A Scalable, Efficient, and Accurate Solution to Non-Rigid Structure from Motion

Antonio Agudo and Francesc Moreno-Noguer

Abstract

Most Non-Rigid Structure from Motion (NRSfM) solutions are based on factorization approaches that allow reconstructing objects parameterized by a sparse set of 3D points. These solutions, however, are low resolution and generally, they do not scale well to more than a few tens of points. While there have been recent attempts at bringing NRSfM to a dense domain, using for instance variational formulations, these are computationally demanding alternatives which require certain spatial continuity of the data, preventing their use for articulated shapes with large deformations or situations with multiple discontinuous objects. In this paper, we propose incorporating existing point trajectory low-rank models into a probabilistic framework for matrix normal distributions. With this formalism, we can then simultaneously learn shape and pose parameters using expectation maximization, and easily exploit additional priors such as known point correlations. While similar frameworks have been used before to model distributions over shapes, here we show that formulating the problem in terms of distributions over trajectories brings remarkable improvements, especially in generality and efficiency. We evaluate the proposed approach in a variety of scenarios including one or multiple objects, sparse or dense reconstructions, missing observations, mild or sharp deformations, and in all cases, with minimal prior knowledge and low computational cost.

Index Terms

Probabilistic trajectory space, Time-varying scenes, Non-Rigid structure from motion, Low-rank representation, Factorization.

1 Introduction

While Structure-from-Motion (SfM) methods have obtained remarkable results in reconstructing rigid scenes from motion cues and perspective cameras [1], [15], [19], the problem of inferring 3D shape of deforming objects is still in its infancy. This task is referred to as Non-Rigid Structure from Motion (NRSfM) and consists of estimating the shape of a time-varying 3D scene from 2D point trajectories acquired with a single color camera. It represents a fundamental problem in computer vision with a number of applications in other fields, including robotics, pattern recognition, computer graphics, mechanical engineering or medical imaging.

The main difficulty to resolve when addressing the NRSfM problem is due to the fact that many different 3D shapes can produce similar image observations, and uniquely considering reprojection constraints is not sufficient to obtain a single solution for the shape. Consequently, additional a priori knowledge about the deformation of the structure and the camera motion is required. In addition, the problem can be simplified by assuming an orthographic camera model, which represents a good approximation when the object depth is much smaller than the distance from the camera. Most existing approaches apply the well-known factorization algorithm for

- *The authors are with the Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Barcelona, 08028, Spain. Email: {aagudo, fmoreno}@iri.upc.edu*

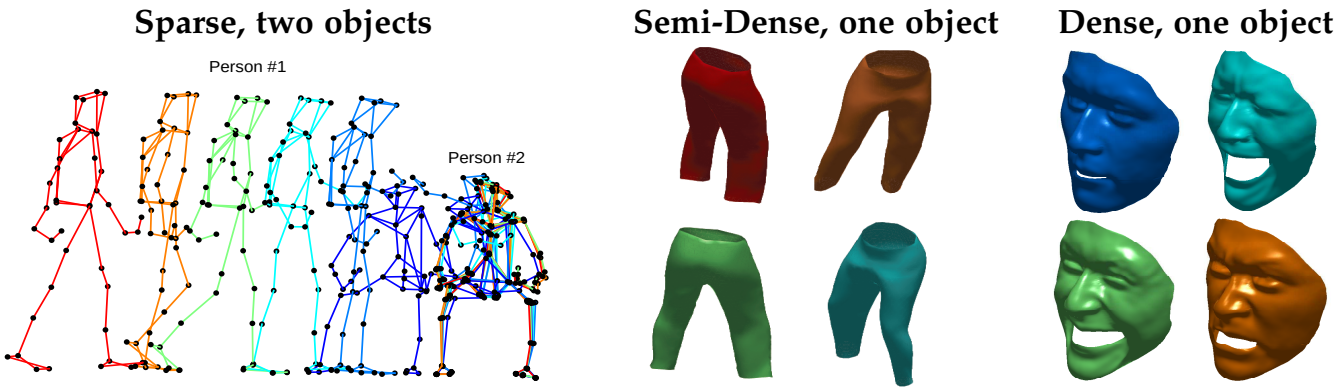

Fig. 1: **Scalable, generic and efficient solution to NRSfM**. We propose a versatile approach that can handle a wide range of scenarios. **Left:** Simultaneous reconstruction of two articulated bodies. Person #2 sits down, and Person #1 walks and approaches until they meet. **Middle:** Semi-dense reconstruction (1,453 points) of deforming pants. **Right:** Dense reconstruction (28,887 points) of a gesturing face. For all cases, the colors encode different temporal configurations of the scenario.

rigid reconstruction [42], and use prior information in form of low-rank shape basis [6], [12], [16], [20], [43], [49]. Similarly, low-rank models have also been proposed to constrain the motion of each point on the object through predefined trajectory bases [24], [25] or dynamic priors on the forces that induce the deformation [5]. However, since these methods need to factorize a matrix of size proportional to the number of input points, they can only be applied to shapes of relatively low resolution.

Recently, NRSfM has been extended to dense reconstruction by formulating the problem as a variational optimization [21] or applying a nuclear minimization algorithm [18]. While these alternatives provide dense 3D estimates at every pixel in the image, they require certain smoothness and continuity constraints on the shape to reconstruct, preventing its use on sparse or articulated configurations, and on scenes composed of multiple objects. Still, these solutions are prone to be computationally expensive and require dedicated GPU-implementations [21].

In this work, we present a probabilistic framework that overcomes most of the aforementioned limitations. For this purpose, we draw inspiration on previous methods that span the trajectory of every object point –or image pixel in the dense case– using a low-rank trajectory model [9], [44], and on those that define Gaussian distributions over shapes [43] or forces [5]. Combining both these ideas, we introduce a new Probabilistic Point Trajectory Approach (PPTA), which allows retrieving object shape and camera pose by just decomposing a matrix whose size is proportional to the dimension of the low-rank space and is independent on the number of object points, all of them in an unsupervised manner. Accordingly, we leverage on the Expectation-Maximization (EM) algorithm, like in [5], [43], although focusing on a point trajectory model instead of a shape or force one. It requires reformulating EM in terms of matrix distributions, and a substantially more elaborate optimization process. In addition, our formulation can incorporate spatial correlation priors that define the similarities between object points, providing better solutions when these relations appear on the data. This new variant of our algorithm is supervised and is denoted as Probabilistic Correlation Point Trajectory Approach (PCPTA). Our result compares favorably to its predecessors in terms of scalability (can handle sparse or dense point configurations), generality (it is applicable to single or multiple objects, and for articulated and continuous deformations) and computational efficiency. Furthermore, in contrast to other factorization-based approaches [9], [16], [21], our approach can naturally incorporate a scheme to handle missing entries, and it is robust against noisy observations.

Figure 1 shows three sample 3D reconstructions obtained with our approach. The left column depicts the estimated 3D shape for two persons interacting but moving independently. The middle column shows several instances of deforming and moving pants, from a sequence of 291 frames and 1,453 points per frame. And finally, the right column corresponds to a dense reconstruction of a deforming face, from a sequence with 99 frames, and 28,887 points per frame. For this last example, we are able to batch process all 2D point tracks and precisely estimate shape and pose for all frames in 8.7 seconds, using unoptimized Matlab code and a CPU-based laptop. We are not aware of any other approach bringing together similar characteristics of versatility, accuracy, and efficiency.

The remainder of this paper is organized as follows. Section 2 discusses the related work in this field and emphasizes on our contribution. In Section 3 we introduce the NRSfM problem, focusing our study on the low-rank trajectory model we use. This is followed in Section 4 by a description of our probabilistic method that can exploit jointly both temporal and spatial priors to simultaneously learn the camera motion and the time-varying 3D shape from 2D trajectories. In Section 5 we present an extensive experimental evaluation on challenging sequences and provide a comparison with respect to state-of-the-art techniques in terms of accuracy and efficiency. Conclusions are described in Section 6.

2 Related Work

Simultaneously reconstructing time-varying 3D shapes along with camera motion from only 2D point tracks is a poorly constrained problem, as different 3D object configurations and camera poses yield very similar 2D image observations. This inherent ambiguity has been tackled by introducing several constraints on the camera motion or type of shape deformation. Most NRSfM methods assume the 3D shape to be spanned by a single low dimensional shape subspace [16], [33], [34], [43], by a dual low-rank shape model [6], or by means of a union of temporal subspaces [49]. Very recently, the concept of compressibility was introduced in NRSfM to enforce a union of subspaces, where a different set of shape bases were employed for each shape instance [29]. As a result of applying the previous models, the NRSfM becomes a trilinear problem that can be solved using factorization techniques [12], [47] or optimization strategies, enforcing spatial [43] or temporal shape smoothness [7], [8], [10], [17], [31], isometry [14], [35], [45], or by imposing the 3D shapes to be closely aligned [30]. The shape deformations were also considered as spatial variations in a shape space, where spatial smoothness, rather than temporal, is enforced [28]. This problem can be reduced by computing the shape basis using training data [33], [39], or by applying modal [3] and spectral [4] analysis in the initial frames for sequential estimation.

Alternatively, a number of approaches have introduced restrictions on the trajectory of every object point using predefined bases which turned the trilinear problem to a bilinear one [9]. This was even further simplified in [36], where additional static points were used to independently solve for the camera motion, resulting finally in a linear problem. In [44], priors on trajectories were introduced in terms of 3D point differentials. Subsequent works have combined shape and trajectory spaces, where the non-rigid shape enforces a smooth time-trajectory of a single point in a linear shape space [24], [25]. In [40], a statistical model of 3D motion based on a Kronecker structure of the spatio-temporal covariance was expressed as a matrix normal distribution. While this approach can independently separate correlations across time and across shape, our formulation can compactly incorporate shape correlations priors if available, in combination to temporal smoothness that is imposed by the trajectory basis.

In any event, the underlying methodologies used by these approaches are not scalable and limit their application domain to low-resolution surfaces or sparse set of points, always for one

single object. Recent approaches, though, have been able to provide dense reconstructions by using variational optimization [21], applying a nuclear minimization algorithm that introduces smoothness priors on the shape [18] or by means of the rediscovered metric projections algorithm [23]. In [38], dense reconstructions have been obtained through piecewise rigid models, but it suffers from the relative low expressiveness of the piecewise models, which limits the applicability to structures with mild deformations. Finally, another category of algorithms have been proposed to retrieve the dense reconstruction in a sequential fashion [2], [3], exploiting physics-based models. However, while these methods represent a step forward on the field, they can only be applied to smooth shapes presenting spatial continuity, and cannot be applied to sparse representations. Additionally, these techniques normally are computationally demanding, and require dedicated GPU-implementations [21].

In this paper, we overcome most of the limitations of previous methods. To this end, we model the well-known trajectory space [9], [44] by means of matrix normal distributions, that enforce spatial smoothness together with the inherent temporal smoothness of the subspace. This results in a novel probabilistic framework which, by construction, does not have a limitation of rank as the previous trajectory-based methods and allow the use of higher frequencies, providing more accurate factorizations in an unsupervised manner. Moreover, our formulation can naturally exploit more sophisticated priors which can not be incorporated in previous formulations, such as similarities between object points, that can be coded by means of a covariance matrix in a supervised manner. We would also like to point out that while the optimization tool we use to learn the trajectory model and the camera pose is the EM algorithm, like other methods [5], [43] in the literature, the underlying model is substantially different. We probabilistically model trajectory distributions instead of shape [43] or force ones [5], which requires having to devise a novel matrix version of the EM algorithm (instead of the standard vectorial one). The experiments section will clearly demonstrate the advantages of this approach with respect to competing techniques, showing its accuracy, versatility, and efficiency.

3 Low-Rank Models on NRSfM

We next review the general description of the NRSfM problem and the standard matrix factorization approach to tackle it.

Consider a T frames video sequence of a time-varying 3D shape and N input 2D point tracks. Let $\mathbf{x}_i^t = [x_i^t, y_i^t, z_i^t]^\top$, with $1 \leq i \leq N$ and $1 \leq t \leq T$, be the 3D coordinates of the i-th point at time t and $\mathbf{u}_i^t = [u_i^t, v_i^t]^\top$ its 2D image projection. If we assume an orthographic camera model we can jointly represent the projection equation of all points for all frames as the following linear system:

$$\underbrace{\begin{bmatrix} \tilde{\mathbf{u}}_1^1 & \dots & \tilde{\mathbf{u}}_N^1 \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{u}}_1^T & \dots & \tilde{\mathbf{u}}_N^T \end{bmatrix}}_{\mathbf{P}\in\mathbb{R}^{2T\times N}} = \underbrace{\begin{bmatrix} \mathbf{R}^1 & & \\ & \ddots & \\ & & \mathbf{R}^T \end{bmatrix}}_{\mathbf{R}\in\mathbb{R}^{2T\times 3T}} \underbrace{\begin{bmatrix} \mathbf{x}_1^1 & \dots & \mathbf{x}_N^1 \\ \vdots & \ddots & \vdots \\ \mathbf{x}_1^T & \dots & \mathbf{x}_N^T \end{bmatrix}}_{\mathbf{S}\in\mathbb{R}^{3T\times N}}$$

where $\tilde{\mathbf{u}}_i^t$ are the zero-mean point coordinates, obtained by subtracting the mean translation vector from the original coordinates, i.e., $\tilde{\mathbf{u}}_i^t = \mathbf{u}_i^t - \mathbf{t}^t$, with $\mathbf{t}^t = \sum_i \mathbf{u}_i^t / N$. The matrix $\mathbf{R}$ is a block diagonal and is made of the T orthographic camera rotations $\mathbf{R}^t \in \mathbb{R}^{2\times 3}$.

In short, the NRSfM problem can be stated as that of simultaneously recovering the pose parameters $\mathbf{R}$ and a deforming shape $\mathbf{S}$, given the 2D trajectories matrix $\mathbf{P}$. Since this is a highly ambiguous problem, one needs to resort to additional constraints, typically in the form of low-rank models. Among these, there exist models spanning the object shape [3], [21], [43], individual point trajectories [9], [36], [44], a combination of shape and trajectory [26], [40], and very recently,

	Deterministic	Probabilistic
Shape	$\mathbf{P}=\mathbf{R}(\mathbf{B}\otimes\mathbf{I}_3)\mathbf{\Phi}$ [12], [17], [47]	$\sum_t \mathbf{p}^t=\sum_t \mathbf{G}^t\mathbf{D}\boldsymbol{\gamma}^t+\mathbf{n}^t,\ \ \boldsymbol{\gamma}^t\sim\mathcal{N}(\mathbf{0};\mathbf{I}_K),\ \ \mathbf{n}^t\sim\mathcal{N}(\mathbf{0};\sigma^2\mathbf{I}_{2N})$ [43]
Trajectory	$\mathbf{P}=\mathbf{R}\mathbf{W}\mathbf{\Phi}$ [9], [48]	$\mathbf{P}=\mathbf{R}\mathbf{W}\mathbf{\Phi}+\mathbf{N},\ \ \text{vec}(\mathbf{\Phi})\sim\mathcal{N}\left(\text{vec}(\mathbf{0});\mathbf{I}_{3K}\otimes\mathbf{C}^{-1}\right),\ \ \text{vec}(\mathbf{N})\sim\mathcal{N}\left(\text{vec}(\mathbf{0});\sigma^2\mathbf{I}_{2T}\otimes\mathbf{C}^{-1}\right)$ Ours
Shape-Trajectory	$\mathbf{P}=\mathbf{R}(\mathbf{E}\mathbf{H}\otimes\mathbf{I}_3)\mathbf{\Phi}$ [24], [26]	$\mathbf{P}=\mathbf{R}(\mathbf{\Theta}\mathbf{\Gamma}\mathbf{\Psi}+\mathbf{N})^{\#}$, $\text{vec}(\mathbf{\Gamma})\sim\mathcal{N}\left(\text{vec}(\mathbf{0});\sigma^2\mathbf{I}_T\otimes\mathbf{I}_{3N}\right),\ \ \text{vec}(\mathbf{N})\sim\mathcal{N}\left(\text{vec}(\mathbf{0});\sigma^2\mathbf{I}_T\otimes\mathbf{I}_{3N}\right)$ [40]
Force	n/a	$\sum_t \mathbf{p}^t=\sum_t \mathbf{G}^t\mathbf{K}\mathbf{F}\boldsymbol{\gamma}^t+\mathbf{n}^t,\ \ \boldsymbol{\gamma}^t\sim\mathcal{N}(\mathbf{0};\mathbf{I}_K),\ \ \mathbf{n}^t\sim\mathcal{N}(\mathbf{0};\sigma^2\mathbf{I}_{2N})$ [5]

TABLE 1: **Low-rank representations to encode NRSfM.** Qualitative comparison of the four subspaces that have been typically proposed on the NRSfM literature: shape, trajectory, shape-trajectory and force models. Both deterministic and probabilistic solutions have been proposed. For easy comparison, we use a unified formulation. Additionally, we define $\mathbf{p}^t=[(\tilde{\mathbf{u}}_1^t)^\top\dots(\tilde{\mathbf{u}}_N^t)^\top]^\top$ and $\mathbf{G}^t=(\mathbf{I}_N\otimes\mathbf{R}^t)$. **Shape models:** In this case, the low-rank coefficients are included in the matrix $\mathbf{B}\in\mathbb{R}^{T\times K}$ or in the vector $\boldsymbol{\gamma}^t\in\mathbb{R}^{K\times 1}$ with $t=\{1,\dots,T\}$ for the deterministic and probabilistic versions, respectively. Due to the shape-trajectory duality theorem [9], the matrix $\mathbf{\Phi}$ also contains the shape basis vectors, the same as $\mathbf{D}\in\mathbb{R}^{3N\times K}$ that rearranges the elements of $\mathbf{\Phi}$. **Trajectory models:** The matrices $\mathbf{W}$ and $\mathbf{\Phi}$ contain the trajectory bases and coefficients, respectively. When spatial correlation priors are not used, the matrix $\mathbf{C}$ can be set to $\mathbf{I}_N$. **Shape-trajectory models:** The matrices $\mathbf{E}\in\mathbb{R}^{T\times D}$ and $\mathbf{H}\in\mathbb{R}^{D\times K}$ include the DCT basis and the corresponding coefficients. For the probabilistic version [40], the matrices $\mathbf{\Theta}\in\mathbb{R}^{T\times T}$ and $\mathbf{\Psi}\in\mathbb{R}^{3N\times 3N}$ account for the trajectory and shape basis; and $\mathbf{\Gamma}\in\mathbb{R}^{T\times 3N}$ is a matrix of mixing coefficients. The operator $(\cdot)^{\#}$ rearranges the elements of a $T\times 3N$ matrix into a $3T\times N$ matrix. **Force models:** These models also learn the compliance matrix $\mathbf{K}\in\mathbb{R}^{3N\times 3N}$ and the force basis $\mathbf{F}\in\mathbb{R}^{3N\times K}$. The probabilistic version of the trajectory model we propose is the only one that works with and without training data, and incorporates temporal, spatial and probabilistic priors. It is suitable for reconstructing scenarios made of sparse or dense point configurations, and for single or multiple objects.

the forces that induce the deformation [5]. A qualitative comparison between the four low-rank variants is shown in Table 1. For each model, we show what we denote as deterministic and probabilistic versions. The former does not explicitly model the uncertainty and builds upon factorization-based approaches, while the latter introduces a probabilistic component in the model, and employs EM-like techniques to resolve each of the components. It is worth pointing out that matrix normal distributions were also used in [40], enforcing a shape-trajectory constraint by means of separable shape and time correlations. In contrast, in our case we present a probabilistic trajectory formulation that, in addition to the temporal constraint, can naturally exploit shape correlations when are available. Moreover, our formulation is more compact since it codes the shape correlations though N-matrices instead of $3N$-matrices as was done in [40]. In this paper, we introduce the probabilistic version of the trajectory-based model, which has not been considered before.

3.1 Low-Rank Trajectory Model

In this paper, we have focused on the trajectory-based low-rank models, because independently modeling point trajectories allows representing larger deformations and a much wider set of scenarios than shape-based models. Conversely, this might have the opposite effect of introducing spatial noise in the reconstructions. However, as we will show in the experimental section, this issue is controlled by the probabilistic point trajectory approach we propose, which in addition to the temporal smoothness due to the trajectory basis, naturally incorporates Gaussian spatial smoothness between object points.

Trajectory-based models [9], [26], [48], approximate the position of every point over time by a linear combination of K low-frequency basis vectors. Although we could model the trajectory basis from different ways (e.g., through the discrete wavelet or Fourier transforms), we use the discrete cosine transform as previously done in [9], [26], [48]. More specifically, for $t = \{1, \dots, T\}$ we define a K-dimensional vector $\mathbf{w}^t = [w_1^t, \dots, w_K^t]^\top$, with:

$$w_k^t = \frac{\rho_k}{\sqrt{T}} \cos\left(\frac{\pi(2t-1)(k-1)}{2T}\right),$$

where $\rho_k = 1$ for $k = 1$, and $\rho_k = \sqrt{2}$, otherwise. The time-varying 3D shape for all T image frames can then be written as $\mathbf{S} = \mathbf{W}\mathbf{\Phi}$, where $\mathbf{W} \in \mathbb{R}^{3T\times 3K}$ is a known matrix with the predefined trajectory basis, that can compactly approximate most real trajectories. $\mathbf{\Phi} \in \mathbb{R}^{3K\times N}$ is a matrix of unknown coefficient vectors $\boldsymbol{\phi}_i \in \mathbb{R}^{3K\times 1}$, for each of the points $i = \{1, \dots, N\}$:

$$\mathbf{W} = \begin{bmatrix} \mathbf{I}_3 \otimes (\mathbf{w}^1)^\top \\ \vdots \\ \mathbf{I}_3 \otimes (\mathbf{w}^T)^\top \end{bmatrix}, \quad \mathbf{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_N \end{bmatrix}, \tag{1}$$

and $\otimes$ denotes the Kronecker product. The linear system of projection equations can then be rewritten as:

$$\mathbf{P} = \mathbf{R}\mathbf{W}\mathbf{\Phi} = \mathbf{A}\mathbf{\Phi} \quad \text{where } \mathbf{A} \in \mathbb{R}^{2T\times 3K}. \tag{2}$$

In matrix factorization, retrieving the factors $\mathbf{A}$ and $\mathbf{\Phi}$ entails performing the Singular Value Decomposition (SVD) $\mathbf{P} = \mathbf{U}_\mathcal{P}\mathbf{\Sigma}_\mathcal{P}\mathbf{V}_\mathcal{P}^\top = (\mathbf{U}_\mathcal{P}\mathbf{\Sigma}_\mathcal{P}^{\frac{1}{2}})(\mathbf{\Sigma}_\mathcal{P}^{\frac{1}{2}}\mathbf{V}_\mathcal{P}^\top)$ and setting to zero all but the largest $3K$ singular values in $\mathbf{\Sigma}_\mathcal{P}$. This solution is non-unique and still requires performing an Euclidean upgrade. We will briefly describe this step for our particular case in Section 5.2.

While matrix factorization approaches are very simple to apply, they have a limitation in the maximum dimension $K_{\text{max}} \geq K$ of the low-rank space [9], [12]. Concretely, this rank is limited, by construction, to the size of the matrix $\mathbf{P}$, which is $2T \times N$, yielding that $K_{\text{max}} = \min(\frac{2T}{3}, \frac{N}{3})$. For large deformations or short but dense sequences this maximum allowed rank may not be sufficient to guarantee accurate 3D reconstructions. A second limitation of the matrix factorization is that when either the number of frames T or points N considered is very large, the computation of the SVD of $\mathbf{P}$ will become very computationally demanding.

The probabilistic trajectory-based formulation we propose in the following section overcomes these limitations since we can use any value of K. This results in a novel method that yields a remarkable speed-up with respect to state-of-the-art techniques while allowing for more accurate 3D reconstructions. In addition, our approach can naturally exploit spatial similarities between feature points, providing better solutions when the deformations include point correlations.

4 Probabilistic NRSfM Model with Spatio-Temporal Priors

In order to give a probabilistic interpretation to the NRSfM problem, let us consider the observed 2D point tracks to be corrupted by a Gaussian noise, which we represent by a matrix $\mathbf{N} \in \mathbb{R}^{2T\times N}$. Equation (2) then becomes:

$$\mathbf{P} = \mathbf{A}\mathbf{\Phi} + \mathbf{N}\,. \tag{3}$$

Our problem consists in simultaneously estimating the camera motion $\mathbf{R}$ and the trajectory coefficients $\mathbf{\Phi}$ (or equivalently, the time-varying 3D shape $\mathbf{S} \equiv \mathbf{W}\mathbf{\Phi}$) given 2D point tracks $\mathbf{P}$ on a monocular video corrupted by Gaussian noise $\mathbf{N}$. It is worth noting that since we are considering an orthographic camera model the translation at each frame can be estimated as the mean position of all observed 2D points (equivalent to the maximum-likelihood estimator), as we do in section 3.

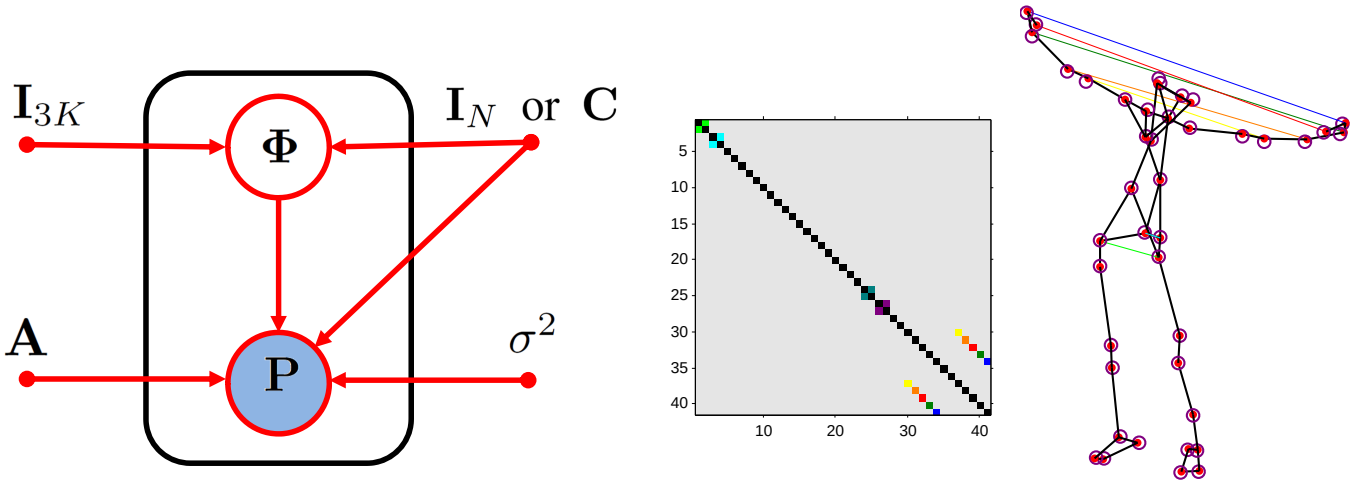

Fig. 2: **Graphical representation of the probabilistic NRSfM problem with point trajectory model.** The observation matrix is represented by $\mathbf{P}$, which assumes centered 2D coordinates. $\mathbf{\Phi}$ corresponds to the latent variable matrix and the model parameters are the observation noise variance σ^2 and $\mathbf{A} = \mathbf{RW}$, where $\mathbf{R}$ are the rotation parameters and $\mathbf{W}$ the predefined trajectory basis. The matrix $\mathbf{C}$ encodes correlation between observations. In the most general case, this correlation is not known, and observations are assumed to be independent and identically distributed. In this case, $\mathbf{C}$ is set to the identity matrix $\mathbf{I}_N$. At the center we represent a toy example of a correlation matrix corresponding to a person stretching in which we use a priori knowledge to link the motion of several joints (e.g., hands, arms, and hips). These links are the non-diagonal entries of $\mathbf{C}$.

Probabilistic strategies have been used before to estimate vector-based normal distributions over shape [43] or force [5] low-rank coefficients, and matrix-based normal distributions over shape-trajectory coefficients (see Table 1). Here, we show that writing the problem in matrix form yields a straightforward manner to describe distribution over trajectories. Maximum Likelihood Estimation (MLE) is then applied to these distributions in order to estimate, using EM, the corresponding pose and noise parameters. Retrieving shape involves a final metric update stage.

In this paper, we present two probabilistic algorithms to solve the NRSfM problem by exploiting spatio-temporal priors. First, we propose a Probabilistic Point Trajectory Approach (PPTA), modeling the unknown coefficients by means of variable normal distributions. In this case, temporal smoothness priors are imposed by the trajectory basis we include in $\mathbf{A}$ by means of the matrix $\mathbf{W}$, and the model parameters are learned in an unsupervised manner. Second, we reformulate our model in terms of a Probabilistic Correlation Point Trajectory Approach (PCPTA), to present a supervised method where the spatial affinities between object points are also exploited.

4.1 Probabilistic Point Trajectory Model (PPTA)

We first consider our data and noise process to be independent and identically distributed. This means the covariance matrix between the instances can be modeled by $\mathbf{I}_N$. As it is standard in PPCA [37], [41], we assume a zero-mean Gaussian prior distribution on the latent variables $\mathbf{\Phi}$, i.e., the trajectory coefficients, and a zero-mean Gaussian distribution with variance σ^2 on the noise

term, such that:

$$p(\text{vec}(\mathbf{\Phi})) \sim \mathcal{N}\left(\text{vec}(\mathbf{0}); \mathbf{I}_{3K} \otimes \mathbf{I}_N\right), \tag{4}$$

$$p(\text{vec}(\mathbf{N})) \sim \mathcal{N}\left(\text{vec}(\mathbf{0}); \sigma^2\mathbf{I}_{2T} \otimes \mathbf{I}_N\right), \tag{5}$$

where $\text{vec}(\cdot)$ indicates the matrix vectorization operator.

Since the linear combination of independent Gaussian random variables is still Gaussian, $\mathbf{P}$ (which linearly combines $\mathbf{\Phi}$ and $\mathbf{N}$) will be normally distributed. Concretely, using simple manipulations on matrix variable normal distributions [27], it can be shown that:

$$p(\text{vec}(\mathbf{P})) \sim \mathcal{N}(\text{vec}(\mathbf{0}); (\mathbf{A}\mathbf{A}^\top + \sigma^2\mathbf{I}_{2T}) \otimes \mathbf{I}_N), \tag{6}$$

$$p(\text{vec}(\mathbf{P}|\mathbf{\Phi})) \sim \mathcal{N}(\text{vec}(\mathbf{A}\mathbf{\Phi}); \sigma^2\mathbf{I}_{2T} \otimes \mathbf{I}_N). \tag{7}$$

These distributions will be used in section 5 within a MLE framework to estimate pose, shape, and noise parameters.

4.2 Probabilistic Correlation Point Trajectory Model (PCPTA)

In the previous model, we have considered the data samples and noise to be independent and identically distributed. However, many real-world deformations, such as the motion of the human face or full body is made of subsets of points that follow similar deformation patterns.

In order to exploit the similarities between object points under non-rigid motion, we now show how to integrate these priors into our formulation. To this end, let us consider a symmetric matrix $\mathbf{C} \in \mathbb{R}^{N\times N}$, that encodes the correlations between the N instances. Note that this correlation is action-specific, since a similar deformation between points is exploited. If we consider the same spatial correlation matrix for both data and noise, we can inject this information via a covariance matrix as:

$$p(\text{vec}(\mathbf{\Phi})) \sim \mathcal{N}\left(\text{vec}(\mathbf{0}); \mathbf{I}_{3K} \otimes \mathbf{C}^{-1}\right), \tag{8}$$

$$p(\text{vec}(\mathbf{N})) \sim \mathcal{N}\left(\text{vec}(\mathbf{0}); \sigma^2\mathbf{I}_{2T} \otimes \mathbf{C}^{-1}\right), \tag{9}$$

where the only difference compared to the previous model is just in the matrix $\mathbf{C}$. Following the analysis for PPTA, in PCPTA we have:

$$p(\text{vec}(\mathbf{P})) \sim \mathcal{N}(\text{vec}(\mathbf{0}); (\mathbf{A}\mathbf{A}^\top + \sigma^2\mathbf{I}_{2T}) \otimes \mathbf{C}^{-1}), \tag{10}$$

$$p(\text{vec}(\mathbf{P}|\mathbf{\Phi})) \sim \mathcal{N}(\text{vec}(\mathbf{A}\mathbf{\Phi}); \sigma^2\mathbf{I}_{2T} \otimes \mathbf{C}^{-1}). \tag{11}$$

The correlation matrix $\mathbf{C}$ could be learned from deformable training data, or considering physical relations between points. In this work, we present a strategy to recover it using time-varying 3D shapes. In Fig. 2, we show the graphical models of our methods. It is worth noting that PCPTA becomes PPTA when $\mathbf{C} = \mathbf{I}_N$, i.e., when the data does not render similarities between object points.

4.3 Spatial Correlation Priors

In this paper, we learn the correlation matrix $\mathbf{C}$ between object points from training data. In order to cope with noisy data, we follow [32] and formulate the problem as:

$$\underset{\mathbf{C},\mathbf{E}}{\arg\min} \|\mathbf{C}\|_* + \lambda\|\mathbf{E}\|_1$$

$$\text{subject to} \quad \mathbf{X} = \mathbf{X}\mathbf{C} + \mathbf{E}$$

where $\mathbf{X}$ contains the time-varying 3D positions of N points obtained from training data and $\mathbf{E}$ represents a residual noise. λ is a positive weight. $\|\cdot\|_*$ indicates the nuclear norm and $\|\cdot\|_1$ the convex approximation to sparse error. This problem can be efficiently solved by the augmented Lagrange multiplier method [11]. Note that we do not use the measurement matrix $\mathbf{W}$ to compute the correlation priors, since this matrix includes a combination of rigid and non-rigid motions, and spatial deformation priors only encode the non-rigid component.

Once $\mathbf{C}$ is learned, we normalize it and enforce unit entries at the diagonal by:

$$\mathbf{C} = \mathbf{C} \odot (\mathbf{1}_N \mathbf{1}_N^\top - \mathbf{I}_N) + \mathbf{I}_N \ , \tag{12}$$

where $\odot$ represents the Hadamard product and $\mathbf{1}_N$ is a vector of ones. Even though the estimated matrix can be directly used to encode the spatial correlations, weak dependencies due to ghost relations could still introduce artifacts. With the purpose of increasing robustness, we set to zero the positions of the corresponding point indexes that are smaller than a threshold (0.6 in our experiments). In Fig. 2 we show an example of this matrix.

5 Learning Pose and Shape

In this section, we describe the general EM-based factorization approach to estimate model parameters with the PCPTA model. Recall that to consider the PPTA model, we simply have to set $\mathbf{C} \equiv \mathbf{I}_N$. After the factorization, we perform a final metric update step.

5.1 Parameter Estimation

Assuming $\mathbf{C}$ to be known, we have to learn the model parameters $\mathbf{A}$ and σ^2 by using an EM algorithm, where the maximum likelihood of the trajectory observations $\mathbf{P}$ is estimated by iterating between E- and M-steps.

***E*-step.** In the E-step, we estimate the conditional distribution of the latent variables $\mathbf{\Phi}$, given the observations $\mathbf{P}$ and the current model parameter estimate. Following [41], we apply the Bayes' rule with Eqs. (8)-(11), and some properties of matrix variate normal distributions [27], it can be shown this distribution is again Gaussian:

$$p(\text{vec}(\mathbf{\Phi}|\mathbf{P}, \mathbf{A}, \sigma^2)) \sim \mathcal{N}(\text{vec}(\mathbf{\Upsilon}_{\mathbf{\Phi}}); \mathbf{\Sigma}_{\mathbf{\Phi}} \otimes \mathbf{C}^{-1}), \tag{13}$$

with:

$$\mathbf{\Upsilon}_{\mathbf{\Phi}} = (\mathbf{A}^\top \mathbf{A} + \sigma^2 \mathbf{I}_{3K})^{-1} \mathbf{A}^\top \mathbf{P}, \tag{14}$$

$$\mathbf{\Sigma}_{\mathbf{\Phi}} = \sigma^2 N (\mathbf{A}^\top \mathbf{A} + \sigma^2 \mathbf{I}_{3K})^{-1}. \tag{15}$$

The expectations over the latent variables can then be defined as $\mathbb{E}[\mathbf{\Phi}] \equiv \mathbf{\Upsilon}_{\mathbf{\Phi}}$ and $\mathbb{E}[\mathbf{\Phi C \Phi}^\top] \equiv \mathbf{\Sigma}_{\mathbf{\Phi}} + \mathbf{\Upsilon}_{\mathbf{\Phi}} \mathbf{C} \mathbf{\Upsilon}_{\mathbf{\Phi}}^\top$.

***M*-step.** In the M-step, we update the model parameters $\mathbf{A}$ and σ^2 by maximizing the following expected log-likelihood[1]:

$$\begin{aligned} \mathcal{L} =& \mathbb{E}\Big[\ln p\big(\mathbf{P}, \mathbf{\Phi}\big)\Big] = -\frac{1}{2\sigma^2}\Big[\text{tr}\big(\mathbf{A}^\top \mathbf{A} \mathbb{E}[\mathbf{\Phi C \Phi}^\top]\big) \\ & -2\ \text{tr}\big(\mathbf{P C} \mathbb{E}[\mathbf{\Phi}]^\top \mathbf{A}^\top\big) + \text{tr}\big(\mathbf{P C P}^\top\big)\Big] - NT\ \ln \sigma^2. \end{aligned} \tag{16}$$

1. The joint distribution on $\mathbf{P}$ and $\mathbf{\Phi}$ is defined as $p(\mathbf{P}, \mathbf{\Phi}) = p(\mathbf{\Phi}) p(\mathbf{P}|\mathbf{\Phi})$.

Independently solving for each of the variables yields the following update rules, at iteration $j+1$:

$$\mathbf{A}_{j+1} \leftarrow \mathbf{D}\mathbf{A}_j \left(\sigma^2 \mathbf{I}_{3K} + \left(\mathbf{A}_j^\top \mathbf{A}_j + \sigma^2 \mathbf{I}_{3K}\right)^{-1} \mathbf{A}_j^\top \mathbf{D}\mathbf{A}_j\right)^{-1}$$
$$\sigma_{j+1}^2 \leftarrow \frac{1}{2T}\text{tr}\left(\mathbf{D} - \mathbf{D}\mathbf{A}_j \left(\mathbf{A}_j^\top \mathbf{A}_j + \sigma^2 \mathbf{I}_{3K}\right)^{-1} \mathbf{A}_{j+1}^\top\right),$$

where the matrix $\mathbf{D}$ corresponds to the covariance matrix of the centered observations, and it is defined as:

$$\mathbf{D} = N^{-1}\mathbf{P}\mathbf{C}\mathbf{P}^\top . \tag{17}$$

5.2 Metric Upgrade

The parameters $\hat{\mathbf{A}} = \mathbf{A}_{j+1}$ and $\hat{\mathbf{\Phi}} = \mathbb{E}[\mathbf{\Phi}]$ estimated using EM factorization are not unique, i.e., we could consider any invertible matrix $\mathbf{Q} \in \mathbb{R}^{3K\times 3K}$ such that $\mathbf{A}\mathbf{\Phi} = \hat{\mathbf{A}}\mathbf{Q}\mathbf{Q}^{-1}\hat{\mathbf{\Phi}}$, providing also valid factorizations. In order to find the rectification matrix $\mathbf{Q}$ that guarantees a metric structure on $\mathbf{S}$ and orthonormality on the rotation matrices that form $\mathbf{A} \equiv \mathbf{A}(\mathbf{R})$, we resort to the Euclidean upgrade strategy proposed in [9], [25]. For completeness, we next review this step on our algorithm.

In practice, it is not necessary to recover the full matrix $\mathbf{Q}$, since the first column triplet of the matrix $\mathbf{Q}$, which we denote as $\mathbf{Q}^* \in \mathbb{R}^{3K\times 3}$, provides enough constraints to retrieve the camera parameters (this is possible because the matrix $\mathbf{W}$ of trajectories is predefined). $\mathbf{Q}^*$ is obtained by performing a simple constrained non-linear minimization routine. Then the rotation matrices can be computed by applying:

$$\hat{\mathbf{A}}\mathbf{Q}^* = \begin{bmatrix} w_1^1\mathbf{R}^1 \\ \vdots \\ w_1^T\mathbf{R}^T \end{bmatrix}. \tag{18}$$

After estimating the rotation matrices $\mathbf{R}^t$ with $t \in \{1,\ldots,T\}$, the first factor on the factorization can be obtained as $\mathbf{A} = \mathbf{R}\mathbf{W}$, and then the time-varying 3D shape can be computed by solving an over-constrained linear system on $\mathbf{\Phi}$, such that:

$$\mathbf{S} = \mathbf{W}\mathbf{A}^\dagger\mathbf{P}, \tag{19}$$

where $(\cdot)^\dagger$ represents the pseudo-inverse operator.

Note that $\mathbf{Q}$ could also be computed by trace-norm minimization [16], [40] assuming the rank of the subspace a priori. However, we discarded this alternative as it is more computationally demanding.

5.3 Initialization

In order to maximize the expected log-likelihood in Sect. 5.1, we propose a simple parameter initialization.

The camera matrix $\mathbf{A} = \mathbf{R}\mathbf{W}$ is initialized using standard factorization on the 2D point trajectories $\mathbf{P} = \mathbf{U}_\mathcal{P}\mathbf{\Sigma}_\mathcal{P}\mathbf{V}_\mathcal{P}^\top$, i.e., $\mathbf{R}\mathbf{W} = \mathbf{U}_\mathcal{P}\mathbf{\Sigma}_\mathcal{P}^{\frac{1}{2}}$ and considering the $3K$ largest singular values in $\mathbf{\Sigma}_\mathcal{P}$. Again, this estimation is not unique and a rectification matrix $\mathbf{Q}$ is required to enforce the orthonormality constraints of $\mathbf{R}$. To estimate $\mathbf{Q}$, in this case, we automatically increase the rank value K on the factorization until no additional improvement in the average camera orthonormality is achieved, i.e., we select the factorization that minimizes $\frac{1}{T}\sum_{t=1}^{T}\|\mathbf{R}^t\mathbf{R}^{t^\top} - \mathbf{I}_2\|_\mathcal{F}^2$, where $\mathcal{F}$ denotes the Frobenius norm. Once the camera matrices $\mathbf{R}^t$ are estimated for $t = \{1,\ldots,T\}$, we

complete the block diagonal matrix $\mathbf{R}$. Since the matrix with the trajectory basis $\mathbf{W}$ is predefined, we can then easily obtain the initialization of $\mathbf{A}$.

Regarding the noise variance, we initially set $\sigma^2 = 10^{-6}$, and hence the latent variables $\mathbf{\Phi}$ can be directly initialized by considering the distribution on Eq. (13).

5.4 Missing Observations

Unlike other techniques for sparse [9], [16] and dense reconstructions [21], our approach can handle missing tracks due to occlusion or outliers. As done in [26], we initially set the missing observations $\hat{\mathbf{u}}_i^t$ to the 2D value predicted by the low-rank trajectory model. These initial predictions are then used to compute $\mathbf{R}$ and $\mathbf{\Phi}$ and are further refined after these parameters have been estimated:

$$\hat{\mathbf{u}}_i^t = \mathbf{R}^t(\mathbf{I}_3 \otimes (\mathbf{w}^t)^\top)\boldsymbol{\phi}_i + \mathbf{t}^t. \tag{20}$$

5.5 Computational Cost

One of the main strengths of the approaches we propose is its efficiency, which allows performing dense estimations with a minimal computational load, since the complexity only linearly depends on the number of points. Our formulation broadly consists of two main stages: the learning of the model parameters using EM, and the metric upgrade.

For the EM phase, we just need to invert $3K$-order matrices and perform several matrix multiplications. Since the rank K of the trajectory subspace is typically smaller than the number of frames T of the sequence, it is the latter that bounds the $\mathcal{O}(T^2KJ)$ complexity, where J is the number of EM iterations until convergence. However, our approach still depends on linearly of the number of points N, due to that the complexity of computing $\mathbf{D}$ is $\mathcal{O}(TN)$. Additionally, this complexity can be further reduced in practice, because matrices like $\mathbf{R}$, $\mathbf{W}$ or $\sigma^2\mathbf{I}_{3K}$, and even $\mathbf{C}$, are highly sparse.

Regarding the metric upgrade stage, the complexity of the non-linear procedure to obtain the rectification matrix is dominated by $\mathcal{O}(K^3)$. Hence the overall computational cost does not strongly depend on the number of points N, being the reason why we can efficiently handle dense observations.

6 Experimental Results

We now present results for a large variety of situations that demonstrate the versatility of our approach. We consider both articulated bodies and dense surfaces, single and multiple objects, mild and more severe deformations. Qualitative and quantitative results will be presented.

For quantitative evaluation, we will follow the metrics already used in [16], [24], and will report the mean rotation error e_R and normalized mean 3D error e_S, defined as:

$$e_R = \frac{1}{T}\sum_{t=1}^{T} \|\bar{\mathbf{R}}^t - \mathbf{R}^t\|_{\mathcal{F}} \ ,$$

where $\mathbf{R}^t$ is the estimated rotation matrix at frame t and $\bar{\mathbf{R}}^t$ is the corresponding ground truth rotation. e_S is computed as:

$$e_S = \frac{1}{\sigma TN}\sum_{t=1}^{T}\sum_{n=1}^{N} e_n^t, \quad \sigma = \frac{1}{3T}\sum_{t=1}^{T}(\sigma_x^t + \sigma_y^t + \sigma_z^t),$$

where e_n^t is the 3D reconstruction error for the n-th point at frame t. σ_x^t, σ_y^t and σ_z^t indicate the standard deviations at frame t of the x-, y- and z-coordinates of the original shape. We provide e_S and e_R whenever ground truth 3D data or rotation is available, respectively. Please, see videos in the supplemental material.

Met. / Data	EM-PPCA [43]		MP [34]		PTA [9]		CSF [24]		KSTA [25]		BMM [16]		PPTA (Ours)	
	e_R	$e_S(K)$	e_R	$e_S(K)$	e_R	$e_S(K)$	e_R	$e_S(K)$	e_R	$e_S(K)$	e_R	$e_S(K)$	e_R	$e_S(K)$
Noise-free observations														
Drink	0.186	0.261(7)	0.330	0.357(12)	0.006	0.025(13)	0.006	0.022(6)	0.006	0.020(12)	0.007	0.027(12)	0.006	**0.011(30)**
Stretch	0.749	0.458(7)	0.832	0.900(8)	0.055	0.109(12)	0.049	0.071(8)	0.049	**0.064(11)**	0.068	0.103(11)	0.058	0.084(11)
Yoga	0.688	0.445(8)	0.854	0.786(2)	0.106	0.163(11)	0.102	0.147(7)	0.102	0.148(7)	0.088	**0.115(10)**	0.106	0.158(11)
Pick-up	0.417	0.423(14)	0.249	0.429(5)	0.155	0.237(12)	0.155	0.230(6)	0.155	0.233(6)	0.121	**0.173(12)**	0.154	0.235(12)
Dance	–	0.339(4)	–	0.271(5)	–	0.296(5)	–	0.271(2)	–	0.249(4)	–	**0.188(10)**	–	0.229(4)
Average error:		0.385		0.549		0.166		0.148		0.143		**0.121**		0.143
Noisy observations														
Drink	0.231	0.250(7)	0.329	0.517(12)	0.043	0.045(13)	0.043	0.044(6)	0.043	0.042(12)	0.044	0.056(12)	0.042	**0.038(30)**
Stretch	0.819	0.886(7)	0.872	0.975(8)	0.091	0.144(12)	0.091	0.121(8)	0.091	0.166(11)	0.098	0.183(11)	0.091	**0.123(11)**
Yoga	0.700	0.507(8)	0.858	0.791(2)	0.124	0.174(11)	0.125	0.168(7)	0.125	**0.172(7)**	0.136	0.195(10)	0.124	**0.174(11)**
Pick-up	0.499	0.807(14)	0.250	0.407(5)	0.148	0.228(12)	0.148	0.224(6)	0.148	0.222(6)	0.141	**0.212(12)**	0.148	0.228(12)
Dance	–	0.336(4)	–	0.282(5)	–	0.299(5)	–	0.266(2)	–	0.248(4)	–	0.236(10)	–	**0.222(4)**
Average error:		0.557		0.594		0.178		0.165		0.170		0.176		**0.157**

TABLE 2: **Quantitative comparison on single full-body motion capture sequences from [9].** Rotation e_R and reconstruction e_S errors for competing techniques: EM-PPCA [43], MP [34], PTA [9], CSF [24], KSTA [25] and BMM [16]; and our PPTA approach without assuming spatial priors. For every method, we also include in parentheses the rank of the linear subspace that gave the lowest e_{3D} error. The symbol "$-$" indicates that ground truth data is not available.

Met. / Data	EM-PPCA [43]		MP [34]		PTA [9]		CSF [24]		KSTA [25]		BMM [16]		PPTA (Ours)	
	e_R	$e_S(K)$	e_R	$e_S(K)$	e_R	$e_S(K)$	e_R	$e_S(K)$	e_R	$e_S(K)$	e_R	$e_S(K)$	e_R	$e_S(K)$
Noise-free observations														
Jump	0.801	1.169(6)	0.378	0.430(2)	0.119	0.319(5)	0.039	**0.079(10)**	0.039	0.113(4)	0.045	0.160(11)	0.051	0.126(13)
Greet	0.274	0.325(6)	0.127	0.242(4)	0.094	0.166(3)	0.060	0.119(5)	0.060	**0.115(5)**	0.063	0.182(11)	0.059	0.134(7)
Chicken	0.062	0.101(3)	0.082	0.208(4)	0.043	0.151(6)	0.117	0.210(6)	0.117	0.202(9)	0.032	**0.083(12)**	0.041	0.143(8)
Meet	0.432	0.786(2)	0.281	0.392(14)	0.052	0.166(10)	0.403	1.049(4)	0.044	0.176(4)	0.042	0.249(12)	0.046	**0.145(12)**
Pull	0.623	0.459(11)	0.521	0.632(5)	0.271	0.358(8)	0.226	0.302(12)	0.226	0.297(6)	0.211	0.287(9)	0.191	**0.247(13)**
Average error:		0.568		0.381		0.232		0.352		0.181		0.192		**0.159**
Noisy observations														
Jump	0.814	1.159(6)	0.789	1.068(2)	0.123	0.286(5)	0.061	0.813(10)	0.061	0.345(4)	0.077	0.246(11)	0.059	**0.105(13)**
Greet	0.277	0.359(6)	0.119	0.247(4)	0.107	0.182(3)	0.074	0.149(5)	0.074	**0.128(5)**	0.086	0.199(11)	0.072	0.136(7)
Chicken	0.068	0.135(3)	0.096	0.228(4)	0.043	0.157(6)	0.119	0.211(6)	0.119	0.205(9)	0.045	0.147(12)	0.042	**0.143(8)**
Meet	0.433	0.826(2)	0.263	0.386(14)	0.054	0.187(10)	0.409	1.056(4)	0.065	0.196(4)	0.063	0.275(12)	0.065	**0.162(12)**
Pull	0.617	0.467(11)	0.588	0.674(5)	0.271	0.397(8)	0.234	0.312(12)	0.234	0.304(6)	0.242	0.312(9)	0.219	**0.283(13)**
Average error:		0.589		0.519		0.242		0.510		0.236		0.236		**0.166**

TABLE 3: **Quantitative comparison on multiple deforming and interacting full-body mocap sequences.** See caption on Table 2.

6.1 Sparse Human Body Reconstruction

Our approach and other NRSfM algorithms will be first evaluated on sparse data acquired with motion capture systems. We next list the datasets and for each of them, we indicate by (T/N) the number of frames and point tracks, respectively. Regarding full-body motion with a single object, we considered the following sequences: *Drink* (1,102/41), *Stretch* (370/41), *Yoga* (307/41), *Pick-up* (357/41) and *Dance* (264/75), all of them taken from [9]. For full-body motion with multiple objects, we use the following five sequences from the CMU motion-capture database with two persons interacting and performing different actions: *Jump* (432/82), people alternating jumping jacks; *Greet* (200/82), people walking and shaking hands; *Chicken* (1,536/82), persons performing the chicken dance; *Meet* (1,253/82), two persons meeting and sitting side by side, and finally, the *Pull* (430/82) sequence where one person pulls the other's elbow. For these experiments, we follow the same evaluation procedure as in state-of-the-art techniques [9], [24], [25], i.e., we synthesized a slow moving orthographic camera and projected 3D data using these rotations to get the image observations. The amount of camera rotation was 5 degrees per frame with the y-axis of the camera pointing towards the center of the scenario.

To make a fair comparison, we employ our PPTA approach without considering correlation

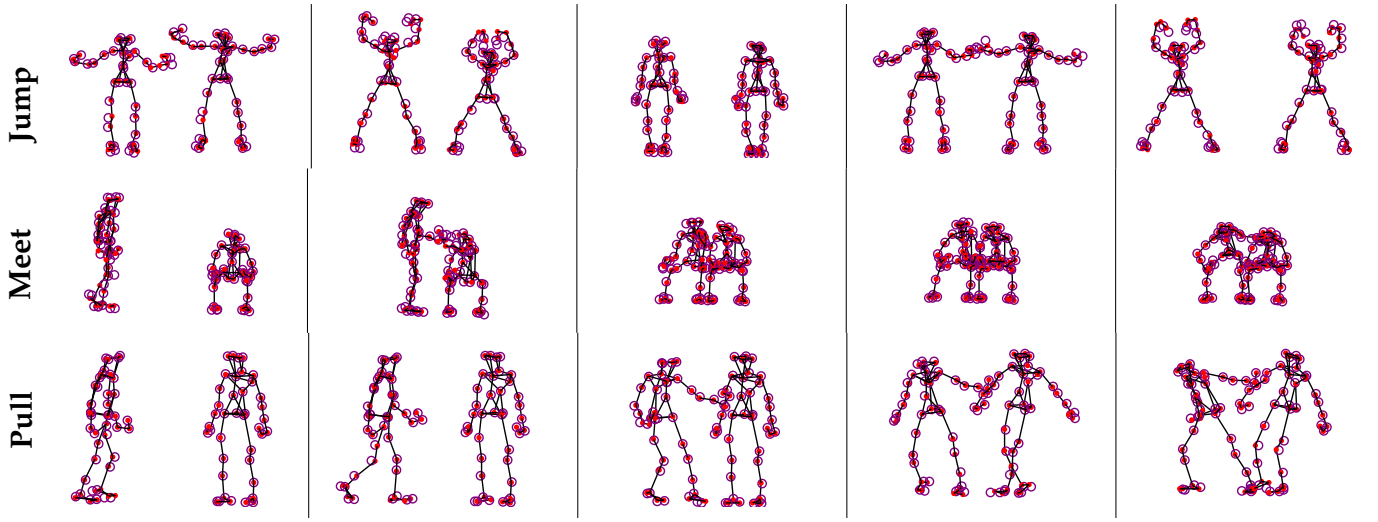

Fig. 3: **Motion capture sequences with multiple and interacting objects.** Sample reconstruction results on the *Jump*, *Meet* and *Pull* sequences. The red dots represent the 3D reconstruction obtained using the proposed approach. Note that they are very close to the ground truth, shown as purple circles. Best viewed in color.

priors, and compare it against the following state-of-the-art methods[2]: EM-PPCA [43], Metric Projections (MP) [34], the Point Trajectory Approach (PTA) [9], the Column Space Fitting (CSF) [24], the Kernel Shape Trajectory Approach (KSTA) [25] and the Block Matrix Method (BMM) proposed in [16]. For every method, we set the parameters according to the values reported in the original papers. As in all other low-rank techniques, our approach requires setting the rank K of the subspace. No other parameter or regularization weight needs to be tuned. Since the solution depends on the rank of the linear subspace, we have chosen, for every experiment and method, the basis rank that gave the lowest e_{3D} error. For our approaches, overall, we have observed that our solution is more accurate as long as the basis rank is increased, but without achieving large values of rank. On the other hand, when no improvement is achieved, we note that the solution is stable, and our 3D reconstructions do not deviate too much.

We then compared all methods in two situations: for noise-free observations, and when the 2D point tracks were artificially corrupted by zero-mean Gaussian noise with standard deviation $\sigma_{noise} = 0.01\rho$, with ρ being the maximum distance of an image point to the centroid of all the points. Tables 2 and 3 summarize the rotation and reconstruction errors for single and multiple full-body sequences, respectively. When reconstructing single objects, our approach obtains comparable results to the most accurate state-of-the-art methods, KSTA [25] and BMM [16], and consistently outperforms them under noisy observations. Our solutions are also comparable to those reported in [40] for noise-free observations. Unfortunately, the source code of this approach is not publicly available, and we cannot complete the comparison for noise observations. In any event, our approach is substantially more efficient, as it only requires estimating a $3K \times N$ matrix of trajectory coefficients. In contrast, [40] needs to recover a $T \times 3N$ matrix of mixed coefficients plus the shape basis matrix of size $3N \times 3N$ (to make a fair comparison, we discard the trajectory basis matrix of size $T \times T$, that could be pre-defined as in our case).

When dealing with several objects, our approach obtains, on average, more accurate solutions than competing techniques for both noise-free and noisy observations. In addition, this accuracy is obtained at a remarkable speed-up. See in Table 4 that our results are obtained between $3\times$ and

2. We also tried to compare against [14], but did not manage to feed the algorithm with all input data it requires.

Data	KSTA [25]	BMM [16]	PPTA	Speed-up
Drink	290	1,128	**129**	2.24/8.74
Stretch	75	398	**9**	8.33/44.22
Yoga	90	247	**8**	11.25/30.87
Pick-up	39	647	**12**	3.25/53.91
Dance	103	502	**3**	34.33/167.33
Jump	167	982	**14**	11.93/70.14
Greet	143	601	**4**	35.75/150.25
Chicken	1,145	5,363	**387**	2.95/13.85
Meet	854	11,894	**226**	3.78/52.63
Pull	282	825	**17**	16.59/48.53

TABLE 4: **Computation Times.** We compare the execution time (in seconds) of our approach with respect to KSTA [25] and BMM [16], the two most accurate state-of-the-art approaches. All methods were executed in plain Matlab. The right-most column shows the speed-up we obtain with respect to these methods.

Met. / Data	PPTA		PC2PTA		PCPTA	
	e_R	e_S	e_R	e_S	e_R	e_S
Drink	0.006	0.011	0.006	0.011	0.005	**0.010**
Stretch	0.058	0.084	0.079	0.108	0.047	**0.068**
Yoga	0.106	0.158	0.122	0.202	0.099	**0.155**
Pick-up	0.154	**0.235**	0.164	0.250	0.154	0.237
Dance	–	0.229	–	0.244	–	**0.212**
Jump	0.051	0.126	0.094	0.212	0.047	**0.119**
Greet	0.059	0.134	0.068	0.139	0.029	**0.131**
Chicken	0.041	0.143	0.058	0.158	0.040	**0.142**
Meet	0.046	0.145	0.053	0.165	0.042	**0.143**
Pull	0.191	0.247	0.204	0.263	0.184	**0.238**

TABLE 5: **Introducing spatial correlation priors.** 3D reconstruction and rotation errors of our PPTA formulation and the same results when deformation similarities are exploited (denoted as PC2PTA and PCPTA, respectively). We report results on the sequences of Tables 2 and 3, using the same rank. Note that the errors are reduced for those sequences that exhibit a certain degree of spatial motion similarity, especially when the correlation matrix codes these similarities properly. Again, the symbol "$-$" indicates that the ground truth is not available.

$167\times$ faster than previous approaches. Figure 3 shows the 3D reconstruction results we obtain on several frames for sequences with multiple objects. Recall that no a priori object segmentation is required for tackling these scenarios.

We have also used the motion-capture sequences to evaluate the effect of introducing a correlation matrix linking the motion of similar points (see Section 4.3), and employ then our PCPTA formulation. As a proof of concept, these correlation matrices have been estimated from a noisy version of the 3D ground truth. In practice they could be readily learned from training data if available. The 2D measurement matrices could also be used for this purpose, marginalizing the non-rigid component. With the purpose of validating our PCPTA formulation, we also include another baseline where a PCA-based generic correlation model is used in combination with our PCPTA approach, that we will denote as PC2PTA. In this case, we independently compute a correlation matrix for every sequence. This means that the only difference between the PCPTA and PC2PTA algorithms is how the correlation matrix is computed. The results when this information is incorporated are summarized in Table 5. Note that the reconstruction results are more accurate when the amount of deformation similarities is larger, such as in the *stretch* sequence, that exhibits repetitive motions. Furthermore, it can be observed that the way in which spatial similarities are encoded is also relevant, since a generic correlation model does not provide more accurate solutions than not exploiting similarities, as happens with our PCPTA-based solutions.

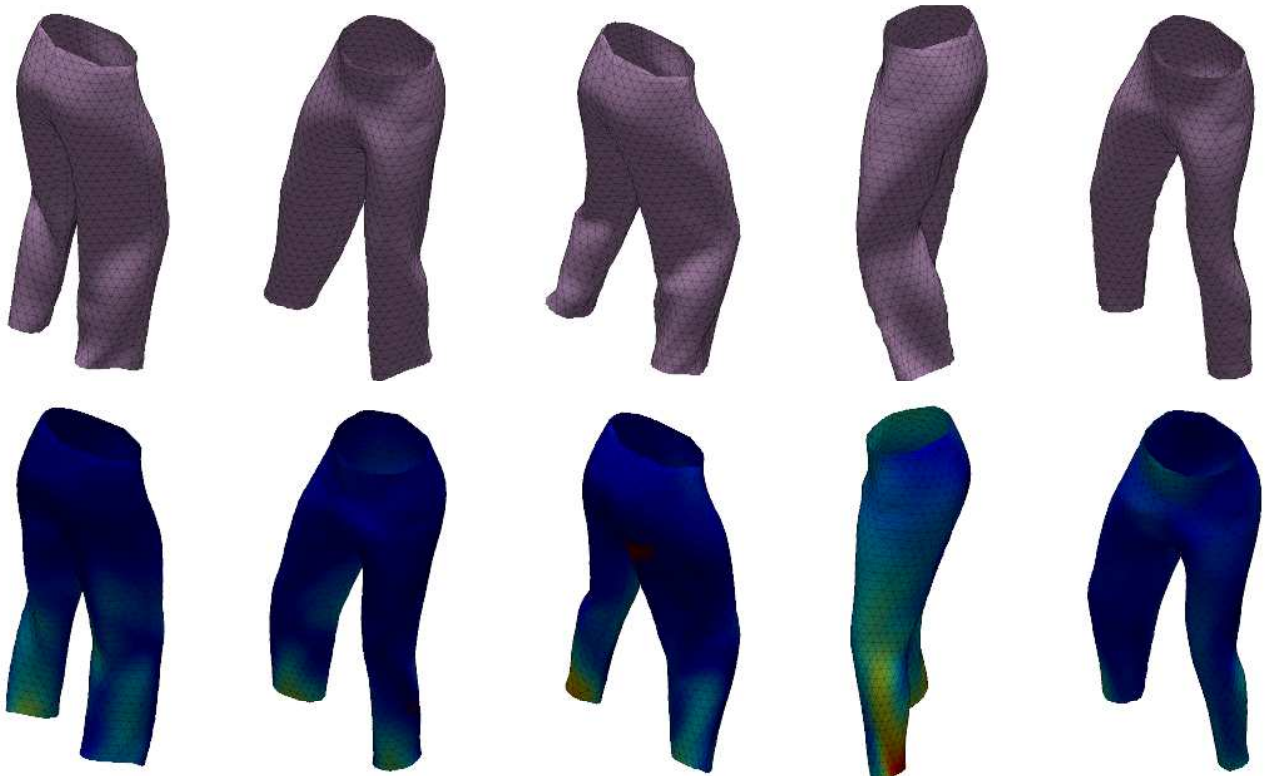

Fig. 4: **Pant sequence.** Sample frames of the *Pant* sequence. For each of them, we show the 3D ground truth shape (top) and color-coded 3D reconstruction (bottom), where reddish areas denote larger errors. Best viewed in color.

6.2 Dense Reconstruction

To highlight the computational efficiency of our approach while still being comparable and even more accurate than competing methods, we performed a series of experiments on four sequences with increasing levels of point density. We next describe each sequence and point to the qualitative results obtained by our approach:

- We first analyze the *Pant* sequence that was originally proposed in [46] and renders a deforming pants while jumping and rotating. It contains 291 frames, and 1,453 point tracks. A qualitative evaluation with respect to 3D ground truth is shown in Fig. 4.
- We next process the *Cloth* sequence taken also from [46], of a cloth being deformed when several coins fall on top of it. This sequence is made of 31 frames and 2,145 point tracks, and it is particularly challenging because the rapidly varying deformations produce strong warps. Our 3D reconstruction for a few sample frames is depicted in Fig. 5.
- In order to demonstrate our approach is also suited to encode highly complex shapes, we process the synthetic *Ogre* sequence from [13]. It shows the face of an Ogre while changing between five facial expressions. The difficulty of this sequence stems on the complex and local details of the shape. It contains 81 frames, and the density grows up to 19,985 point tracks. Our results are shown in Fig. 6.
- Finally, we process a very dense sequence of a human face, that we denote as the *Face* sequence, changing expression from a smile to anger. It has 99 frames, and 28,887 point tracks. In this case, a qualitative visualization of the results we obtain is shown in Fig. 7.

We next discuss the quantitative results. For the first three experiments (*Pant*, *Cloth*, and *Ogre*), we compare against the methods with better performance in the sparse sequences of the previous

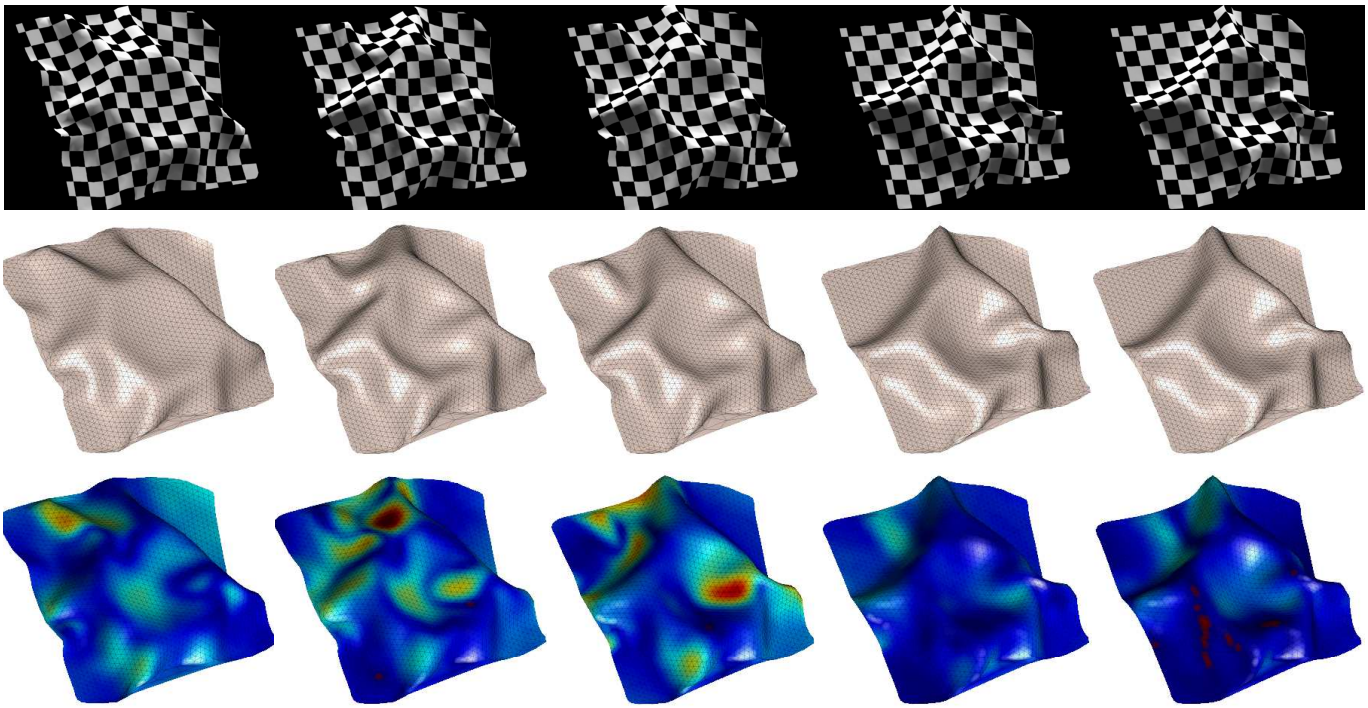

Fig. 5: **Cloth sequence. Top:** Sample frames of the sequence. **Middle:** 3D ground truth. **Bottom:** A color-coded 3D reconstruction where reddish areas denote larger errors. Best viewed in color.

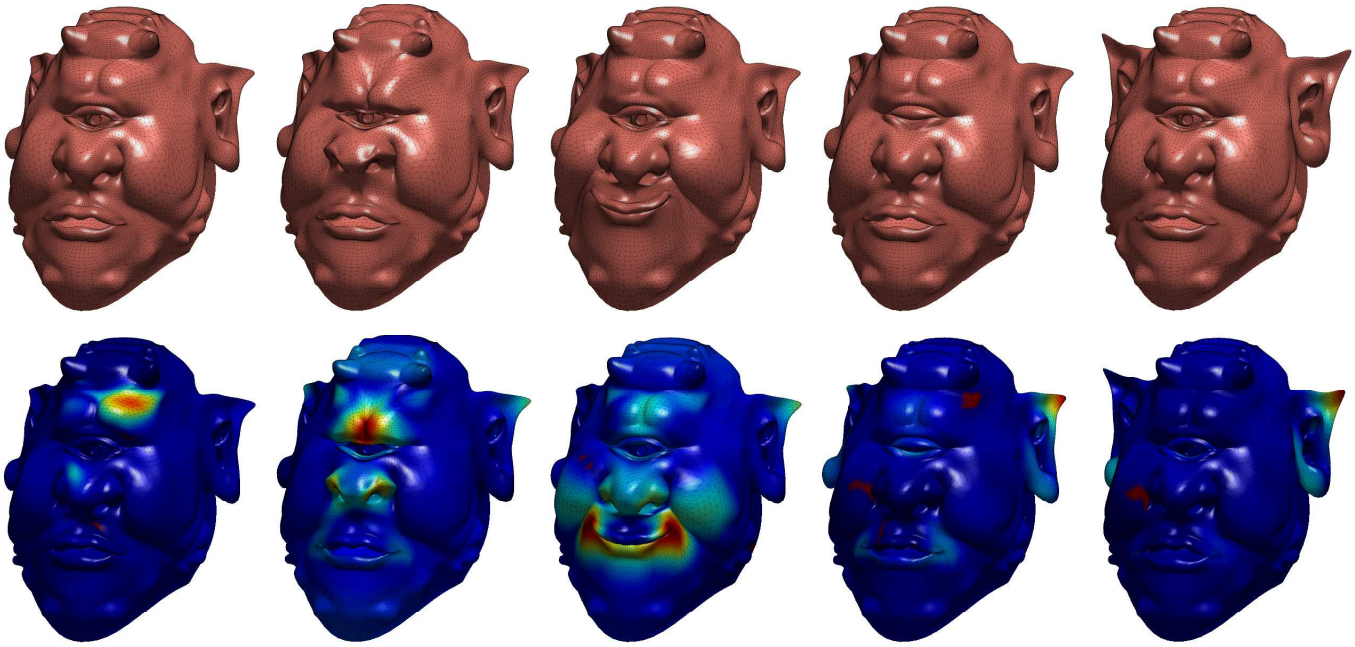

Fig. 6: **Ogre sequence.** Sample frames of the *Ogre* sequence. For each of them, we show the 3D ground truth shape (top) and color-coded 3D reconstruction (bottom), where reddish areas denote larger errors.

section, i.e., KSTA [25] and BMM [16]. In Table 6 we report the accuracy results in terms of rotation and 3D error, together with the computation times. Note that our approach yields similar results as the two competing algorithms (when they obtain a solution), but with a tremendous speed-up, that reaches several orders of magnitude when compared against BMM [16]. These results show

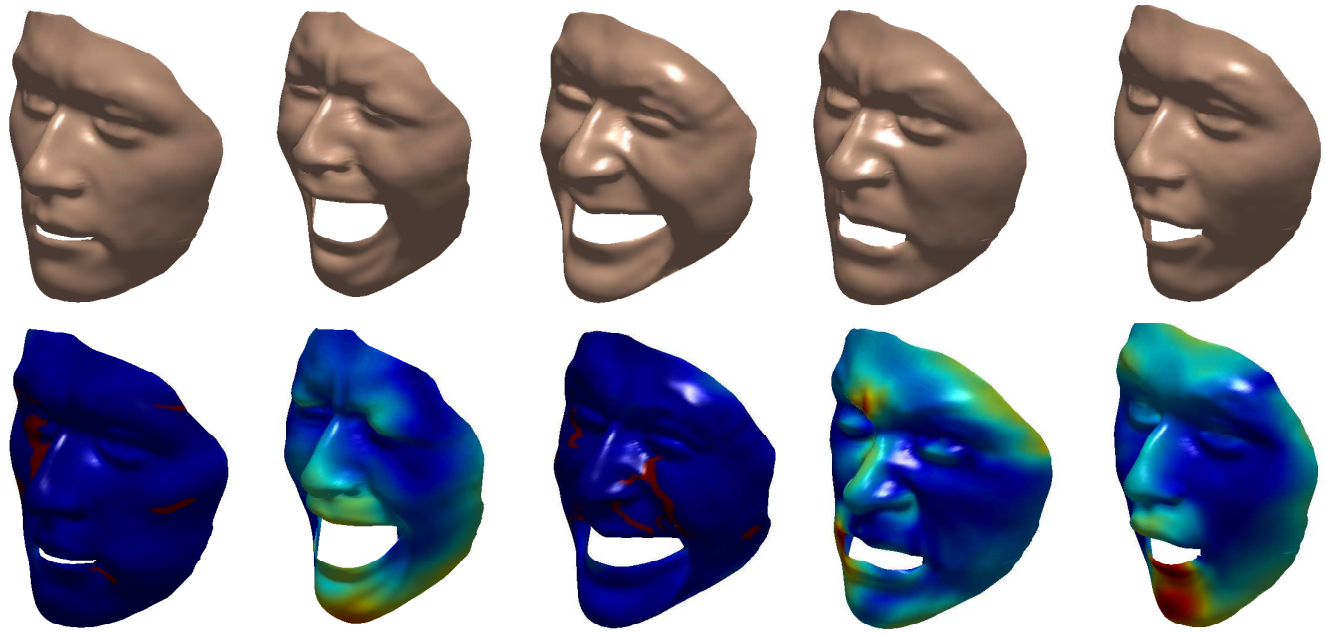

Fig. 7: **Face sequence.** Sample frames of the *Face* sequence. For each of them, we show the 3D ground truth shape (top) and color-coded 3D reconstruction (bottom), where reddish areas denote larger errors.

Data	KSTA [25]			BMM [16]			PPTA		
	e_R	$e_S(K)$	c_t	e_R	$e_S(K)$	c_t	e_R	$e_S(K)$	c_t
Pant	–	0.220(3)	1,258.7	–	**0.188(2)**	19,571.9	–	0.203(2)	**4.2**
Cloth	0.054	0.092(3)	20.1	0.042	**0.053(3)**	25,445.1	0.032	**0.057(2)**	**1.8**
Ogre	0.005	**0.011(4)**	597.3	¬	¬	¬	0.006	**0.013(8)**	**2.1**

TABLE 6: **Dense Sequences.** Rotation error e_R, reconstruction error e_S, and computation time c_t (in seconds) of our PPTA approach, KSTA [25] and BMM [16]. The symbol "¬" indicates the algorithm did not manage to process the sequence, and "−", that the ground truth is not available.

that our method exhibits the best trade-off between accuracy and computational budget, while it can be indistinctly used for sparse or dense data.

For a direct comparison against the dense variational approach of [21], we consider the *Face* sequence used in this paper and compute the mean 3D reconstruction error. Our PPTA yields an error of 3.07% (for $K = 15$), and we can process the whole sequence in 8.7 seconds, using non-optimized Matlab code on a standard laptop. The accuracy we obtain is considerably better than the rest of methods evaluated in [21]: MP [34] (5.13%)(6) and PTA [9] (4.50%)(4); than the sequential dense methods BA-FEM [3] (4.64%)(4) and EM-FEM [2] (4.53%)(4), and it is very close to the variational approach VNR [21] (2.60%)(9). Nevertheless, while the computational time of VNR [21] is not provided in this paper, the fact that this approach optimizes a shape matrix whose dimension is proportional to the number of points makes us believe that it is quite demanding. Indeed, this approach uses GPU-based computations. For this particular experiment, we were not able to provide results for KSTA [25] and BMM [16], as they could not handle the large dimensionality of the problem.

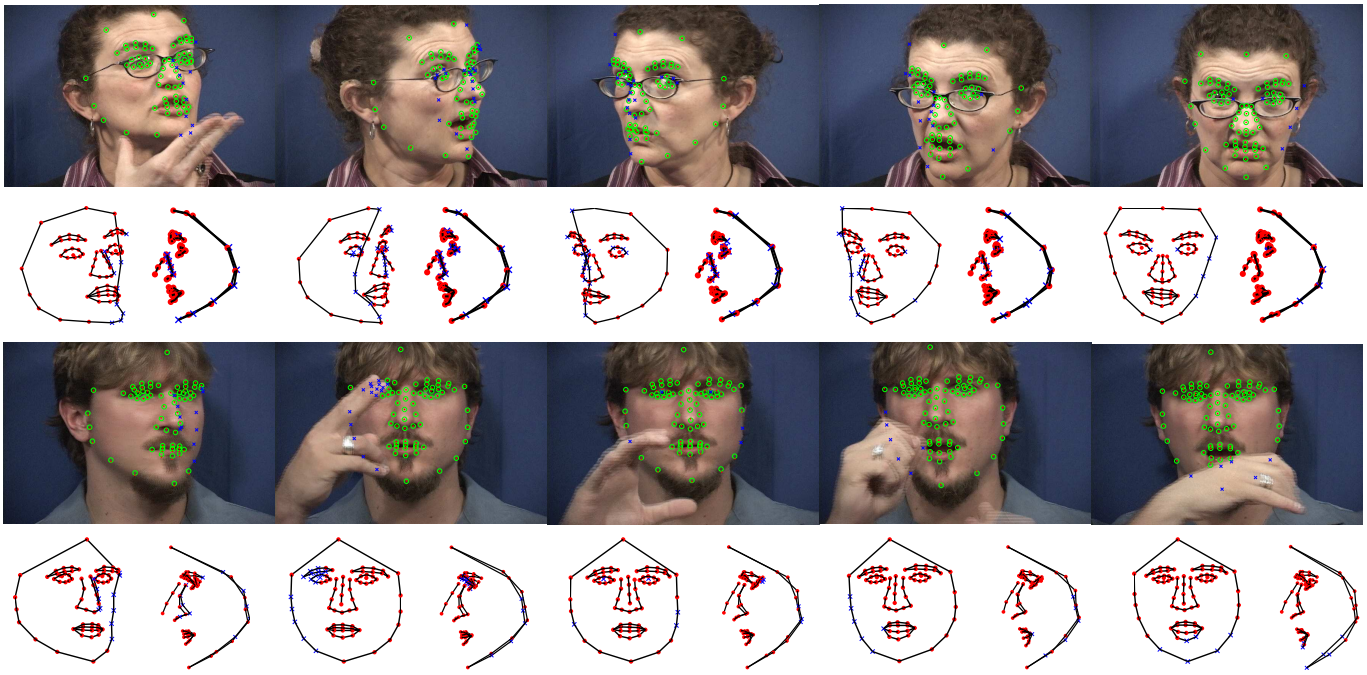

Fig. 8: **ASL video sequences.** In both cases, **Top:** Images with 2D tracking data in green circles, and reprojected 3D structure in red dots (visible points) and blue crosses (occluded points). **Bottom:** Camera and side-views of the reconstructed 3D face structure.

6.3 Real Videos

We have also evaluated our approach on real sequences, which despite not having ground truth, allow for a qualitative evaluation in different real-world conditions such as the presence of structured occlusions or noisy point tracks.

First, we have processed two *ASL sequences* of an American Sign Language (ASL), originally used in [24], and consisting of a person moving the head while talking and hand gesturing. The challenge of this sequence is to handle the partial occlusions of the face produced by the hands or by a lack of visibility due to face motion. The first sequence, of 115 frames and 77 landmarks, has 17.4% of missing observations. The second one, of 114 frames and also with 77 landmarks, has 11.5% of missing observations. Figure 8 shows our 3D reconstruction on a few sample frames for both sequences, when we set K=4. Note that our approach provides a correct estimation of all points, even under structured occlusions.

As a final experiment, we processed a real sequence of a beating *Heart*, consisting of 79 frames and 68,295 points. In this case, we obtain dense 2D trajectories from optical flow computed by [22] and used as input to our approach. In Fig. 9, we show some frames and our dense 3D reconstruction using $K = 2$. Note that even though the estimated 2D flow is quite noisy, the reconstructed 3D shape seems to be physically correct. This result, in conjunction with the efficiency of our approach (the whole sequence is processed in 3.84 seconds) opens new directions in which the flow and the 3D shape can be simultaneously estimated.

7 Conclusion

In this paper, we have addressed the NRSfM problem using an approach that incorporates low-rank point trajectory models into a probabilistic framework on matrix distributions, and that can also exploit correlation priors. This results in a technique that combines the advantages of local and global representations, allowing to reconstruct both articulated/sparse shapes and

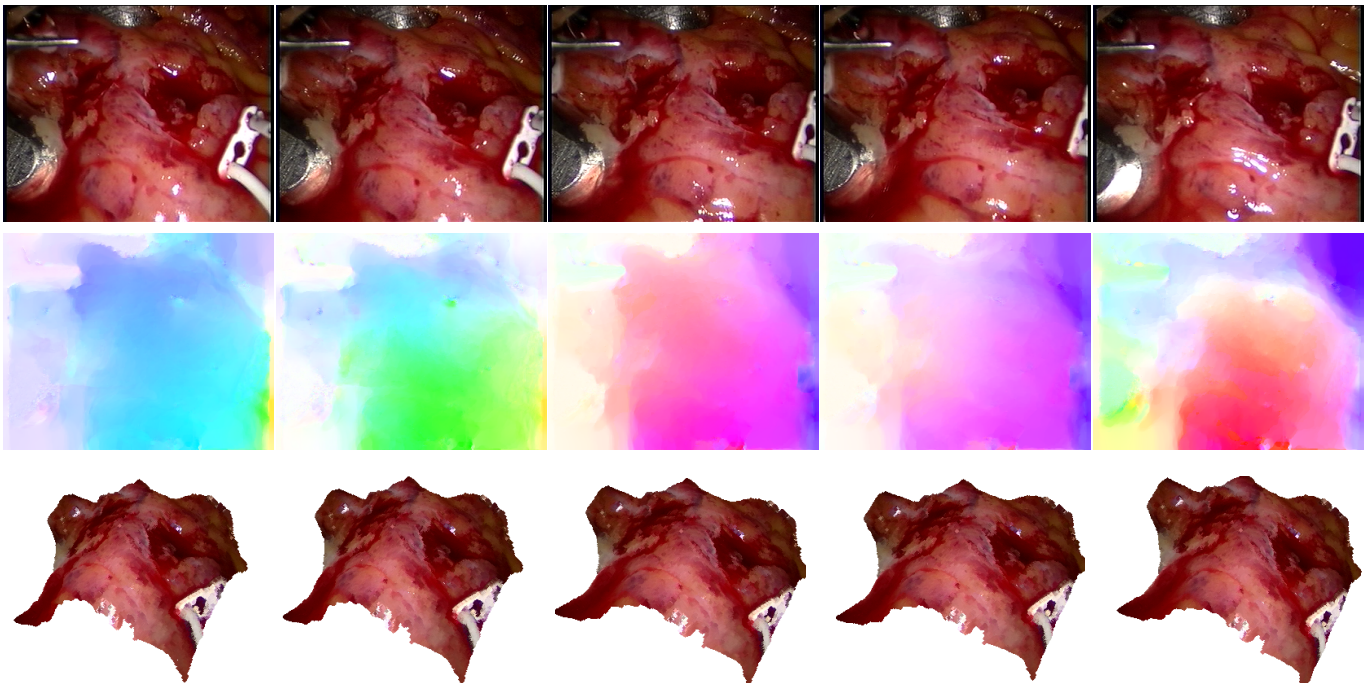

Fig. 9: **Heart sequence. Top:** Input images. **Middle:** Color coded optical flow. **Bottom:** Textured 3D reconstruction from a general view. Best viewed in color.

dense surface, works with and without training data and dealing with observations corrupted by noise and occlusions. Most importantly, all this is achieved using very low computational resources, handling sequences with per-pixel observations in a matter of a few seconds. All our claims have been extensively validated on both mocap and real videos showing improved performance to state-of-the-art techniques at much less cost. Our future work is oriented to integrate this approach into an optical flow framework, taking NRSfM a step forward and solving it for unknown point tracks.

ACKNOWLEDGMENTS

This work has been partially supported by the Spanish Ministry of Science and Innovation under projects RobInstruct TIN2014-58178-R and HuMoUR TIN2017-90086-R, and by a Google Faculty Award.

REFERENCES

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building Rome in a day. In *IEEE International Conference on Computer Vision*, pages 72–79, 2009.

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

[3] A. Agudo, J. M. M. Montiel, L. Agapito, and B. Calvo. Modal space: A physics-based model for sequential estimation of time-varying shape from monocular video. *Journal of Mathematical Imaging and Vision*, 57(1):75–98, 2017.

[4] A. Agudo, J. M. M. Montiel, B. Calvo, and F. Moreno-Noguer. Mode-shape interpretation: Re-thinking modal space for recovering deformable shapes. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1–8, 2016.

[5] A. Agudo and F. Moreno-Noguer. Learning shape, motion and elastic models in force space. In *IEEE International Conference on Computer Vision*, pages 756–764, 2015.

[6] A. Agudo and F. Moreno-Noguer. Recovering pose and 3D deformable shape from multi-instance image ensembles. In *Asian Conference on Computer Vision*, pages 271–307, 2016.

[7] A. Agudo and F. Moreno-Noguer. Combining local-physical and global-statistical models for sequential deformable shape from motion. *International Journal of Computer Vision*, 122(2):371–387, 2017.

[8] A. Agudo, F. Moreno-Noguer, B. Calvo, and J. M. M. Montiel. Real-time 3D reconstruction of non-rigid shapes from single moving camera. *Computer Vision and Image Understanding*, 153(12):37–54, 2016.

[9] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Non-rigid structure from motion in trajectory space. In *Neural Information Processing Systems*, pages 41–48, 2008.

[10] A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S. Olsen, and P. Sayd. Coarse-to-fine low-rank structure-from-motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[12] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 690–696, 2000.

[13] N. A. Carr, J. Hoberock, K. Crane, and J. C. Hart. Fast GPU ray tracing of dynamic meshes using geometry images. In *Graphics Interface*, pages 203–209, 2006.

[14] A. Chhatkuli, D. Pizarro, and A. Bartoli. Non-rigid shape-from-motion for isometric surfaces using infinitesimal planarity. In *British Machine Vision Conference*, 2014.

[15] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. SfM with MRFs: Discrete-continuous optimization for large-scale structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2841–2853, 2013.

[16] Y. Dai, H. Li, and M. He. A simple prior-free method for non-rigid structure from motion factorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2018–2025, 2012.

[17] A. Del Bue, X. Llado, and L. Agapito. Non-rigid metric shape and motion recovery from uncalibrated images using priors. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1191–1198, 2006.

[18] K. Fragkiadaki, M. Salas, P. Arbeláez, and J. Malik. Grouping-based low-rank trajectory completion and 3D reconstruction. In *Neural Information Processing Systems*, pages 55–63, 2014.

[19] J.M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.H. Jen, E. Dunn, B. Clipp, S. Lazabnik, and M. Pollefeys. Building rome on a cloudless day. In *European Conference on Computer Vision*, pages 368–381, 2010.

[20] Y. Gao and A. L. Yuille. Symmetric non-rigid structure from motion for category-specific object structure estimation. In *European Conference on Computer Vision*, pages 408–424, 2016.

[21] R. Garg, A. Roussos, and L. Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1272–1279, 2013.

[22] R. Garg, A. Roussos, and L. Agapito. A variational approach to video registration with subspace constraints. *International Journal of Computer Vision*, 104(3):286–314, 2013.

[23] V. Golyanik and D. Stricker. Dense batch non-rigid structure from motion in a second. In *IEEE Winter Conference on Applications of Computer Vision*, pages 254–263, 2017.

[24] P. F. U. Gotardo and A. M. Martinez. Computing smooth time-trajectories for camera and deformable shape in structure from motion with occlusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):2051–2065, 2011.

[25] P. F. U. Gotardo and A. M. Martinez. Kernel non-rigid structure from motion. In *IEEE International Conference on Computer Vision*, pages 802–809, 2011.

[26] P. F. U. Gotardo and A. M. Martinez. Non-rigid structure from motion with complementary rank-3 spaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3065–3072, 2011.

[27] A. K. Gupta and D. K. Nagar. *Matrix Variate Distributions*. Monographs and Surveys in Pure and Applied Mathematics, Chapman & Hall/CRC, 2000.

[28] O. Hamsici, P. F. U. Gotardo, and A. M. Martinez. Learning spatially-smooth mappings in non-rigid structure from motion. In *European Conference on Computer Vision*, pages 260–273, 2012.

[29] C. Kong and S. Lucey. Prior-less compressible structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4123–4131, 2016.

[30] M. Lee, J. Cho, C. H. Choi, and S. Oh. Procrustean normal distribution for non-rigid structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1280–1287, 2013.

[31] M. Lee, C. H. Choi, and S. Oh. A procrustean markov process for non-rigid structure recovery. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1550–1557, 2014.

[32] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013.

[33] F. Moreno-Noguer and J. M. Porta. Probabilistic simultaneous pose and non-rigid shape recovery. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1289–1296, 2011.

[34] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2898–2905, 2009.

[35] S. Parashar, D. Pizarro, and A. Bartoli. Isometric non-rigid shape-from-motion in linear time. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4679–4687, 2016.

[36] H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh. 3D reconstruction of a moving point from a series of 2D projections. In *European Conference on Computer Vision*, pages 158–171, 2010.

[37] S. Roweis. EM algorithms for PCA and SPCA. In *Neural Information Processing Systems*, pages 626–632, 1998.

[38] C. Russell, R. Yu, and L. Agapito. Video pop-up: Monocular 3D reconstruction of dynamic scenes. In *European Conference on Computer Vision*, pages 583–598, 2014.

[39] M. Salzmann, R. Urtasun, and P. Fua. Local deformation models for monocular 3D shape recovery. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[40] T. Simon, J. Valmadre, I. Matthews, and Y. Sheikh. Separable spatiotemporal priors for convex reconstruction of time-varying 3D point clouds. In *European Conference on Computer Vision*, pages 204–219, 2014.

[41] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analysers. *Neural Computation*, 11(2):443–482, 1999.

[42] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization approach. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[43] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from-motion: estimating shape and motion with hierarchical priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):878–892, 2008.

[44] J. Valmadre and S. Lucey. General trajectory prior for non-rigid reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1394–1401, 2012.

[45] A. Varol, M. Salzmann, E. Tola, and P. Fua. Template-free monocular reconstruction of deformable surfaces. In *IEEE International Conference on Computer Vision*, pages 1811–1818, 2009.

[46] R. White, K. Crane, and D. Forsyth. Capturing and animating occluded cloth. In *International Conference and Exhibition on Computer Graphics and Interactive Techniques*, 2007.

[47] J. Xiao, J. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion. *International Journal of Computer Vision*, 67(2):233–246, 2006.

[48] A. Zaheer, I. Akhter, M. H. Baig, S. Marzban, and S. Khan. Multiview structure from motion in trajectory space. In *IEEE International Conference on Computer Vision*, pages 2447–2453, 2011.

[49] Y. Zhu, D. Huang, F. De La Torre, and S. Lucey. Complex non-rigid motion 3D reconstruction by union of subspaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1542–1549, 2014.