise strategies [11, 28, 31, 34] allow recovering larger deformations, although their performance highly depends on having overlapping features in neighboring patches, or require large number of correspondences to enforce local rigidity constraints [11, 31, 34], which can be hard to obtain in practice. In any event, these previous approaches batch process all frames of the sequence at once, after video capture, preventing them from being used on-line and in real-time applications. This has been recently addressed in [2, 4, 24], which, however, still focus on global models only valid for relatively small deformations [24] or continuous surfaces [2, 4].

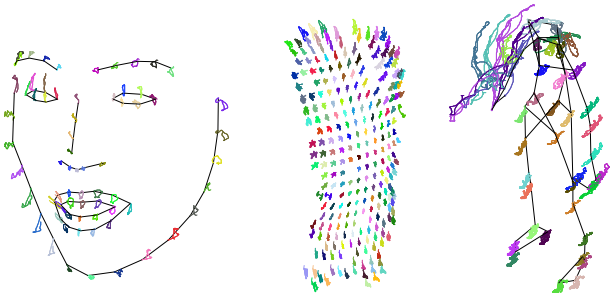

Figure 1. 3D Reconstruction using our physically-inspired velocity model for different types of deformations: face, human torso and articulated motion. Each line represents the per point non-rigid motion detected by our algorithm. Best viewed in color.

An alternative to statistical and low-rank approaches is to directly model the physical laws that locally govern object kinematics. Drawing inspiration from computer graphics [27], there have been several attempts at using these models for tracking non-rigid motion [20] and human activities [10]. Unfortunately, these methods are usually focused to specific types of motion, and their underlying laws rely on non-linear relations complex to optimize. An interesting exception is [29], which directly uses the Newton's second law of motion to build a convex formulation for tracking purposes. This work, though, is not sequential, does not estimate the camera pose, as we do, and still holds on priors

from training data when dealing with complex models (e.g. human motion).

In this paper, we also exploit Newton's second law of motion, but in contrast to [29], we do not directly optimize over these constraints, but leverage them to introduce a *force perturbed* second-order Markov model that rules the local motion of every particle conforming the shape. The joint dynamics are then optimized using a Bundle Adjustment (BA) framework, with simple regularization terms that ensure temporal and global spatial consistency of the estimated shape and camera poses. The resulting approach is sequential, fast, can cope with missing data and with different types of deformations such as articulated, isometric and stretchable, without requiring pre-trained data. We demonstrate the effectiveness on both synthetic and real monocular video sequences, such as those depicted in Fig. 1, and show comparable results to competing batch algorithms, but at a much smaller cost. Additionally, our approach yields remarkable improvement when compared to other sequential NRS*f*M methods.

2. Related work

NRS*f*M is an inherently ambiguous problem that to be solved requires a priori knowledge of either the nature of the deformations or the camera path. Early NRS*f*M approaches extended the Tomasi and Kanade's factorization algorithm [32] to the non-rigid case by representing deformations as linear combinations of basis shapes under orthography [9, 36]. On top of this, spatial [33] and temporal [8, 13, 33] smoothness priors have been considered to further limit the solution space. Later, [12] relaxed the amount of extra prior knowledge by directly imposing a low-rank constraint on the factorization of the measurement matrix. Other approaches have modeled deformation using a low-rank trajectory basis per 3D point [6] and enforcing smoothness on their paths [16]. One inherent limitation of these methods, is that they are highly sensitive to the number of bases chosen to represent the trajectory, making them very problem specific. Additionally, while being adequate to encode global deformations, low-rank methods' applicability is limited to smoothly deforming objects.

Recently, results from this field have significantly advanced. Stronger deformations have been tackled using piecewise models [11, 28, 31], or eliminating the rank dependency by means of Procustean normal distributions [17]. In [14], a variational approach combining a low-rank shape model with local smoothness allowed per-pixel dense reconstructions.

In any event, all aforementioned NRS*f*M works are batch and they need all the frames in the sequence at once, preventing thus, online and real-time computations. While sequential solutions exist for the rigid case [22, 23], sequential estimation of deformable objects based only on the measurements up to that moment remains a challenging and unsolved problem. There are just a few attempts along this direction [2, 3, 4, 24]. Specifically, Paladini *et al*. [24] proposed a 3D-implicit low-rank model to encode the time-varying shape, estimating the remaining model parameters by BA over a temporal sliding window. Agudo *et al*. [4] introduced linear elasticity by means of finite element models into an extended Kalman filter to encode extensible deformations in real-time. Very recently, [2, 5] presented the first approach to reconstruct both sparse and dense 3D shapes in a sequential fashion, relying on a linear subspace of mode shapes computed by modal analysis. However, despite being very promising, these methods are only valid to handle smoothly deforming objects, as is the case of [24], and cannot be applied to articulated motion [4, 5].

An alternative to these approaches is to consider the object as a system of individual particles and represent global deformation by locally modeling the underlying physical laws that govern each of the particles. This has been typically used in computer graphics for simulation purposes [7, 27], and further exported to computer vision applications, for non-rigid tracking of surfaces [20] or articulated bodies [10, 29, 35]. Yet, none of these approaches tackles the problem of besides retrieving shape, estimating the camera pose parameters.

Contribution: In this paper we overcome most of the limitations of previous approaches. We propose a sequential solution to simultaneously recover camera motion and non-rigid shape from point tracks in a monocular video. To this end, we represent the object as an ensemble of particles and employ the Newton's second law to constrain their motion, according to a constant velocity model with acting forces. Global and temporal consistency is enforced by combining this dynamical model with simple regularization terms into a BA framework. Our method can handle both articulated and non-rigid motion without requiring any training data, achieving similar accuracies as batch methods, or approaches relying on pre-trained models.

3. Classical mechanics motion model

The deformation model we propose holds on the Newton's second law of motion, which is satisfied by any real-world object. We next review its general formulation.

We assume our object is represented by a system of n particles (as shown in Fig. 2). Let $\mathbf{y}_i^t \in \mathbb{R}^3$ be the 3D position of the i-th particle at a time instant t and m_i its mass, assumed to be constant. When a force $\mathbf{f}_i^t$ is applied to this particle, Newton's second law of motion states that it produces a proportional acceleration:

$$\mathbf{f}_i^t = m_i \mathbf{a}_i^t = m_i \frac{d\mathbf{v}_i^t}{dt} \,, \tag{1}$$

where $\mathbf{v}_i^t$ is the instantaneous velocity of the particle, and $\mathbf{f}_i^t$

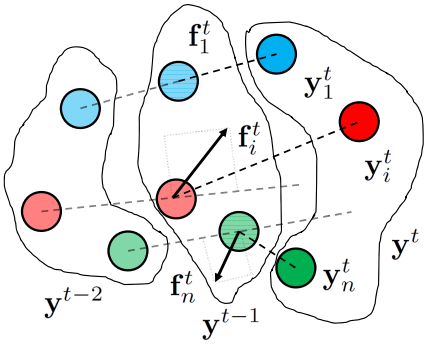

Figure 2. **Force-perturbed motion model for a system of particles.** We use a kinematic model derived from Newton's second law of motion. A particle is moving with constant velocity while no forces are acting on it (blue particle). External forces $\mathbf{f}^t$ can change the dynamical behavior of a single particle (red and green particles), and hence, change the configuration $\mathbf{y}^t$ of the deformable object.

is the sum of all external forces applied to the particle.

In order to derive the formulation of our kinematic model we first approximate the acceleration at time t using backward second-order finite differences:

$$\mathbf{f}_i^t \approx m_i \left[\frac{\mathbf{y}_i^{t-2} - 2\mathbf{y}_i^{t-1} + \mathbf{y}_i^t}{(\Delta t)^2} \right] , \tag{2}$$

that relates the current force $\mathbf{f}^t$ with the current 3D location $\mathbf{y}^t$ and the locations at previous time instances $\mathbf{y}^{t-1}$ and $\mathbf{y}^{t-2}$. We next extend the model to all the n particles of the deformable object.

Let $\mathbf{y}^t = [(\mathbf{y}_1^t)^\top, \dots, (\mathbf{y}_n^t)^\top]^\top$ be a $3n$ dimensional vector composed of the 3D locations of all particles at time t; and $\mathbf{f}^t = [(\mathbf{f}_1^t)^\top, \dots, (\mathbf{f}_n^t)^\top]^\top$ a $3n$ dimensional vector containing all instantaneous forces. We can then re-write Eq. (2) for all the particles using the following linear system:

$$\mathbf{f}^t = \begin{bmatrix} \mathbf{M} & -2\mathbf{M} & \mathbf{M} \end{bmatrix} \begin{bmatrix} \mathbf{y}^{t-2} \\ \mathbf{y}^{t-1} \\ \mathbf{y}^t \end{bmatrix} , \tag{3}$$

where $\mathbf{M}$ is a $3n \times 3n$ diagonal matrix with entries being the masses of each particle. In practice, we omit them and set $\mathbf{M} = \mathbf{I}$, the $3n \times 3n$ identity matrix. We also omit the term Δt in Eq. (2). By doing this, the forces we estimate will be up to scale, and will be expressed per unit of mass and increment of time, or equivalently, in length units. This lets us to directly relate forces applied to the particles to their displacement. More specifically, the 3D position of the particles at time t can be written according to the following dynamical model:

$$\mathbf{y}^t = \mathbf{f}^t + 2\mathbf{y}^{t-1} - \mathbf{y}^{t-2} = \mathbf{f}^t + \mathbf{d}^t , \tag{4}$$

where $\mathbf{d}^t = 2\mathbf{y}^{t-1} - \mathbf{y}^{t-2}$ is a displacement vector. Observe that when $\mathbf{f}^t = \mathbf{0}$ this dynamical model boils down to a second-order Markov model in which each particle will move with a constant velocity $\mathbf{d}^t$ (see the blue particles in Fig. 2). However, when external forces are acting $\mathbf{f}^t \neq \mathbf{0}$, the particles can change their dynamics, accelerating or even reaching the rest. It is worth to point that a similar kinematic model was already used in [4], but in contrast to our paper, it was a first order Markov model and used to encode the camera motion, and not to encode the motion of each particle conforming the time-varying shape, as we do in this paper.

4. Sequential non-rigid shape and camera pose

In this section, we describe how to exploit the proposed dynamic model to simultaneously, and in a sequential manner, estimate deformable shape and camera pose.

4.1. Problem formulation

Let us consider a deformable object as an ensemble of n particles. At time t we represent the 3D position of all particles with the (previously defined) $3n$ dimensional vector $\mathbf{y}^t$. If we assume an orthographic camera model, the projection of this object can be written as:

$$\mathbf{P}^t = [\mathbf{p}_1^t, \dots, \mathbf{p}_n^t] = \mathbf{R}^t \mathbf{Y}^t + \mathbf{T}^t, \tag{5}$$

where $\mathbf{P}^t$ is the $2 \times n$ measurement matrix, $\mathbf{p}_i^t = [u_i^t, v_i^t]^\top$ are the image coordinates of the i-th particle, $\mathbf{R}^t$ is a 2×3 truncated version of the rotation matrix, and $\mathbf{T}^t$ is a $2 \times n$ matrix that stacks n copies of the bidimensional translation vector $\mathbf{t}^t$. To represent the 3D shape $\mathbf{Y}^t$, we use a permutation operator $\mathcal{P}(\mathbf{y}^t)$ that rearranges the entries of $\mathbf{y}^t$ into a $3 \times n$ matrix such that the i-th column of $\mathbf{Y}^t$ corresponds to the 3D coordinates of the point i.

Given 2D point tracks up to frame t of a monocular video, our problem consists in sequentially and simultaneously estimating the camera motion $(\mathbf{R}^t, \mathbf{t}^t)$ and the deformable 3D shape $\mathbf{Y}^t$.

4.2. Non-linear optimization

We represent the deformable object using Eq. (4), which after applying the operator $\mathcal{P}(\cdot)$, can be rewriten as $\mathbf{Y}^t = \mathbf{F}^t + \mathbf{D}^t$. Note that at frame t the displacement $\mathbf{D}^t = 2\mathbf{Y}^{t-1} - \mathbf{Y}^{t-2}$ is already known, as it only involves the particles position at previous time instances. Therefore, the current 3D shape estimation is reduced to estimating the forces $\mathbf{F}^t$.

In order to solve for $\mathbf{F}^t$ and the pose parameters $\mathbf{R}^t$ and $\mathbf{T}^t$, we perform a BA over a temporal sliding window on the last fames. This is indeed similar to what was done in other sequential NRS*f*M approaches [2, 5, 24], with the key difference that we do not rely on a low-rank model to parameterize the object deformation. The use of the Newton's

second law of motion yields to our method higher generalization properties and major resilience to large non-linear deformations.

More specifically, we consider a temporal window on the last three frames, and jointly represent the projection equations as:

$$\begin{bmatrix}\mathbf{P}^{t-2}\\ \mathbf{P}^{t-1}\\ \mathbf{P}^{t}\end{bmatrix} = \begin{bmatrix}\mathbf{R}^{t-2} & & \\ & \mathbf{R}^{t-1} & \\ & & \mathbf{R}^{t}\end{bmatrix}\begin{bmatrix}\mathbf{Y}^{t-2}\\ \mathbf{Y}^{t-1}\\ \mathbf{F}^{t}+\mathbf{D}^{t}\end{bmatrix} + \begin{bmatrix}\mathbf{T}^{t-2}\\ \mathbf{T}^{t-1}\\ \mathbf{T}^{t}\end{bmatrix}.$$

Since the measurement matrix $\mathbf{P}^t$ may contain lost tracks due to occlusions or outliers, we define $\mathcal{V}^t$ as the set of visible points at time t. We then estimate the model parameters by minimizing the following energy function in terms of $\{\mathbf{R}^j, \mathbf{t}^j, \mathbf{F}^t\}$, with $j = \{t-2, t-1, t\}$:

$$E = E_{img} + \alpha_p E_{pose} + \alpha_s E_{shape} + \alpha_e E_{ext} \qquad (6)$$

where:

$$E_{img} = \sum_{j=t-2}^{t} \sum_{\nu \in \mathcal{V}^j} \|\mathbf{p}_\nu^j - \mathbf{R}^j(\mathbf{q}^j)\mathbf{y}_\nu^j - \mathbf{t}^j\|_{\mathcal{F}}^2$$

minimizes de reprojection error of all observed points in $\mathcal{V}^j$. $\|\cdot\|_{\mathcal{F}}$ represents the Frobenius norm and $\mathbf{R}^j$ are the rotation matrices, which are parameterized using quaternions, $\mathbf{R}^j(\mathbf{q}^j)$, to guarantee orthonormality $\mathbf{R}^j\mathbf{R}^{j\top} - \mathbf{I}_2 = \mathbf{0}$. A second energy term, E_{pose}, serves as a regularizer for the estimated pose indicating the rotation matrices and translation vectors of consecutive frames should agree with one another:

$$E_{pose} = \sum_{j=t-1}^{t} \|\mathbf{q}^j - \mathbf{q}^{j-1}\|_{\mathcal{F}}^2 + \alpha_t \sum_{j=t-1}^{t} \|\mathbf{t}^j - \mathbf{t}^{j-1}\|_{\mathcal{F}}^2 ,$$

where α_t is the specific weight for the translation energy term. Similarly, we have introduced a regularization for the shape, to penalize strong variations in consecutive frames:

$$E_{shape} = \|\mathbf{Y}^t\left(\mathbf{F}^t\right) - \mathbf{Y}^{t-1}\|_{\mathcal{F}}^2 ,$$

where the current shape $\mathbf{Y}^t$ is only function of the estimated force. Finally, we have also considered spatial priors to control the extensibility of the surface. To this end, we regularize the change in the euclidean distance over n_e edges of the object using a Gaussian kernel, where d_e^r represents the initial estimated length for edge e and d_e^t is the length for current frame:

$$E_{ext} = \sum_{e=1}^{n_e} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{d_e^{r\,2}}{2\sigma^2}\right) |d_e^r - d_e^t(\mathbf{F}^t)| .$$

Note that this prior is not a hard constraint, and hence it still permits non-isometric deformations.

We optimize the function $E(\mathbf{R}^j, \mathbf{t}^j, \mathbf{F}^t)$ using sparse Levenberg-Marquardt. The regularization weights α_p, α_t, α_s and α_e are determined empirically, but kept constant in all experiments we describe in the experimental section. Note, again, that in contrast to competing approaches [9, 12], we can deal with missing data and do not require all points to be tracked throughout the whole sequence.

4.3. Initialization upon the arrival of a new image

The optimization function we have presented involves seven different parameters within a temporal window of size three: $\mathbf{R}^{t-2}$, $\mathbf{R}^{t-1}$, $\mathbf{R}^t$, $\mathbf{t}^{t-2}$, $\mathbf{t}^{t-1}$, $\mathbf{t}^t$ and $\mathbf{F}^t$. Upon the arrival of a new image, and its associated measurement matrix $\mathbf{P}^t$, these parameters need to be given an initial value, and since Eq. (6) is highly non-linear, it is important not to initialize their values at random. In particular $\mathbf{R}^{t-2}$, $\mathbf{R}^{t-1}$, $\mathbf{t}^{t-2}$ and $\mathbf{t}^{t-1}$ are initialized to the values we have estimated when evaluating frames $t-2$ and $t-1$. The translation vector $\mathbf{t}^t$ is simply initialized to the mean of the measurement matrix $\mathbf{P}^t$. The initialization of $\mathbf{R}^t$ and $\mathbf{F}^t$ is a bit trickier. We next describe how we do it.

Initialization of $\mathbf{R}^t$: Even though we could initialize $\mathbf{R}^t$ to $\mathbf{R}^{t-1}$, we decided not doing so, and start with a better initial value that yields the best fit of $\mathbf{Y}^{t-1}$ onto the current observations $\mathbf{P}^t$, assuming just a rigid motion. This brings faster convergence rate to the subsequent bundle adjustment procedure. More specifically, we seek to retrieve the initial value of $\mathbf{R}^t$ such that:

$$\arg\min_{\mathbf{R}^t} \sum_{\nu \in \mathcal{V}^t} \|\mathbf{p}_\nu^t - \mathbf{R}^t\mathbf{y}_\nu^t - \mathbf{t}^t\|_{\mathcal{F}}^2 \qquad (7)$$

where all parameters but $\mathbf{R}^t$ are known. Recall that $\mathbf{R}^t$ is a 2×3 truncated matrix, which can be computed from a full rotation matrix $\mathbf{Q}^t \in$ SO(3) using $\mathbf{R}^t = \mathbf{\Pi}\mathbf{Q}^t$, and where $\mathbf{\Pi}$ is the orthographic camera matrix. In oder to solve Eq. (7), while ensuring the resulting $\mathbf{Q}^t$ to lie on SO(3) group, we have followed a standard Newton algorithm for optimizing on manifolds [18, 30], which usually converges in one single iteration. We refer the reader to these papers for further details.

Initialization of $\mathbf{F}^t$: Let $\bar{\mathbf{P}}^t$ and $\bar{\mathbf{D}}^t$ be the known measurement and displacement matrices for the set of visible points $\mathcal{V}^t$, and $\mathbf{R}^t$, $\mathbf{T}^t$ the initialization values for the pose. In order to estimate an initial value for the force matrix $\bar{\mathbf{F}}^t$ (also for the visible particles) we minimize the reprojection error:

$$\arg\min_{\bar{\mathbf{F}}^t} \|\bar{\mathbf{P}}^t - \mathbf{R}^t\left(\bar{\mathbf{F}}^t + \bar{\mathbf{D}}^t\right) - \mathbf{T}^t\|_{\mathcal{F}}^2 \qquad (8)$$

We solve this minimization in closed form. To this end, we first rewrite our problem as that of estimating $\bar{\mathbf{F}}^t$ such that $\bar{\mathbf{P}}^t - \mathbf{R}^t\bar{\mathbf{D}}^t - \mathbf{T}^t = \mathbf{R}^t\bar{\mathbf{F}}^t$. Then, $\bar{\mathbf{F}}^t$ can be computed as:

$$\bar{\mathbf{F}}^t = ((\mathbf{R}^t)^\top\mathbf{R}^t)^{-1}(\mathbf{R}^t)^\top(\bar{\mathbf{P}}^t - \mathbf{R}^t\bar{\mathbf{D}}^t - \mathbf{T}^t) \qquad (9)$$

Since the matrix $(\mathbf{R}^t)^\top \mathbf{R}^t$ is ill-conditioned, we add a damping term on its diagonal before computing the actual inverse.

On the other hand, for the subset of occluded points we set their initial vector of forces to those estimated in the previous frame, i.e., $\hat{\mathbf{F}}^t \equiv \hat{\mathbf{F}}^{t-1}$. Finally, we take $\mathbf{F}^t \equiv \bar{\mathbf{F}}^t \cup \hat{\mathbf{F}}^t$.

4.4. Initial model estimation

We next describe how the shape at rest and the initial pose values are set at the beginning of the sequence. For this purpose, we follow [2, 5, 24], and assume that the sequence contains a few initial frames where the object does not undergo large deformations. We use a standard practice done in NRS*f*M, that is running a rigid factorization algorithm [19] on these first frames –instead of using all sequence– to obtain a shape and pose estimate. Once this initialization is done, we then run our approach, which just for the first incoming image uses the assumption that $\mathbf{y}^{t-2} = \mathbf{y}^{t-1}$, i.e., it assumes each particle has null velocity.

4.5. Computational cost

Since we estimate a perturbation force per point, the complexity of our BA algorithm is dominated by the solution of the linear system within the Levenberg-Marquardt process with $\mathcal{O}(\mathcal{W}^3 n^3)$ cost, being $\mathcal{W}$ the size of the temporal window. Indeed, as we only consider a window of size $\mathcal{W} = 3$ this term is negligible in this analysis and our complexity is $\mathcal{O}(n^3)$. With these values, we can achieve real-time performance for models of about one hundred points. For instance, in the experiments we report in the next section, we achieve a frame rate of about 5 fps when dealing with a model of approximately 40 points. Since these results are obtained with unoptimized Matlab code, they can still be significantly speeded up.

5. Experimental evaluation

In this section we present experimental results for different types of deformations, including articulated and non-rigid motion (some examples are shown in Fig. 1). We provide both qualitative results[1] and quantitative evaluation, where we compare our method to several state-of-the-art approaches. In particular, we report the standard 3D reconstruction error given by:

$$e_{3\mathcal{D}} = \frac{1}{n_f} \sum_{t=1}^{n_f} \frac{\|\tilde{\mathbf{Y}}^t - \tilde{\mathbf{Y}}^t_{GT}\|_{\mathcal{F}}}{\|\tilde{\mathbf{Y}}^t_{GT}\|_{\mathcal{F}}}, \tag{10}$$

where $\tilde{\mathbf{Y}}^t$ is the estimated 3D reconstruction, $\tilde{\mathbf{Y}}^t_{GT}$ is the corresponding ground truth, and n_f is the total number of non-rigid frames in the sequence. The $e_{3\mathcal{D}}$ is computed after aligning the estimated 3D shape with the 3D ground truth using Procrustes analysis over all frames.

[1] Videos of the experimental results can be found on website `http://webdiis.unizar.es/~aagudo`

5.1. Motion capture data

We first evaluate our method on several existing datasets with 3D ground truth. We use the following motion capture sequences: *Drink*, *Stretch* and *Yoga* from [6], for evaluating articulated motion; the face deformation sequences *Jacky* and *Face*, from [33] and [25], respectively; and finally the synthetic bending *Shark* sequence from [33].

We compare our approach (denoted PSMM, from Particle Sequential Motion Model) against eight state-of-the-art methods, both batch and sequential approaches. Among the batch algorithms we consider: EM-PPCA [33], the Metric Projections (MP) [25], the DCT-based 3D point trajectory (PTA) [6], the Column Space Fitting (CSF2) [16], the Kernel Shape Trajectory Approach (KSTA) [15] and the block matrix method for SPM [12]. We also consider the following sequential methods: Sequential BA (SBA) [24], and the BA with Finite Elements formulation (BAFEM) of [2]. The parameters of these methods were set in accordance with their original papers. We exactly use the same initialization for our proposed method, SBA [24] and BAFEM [2].

Table 1 summarizes the results. It can be seen that our approach consistently outperforms the other sequential methods, specially SBA [24] while being more generally applicable than BAFEM [2], that cannot model articulated motion. Our results are also comparable to batch methods, where all frames need to be available in advance. Additionally, most of these methods are very sensitive to the choice of the specific rank of the deformation model. We do not require any of this fine tuning. Fig. 3 shows the 3D reconstruction results on several frames of these mocap evaluation sequences.

5.2. Real videos

In this section, we evaluate our approach on several available real sequences. We next provide qualitative evaluation on four different sequences, going from smooth continuous warps to abrupt deformations produced by a newspaper being torn apart.

The *Actress* sequence, is made of 102 frames showing a woman simultaneously talking and moving her head. We rely on the sequence tracks from [8], and as is also done in sequential methods [2, 24], we use the first 30 frames to compute the initial model. Fig. 5, shows the 3D reconstruction we obtain rotated according to the estimated rotation matrices, that is comparatively very similar to those obtained by [2, 24]. Fig. 4 depicts the camera rotation we estimated, showing a smooth motion.

The *Tear* sequence [31] contains 167 frames of a paper being split in two parts. We use the point tracks provided

Met. / Seq.	*Batch Methods*						*Sequential Methods*		
	EM-PPCA [33]	MP [25]	PTA [6]	CSF2 [16]	KSTA [15]	SPM [12]	SBA [24]	BAFEM [2]	PSMM
Drink [6]	5.56(5)	4.14(6)	1.38(13)	1.14(6)	0.94(12)	1.60(12)	11.25(12)	-	1.93
Stretch [6]	13.72(15)	8.13(5)	3.85(8)	2.46(8)	2.00(7)	1.86(11)	17.61(20)	-	5.76
Yoga [6]	11.89(14)	12.98(8)	2.42(8)	1.84(7)	2.12(7)	1.65(10)	15.84(20)	-	6.65
Shark [33]	1.82(2)	9.34(23)	5.91(6)	1.09(5)	1.03(3)	6.29(2)	8.81(5)	-	6.99
Jacky [33]	1.80(5)	2.74(5)	2.69(3)	1.93(5)	2.12(4)	1.82(7)	2.90(16)	3.43(15)	2.80
Face [25]	7.30(9)	3.77(7)	5.79(2)	6.34(5)	6.14(8)	2.67(9)	6.92(27)	6.89(2)	4.49

Table 1. **Quantitative comparison on motion capture sequences.** We show $e_{3\mathcal{D}}[\%]$ for batch methods EM-PPCA [33], MP [25], PTA [6], CSF2 [16], KSTA [15] and SPM [12]; and for sequential methods SBA [24], BAFEM [2] and our approach denoted as PSMM. For low-rank based methods, we have selected the rank in the basis (in brackets) that gave the lowest $e_{3\mathcal{D}}$ error.

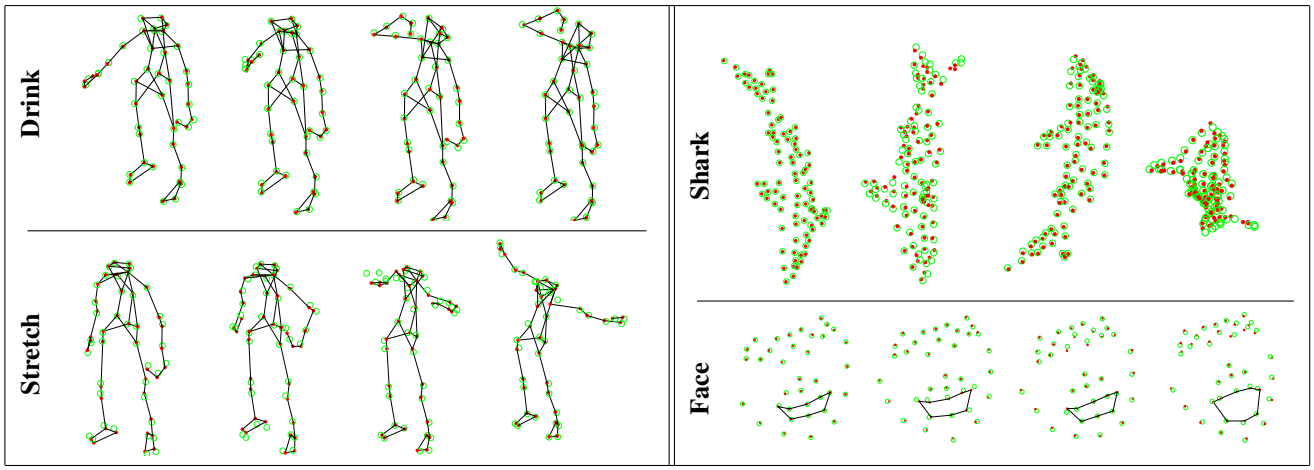

Figure 3. **Motion capture sequences.** We show our 3D reconstruction with red dots and 3D ground truth with green circles. **Left:** Articulated motion for *drink* and *stretch* sequences. **Right:** Non-rigid motion for synthetic bending *shark* and *face* sequences.

by [28]. Again, the first 30 frames of the sequence are used to initialize the model. For this specific experiment we set the weight α_e of the extensibility term in Eq. (6) to zero, to allow the model to be split in two, without the need of exactly knowing the edges that suffer the cut. Fig. 6 shows a few 3D reconstructions obtained with our approach and with CSF2 [16] using a rank in the basis of 5. Although both solutions are similar, the batch method CSF2 [16] produces a cut before the actual separation in two parts is produced.

The *Back* sequence consists of 150 frames showing the back of a person deforming sideways and flexing. We use the sparse point tracks of [28] and the first 20 frames to compute the initial model. Fig. 7 shows a few 3D reconstructions obtained with our approach and with CSF2 [16] with a rank in the basis of 5. This is one of the batch methods with better performance in the mocap experiments of the previous section, specially under significant changes of the camera rotation, as is this experiment (see Fig. 4, bottom-left). Observe, again from Fig. 7, that qualitatively the two approaches are very similar, despite CSF2 [16] produces some combinations of concave/convex regions (marked with magenta) which do not seem very realistically plausible.

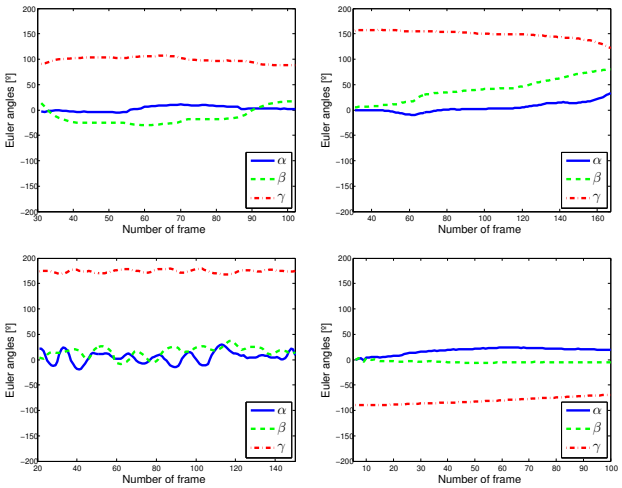

Figure 4. **Rotation estimation on real videos.** We display the estimate Euler angles. **Top:** Actress and Tear sequence. **Bottom:** Back and Bending sequence.

Finally, we process a *Paper Bending* sequence of 100 frames already used in [8]. In this experiment we show a qualitative evaluation with respect to missing data, which

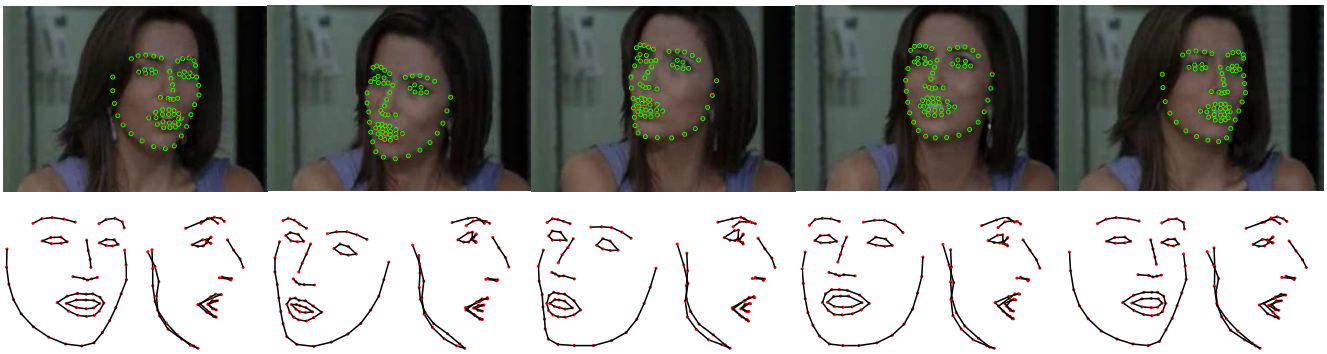

Figure 5. **Actress sequence. Top:** Frames #31, #48, #66, #84 and #102 with 2D tracking data and reprojected 3D shape with green circles and red dots respectively. **Bottom:** Original viewpoint and side views of our 3D reconstruction.

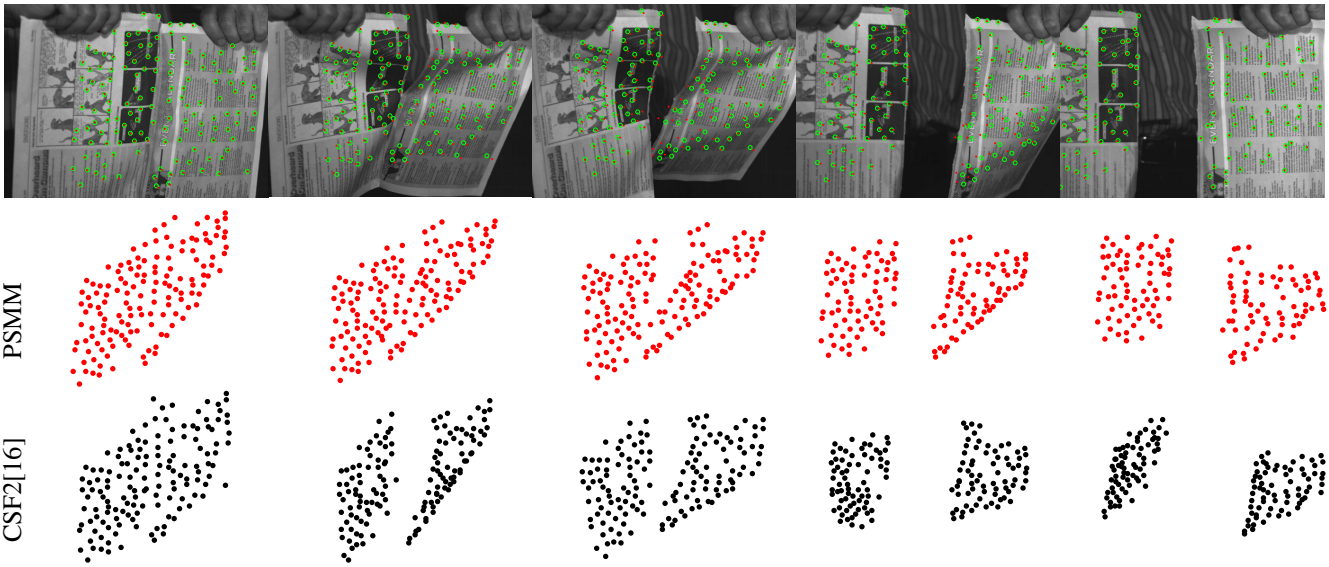

Figure 6. **Tear sequence. Top:** Frames #31, #52, #64, #82 and #123 with 2D tracking data and reprojected 3D shape with green circles and red dots respectively. **Bottom:** General views of our 3D reconstruction and CSF2 [16]. Note that the batch method CSF2 [16] splits the paper in two parts before observing it.

our BA-based approach can naturally handle. In particular, we add a random pattern of 20% of missing data in the measurement matrix. In Fig. 8, we report our 3D reconstruction. Again, we include the reconstruction result obtained with the batch CSF2 [16]. Note, however, that in this case the performance of this algorithm drops significantly, even without the presence of outliers. This is due, as pointed in [14], that trajectory-based algorithms become unstable when dealing with small camera rotations, as is the case of this experiment (see bottom-right graph in Fig. 4).

6. Conclusions

In this paper we have exploited Newton's second law of motion to model the non-rigid deformation of an object represented by a system of particles. We have introduced this simple physics-based dynamical model into a bundle adjustment framework, yielding an approach that allows to simultaneously and on-the-fly recover camera motion and time-varying shape. Our system can handle different types of deformations, including articulated, non-rigid, isometric and extensible cases. Additionally, we do not require of any learning data and the overall solution is remarkably fast. All our claims have been experimentally validated on mocap and real sequences showing a similar performance to computationally intensive batch approaches, and being remarkably more accurate than state-of-the-art sequential approaches. Regarding real-time capability, our approach ensures that the computational cost per frame is bounded and does not grow with the number of frames. We believe our method is a suitable groundwork for later exploitation in real-time applications. Our future work is oriented to generalize our model to full perspective projection cameras and incorporating the feature tracking and outlier detection into a single process.

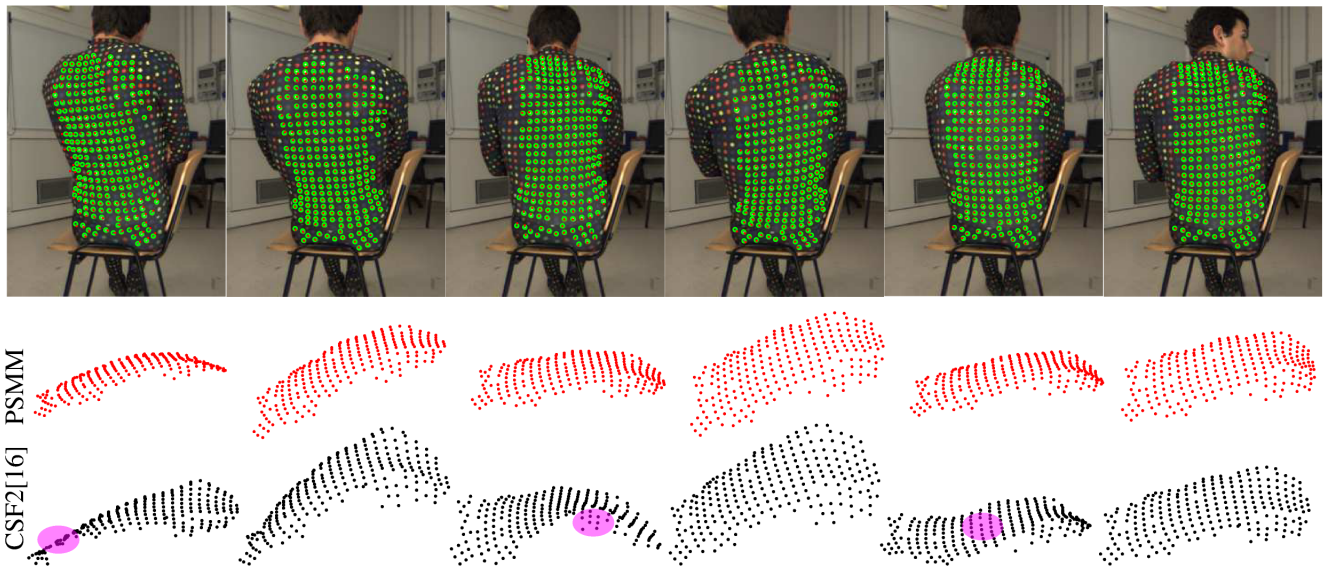

Figure 7. **Back sequence. Top:** Frames #30, #53, #82, #113, #137 and #148 with 2D tracking data and reprojected 3D shape with green circles and red dots respectively. **Bottom:** General view of the 3D reconstruction obtained with our sequential method and CSF2 [16], that batch processes all frames. In magenta we highlight small artifacts of the reconstruction. Best viewed in color.

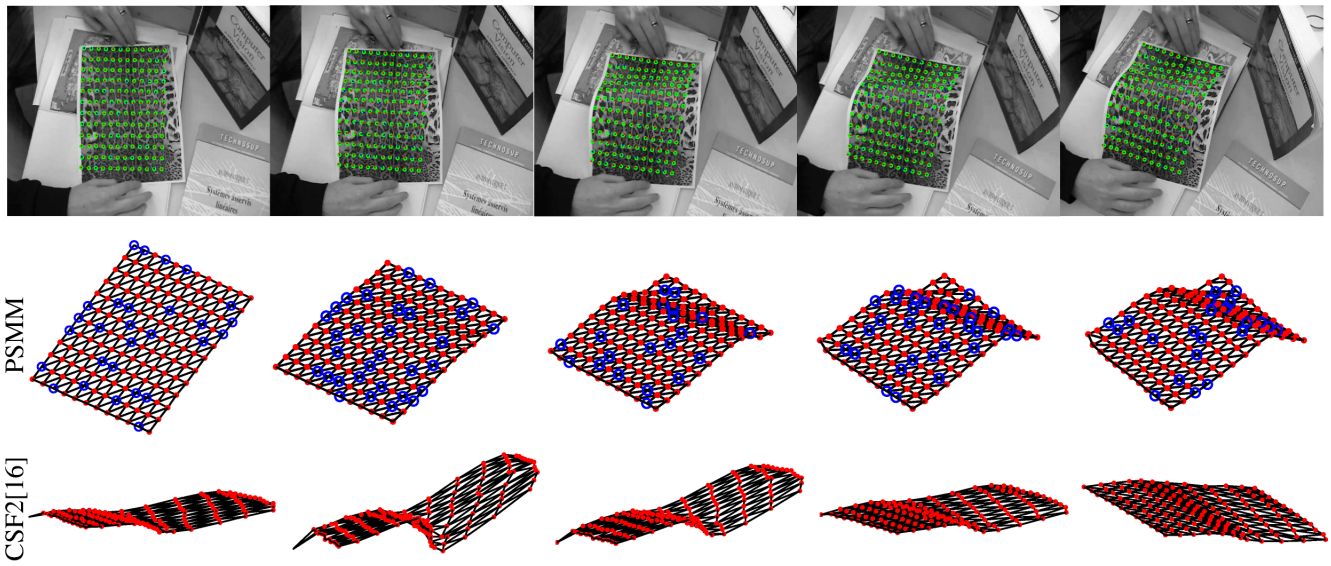

Figure 8. **Paper Bending sequence. Top:** Frames #20, #40, #60, #80 and #100 with 2D tracking data in green circles, and reprojected 3D shape with red and blue circles. Blue circles correspond to missing data. **Bottom:** General view of the 3D shape obtained with our sequential method and CSF2 [16], that batch processes all frames simultaneously. Since this sequence only shows small changes in the rotation, CSF2 [16] becomes highly unestable. Best viewed in color.

Acknowledgments

This work was partly funded by the MINECO projects DIP2012-32168 and DPI2011-27510; by the ERA-net CHISTERA project VISEN PCIN-2013-047; by the EU Projects PopCorn FP7-SME-2013 606634 and ARCAS FP7-ICT-2011-287617; and by a scholarship FPU12/04886 of the Spanish MECD. The authors also thank Chris Russell and Lourdes Agapito for making their data available.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building Rome in a day. In *ICCV*, 2009.

[2] A. Agudo, L. Agapito, B. Calvo, and J. M. M. Montiel. Good vibrations: A modal analysis approach for sequential non-rigid structure from motion. In *CVPR*, 2014.

[3] A. Agudo, B. Calvo, and J. M. M. Montiel. 3D reconstruction of non-rigid surfaces in real-time using wedge elements.

In *ECCVW*, 2012.

[4] A. Agudo, B. Calvo, and J. M. M. Montiel. Finite element based sequential bayesian non-rigid structure from motion. In *CVPR*, 2012.

[5] A. Agudo, J. M. M. Montiel, L. Agapito, and B. Calvo. Online dense non-rigid 3D shape and camera motion recovery. In *BMVC*, 2014.

[6] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Non-rigid structure from motion in trajectory space. In *NIPS*, 2008.

[7] D. Baraff. Analytical methods for dynamic simulation of non-penetrating rigid bodies. In *ACM SIGGRAPH*, 1989.

[8] A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S. Olsen, and P. Sayd. Coarse-to-fine low-rank structure-from-motion. In *CVPR*, 2008.

[9] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *CVPR*, 2000.

[10] M. Brubaker, L. Sigal, and D. Fleet. Estimating contact dynamics. In *ICCV*, 2009.

[11] A. Chhatkuli, D. Pizarro, and A. Bartoli. Non-rigid shape-from-motion for isometric surfaces using infinitesimal planarity. In *BMVC*, 2014.

[12] Y. Dai, H. Li, and M. He. A simple prior-free method for non-rigid structure from motion factorization. In *CVPR*, 2012.

[13] A. Del Bue, X. Llado, and L. Agapito. Non-rigid metric shape and motion recovery from uncalibrated images using priors. In *CVPR*, 2006.

[14] R. Garg, A. Roussos, and L. Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *CVPR*, 2013.

[15] P. F. U. Gotardo and A. M. Martinez. Kernel non-rigid structure from motion. In *ICCV*, 2011.

[16] P. F. U. Gotardo and A. M. Martinez. Non-rigid structure from motion with complementary rank-3 spaces. In *CVPR*, 2011.

[17] M. Lee, J. Cho, C. H. Choi, and S. Oh. Procrustean normal distribution for non-rigid structure from motion. In *CVPR*, 2013.

[18] Y. Ma, J. Kosecka, and S. Sastry. Optimization criteria and geometric algorithms for motion and structure estimation. *IJCV*, 44(3):219–249, 1999.

[19] M. Marques and J. Costeira. Optimal shape from estimation with missing and degenerate data. In *WMVC*, 2008.

[20] D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *TPAMI*, 15(6):580–591, 1993.

[21] F. Moreno-Noguer and J. M. Porta. Probabilistic simultaneous pose and non-rigid shape recovery. In *CVPR*, 2011.

[22] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Generic and real-time structure from motion using local bundle adjustment. *IMAVIS*, 27(8):1178–1193, 2009.

[23] R. Newcome and A. J. Davison. Live dense reconstruction with a single moving camera. In *CVPR*, 2010.

[24] M. Paladini, A. Bartoli, and L. Agapito. Sequential non rigid structure from motion with the 3D implicit low rank shape model. In *ECCV*, 2010.

[25] M. Paladini, A. Del Bue, M. Stosic, M. Dodig, J. Xavier, and L. Agapito. Factorization for non-rigid and articulated structure using metric projections. In *CVPR*, 2009.

[26] H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh. 3D reconstruction of a moving point from a series of 2D projections. In *ECCV*, 2010.

[27] Z. Popovic and A. Witkin. Physically based motion transformations. In *ACM SIGGRAPH*, 1999.

[28] C. Russell, J. Fayad, and L. Agapito. Energy based multiple model fitting for non-rigid structure from motion. In *CVPR*, 2011.

[29] M. Salzmann and R. Urtasun. Physically-based motion models for 3D tracking: A convex formulation. In *ICCV*, 2011.

[30] A. Shaji and S. Chandran. Riemannian manifold optimisation for non-rigid structure from motion. In *CVPRW*, 2008.

[31] J. Taylor, A. D. Jepson, and K. N. Kutulakos. Non-rigid structure from locally-rigid motion. In *CVPR*, 2010.

[32] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization approach. *IJCV*, 9(2):137–154, 1992.

[33] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: estimating shape and motion with hierarchical priors. *TPAMI*, 30(5):878–892, 2008.

[34] A. Varol, M. Salzmann, E. Tola, and P. Fua. Template-free monocular reconstruction of deformable surfaces. In *ICCV*, 2009.

[35] M. Vondrak, L. Sigal, and O. C. Jenkins. Physical simulation for probabilistic motion tracking. In *CVPR*, 2008.

[36] J. Xiao, J. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion. *IJCV*, 67(2):233–246, 2006.