J Math Imaging Vis (2017) 57:75–98
DOI 10.1007/s10851-016-0668-2

Modal Space: A Physics-Based Model for Sequential Estimation of Time-Varying Shape from Monocular Video

Antonio Agudo[1] · José M. Martínez Montiel[2] · Lourdes Agapito[3] · Begoña Calvo[2]

Received: 21 December 2015 / Accepted: 6 June 2016 / Published online: 22 June 2016

Abstract This paper describes two sequential methods for recovering the camera pose together with the 3D shape of highly deformable surfaces from a monocular video. The nonrigid 3D shape is modeled as a linear combination of mode shapes with time-varying weights that define the shape at each frame and are estimated on-the-fly. The low-rank constraint is combined with standard smoothness priors to optimize the model parameters over a sliding window of image frames. We propose to obtain a physics-based shape basis using the initial frames on the video to code the time-varying shape along the sequence, reducing the problem from trilinear to bilinear. To this end, the 3D shape is discretized by means of a soup of elastic triangular finite elements where we apply a force balance equation. This equation is solved using modal analysis via a simple eigenvalue problem to obtain a shape basis that encodes the modes of deformation. Even though this strategy can be applied in a wide variety of scenarios, when the observations are denser, the solution can become prohibitive in terms of computational load. We avoid this limitation by proposing two efficient coarse-to-fine approaches that allow us to easily deal with dense 3D surfaces. This results in a scalable solution that estimates a small number of parameters per frame and could potentially run in real time. We show results on both synthetic and real videos with ground truth 3D data, while robustly dealing with artifacts such as noise and missing data.

Keywords Sequential nonrigid structure from motion · Dense reconstruction · Modal analysis · Finite elements

1 Introduction

The combined inference of 3D scene structure and camera motion from monocular image sequences, or rigid Structure *f*rom Motion (S*f*M), is one of the most active areas of research in computer vision. In the last decade, S*f*M methods have made significant progress in simultaneously retrieving camera motion and 3D shape in real time for a sparse set of salient points [16,27,34] and even per-pixel dense reconstructions from video sequences acquired with a hand-held camera [36,37] or with a micro aerial vehicle [62]. While S*f*M is now considered to be a mature field, these methods cannot be applied to structures undergoing nonrigid deformations, where the problem remains a challenge. In these cases, recovering the 3D structure of a deformable object from a monocular image sequence is an ill-posed problem since many different 3D shapes can have the same image measurements, producing severe ambiguities.

Nonrigid structure *f*rom motion (NRS*f*M) addresses this limitation and methods from this field are now capable of estimating simultaneously accurate 3D reconstructions of nonrigid objects and camera motion from monocular

Part of this work was done while A. Agudo was at UCL.

✉ Antonio Agudo
aagudo@iri.upc.edu

José M. Martínez Montiel
josemari@unizar.es

Lourdes Agapito
l.agapito@cs.ucl.ac.uk

Begoña Calvo
bcalvo@unizar.es

[1] Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Barcelona 08028, Spain

[2] I3A-Universidad de Zaragoza, Mariano Esquillor s/n, Zaragoza 50018, Spain

[3] University College London, Gower Street, London WC1E 6BT, UK

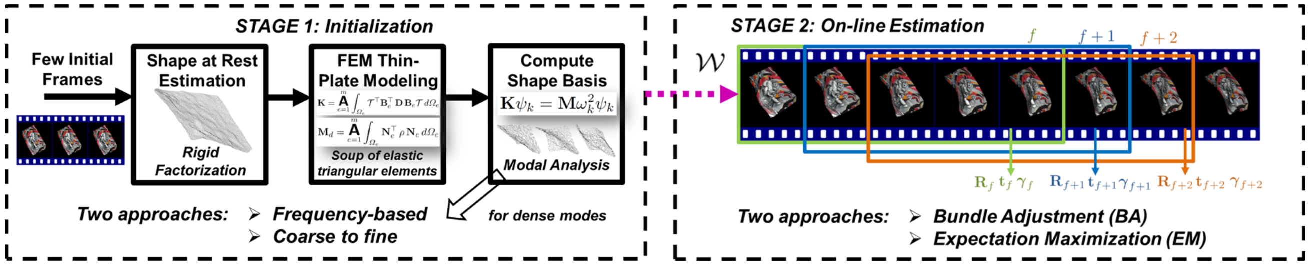

STAGE 1: Initialization
Few Initial Frames
Shape at Rest Estimation
Rigid Factorization
FEM Thin-Plate Modeling
$\mathrm{K} = \mathop{\mathbf{A}}_{e=1}^{m} \int_{\Omega_e} \mathcal{T}^\top \mathbf{B}_e^\top \mathbf{D} \mathbf{B}_e \mathcal{T} \, d\Omega_e$
$\mathrm{M}_d = \mathop{\mathbf{A}}_{e=1}^{m} \int_{\Omega_e} \mathbf{N}_e^\top \rho \mathbf{N}_e \, d\Omega_e$
Soup of elastic triangular elements
Compute Shape Basis
$\mathbf{K}\psi_k = \mathbf{M}\omega_k^2\psi_k$
Modal Analysis
Two approaches:
➢ Frequency-based
➢ Coarse to fine
for dense modes
STAGE 2: On-line Estimation
$\mathcal{W}$
f
$f+1$
$f+2$
$\mathbf{R}_f \, \mathbf{t}_f \, \gamma_f$
$\mathbf{R}_{f+1} \, \mathbf{t}_{f+1} \gamma_{f+1}$
$\mathbf{R}_{f+2} \, \mathbf{t}_{f+2} \, \gamma_{f+2}$
Two approaches:
➢ Bundle Adjustment (BA)
➢ Expectation Maximization (EM)

Fig. 1 On-line camera motion and nonrigid 3D shape from monocular image sequence. Our approaches consist of two stages: initialization and on-line estimation. *Left* In stage one, we use a few initial frames to estimate a shape at rest that we model by means of a soup of elastic triangular finite elements and to estimate the stiffness and mass matrices. Finally, we compute a shape basis by solving a simple eigenvalue problem using modal analysis. For dense cases, we present two strategies: a frequency-based and a coarse-to-fine approach. *Right* In stage two, we estimate camera motion and deformable 3D shape in a sequential manner over a sliding temporal window of frames. When a new frame $f+1$ is available, the temporal window shifts—*blue* window—and the new frame is processed. For this stage, we propose two approaches: a BA based using temporal smoothness priors and an EM-based using spatial ones (Color figure online)

video. The underlying principle behind most approaches is to model deformations using a low-rank shape basis [9,12,15,21,39,40,57] or a trajectory basis [8,41]. However, NRS*f*M methods remain behind their rigid counterparts when it comes to real-time performance. The main reason behind this is that they are typically limited to batch operations where all the frames in the sequence are processed at once, after the whole acquisition, preventing them from on-line and real-time performance applications. Only recently, have NRS*f*M methods been extended to sequential processing [3,40,51] using a small set of salient points. However, they remain slow [40,51] or do not scale to the use of a large number of points [3,40,51].

In this paper, we push monocular NRS*f*M forward toward real-time operation by proposing two fast sequential algorithms to simultaneously recover the nonrigid 3D shape and camera pose of strongly deforming surfaces under realistic real-world assumptions. Our approaches can deal with significant occlusions; they are suitable for modeling a wide variety of deformations from inextensible to highly extensible without the need for a pre-trained model, and they can be used to handle both sparse and dense data. To this end, we employ a linear combination of mode shapes—computed using continuum mechanics from a 3D rest shape estimation—to model the nonrigid 3D shape. Our on-line approaches work in two stages (see Fig. 1). In stage one, we estimate a shape at rest from the first few frames by means of a rigid factorization and then obtain a physics-based shape basis by solving a simple eigenvalue problem. In stage two, equipped with this low-rank shape constraint, the operation in a sequential manner over a sliding temporal window of frames is possible. To perform this stage, we propose two algorithms. The first is based on bundle adjustment (BA) with temporal smoothness priors, where the only parameters to estimate per-frame are the camera pose and the basis coefficients. The second uses a probabilistic model of the linear subspace with a Gaussian prior on each mode to encode the nonrigid 3D shape, i.e., spatial smoothness priors are assumed. Since the basis weights in the subspace are marginalized out, this method only optimizes per every frame the camera pose and a measurement noise by using expectation maximization (EM). In both cases, the number of parameters to estimate per frame is small, and our methods are fast and may potentially run in real time at frame rate. Although our shape basis is computed considering only the rest shape, it has proven experimentally to be able to code subsequent scene deformations without the need for 3D pre-learned data. It is worth pointing out that this shape basis is able to describe future deformations of the shape at rest (see experimental section). So for the same rest shape, our shape basis can code very different types of deformations.

Regarding the computation of mode shapes, while our methods can be directly applied from sparse to dense reconstructions, solving the stage one for dense cases could become prohibitive (see Fig. 1) in terms of computational load. To efficiently encode the deformations in these cases, we also propose two methods: (1) a frequency-based method where the dense mode shapes are computed using an interval of frequencies obtained in a low-resolution mesh, and (2) a coarse-to-fine approach where the dense shape at rest is down-sampled to a sparse mesh where modal analysis is applied at a low computational cost. These sparse 3D shapes are grown back to dense by exploiting the shape functions[1] to encode the geometry into the finite elements.

This paper combines and extends two conference publications [4,5]. Here, we integrate these preliminary publications into a comprehensive presentation of our sequential approach to retrieve 3D nonrigid shapes together with camera motion from monocular video. We also present a novel frequency-

[1] In this work, we denote shape functions as the approximation functions that interpolate the solution using the discrete values obtained at the mesh nodes.

based algorithm to compute stretching modes at low computational cost. In addition, we conduct new experiments to measure our algorithm's resilience to corrupted observations such as noise and missing data, and to compare the performance of its different variants. We also show more extensive comparisons of our results with other methods reported in the literature. The paper is organized as follows. In section two, we discuss the related work and describe the contributions of our work. Section three defines our novel physics-based deformation model. In section four, we formulate the problem of on-line recovery of camera motion and time-varying shape from a monocular sequence. This is followed in section five by a description of our sequential algorithms used to solve the previous problem using different priors. In section six, we show our experimental results and present a comparison with respect to state-of-the-art techniques. Our conclusions are set out in section seven. Finally, to keep the paper self-contained, we provide an appendix with mathematical details.

2 Related Work

NRS*f*M is an inherently ill-posed problem unless additional a priori knowledge of the shape and camera motion is considered. A seminal work by [12] proposed a low-rank shape constraint as an extension of the [54] factorization algorithm to the nonrigid case. Their key insight was to model time-varying shape as a linear combination of an unknown shape basis under orthography. Although this prior has proved to be a powerful constraint, it is insufficient to solve the inherent ambiguities in NRS*f*M. It was shown in [7] that the low-rank shape prior in addition to orthonormality constraints on camera motion are sufficient for noise-free observations. Recently, [15] imposed the low-rank shape constraint directly on the time-varying shape matrix via a trace norm minimization approach.

Most approaches have required the use of additional priors using different optimization schemes to include temporal smoothness [3,9,17,57,61], smooth-time trajectories [8,25], inextensibility constraints [61], rigid priors [6], and spatial smoothness [21,57]. BA has become a popular optimization tool for refining an initial rigid solution [54] optimizing the pose, shape basis, and coefficients while incorporating both motion and deformation priors [9,17]. A solution in [39] recovers motion matrices that lie on the correct motion manifold where the metric constraints are exactly satisfied. More recently, [21] combined a low-rank shape and local smoothness priors within a variational approach to produce per-pixel dense vivid reconstructions. A dual formulation using a trajectory basis was proposed in [8], where predefined discrete cosine trajectory bases are used to express the trajectory of each 3D point, considering each point as an independent object. Smoothness for the time-trajectory of each single point was assumed in [25].

Piecewise models have been proposed to encode more accurately strong local deformations. Piecewise planar [60], locally rigid [13,52], or quadratic [20] approaches rely on common features shared between patches to enforce global consistency. [45] proposed a formulation to automate the best division of the surface into local patches. Later, [3] proposed a finite element method (FEM) formulation providing support for a piecewise approach where the local elements are stretchable triangles. The FEM modeling assures deformable surface continuity without additional constraints.

The alternative strand of methods known as template based [33,38,46,47] rely on correspondences between the 2D points in the current image and a reference 3D shape which is assumed to be known in advance. In [33,47], the unknown shape was encoded as a linear combination of deformation modes learnt in advance from a relatively large set of training data. [38] introduced the Laplacian formalism in computer vision to regularize 3D meshes without requiring any training data. To avoid inherent ambiguities, additional shape constraints are required such as inextensibility [33,38]. Recently, [61] showed the exclusive use of these constraints, and it is sufficient to perform nonrigid reconstruction but only for isometric deformations, limiting the applicability of method.

On the other hand, spatial smoothness can be coded by means of probabilistic priors, although they have not been extensively used to recover nonrigid structure and camera motion. These priors allow the marginalization of hidden data that does not have to be explicitly computed, simplifying the optimization problem and avoiding over-fitting. Probabilistic Bayesian priors have been used in NRS*f*M to model deformation weights in a low-dimensional subspace based on principal component analysis (PCA) [56], the remaining model parameters being estimated by EM. These priors have also been used in template-based methods, where a Gaussian process latent variable model was employed to learn a prior over the deformations of local surface patches [47]. More recently, [28] proposed a Procrustean normal distribution to model shape deformation without additional constraints. The unknown shape is encoded as a linear combination of deformation modes learned on-the-fly for a relatively small deformation [56], or in advance from a relatively large set of training data [47]. However, the large deformations of real-world shapes may need larger values of rank, and hence the reconstruction becomes under-constrained for methods that learn deformation modes on the fly [56]. This ambiguity can be reduced using a predefined basis in terms of shape or trajectory which acts as a representative basis while reducing the number of parameters to be estimated. A shape basis could be estimated with a learning step over nonrigid 3D training data, but this information is not always available in advance in which case alternative methods to obtain a predefined basis are necessary. Probabilistic priors have also been used to model nodal forces in a physics-based deformation model combining an extended kalman filter (EKF) with the FEM to predict nonrigid displacements [2,3].

Mechanical priors have also been proposed to constrain the deformations of nonrigid objects. Early approaches used deformable superquadrics [32], balloons [14] or spring meshes [26], although these approaches were only valid to code relatively small deformations. The FEM was proposed to accurately represent specific materials [59,65] with known material properties. To tackle the high dimensionality of these physics-based models, a low-rank representation was proposed by applying modal analysis over a known structure discretized in 3D finite elements [35,42]. This method was then applied to image segmentation [43], medical imaging [35], and deformable 2D motion tracking [48,50]. [42] integrated the incremental updates of the basis coefficients over time using an EKF. This resulted in inevitable drift that eventually destroyed the ability to accurately reconstruct the nonrigid object. Recently, linear elasticity models have been proposed in [3,30] to code the extensible deformation of nonrigid objects, factorizing out the material properties into a normalized forces vector [3] or optimizing a stretching energy with respect to these [30].

Despite these advances, previous approaches to NRS*f*M typically remain batch and process all the frames in the sequence at once, after video capture. While sequential real-time S*f*M [27,34,36] solutions exist for rigid scenes, on-line estimation of nonrigid shape from a single camera based only on the measurements up to that moment remains a challenging problem. Sequential formulations have emerged only recently [3,40,51]. The first sequential NRS*f*M system was proposed in [40], based on BA over a sliding window with an implicit low-rank shape model. In [51], an incremental principal component analysis was proposed to model the shape basis. However, these approaches did not achieve real-time performance and were only demonstrated for a small number of feature points and small deformations. The first real-time on-line solution for NRS*f*M including feature tracking and outliers detection in a single process was proposed in [2,3], combining an EKF with FEM to estimate a small set of salient points which belong to a deformable object.

In this work, we present two sequential approaches over a sliding window to solve the NRS*f*M problem as the data arrive. In both cases, we use a predefined physics-based model to define a number of meaningful deformation mode shapes to obtain a low-rank shape model. To compute this shape basis, we only need an estimation of the shape at rest from a few initial frames of the image sequence. We discretize the scene into elastic triangular finite elements to code the behavior of the object with a thin-plate model and unknown material properties (see Fig. 1). We then propose two sequential algorithms to solve for the parameters of the model. The first combines the low-rank shape model with temporal smoothness priors and uses BA to estimate the parameters. The second combines the shape basis with probabilistic priors and solves the latent variable problem by EM. In both cases, the number of parameters to be estimated is small, resulting in a system with low computational cost that may potentially run in real time. Similarly to [3], our methods also use FEM to encode the nonrigid shape. However, while [3] had to compute a stiffness matrix at every step, in contrast, we here use FEM to solve a single modal analysis problem which provides a shape basis able to encode scene deformations. Our methods are valid for large displacements caused by strong deformations and can cope with both isometric and elastic warps. Note that most deformations in NRS*f*M, such as a smiling face or a waving flag, are produced with respect to a mean shape, and hence our method can model this type of deformations without updating the shape basis per frame. Exploring this way forward will be part of our future work.

3 Continuum Mechanics Deformation Model

A common way to model nonrigid 3D shapes in computer vision is to represent the 3D shape as a linear combination of shape basis [9,12,15,21,39,40,47,55]. The complexity of the problem can be reduced using dimensionality reduction techniques such as PCA [11,33] or modal analysis [42,48,50]. In this work, we only need an estimation of the shape at rest (from rigid structure-from-motion) to apply modal analysis and compute a shape basis, instead of using nonrigid 3D training data. We propose a method to compute the shape basis from a continuum mechanics physics-based model of the scene. Our approach is valid for both sparse and dense data and for encoding both bending and stretching deformations. We first review some dynamics concepts from continuum mechanics that will lead to our formulation of the estimation of the deformable shape basis.

3.1 Dynamics Overview

The dynamic response of a nonrigid solid under external actions has been widely studied in mechanical engineering [10,66]. Numerical methods are mandatory to approximately solve the partial differential equations modeling this behavior, with FEM being the most common approach.

A continuous object Ω can be discretized into a number of finite elements Ω_e (Fig. 2). Each element is defined by the 3D location of its nodes, $\bar{\mathbf{s}}_j$ for a generic node j. Hence the geometry of the undeformed solid is encoded as $\bar{\mathbf{S}} \in \mathbb{R}^{3\times p}$ where the matrix columns are the 3D coordinates defining the location of the p discretization nodal points:

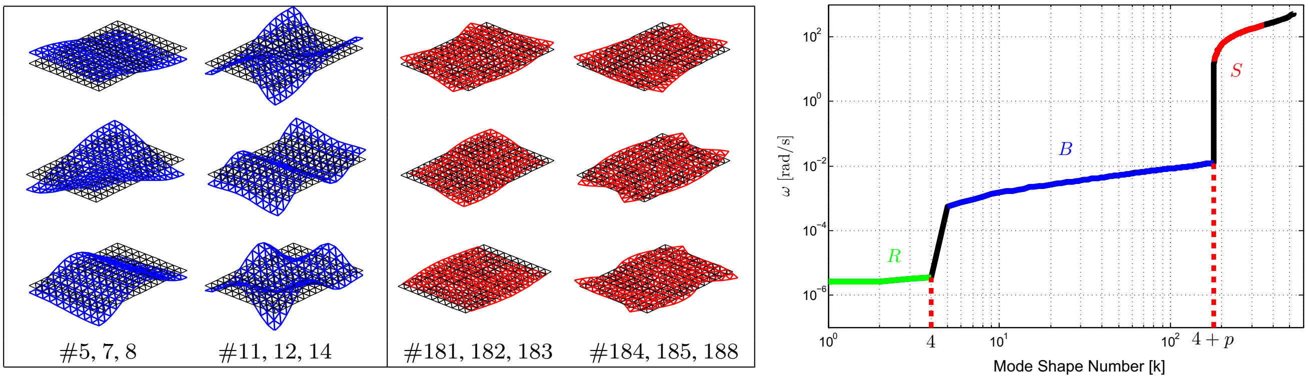


Fig. 2 *Left* Representation of some nonrigid mode shapes of a plate surface discretized into triangular elements with #176 nodes. We show the effect of adding the mode shape (*colored mesh*) to the shape at rest (*black mesh*) using an arbitrary weight. Note that the effect of subtracting can be obtained using the opposite weight. First and second column: bending mode shapes. Third and fourth column: stretching mode shapes. *Right* Eigen-frequencies ω_k for the previous *black mesh* in logarithmic scale. Rigid, bending, and stretching mode shape intervals in *green*, *blue* and *red line*, respectively. Best viewed in color (Color figure online)

$$\bar{\mathbf{S}} = \left[\bar{\mathbf{s}}_1 \dots \bar{\mathbf{s}}_j \dots \bar{\mathbf{s}}_p\right] = \begin{bmatrix} X_1 & \dots & X_j & \dots & X_p \\ Y_1 & \dots & Y_j & \dots & Y_p \\ Z_1 & \dots & Z_j & \dots & Z_p \end{bmatrix}. \tag{1}$$

Alternatively, for later computations, we also define the vectorized form of $\bar{\mathbf{S}}$ as $\bar{\mathbf{S}} \in \mathbb{R}^{3p\times 1}$.

The fundamental equations of discrete structural mechanics are elaborate versions of Newtonian mechanics formulated as force balance statements. The dynamic response of the object Ω is governed by the laws of physics and can be modeled using the general *damped forced vibrations* equation as

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{r}, \tag{2}$$

where $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ are the global mass, damping, and stiffness $3p \times 3p$ matrices, respectively. $\mathbf{u}$ and $\mathbf{r}$ are the $3p \times 1$ 3D nodal displacement and external forces vectors, respectively. The motion at each node j is specified by means of $\mathbf{u}_j = \left(\Delta X_j, \Delta Y_j, \Delta Z_j\right)^\top$. Derivatives with respect to t are denoted by $\dot{\mathbf{u}}(t) \equiv \frac{d\mathbf{u}(t)}{dt}$ and $\ddot{\mathbf{u}}(t) \equiv \frac{d^2\mathbf{u}(t)}{dt^2}$. In the linear case, both matrices $\mathbf{K}$ and $\mathbf{M}$ are independent of $\mathbf{u}$ and $\mathbf{r}$ since they are evaluated at the nondeformed state $\mathbf{u} = \mathbf{0}$.

However, the previous force balance can be replaced in practice by an *undamped free vibrations* equation as

$$\mathbf{M}\ddot{\mathbf{u}}(t) + \mathbf{K}\mathbf{u}(t) = \mathbf{0}. \tag{3}$$

In this case, in the absence of external loads $\mathbf{r} = \mathbf{0}$, the displacement can be determined from the initial conditions, i.e., only knowing a shape at rest $\bar{\mathbf{S}}$. The internal elastic forces $\mathbf{K}\mathbf{u}$ balance the negative of the inertial forces $\mathbf{M}\ddot{\mathbf{u}}$ which can be interpreted as assigning a certain mass and a certain stiffness between nodal points. This undamped model gives conservative answers and is easier to handle numerically. The equation is linear and homogeneous, and its solution is a linear combination of exponentials modulated by the mode shapes, i.e., of the form $\mathbf{u} = \boldsymbol{\psi} \sin(\omega t)$ with $\boldsymbol{\psi}$ a $3p$-dimensional vector and ω a real scalar.

3.2 Proposed Model Matrices

This section is devoted to the computation of the matrices $\mathbf{K}$ and $\mathbf{M}$ in Eq. (3). We discretize the surface of the observed solid into m linear triangular elastic finite elements with $\mathcal{E}$ connectivity. We compute the stiffness matrix $\mathbf{K}$ by means of a model for thin-plate elements. The deformation is modeled as a combination of plane-stress and Kirchhoff's plate, using the free-boundary conditions matrix for linear elastostatic objects with isotropic and homogeneous material properties, as proposed by [3]:

$$\mathbf{K} = \mathop{\mathbf{A}}_{e=1}^{m} \int_{\Omega_e} \mathcal{T}^\top \mathbf{B}_e^\top \mathbf{D}\, \mathbf{B}_e \mathcal{T}\, d\Omega_e, \tag{4}$$

where $\mathbf{B}_e$ is the strain-displacement matrix defined in terms of the approximation function derivatives, i.e., this matrix relates the strain and displacement fields. $\mathbf{D}$ is the constitutive matrix depending on the elastic properties of the material: Young's modulus E and Poisson's ratio ν (for more details, see [10]). We assume near incompressible materials $\nu \approx 0.5$, valid for rubber, papers, and human tissue such as a face. $\mathcal{T}$ is the local-to-global displacement transformation matrix. Finally, $\mathbf{A}$ represents the assembly operator, i.e., the global matrix is assembled from the piecewise contributions.

To model the mass matrix $\mathbf{M}$, we assume a lumped mass at the nodes, leading to the mass matrix being computed as

$$\mathbf{M} = \mathop{\mathsf{A}}_{e=1}^{m} \frac{\rho h A_e}{3} \,\mathbf{diag}\big([1\,1\,1\,1\,1\,1\,1\,1\,1]^\top\big), \tag{5}$$

preserving the total element mass $\sum_a \tilde{M}^e_{aa} = \int_{\Omega_e} \rho \, d\Omega_e$, where $\tilde{M}^e_{aa}$ is the mass per component, and ρ is the material density (mass-density). For simplicity, the surface thickness h is the same for all elements. A_e represents the element area. While a lumped mass provides less rigidity to the shape with respect to a distributed mass (the physical configuration), in our domain this difference is negligible and the accuracy of both models is roughly the same. Since a lumped mass provides a more efficient model, we always use this model.

3.3 Modal Analysis

According to structural engineering FEM analysis [10], the deformed object at a given sample time can be approximated as a linear combination of some mode shapes which can be computed by solving a generalized eigenvalue problem from the undamped free vibration dynamics (see Eq. (3)). Modal analysis is standard in structural engineering and has also been applied in computer vision to the spring mesh model in medical imaging [43]. In [35,42], it was used for motion analysis and to track and recover heart motion, and in [48,50] for nonrigid 2D tracking. Modal analysis was used to decouple the equilibrium equations by obtaining a closed-form solution of Eq. (2).

In this work, we propose to apply modal analysis to a soup of elastic triangles with unknown material properties (E, ρ). Hence, the stiffness matrix $\mathbf{K}$ and mass matrix $\mathbf{M}$ are computed per unit of E and ρ, respectively. This does not limit the generality of our model. The only implication is that the estimated displacement will be up to scale. Our algorithm does not require boundary conditions, i.e., rigid points, but we could exploit them if they were available by solving a constrained eigenvalue problem with Dirichlet constraints to fix the values of several points $\mathbf{u}_j = \mathbf{0}$. It is worth pointing that since our method employs a triangulated mesh of 2D finite elements, a subset of deformations like articulated motion cannot be modeled. To solve this, we would have to incorporate another type of finite elements, such as springs or beams. In a similar way, to model balloon-like objects 3D finite elements are required. Exploring this way is part of our future work. Finally, we directly use the mode basis without using the decoupled system in Eq. (2), avoiding having to estimate the applied forces.

The undamped free vibration response of the 3D structure caused by a disturbance with respect to the shape at rest—modeled by Eq. (3)—can be estimated by solving the generalized eigenvalue problem in ω^2:

$$\mathbf{K}\boldsymbol{\psi}_k = \mathbf{M}\omega_k^2\boldsymbol{\psi}_k \,, \tag{6}$$

where $\{\boldsymbol{\psi}_k, \omega_k^2\}$, $k \in N_m$ are the mode shapes (eigenvectors) and frequencies (eigenvalues), respectively. $N_m := \{1 \ldots 3p\}$ is the index set. Each eigenmode $\boldsymbol{\psi}_k$ is a $3p \times 1$ vector and its components are the displacements for all p nodes in the discretization. The modes are normalized to satisfy the orthonormality conditions $\boldsymbol{\psi}_k^\top \mathbf{M}^{-1}\mathbf{K}\boldsymbol{\psi}_l = \omega_k^2 \boldsymbol{\psi}_k^\top \boldsymbol{\psi}_l$ and $\boldsymbol{\psi}_k^\top \boldsymbol{\psi}_l = \delta_{kl}$, where δ_{kl} is the Kronecker delta and $l \in N_m$, such that $\|\boldsymbol{\psi}_k\|_2 = 1$.

3.4 Mode Shape Basis: Analysis and Selection

Modal analysis yields $3p$ orthonormal modes—provided $\mathbf{K}$ and $\mathbf{M}$ are symmetric positive definite—[see some examples in Fig. 2(left)]. To analyze the mode shapes or vibration modes, we sort them according to the energy they need to be excited, using a frequency spectrum from lower to higher frequencies (see Fig. 2(right)). In the case, where no boundary conditions are imposed, we can approximately identify three practical mode families, instead of the two proposed in [48], separating the intermediate modes into bending and stretching ones as follows:

- *Rigid motion modes (R)* Theoretically the first 6 frequencies should be zero, because they correspond to 6 degree of freedom rigid body motions. However, in practice, the rank of stiffness matrix $\mathbf{K}$ is 4 instead of 6 due to the thin-plate approximation [3]. Hence, the first 4 frequencies are zero up to numerical error. This is consistent with our model having 4 rigid modes instead of the normal 6 in a full 3D FEM model [42]. These 4 rigid motion modes are excluded from the basis when coding nonrigid deformations.
- *Bending modes (B)* Bending, out-of-plane deformations, are mainly represented by the modes localized in the interval $[5, p+4]$ (see Fig. 2). These modes can represent elastic bending deformations (with low stretching in-plane). Moreover, selecting a few of the first bending modes provides an accurate mode basis to model bending deformations.
- *Stretching modes (S)* Stretching deformations can be modeled as a linear combination of the modes in the interval $[p+5, 2p]$. Similarly, selecting only the first stretching modes provides a basis to accurately represent the stretching in-plane deformations.

The rest of the mode shapes $[2p+1, 3p]$—the higher frequencies—do not correspond to physical deformations but to artifacts resulting from the discretization process. When at least three noncollinear rigid points are considered, the rigid motion of the modes does not appear in the frequency spectrum. The first modes are similar to the linear, quadratic, and twist modes in [20,45], but our shape basis is more general including additional modes such as high order modes, not available in the quadratic model. The mode shapes capture decreasingly important details in the shape deformation, following a coarse-to-fine approach. Hence, the lower frequencies within each interval model the global deformations and the higher frequencies the local details.

To sum up, any nonrigid 3D displacement $\mathbf{u}$ can be modeled by a basis that contains only $\{k_1, \ldots, k_r\}, r << 3p$ mode shapes. For notational simplicity, it is assumed that the r selected modes are renumbered $k = 1, \ldots, r$, so any displacement field vector can be approximated by a linear combination of the mode shapes expressed as

$$\mathbf{u} \approx \mathcal{S}\boldsymbol{\gamma} = \left[\boldsymbol{\psi}_1 \ldots \boldsymbol{\psi}_k \ldots \boldsymbol{\psi}_r\right]\left[\gamma_1 \ldots \gamma_k \ldots \gamma_r\right]^\top, \qquad (7)$$

where the transformation matrix $\mathcal{S} \in \mathbb{R}^{3p \times r}$ concatenates r mode shapes, and $\boldsymbol{\gamma}$ is a weight vector to obtain a low-rank representation. It is worth pointing that the mode shapes $\boldsymbol{\psi}_k$ do not depend on the material properties (E, ρ). For the same shape at rest, the normalized modes are the same irrespective of (E, ρ). Different (E, ρ) would produce the same deformation but with a different amplitude, which can be absorbed into the deformation weights γ_k.

3.5 Fitting Real Deformations

To empirically demonstrate the suitability of the mode shape basis for coding real deformations, we propose to use our basis to fit 3D deforming objects. To this end, we use two datasets with 3D ground truth acquired from motion capture systems. Particularly, we employ the *serviette* and *carton* datasets[2] that consist in 102 shapes with 63 nodal points and in 53 shapes with 81 nodal points, respectively. Figure 3 show the effect on 3D errors of varying the rank of the subspace, observing a consistent reduction of the error as more mode shapes r are considered (some examples are also shown in figure). It is worth pointing that just a few mode shapes reduce the error by half, observation that we exploit in this paper by proposing a low-rank shape constraint. To make a fair comparison, we also consider two configurations to train a PCA-based approach. The PCA-1 in which the PCA basis is trained with first 10 samples; the PCA-2 where all data are used to train. Note that our approach consistently outperforms PCA-1, without explicitly using 3D deformable training data. Even though PCA-based methods can become very accurate if appropriate learning data are available (such as PCA-2), this method have used an accurate deformation model learned from the ground truth of the whole data, which may be difficult to obtain in real applications such as medical videos we process in experimental section. This limitation is outperformed by our method, which in contrast just needs a rest shape estimation.

3.6 Computational Cost

In this section, we analyze the computational complexity of obtaining mode shapes when solving the eigenvalue problem. To do this, we first compute $\mathbf{M}^{-1}\mathbf{K}$ to transform the generalized eigenvalue problem into a standard one. In the case of the lumped mass matrix—a diagonal matrix—in Eq. (5), the inverse computation cost is negligible. The actual computation and assembly of both $\mathbf{K}$ and $\mathbf{M}$ has a $\mathcal{O}(p)$ complexity that is only significant for dense maps, where the process could easily be parallelized.

The second step is the computation of the mode shapes as eigenvectors. For computational efficiency, we propose to use orthogonal iteration with Rayleigh–Ritz acceleration [18,24]. This returns the eigenvectors (mode shapes), sequentially in ascending frequency order, hence the complexity scales with the number of computed modes. If the r mode shapes to be included in the basis correspond only to low frequencies (bending modes), the computation is efficient with a complexity scaling with $4 + r$ even for dense meshes. However, if both bending and stretching modes have to be computed, the complexity scales with $4 + r + p$, which may become prohibitive in both computation time and memory requirements, especially for dense meshes.

3.7 Computation of Mode Shapes for Dense Meshes

The mode shapes have been classified into two families: bending and stretching modes. The first family of modes is affordable to compute even in the dense case by solving Eq. (6), but it can only encode out-of-plane bending deformations. In contrast, for scenes with stretching in-plane deformations, a few stretching modes have to be included to have a representative basis. Unfortunately, computing these dense modes may become prohibitive in terms of computational and memory requirements. To solve this limitation, in this work, we propose two algorithms to compute mode shape at a lower cost. The first computes the eigenvalue problem in a sparse mesh to obtain the frequencies of vibration and then computes the stretching modes applying the *frequency-based mode shapes* method to the dense mesh. The second approach computes the eigenvalue problem in a sparse mesh and, exploiting the approximation functions within finite elements, computes modes in the dense mesh applying a robust *coarse-to-fine approach*. This enables dense problems to be easily solved and drastically reduces the computational and memory requirements to compute all frequencies in the spectrum.

[2] We use the motion capture data from: http://cvlab.epfl.ch/data/dsr. These data were acquired with a Vicon motion capture system. It contains one sequence of a deforming piece of cloth (*serviette* data) and one sequence of a deforming piece of cardboard (*carton* data).

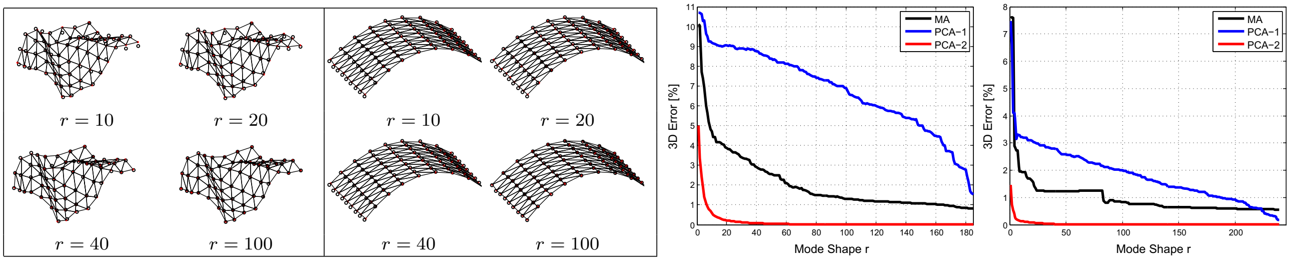


Fig. 3 Fitting real 3D models with the proposed mode shape basis as a function of the number of mode shapes. *Left* We display a specific shape for each dataset using different values of $r = \{10, 20, 40, 100\}$. Reconstructed 3D shape and the corresponding 3D ground truth are shown with *red dots* and *black circles*, respectively. *Right* Results for *serviette* and *carton* datasets, respectively (Color figure online)

3.7.1 Frequency-based Dense Mode Shape Estimation

In this section, we present our frequency-based method to compute dense stretching mode shapes. Our approach begins by solving the eigenvalue problem in Eq. (6) for a sparse mesh of q points where $q << p$, allowing us to easily obtain all the frequencies in the spectrum. As we can locate the stretching modes in the ordered frequency spectrum (see section 3.4), we can obtain a finite interval of frequencies for these modes situated between ω_{low} and ω_{high}. Next, we can formulate the problem to estimate mode shapes over the dense mesh of p points as

$$\begin{aligned} &\arg\min_{\boldsymbol{\psi}_k} \; \| \left(\mathbf{M}^{-1}\mathbf{K} - \omega_k^2 \mathbf{I}\right) \boldsymbol{\psi}_k \|^2 \\ &\text{subject to} \quad \omega_{low} \leq \omega_k \leq \omega_{high}, \end{aligned} \tag{8}$$

where $\boldsymbol{\psi}_k$ are the mode shapes with frequencies between ω_{low} and ω_{high}. Although this approach is valid for both bending and stretching modes, our aim is to obtain the stretching modes while avoiding having to compute the bending modes. Figure 4 displays the frequency spectrum for three meshes and some stretching modes applying this approach over the dark cyan sparse mesh. It can be seen that the frequency ω_{low} remains unchanged for all meshes. We can obtain this frequency over a sparse mesh, and then solve Eq. (8) over a dense mesh by linear least squares. The computed mode shapes using this method are not an approximation and they can code both low and high frequencies.

While this approach offers new theoretical insights, it is very sensitive to changes in the rest shape. Due to this sensitive, this approach is particularly relevant for regular meshes (such as the used in registration) where the sub-sampling process is trivial. For instance, it could easily be employed in template-based methods [33,46,47] using multiple regular meshes and a template where the 3D shape is accurately known.

3.7.2 Coarse-to-Fine Mode Shape Estimation

In this section, we propose our second efficient approach to compute dense mode shapes. We exploit the approximation functions used to define the displacement field within a finite element [10]. Therefore, as in the previous method, we solve the eigenvalue problem for $q << p$ scene points obtaining $\mathcal{S}^* \in \mathbb{R}^{3q\times r}$ basis shapes and then compute $\mathcal{S} \in \mathbb{R}^{3p\times r}$ for p points using the approximation functions. First, we subsample the scene points to convert the p-dimensional map—dense mesh—(see Fig. 5b) into a q-dimensional map–sparse mesh—(see Fig. 5c, d). Each point in the dense mesh

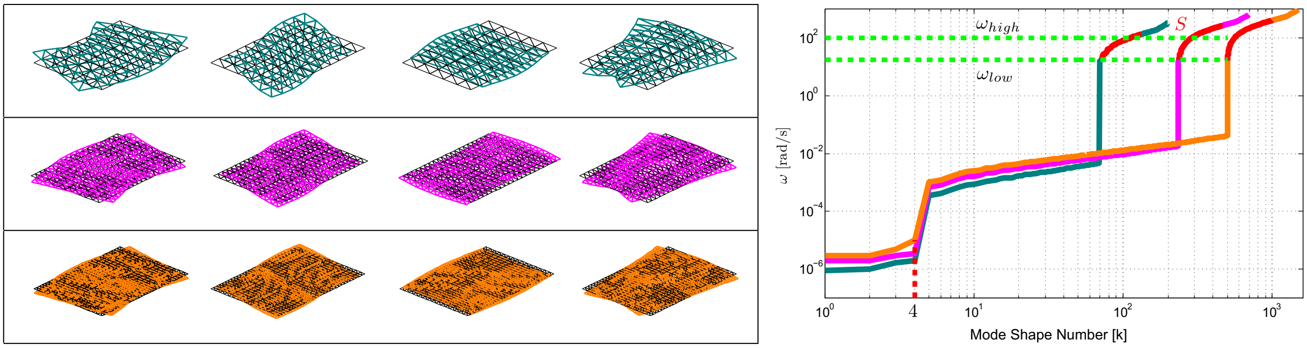


Fig. 4 *Left* Representation of some stretching mode shapes of a plate surface discretized into triangular elements with three meshes. *First row* 5 × 10 mesh in *dark cyan*. *Second row* 10 × 20 mesh in magenta. *Third row* 15 × 30 in orange. The shape at rest is always displayed with a black mesh. *Right* Eigen-frequencies ω_k for the previous black multiple meshes in logarithmic scale. We display the interval $[\omega_{low}, \omega_{high}]$ to find stretching mode shapes in green, being similar for all meshes. Best viewed in *color* (Color figure online)

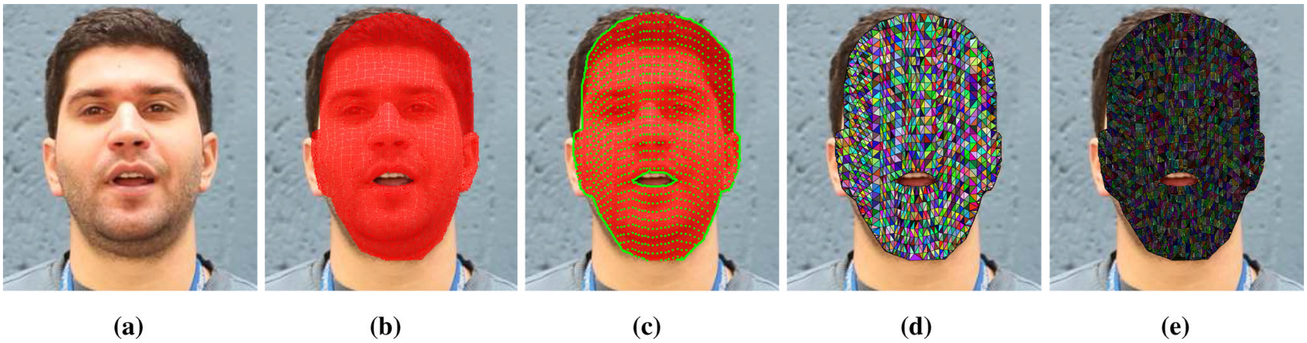


Fig. 5 Coarse-to-fine approach to modal analysis. **a** Reference image plane to compute optical flow. **b** Dense 2D tracking of p points. **c** Subsample of dense shape into q points (*green points*). **d** Delaunay triangulation for sparse mesh. **e** Active search to match every point in the sparse mesh. Please zoom into the electronic images for a detailed view (Color figure online)

must then be matched with an element of the sparse mesh (see Fig. 5e). The q-dimensional map has to ensure a sparse mesh where every p point can be matched. To find an element in the sparse mesh $\triangle(\bar{\mathbf{s}}_a\bar{\mathbf{s}}_b\bar{\mathbf{s}}_c)$ with nodal labels $\{a, b, c\}$ per point $\bar{\mathbf{s}}_j$, we suggest an active search by computing several cross products over a two-dimensional space:

$$\sum_{\substack{\tau=0\\ \tau+1\in\text{mod}(3)}}^{2}\left(\chi_\iota\left(\overrightarrow{\bar{\mathbf{s}}_\tau\,\bar{\mathbf{s}}_j}\times\overrightarrow{\bar{\mathbf{s}}_\tau\,\bar{\mathbf{s}}_{\tau+1}}\right)=\begin{cases}1 & \text{if}\left(\overrightarrow{\bar{\mathbf{s}}_\tau\,\bar{\mathbf{s}}_j}\times\overrightarrow{\bar{\mathbf{s}}_\tau\,\bar{\mathbf{s}}_{\tau+1}}\right)\in\iota\\ 0 & \text{if}\left(\overrightarrow{\bar{\mathbf{s}}_\tau\,\bar{\mathbf{s}}_j}\times\overrightarrow{\bar{\mathbf{s}}_\tau\,\bar{\mathbf{s}}_{\tau+1}}\right)\notin\iota\end{cases}\right)$$
$$=\begin{cases}3 & \text{if } \bar{\mathbf{s}}_j\in\triangle(\bar{\mathbf{s}}_0\bar{\mathbf{s}}_1\bar{\mathbf{s}}_2)\\ \leq 2 & \text{if } \bar{\mathbf{s}}_j\notin\triangle(\bar{\mathbf{s}}_0\bar{\mathbf{s}}_1\bar{\mathbf{s}}_2)\end{cases} \tag{9}$$

where the labels $\{a, b, c\}\equiv\{0, 1, 2\}$ are renumbered and χ_ι represents a step function with $\iota\equiv[0,\infty)$. $\bar{\mathbf{s}}_j$ is inside the triangle element $\triangle(\bar{\mathbf{s}}_a\bar{\mathbf{s}}_b\bar{\mathbf{s}}_c)$ when all the cross products are nonnegative, three in our case. Note that this two-dimensional space is a dimensional reduction of the shape at rest, and hence it can be obtained either by projecting the shape at rest onto the image plane or using the reference image plane (see Figs. 5a, 6). When the active search is completed, we need to compute its natural coordinates (ξ_j, η_j) within the element $\triangle(\bar{\mathbf{s}}_a\bar{\mathbf{s}}_b\bar{\mathbf{s}}_c)$. First, we transform from the global system to a $\mathcal{L}$ local one $\triangle\left(\bar{\mathbf{s}}_a^{\mathcal{L}}\bar{\mathbf{s}}_b^{\mathcal{L}}\bar{\mathbf{s}}_c^{\mathcal{L}}\right)$ defined on the plane of each triangle element and then obtain the natural coordinates as

$$\begin{bmatrix}\xi_j\\ \eta_j\end{bmatrix}=\left[\left[\bar{\mathbf{s}}_b^{\mathcal{L}}\ \bar{\mathbf{s}}_c^{\mathcal{L}}\right]-\left[\bar{\mathbf{s}}_a^{\mathcal{L}}\otimes\mathbf{1}_2^\top\right]\right]^{-1}\left[\bar{\mathbf{s}}_j^{\mathcal{L}}-\bar{\mathbf{s}}_a^{\mathcal{L}}\right], \tag{10}$$

where $\mathbf{1}_2$ is a vector of ones and $\otimes$ indicates the Kronecker product. The 3D displacement can be obtained using the linear approximation functions $\mathfrak{N}^l(\xi_j, \eta_j)\equiv\left[N_1^l\ N_2^l\ N_3^l\right]$ [10] within the element. The 3D displacement for every mode shape can be computed as

$$\mathcal{S}_j=\left[\mathfrak{N}^l(\xi_j, \eta_j)\otimes\mathbf{I}_3\right]\left[\mathcal{S}_a^*\ \mathcal{S}_b^*\ \mathcal{S}_c^*\right]^\top, \tag{11}$$

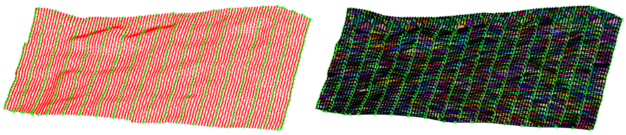


Fig. 6 *Left* A dense shape of p nodes (*red points*) is reduced to q nodes (*green points*) to solve the eigenvalue problem. In this case, the 2D space can be obtained by projecting the 3D shape at rest estimation. *Right* Active search to match the p points with triangular elements of the sparse mesh. Please zoom into the electronic images for a detailed view (Color figure online)

where $\mathcal{S}_a^*$, $\mathcal{S}_b^*$, and $\mathcal{S}_c^*$ are $3\times r$ displacement vectors for the mode shapes basis corresponding to the triangle element $\{a, b, c\}$ to which $\bar{\mathbf{s}}_j$ belongs. Finally, $\mathcal{S}_j$ is placed in rows $3j-2$ through $3j$ in $\mathcal{S}$. Note that for coding nonrigid shapes, the high-frequency details are included in the shape at rest with dense mesh, while these have been lost in the shape basis for the approximation of this method. However, our approximation is accurate even using a sparse mesh with a 10 % of points in the dense mesh, as we show in the experimental section.

Regarding complexity, we drastically reduce the computational cost of computing dense mode shapes using this approach. To compute and assemble both FEM matrices, the complexity is reduced from $\mathcal{O}(p)$ to $\mathcal{O}(q)$, with q being the number of points in the sparse mesh, representing under 10 % of the points in the dense mesh. This allows us to easily compute the full frequency spectrum—including stretching shapes with high frequencies—solving a simple eigenvalue problem of low dimension. It is worth to point that for the frequency-based algorithm, the complexity to solve the eigenvalue problem over a sparse mesh is similar, but later it is necessary to compute a few mode shapes over the dense mesh and the performance is reduced.

4 Problem Formulation

Let us consider a 3D structure $\mathbf{S}_f$ of p points on an image frame f. The orthographic projection $\mathbf{W}_f$ is expressed as

$$\mathbf{W}_f = \begin{bmatrix} u_{f1} \dots u_{fj} \dots u_{fp} \\ v_{f1} \dots v_{fj} \dots v_{fp} \end{bmatrix} = \mathbf{\Pi}\mathbf{Q}_f\mathbf{S}_f + \mathbf{T}_f + \mathbf{N}_f, \quad (12)$$

where $\mathbf{\Pi}$ is the 2×3 orthographic camera matrix, and $\mathbf{Q}_f$ is the 3×3 rotation matrix. Next, $\mathbf{R}_f = \mathbf{\Pi}\mathbf{Q}_f$ are the first two rows of a full rotation matrix (i.e., $\mathbf{R}_f\mathbf{R}_f^\top = \mathbf{I}_2$). Due to the orthographic projection, the depth coordinate of the translation vector cannot be resolved. Considering this ambiguity, we model 2D translations $\mathbf{t}_f$ defined as $\mathbf{T}_f = \mathbf{1}_p^\top \otimes \mathbf{t}_f$ with $\mathbf{1}_p$ a vector of ones and $\mathbf{t}_f = \mathbf{R}_f\mathbf{d}_f$, where $\mathbf{d}_f$ is a 3D-translation vector. Finally, $\mathbf{N}_f$ is a $2 \times p$ zero-mean Gaussian noise process matrix for modeling the noise in image tracks. The noise vector for a generic point j is $\mathbf{n}_{fj} \sim \mathcal{N}\left(0; \sigma^2\mathbf{I}\right)$ with variance of the measurements σ in each dimension.

Alternatively, the orthographic projection $\mathbf{w}_f$ into vectors can be expressed as

$$\mathbf{w}_f = \begin{bmatrix} u_{f1} & v_{f1} & \dots & u_{fp} & v_{fp} \end{bmatrix}^\top = \mathbf{G}_f\mathbf{S}_f + \bar{\mathbf{t}}_f + \mathbf{n}_f, \quad (13)$$

where $\mathbf{G}_f = \mathbf{I}_p \otimes \mathbf{R}_f$ with $\mathbf{I}_p$ a $p \times p$ identity matrix, and $\bar{\mathbf{t}}_f$ and $\mathbf{n}_f$ are the vectorized form of $\mathbf{T}_f$ and $\mathbf{N}_f$, respectively.

Our problem is to simultaneously estimate camera motion $\left(\mathbf{R}_f, \mathbf{t}_f\right)$ and the nonrigid 3D shape $\mathbf{S}_f$ as the data arrive, in every frame f from uncalibrated 2D point correspondences in a monocular video $\mathbf{W}_f$. Note that our measurements can have occlusions and lost tracks, and the measurement matrix $\mathbf{W}_f$ is not full. For these cases, only a set $\mathcal{V}$ of visible points ϱ is observed, but always all p scene points have to be reconstructed.

4.1 Proposed Linear Subspace Deformation Model

We propose to model the nonrigid shape at each instant as a linear combination of a mean shape and r deformation mode shapes from modal analysis. Hence the estimation of the 3D structure at each frame f comes down to estimating the corresponding weight vector $\boldsymbol{\gamma}_f$. It is also useful in our formulation to represent $\mathcal{S}$ using a permutation operator $\mathcal{R}(\mathcal{S})$ that rearranges the entries of $\mathcal{S}$ into a $3r \times p$ matrix such that the j-th column of $\mathcal{R}(\mathcal{S})$ contains all the displacement $\Delta X_{rj}, \Delta Y_{rj}, and \Delta Z_{rj}$ coordinates of the point j for all r modes. Considering Eq. (7), the 3D displacement field into matrices can be rewritten as $\mathbf{U} = \left(\mathbf{I}_3 \otimes \boldsymbol{\gamma}^\top\right)\mathcal{R}(\mathcal{S})$ where $\mathbf{I}_3$ is a 3×3 identity matrix. The deformed structure at frame f can be written as

$$\mathbf{S}_f = \bar{\mathbf{S}} + \mathbf{U}_f = \bar{\mathbf{S}} + \left(\mathbf{I}_3 \otimes \boldsymbol{\gamma}_f^\top\right)\mathcal{R}(\mathcal{S}). \quad (14)$$

Assuming noise-free observations, we can rewrite the orthographic projection Eq. (12) as

$$\mathbf{W}_f = \mathbf{R}_f\left(\bar{\mathbf{S}} + \left(\mathbf{I}_3 \otimes \boldsymbol{\gamma}_f^\top\right)\mathcal{R}(\mathcal{S})\right) + \mathbf{T}_f. \quad (15)$$

In section 5.1, we propose an on-line BA-based algorithm to solve this nonlinear problem as the data arrive.

4.2 Proposed Probabilistic Modal Analysis Model

We also propose to replace the linear subspace model described above with a probabilistic model using a Gaussian prior on each shape in the subspace, inspired by probabilistic PCA [44,53,56]. The weight coefficients $\boldsymbol{\gamma}_f$ are modeled with a Gaussian prior distribution with zero-mean as

$$\boldsymbol{\gamma}_f \sim \mathcal{N}\left(\mathbf{0}; \mathbf{I}_r\right). \quad (16)$$

These deformation weights $\boldsymbol{\gamma}_f$ become latent variables that can be marginalized out and are never explicitly computed. Employing this Gaussian prior over $\boldsymbol{\gamma}_f$, the weights for each shape are similar to each other, the nonrigid shape distribution being $\mathbf{s}_f \sim \mathcal{N}\left(\bar{\mathbf{s}}; \mathcal{S}\mathcal{S}^\top\right)$, producing smooth deformations with respect to shape at rest. The r principal axes are modeled using an orthogonal fixed basis $\mathcal{S}$. In this case, the nonrigid structure at frame f, considering Eq. (7) can be written as

$$\mathbf{s}_f = \bar{\mathbf{s}} + \mathbf{u}_f = \bar{\mathbf{s}} + \mathcal{S}\boldsymbol{\gamma}_f = \tilde{\mathcal{S}}\tilde{\boldsymbol{\gamma}}_f, \quad (17)$$

where $\tilde{\mathcal{S}} = [\bar{\mathbf{s}} \;\; \mathcal{S}]$ is the concatenation matrix of the shape at rest together with the mode shapes, and $\tilde{\boldsymbol{\gamma}}_f = \begin{bmatrix} 1 & \boldsymbol{\gamma}_f^\top \end{bmatrix}^\top$.

By assuming Gaussian noise over the shape and the observations, considering Eq. (13) the distribution to be estimated over the projected points $\mathbf{w}_f$ is also Gaussian:

$$\mathbf{w}_f \sim \mathcal{N}\left(\mathbf{G}_f\bar{\mathbf{s}} + \bar{\mathbf{t}}_f; \mathbf{G}_f\mathcal{S}\mathcal{S}^\top\mathbf{G}_f^\top + \sigma^2\mathbf{I}\right). \quad (18)$$

This is equivalent to solving the NRS*f*M problem. In Sect. 5.2, we propose an on-line EM-based algorithm to solve the maximum likelihood estimation as the data arrive in this latent variable problem.

5 Sequential NRS*f*M

Our aim is to sequentially estimate camera motion and time-varying 3D shape from 2D point tracks. This section is devoted to describing the details of our sequential approaches to NRS*f*M. Note that we just need to tune the number of mode shapes r in the subspace. While most state-of-the-art techniques are very sensitive to the choice of the specific rank, our methods do not require fine tuning (see experimental Sect. 6). Obtaining a sequential system in real time could easily be fixed in terms of computational cost.

5.1 On-line BA Algorithm

We use a sliding temporal window approach as proposed in [40] to perform BA [58] on the last $\mathcal{W}$ frames. As the deformation modes, $\mathcal{S}$ are computed in a previous step following the modal analysis described in Sect. 3, and only an estimate of the shape at rest $\bar{\mathbf{s}}$ is required, the trilinear sequential NRS*f*M problem is reduced to the estimation of per-frame camera motion $(\mathbf{R}_i, \mathbf{t}_i)$ and deformation weights $\boldsymbol{\gamma}_i$, i.e., a bilinear problem. This involves the estimation of a very small number of parameters r to encode the shape at each frame, which leads to a low computational cost system. The model parameters are estimated on the fly by minimizing the image re-projection error of all the observed points ϱ over all frames in the current temporal window $\mathcal{W}$ by means of the following cost function $\mathcal{A}\left(\mathbf{R}_i, \mathbf{t}_i, \boldsymbol{\gamma}_i\right)$:

$$\begin{aligned}
&\underset{\mathbf{R}_i,\mathbf{t}_i,\boldsymbol{\gamma}_i}{\arg\min}\, \lambda_\gamma \sum_{i=f-\mathcal{W}+2}^{f} \|\boldsymbol{\gamma}_i^\top - \boldsymbol{\gamma}_{i-1}^\top\|_{\mathcal{F}}^2 \\
&\quad + \lambda_t \sum_{i=f-\mathcal{W}+2}^{f} \|\mathbf{t}_i - \mathbf{t}_{i-1}\|_{\mathcal{F}}^2 \\
&\quad + \sum_{i=f-\mathcal{W}+1}^{f} \sum_{\varrho\in\mathcal{V}} \|\mathbf{W}_{i\varrho} - \mathbf{R}_i \left(\bar{\mathbf{S}}_\varrho + \left(\mathbf{I}_3 \otimes \boldsymbol{\gamma}_i^\top\right) \mathcal{R}(\mathcal{S}_\varrho)\right) \\
&\quad - \mathbf{T}_i\|_{\mathcal{F}}^2 + \lambda_q \sum_{i=f-\mathcal{W}+2}^{f} \|\mathbf{R}_i(\mathbf{q}_i) - \mathbf{R}_{i-1}(\mathbf{q}_{i-1})\|_{\mathcal{F}}^2, \quad (19)
\end{aligned}$$

where $\|\cdot\|_{\mathcal{F}}$ is the Frobenius norm. The rotation matrices $\mathbf{R}_i(\mathbf{q}_i)$ are parameterized using quaternions to guarantee orthonormality $\mathbf{R}_i\mathbf{R}_i^\top = \mathbf{I}_2$. We add temporal smoothness priors to penalize strong variations in the deformation weights $\boldsymbol{\gamma}_i$, translations $\mathbf{t}_i$, and camera matrices $\mathbf{R}_i(\mathbf{q}_i)$. The positive regularization weights λ_γ, λ_t, and λ_q govern the relative importance of the regularization terms, and they are determined empirically. These coefficients are fixed in all the experiments we describe in the experimental section. This problem can be solved using a sparse Levenberg–Marquardt nonlinear minimization algorithm.

To initialize the parameters for a new incoming frame, we consider the smoothness priors for both camera motion and nonrigid deformation, penalizing far solutions from the previous frame. The mode shape weights $\boldsymbol{\gamma}_i$ are initialized assuming rigid motion $\boldsymbol{\gamma}_i = \boldsymbol{\gamma}_{i-1}$, and the camera pose is initialized as $\mathbf{R}_i = \mathbf{R}_{i-1}$ and $\mathbf{t}_i = \mathbf{t}_{i-1}$.

5.2 On-line EM Algorithm

We propose an on-line version of the EM algorithm—similar to EM for factor analysis [23]—over a sliding window on the last $\mathcal{W}$ frames, similar to the algorithm described above. In this case, we denote the set of model parameters to be estimated by $\Theta_f \equiv \{\mathbf{R}_f, \mathbf{t}_f, \sigma^2\}$, the hidden data by $\boldsymbol{\gamma}_f$, and the complete data as $\{\mathbf{w}_f, \boldsymbol{\gamma}_f\}$. Given the observable data $\mathbf{w}_{f-\mathcal{W}+1:f}$ over the sliding temporal window of frames with indexes $_{f-\mathcal{W}+1:f}$, we estimate the model parameters over all the frames in the current window denoted as $\hat{\mathcal{W}}$. For dealing with missing data, we consider all the observed points $\varrho \in \mathcal{V}$ over all the frames in the current window. Note that for full measurements, the number of visible points ϱ is equal to p. The joint probability of $\mathbf{w}$ over a sliding window, assuming that the samples are independent and identically distributed, may be computed considering the Gaussian distribution per frame, Eq. (18), as

$$p\left(\mathbf{w}_{\hat{\mathcal{W}}}|\mathbf{G}_{\hat{\mathcal{W}}}, \bar{\mathbf{t}}_{\hat{\mathcal{W}}}, \sigma^2\right) = \prod_{i=f-\mathcal{W}+1}^{f} p\left(\mathbf{w}_{i\varrho}|\mathbf{G}_i, \bar{\mathbf{t}}_i, \sigma^2\right). \quad (20)$$

The EM algorithm estimates iteratively until convergence the likelihood, alternating between two steps: the *E-step* and *M-step*. In the *E-step*, the expectation of the data likelihood with respect to the latent variable distribution is computed. For this, we compute the posterior distribution over latent variables $p\left(\boldsymbol{\gamma}_{\hat{\mathcal{W}}}|\mathbf{w}_{\hat{\mathcal{W}}}, \Theta_{\hat{\mathcal{W}}}\right) \propto p\left(\mathbf{w}_{\hat{\mathcal{W}}}|\boldsymbol{\gamma}_{\hat{\mathcal{W}}}, \Theta_{\hat{\mathcal{W}}}\right) p\left(\boldsymbol{\gamma}_{\hat{\mathcal{W}}}|\Theta_{\hat{\mathcal{W}}}\right)$ given the measurements and the current parameter model on the sliding window as

$$p\left(\boldsymbol{\gamma}_{\hat{\mathcal{W}}}|\mathbf{w}_{\hat{\mathcal{W}}}, \Theta_{\hat{\mathcal{W}}}\right) \sim \prod_{i=f-\mathcal{W}+1}^{f} \mathcal{N}\left(\boldsymbol{\mu}_i; \boldsymbol{\Sigma}_i\right),$$

where $\boldsymbol{\mu}_i = \beta_i\left(\mathbf{w}_i - \mathbf{G}_i\bar{\mathbf{s}} - \bar{\mathbf{t}}_i\right)$ and $\boldsymbol{\Sigma}_i = \mathbf{I}_r - \beta_i\mathbf{G}_i\mathcal{S}$, with $\beta_i = \mathcal{S}^\top\mathbf{G}_i^\top\left(\sigma^2\mathbf{I}_{2p} + \mathbf{G}_i^\top\mathcal{S}(\mathbf{G}_i\mathcal{S})^\top\right)^{-1}$ that is efficiently computed using the Woodbury matrix identity [63].

Note that for the missing data case, $\mathbf{G}_i = \mathbf{I}_\varrho \otimes \mathbf{R}_i$ with $\mathbf{I}_\varrho$ being a ϱ-order identity matrix. $\mathcal{S}$, $\bar{\mathbf{s}}$, and $\mathbf{w}_i$ are only evaluated for visible points.

In the *M-step*, the expected value of the log-likelihood function is optimized by replacing the latent variables by their expected values to update the model parameters. We update motion variables maximizing the likelihood with respect to parameters Θ holding hidden distribution fixed, or equivalently minimizing its expected negative log-likelihood function $\nabla_{\Theta_i^t}\, \mathcal{B}\left(\Theta_i^{t-1}\right)|_{\Theta_i^t=\Theta_i^{t-1}} = 0$, because the logarithm is strictly monotonous:

$$\begin{aligned}
\mathcal{B}(\Theta_i) &= \underset{\Theta_i}{\arg\min}\ \mathbb{E}\left[-\sum_{i=f-\mathcal{W}+1}^{f}\sum_{\varrho\in\mathcal{V}} \log p\left(\mathbf{w}_{i\varrho}|\Theta_i\right)\right] \\
&= \underset{\mathbf{R}_i,\mathbf{t}_i,\sigma^2}{\arg\min}\ \frac{1}{2\sigma^2}\sum_{i=f-\mathcal{W}+1}^{f}\sum_{\varrho\in\mathcal{V}} \mathbb{E}\Big[\|\mathbf{w}_{i\varrho}
\end{aligned}$$

$$-\mathbf{G}_i\left(\bar{\mathbf{s}}_{\varrho}+\mathcal{S}_{\varrho}\boldsymbol{\gamma}_i\right)-\bar{\mathbf{t}}_i\|_2^2\Big]$$
$$+\varrho\mathcal{W}\log\left(2\pi\sigma^2\right), \qquad (21)$$

where $\mathbb{E}[\cdot]$ represents the expectation operator.

Note that using the Gaussian prior, we do not need to add additional temporal smoothness priors to penalize strong variations, avoiding tuning regularization weights [21,40,45]. This function cannot be optimized in closed form to compute a global optimum, and partial M-steps are necessary. The model parameters are individually updated in closed form, except for camera rotation (see Appendix for update rules).

In order to initialize the model parameters for a new incoming frame, the camera pose is initialized as $\mathbf{R}_i=\mathbf{R}_{i-1}$ and $\mathbf{t}_i=\mathbf{t}_{i-1}$, while the latent variables are initialized assuming a rigid motion as $\mathbb{E}\left[\boldsymbol{\gamma}_i\right]=\mathbb{E}\left[\boldsymbol{\gamma}_{i-1}\right]$.

5.3 Missing Data on BA-FEM and EM-FEM Algorithms

Unlike other methods [8,12,15], our formulations can cope with incomplete image tracks resulting from occlusions or tracking outliers, always reconstructing all the scene points. Both algorithms have the capability of dealing with missing data since the cost function is only evaluated on the set of visible points $\mathcal{V}$. After that, we exploit our global model to estimate the 2D location $\hat{\mathbf{W}}_{fj}$ of the missing data entries in the observation matrix:

$$\hat{\mathbf{W}}_{fj}=\mathbf{R}_f\left(\bar{\mathbf{S}}_j+\left(\mathbf{I}_3\otimes\boldsymbol{\gamma}_f^{\top}\right)\mathcal{R}(\mathcal{S}_j)\right)+\mathbf{t}_f. \qquad (22)$$

5.4 Shape at Rest Estimation

Our sequential approaches assume that the shape at rest can be estimated similarly to batch [20,45] and sequential [1,40] approaches, using a rigid factorization algorithm [31] on a few initial frames. To this end, we consider that the observed sequence contains some nr initial frames where the object is mostly rigid and does not deform substantially. Our method does not require the surfaces to be planar or developable, and modal analysis can be applied on general nonplanar rest shapes. Note that using rigid SfM to initialize is standard practice in the NRSfM community and, moreover, initialization is one of the challenges in sequential methods, including the rigid case [27].

The rest shape is a tuple $(\mathcal{P},\mathcal{E})$ where $\mathcal{P}=(n_1,\ldots,n_p)$ is a finite set of p nodes and $\mathcal{E}=(e_1,\ldots,e_m)$ is a finite set of m triangular elements—over q points for dense cases—obtained by means of a Delaunay triangulation [10] on the image plane or on the reference image for dense cases [21,22]. For simplicity, we have used the Delaunay triangulation, although we could take advantage of having an estimation of the 3D shape at rest and easily use alternative connectivity algorithms.

5.5 On-line Complexities

In this section, we show both the complexity of the memory and the computational complexity of our sequential approaches. Our NRSfM methods ensure that the computation time per image is bounded and does not grow with the number of frames, as occurs with batch approaches [20,21,39,52]. We show how our methods could potentially achieve real-time performance at frame rate and, more importantly, how both methods process the video sequence as the data arrive, frame by frame.

First, we analyze the memory complexity. Our on-line algorithms store only the latest $\mathcal{W}$ frames with complexity $\mathcal{O}(\mathcal{W})$, unlike batch algorithms where all the frames are stored with complexity $\mathcal{O}(f)$. This is an important feature for real applications where the computational resources are often much lower than in a desktop setting [62]. Next, we analyze the computational complexity for the BA-FEM and EM-FEM algorithms.

5.5.1 BA-FEM Algorithm

An efficient BA implementation consists of three principal parts: the computation of the residuals and sparse Jacobians used for the Levenberg–Marquardt algorithm with $\mathcal{O}(p\mathcal{W}r)$, building the linear system with $\mathcal{O}\left(p\mathcal{W}^2r^2\right)$ and, finally, solving the linear system with $\mathcal{O}\left(\mathcal{W}^3r^3\right)$. Since in our domain the number of frames on the sliding window $\mathcal{W}$ is relatively small, and the number of modes r is significantly smaller than the number of points in the structure shape p, and the cubic term of solving the linear system is negligible. Thus, the complexity of sequential-mode BA over a sliding window is dominated by building the linear system and computing residuals and Jacobians, dominated by $\mathcal{O}\left(p\mathcal{W}^2r^2\right)$.

5.5.2 EM-FEM Algorithm

The complexity of sequential-mode EM over the sliding window is dominated by the *E-step* and can be approximated on the order of $\mathcal{O}\left(t\mathcal{W}pr^2\right)$ as a function of the number of iterations t per frame until convergence to the desired precision. The likelihood function is increased at each iteration until convergence, i.e., $\mathcal{B}\left(\Theta_i^{t+1}\right)-\mathcal{B}\left(\Theta_i^{t}\right)<\epsilon$, where the superscript ^{t} represents the EM iteration. In practice, only a few iterations are necessary to achieve convergence. Moreover, in our on-line EM-FEM algorithm, we use a temporal smoothness by means of the sliding window that is initialized as the last frame, but we do not implicitly use regularizations within the sliding window as in our BA-FEM algorithm, reducing the complexity order from $\mathcal{W}^2$ to $\mathcal{W}$. The total complexity becomes $\mathcal{O}\left(pr^2\right)$ considering $\mathcal{W}$ to be negligible, 5 in our experiments. Furthermore, this complexity over r can be reduced when sparsity is exploited. Our sliding window approach may potentially achieve real-time performance since it could easily be parallelized for the *i.i.d.* assumption.

6 Experimental Results

In this section, we show the experimental results for both synthetic and real sequences, providing comparisons with respect to state-of-the-art methods. We present results from sparse to dense shapes, for isometric and elastic deformations, and finally we show our performance against corrupted observations, such as noise or missing data. In all the experiments, we report the error $e_{3\mathcal{D}} = \frac{1}{f}\sum_{i=1}^{f}\frac{\|\mathbf{S}_i - \mathbf{S}_i^{GT}\|_{\mathcal{F}}}{\|\mathbf{S}_i^{GT}\|_{\mathcal{F}}}$ where $\mathbf{S}_i$ is the 3D reconstruction and $\mathbf{S}_i^{GT}$ is the ground truth. Before computing this 3D error, the 3D reconstruction is aligned with the corresponding ground truth using Procrustes analysis over all the frames.

6.1 Synthetic Data

In this section, we compare our sequential methods BA-FEM and EM-FEM to state-of-the-art techniques, for both sparse and dense observations. In addition, we also analyze the two proposed strategies to compute stretching modes. In all cases, the synthetic deformation is observed by an orthographic camera.

6.1.1 Elastic Sequences

First, we propose two synthetic sequences of a deforming elastic plate with irregular (Syn. 1) and regular (Syn. 2) discretization, respectively. In both cases, we have considered a nonlinear hyperelastic Yeoh material [64] to model large deformations, but just consider a different discretization. The Yeoh model for incompressible materials is a function only of the first strain invariant I_1, and its deformation energy is defined as $W = C_{10}(I_1 - 3) + C_{20}(I_1 - 3)^2 + C_{30}(I_1 - 3)^3$, where the parameters $\{C_{10}, C_{20}, C_{30}\}$ define the material properties. Particularly, we set $C_{10} = 10$MPa, $C_{20} = -0.01$MPa and $C_{30} = 10$MPa for these experiments. We refer the reader to previous paper for further details. Both sequences were generated using Abaqus for which the material properties, boundary conditions and nodal forces are required a priori to compute the deformation. In contrast, our methods based on linear elasticity do not require any prior knowledge to constrain the deformation.

We use our BA-FEM and EM-FEM algorithms for this test and present a comparison against state-of-the-art NRS*f*M methods, both batch and sequential algorithms. For batch methods, we consider EM-LDS [57], MP [39], PTA [8], KSTA [25], the block matrix approach SPM [15], and EM-PND [28]. We also compare with the sequential BA based on implicit models SBA [40]. The parameters of these methods were set in accordance with their original papers. Table 1 summarizes the comparison of the results for each method, while the 3D reconstruction of a typical frame for some methods are displayed in Fig. 7. Figure 8 displays the effect on 3D errors of varying the rank of the subspace, showing the consistent reduction of the error as more modes r are considered. We do not address the selection of r in this paper, since our results suggest that the proposed methods are not extremely sensitive to this choice.

We also evaluate the sensibility of including nonrigid motion in the initial frames that affects to the initialization and hence to the shape basis computation. To this end, we quantify this nonrigid motion by the metric $\|\mathbf{U}_{nonrigid}\|_{\mathcal{F}}$, where $\mathbf{U}_{nonrigid}$ represents the nonrigid 3D displacement of the shape in these frames. Even though in both experiments, the deformation is not produced concerning a mean shape, causing rest shapes more inaccurate, our methods produce accurate results and even outperform the rest of techniques evaluated. Figure 9 shows this effect on 3D errors varying the level of deformation in the initial frames. We can conclude that our methods outperform the state-of-the-art methods in terms of accuracy, with the additional advantage of being sequential, allowing us process frames in an on-line fashion. In addition, we achieve with an unoptimized Matlab code a frame rate of about 3 fps when dealing with a model of 81 points using 10 modes, and hence these results could still be significantly speeded up. We consider our method may potentially achieve real-time performance at frame rate with an efficient implementation.

Parameter selection The contribution of each smoothness term in the cost function of Eq. (19) can be controlled by means of the weights $\lambda_\gamma = 0.15$, $\lambda_t = 0.03$, and $\lambda_q = 0.03$. We have tuned these parameters on these sequences, and used the same values for the rest of experiments. Yet, these parameters do not need to be carefully tuned. For instance, we test the 3D reconstruction error over the two elastic sequences after changing these weights by a $\div/\times 10$ their original value. We obtain an error of 3.90(10)–4.01(10) for Syn.1 and 3.06(10)–3.17(10) for Syn.2. Observe that the reconstruction results barely change.

6.1.2 Dense Face Sequences

We now present results applying boundary conditions, i.e., rigid points, to show how our model can easily incorporate these constraints if they are available. We use only these priors to obtain the shape basis, managing both kinds of points—rigid and nonrigid points—in a single framework, unlike [17,

Table 1 Quantitative comparison for synthetic sequences

Data	Batch Methods Met.						Sequential Methods		
	EM-LDS	PTA	MP	KSTA	SPM	EM-PND	SBA	BA-FEM	EM-FEM
Syn. 1	11.12(2)	9.25(2)	12.42(2)	5.56(2)	–	–	14.03(16)	3.89(10)	3.98(10)
Syn. 2	10.92(2)	11.81(5)	18.84(2)	–	–	7.60	20.90(8)	3.04(10)	3.01(10)

We show $e_{3\mathcal{D}}$[%] for EM-LDS [57], PTA [8], MP [39], KSTA [25], SPM [15], EM-PND [28] and SBA [40]; and for our methods BA-FEM and EM-FEM. For state-of-the-art methods, we have selected the rank of the subspace in the basis (in brackets) that gave the lowest $e_{3\mathcal{D}}$ error. For our methods, we provide results with 10 mode shapes. When an algorithm does not converge, its result is denoted as "–"

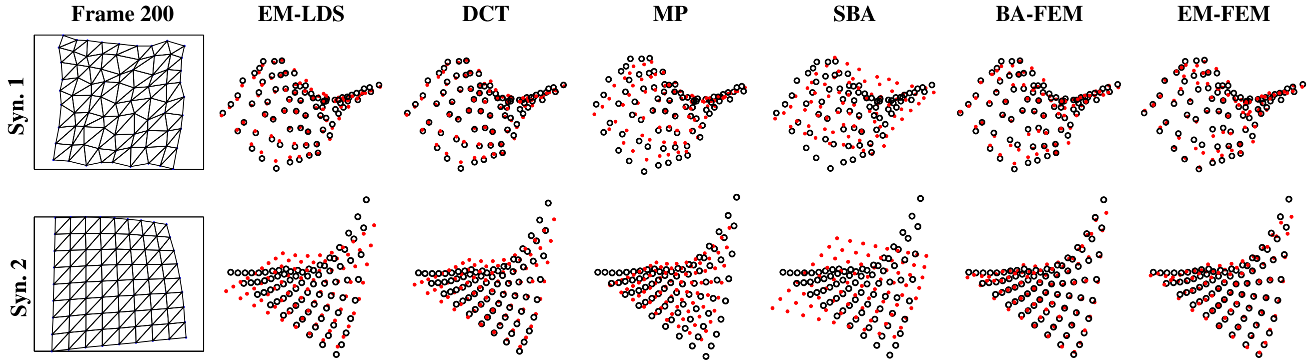


Fig. 7 Synthetic sequences. Comparing results of our BA-FEM and EM-FEM algorithms with respect to EM-LDS [57], PTA [8], MP [39] and SBA [40] for frame #200. Reconstructed 3D shape and the 3D ground truth are shown with red dots and black circles, respectively. *Top* Results for Syn. 1 sequence. *Bottom* Results for Syn. 2 sequence

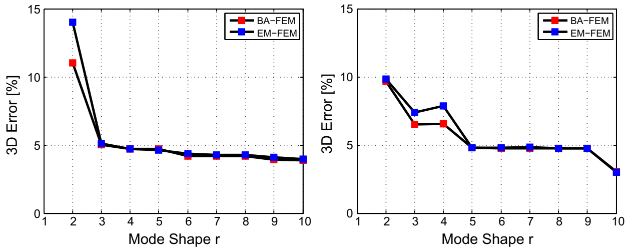


Fig. 8 Mean normalized error $e_{3\mathcal{D}}$ with varying number of mode shapes for synthetic sequences. *Left* Results for Syn. 1 sequence. *Right* Results for Syn. 2 sequence (Color figure online)

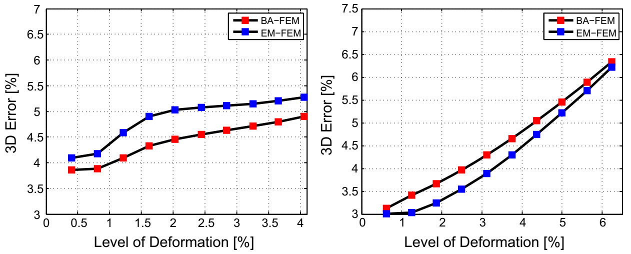


Fig. 9 Mean normalized error $e_{3\mathcal{D}}$ **with varying level of deformation in initial frames** for synthetic sequences. *Left* Results for Syn. 1 sequence. *Right* Results for Syn. 2 sequence (Color figure online)

30] where both kinds of points are independently considered. Note that these constraints do not fix the absolute location of these points in the space, but produce sensitive changes in the mode shape basis. We only use these priors in these sequences to show the qualities of our shape basis estimation.

We use both 3 and 4 synthetic dense face sequences with $p = 28{,}887$ points as proposed in [21] that we denominate Dense Face 1 and Dense Face 2, respectively. Both sequences are challenging owing to the density of the data and the strong deformations that combine bending and stretching, and they are similar with the exception of the camera motion. Thanks to the ability of our method to handle boundary point priors, we can compute just the first $r = 4$ modes that combine both stretching and bending deformations, and experimentally show that they are sufficient to encode the face deformations. In this case, boundary conditions correspond to face points that have a null displacement with respect to the shape at rest. They are selected by means of a connectivity analysis. We show the error $e_{3\mathcal{D}}$ and a quantitative comparison with respect to batch and sequential state-of-the-art methods in Table 2, applying our algorithms. It is worth pointing out that we are not able to provide results for KSTA [25], SPM [15], and EM-PND [28], as they could not handle the large dimensionality of the problem. Our sequential approaches outperform in average the batch PTA [8] and MP [39] approaches, and the sequential approach SBA [40], but they provide an error higher than that of the variational batch method VNR [21], where all frames need to be available in advance. We show our 3D reconstruction for few frames in Fig. 10, using the BA-FEM algorithm.

Table 2 Quantitative comparison for dense face sequences

Data	Batch Methods Met.						Sequential Methods		
	PTA	MP	KSTA	SPM	EM-PND	VNR	SBA	BA-FEM	EM-FEM
Dense Face 1	4.50(4)	5.13(6)	–	–	–	2.60(9)	4.35(4)	4.65(4)	4.53(4)
Dense Face 2	6.61(4)	5.81(6)	–	–	–	2.81(9)	7.03(4)	4.92(4)	4.86(4)

We show $e_{3\mathcal{D}}$[%] for batch methods PTA [8], MP [39], KSTA [25], SPM [15], EM-PND [28] and VNR [21]; for the sequential method SBA [40]; and for our sequential methods BA-FEM and EM-FEM. In all cases, we have selected the number of shapes in the basis (in brackets) that gave the lowest $e_{3\mathcal{D}}$ error. When an algorithm does not handle this experiment, its result is denoted as "–"

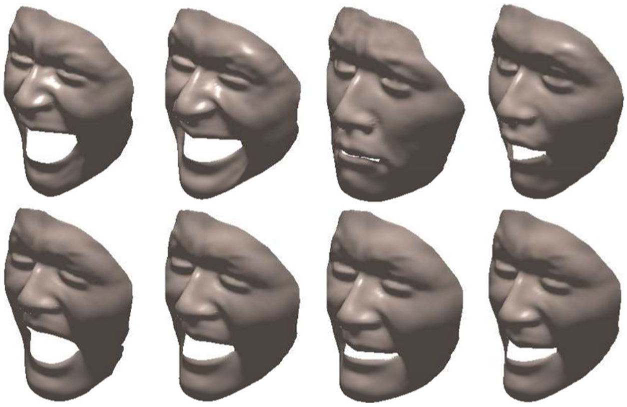


Fig. 10 Dense Face 1 sequence. Reconstruction of the dense face for selected frames: #30, #40, #79, and #95. *Top* Ground truth 3D shapes. *Bottom* Our dense 3D reconstruction (Color figure online)

6.1.3 Efficient Dense Mode Shape Computation

Finally, we propose a synthetic sequence of 109 frames where we simulate an elastic ribbon deformation of $p = 275$ points with a nonlinear Yeoh hyperelastic material [64]. A dense shape basis is computed applying both the frequency-based and the coarse-to-fine approaches. In both cases, we use a low-dimensional mesh of $q = 78$ points and then obtain a mesh of $p = 275$ points. In order to estimate the nonrigid shape for this challenging 43 % stretching deformation, we use our EM-FEM algorithm—similar results are obtained by BA-FEM—and achieve the following performance $e_{3\mathcal{D}}$: (i) for the frequency-based method, we obtain 3.15 % for 5 modes, 1.08 % for 20 modes, and 0.86 % for 40 modes; (ii) for the coarse-to-fine approach, we obtain 3.11 %, 1.57 % and 0.84 % for 5, 20, and 40 stretching modes, respectively. Figure 11 shows our 3D reconstructions with $r = 40$ for a few selected frames including where the stretching is maximum, as well as a qualitative comparison with respect to ground truth. We conclude both methods provide similar 3D reconstructions, but note that the frequency-based method is more sensitive—due to the rigid factorization—and slightly more expensive. For this reason, we propose always to use the coarse- to-fine approach.

6.2 Motion Capture Data

In this section, we quantitatively evaluate our approaches using several existing datasets of motion capture (MoCap) to

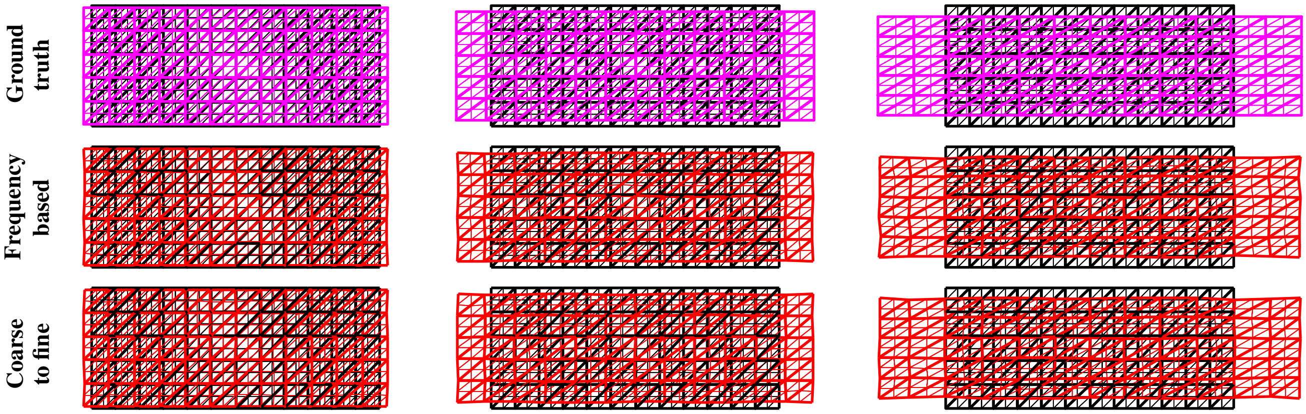


Fig. 11 Stretching ribbon sequence. 3D reconstruction of the 275 point ribbon at frames #20, #60, and maximum deformation at #109. Shape at rest is displayed behind with a black mesh. *Top* Ground truth deformation with a magenta mesh. *Middle* 3D reconstruction using the frequency-based method. *Bottom* 3D reconstruction using the coarse-to-fine approach. In both cases, the sparse mesh is shown in thick lines. Best viewed in color (Color figure online)

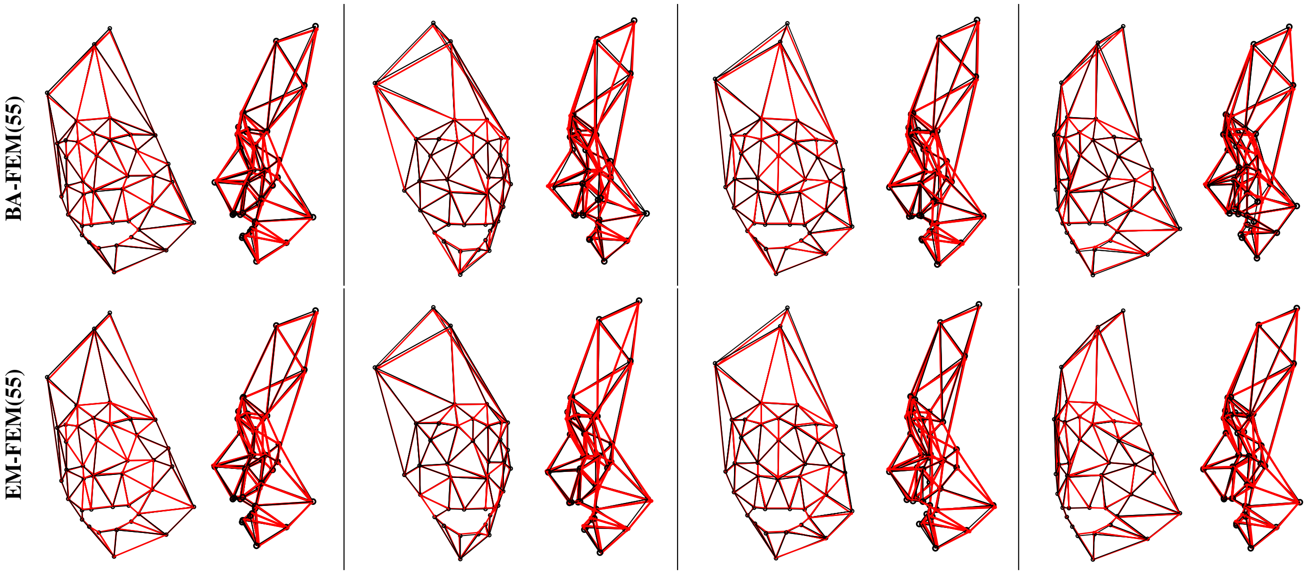


Fig. 12 Facial MoCap sequence. Original viewpoint and side views for selected frames #71, #188, #252, and #288 with ground truth (*black mesh*) and 3D reconstruction (*red mesh*) using r=55. *Top* BA-FEM algorithm. *Bottom* EM-FEM algorithm. Best viewed in color (Color figure online)

Table 3 Quantitative comparison for MoCap sequences using sequential methods

Data	Face Met. 40 points			Sparse Flag 594 points			Dense Flag 9,622 points				
	ε	$e_{3\mathcal{D}}[\%]$	in/op (sec)	ε	$e_{3\mathcal{D}}[\%]$	in/op (sec)	ε	$e_{3\mathcal{D}}[\%]$	CtF	in/op (sec)	CtF
SBA	0	2.90(48*)†	0.15/1.00†	0	7.10(114*)	0.58/82.32	0	13.48(114*)	–	25.67/895	-
	1	6.38(114*)	0.15/16.98	1	7.48(360*)	0.58/870	1	–	–	–	–
BA-FEM	0	3.43(15)	0.29/0.38	0	3.72(10)	19.50/1.96	0	3.50(10)	3.67(10)	300/75	44.62/73
	0	3.92(55)	0.29/1.28	0	3.49(40)	19.50/24.83	0	3.29(25)	3.96(25)	300/186	44.62/182
	1	3.84(15)	0.29/0.42	1	3.73(10)	19.50/1.97	1	3.50(10)	3.68(10)	300/72	44.62/77
	1	4.30(55)	0.29/1.52	1	3.56(40)	19.50/25.39	1	3.25(25)	3.96(25)	300/187	44.62/188
EM-FEM	0	3.36(15)	0.29/0.10	0	3.28(10)	19.50/1.53	0	2.94(10)	3.41(10)	300/75	44.62/62
	0	3.05(55)	0.29/0.19	0	2.81(40)	19.50/2.28	0	2.50(25)	3.08(25)	300/69	44.62/68
	1	3.39(15)	0.29/0.10	1	3.32(10)	19.50/1.53	1	2.96(10)	3.51(10)	300/75	44.62/62
	1	3.49(55)	0.29/0.20	1	2.92(40)	19.50/2.30	1	2.52(25)	3.30(25)	300/70	44.62/69

We show error $e_{3\mathcal{D}}[\%]$ for SBA [40], and for both BA-FEM and EM-FEM methods. In all cases, we show in brackets the number of shapes in the basis. We show computation time for initialization in and optimization process per frame op. We also show the results for the noise-free case $\varepsilon = 0$ and for the noise case $\varepsilon = 1$. Finally, we show the results applying our coarse-to fine-approach (CtF) for dense cases. †: value reproduced from [40]. $*$: SBA reports the rank r, we detail $3r$ in brackets because it is equivalent to the number of weights in both BA-FEM and EM-FEM algorithms. In all cases, $\mathcal{W} = 5$

present an analysis of scalability with respect to sequential state-of-the-art NRSfM. We consider our approaches BA-FEM and EM-FEM, and the sequential method SBA [40]. Moreover, the performance of our methods is also analyzed when adding noise. We report results when zero-mean Gaussian noise with standard deviation $\sigma = \frac{\varepsilon}{100}\kappa$ is added to every point in the mesh following [39], where ε is the noise percentage, and κ is the maximum distance of an image point to the centroid of all the points. For each method, we use exactly the same $\mathbf{W}_{1:nr}$ to compute the shape at rest, i.e., the same initialization. Finally, we also report a comparison w.r.t. batch baselines.

6.2.1 Facial MoCap Sequence

First, we use 316 frames from the facial MoCap sequence [57] with ground truth, where a subject is talking and moving his head while wearing 40 markers. With this sequence, we show the performance of our mesh-based approach using a reduced set of points. We show quantitative results for this

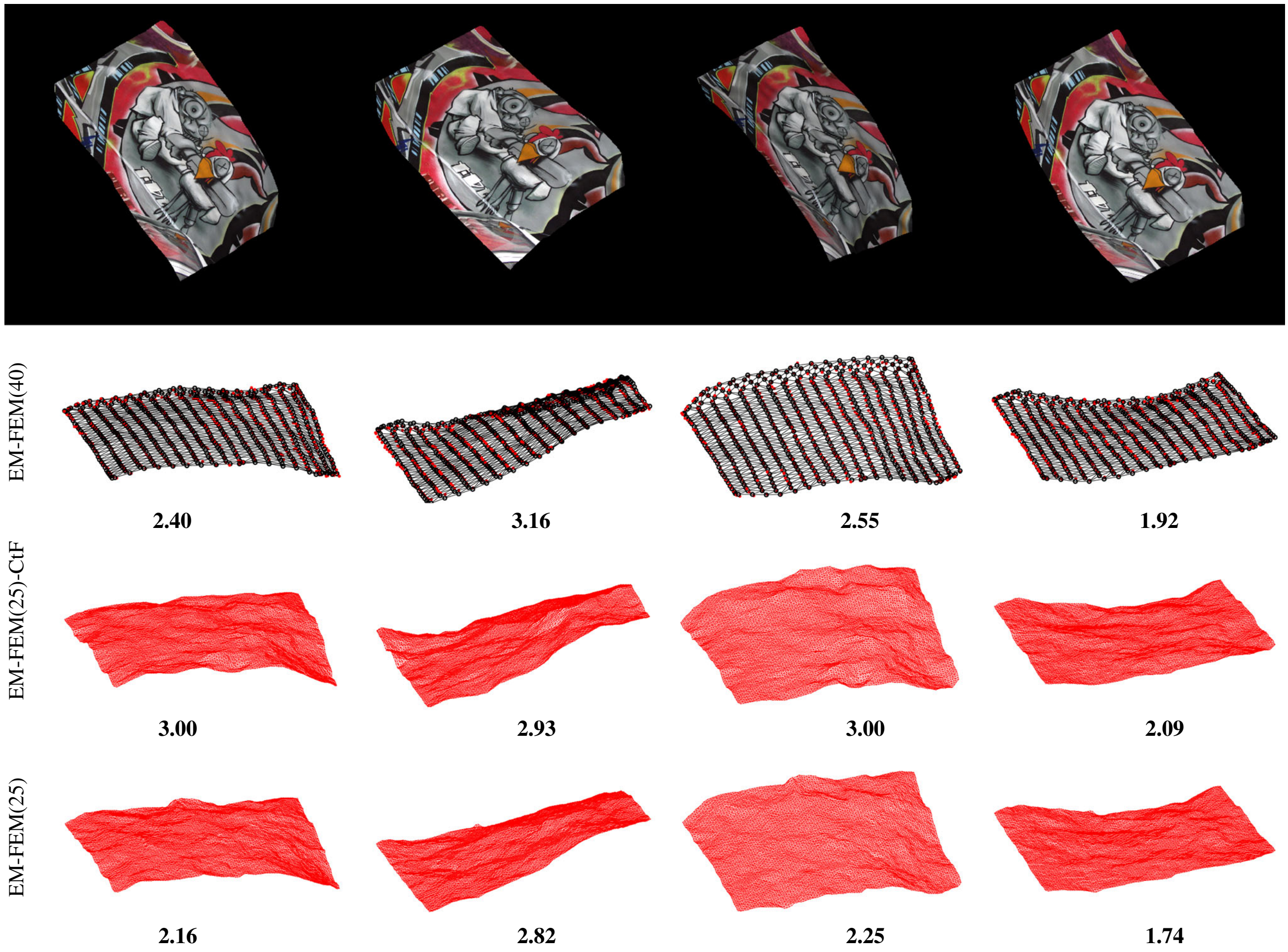


Fig. 13 Flag MoCap sequence. 3D Reconstruction for a few frames with *red dots*. We display a few examples in Table 3 using the EM-FEM algorithm. We also show the corresponding 3D reconstruction error $e_{3\mathcal{D}}[\%]$. *First row* Selected frames #10, #20, #32, and #45. *Second row* Sparse 594 points flag. Ground truth is overlaid with a *black mesh* and *black circles*. *Third row* Dense 9,622 points flag using the coarse-to-fine approach (CtF). *Fourth row* Dense 9,622 points flag without CtF. Please zoom in to the electronic images for a detailed view (Color figure online)

sequence in Table 3 and qualitative results for both BA-FEM and EM-FEM in Fig. 12. We can achieve similar accuracy at a lower computational cost with respect to SBA [40]. However, when the measurements are noisy, the performance of SBA is worse since the shape basis is estimated on-the-fly while our algorithms provide a similar solution with respect to the noise-free case. It is worth noting that with the same basis, our EM-FEM algorithm is faster and more accurate compared to the BA-FEM algorithm, since this deformation is well modeled by Gaussian priors. However, our methods obtain less performance than batch methods in terms of error $e_{3\mathcal{D}}[\%]$. Their results are 2.74(5), 2.69(3), 2.12(4), 1.82(7), and 1.40 for the methods MP [39], PTA [8], KSTA [25], SPM [15], and EM-PND [28], respectively. This is due to the low resolution of the object, which is insufficient to establish the FEM constraints. Despite this, our methods can obtain a competitive solution for this sequence w.r.t. batch ones, without requiring all frames to be available in advance.

6.2.2 Flag MoCap Sequence

We now evaluate our approaches using a challenging dense sequence [22] with $p = 9{,}622$, corresponding to a fabric waving in the wind. To validate the scalability of our method, we also propose a $p = 594$ sparse sequence version of the dense flag sequence, a result of the subsample process used to apply the coarse- to-fine approach (see Fig. 6). We have included a few initial frames corresponding to the camera observing the rigid shape. As the deformation contains little stretching, the first bending mode shapes can accurately encode the deforming scene.

We show quantitative 3D reconstructions for both sparse and dense flag sequences in Table 3 and qualitative results in

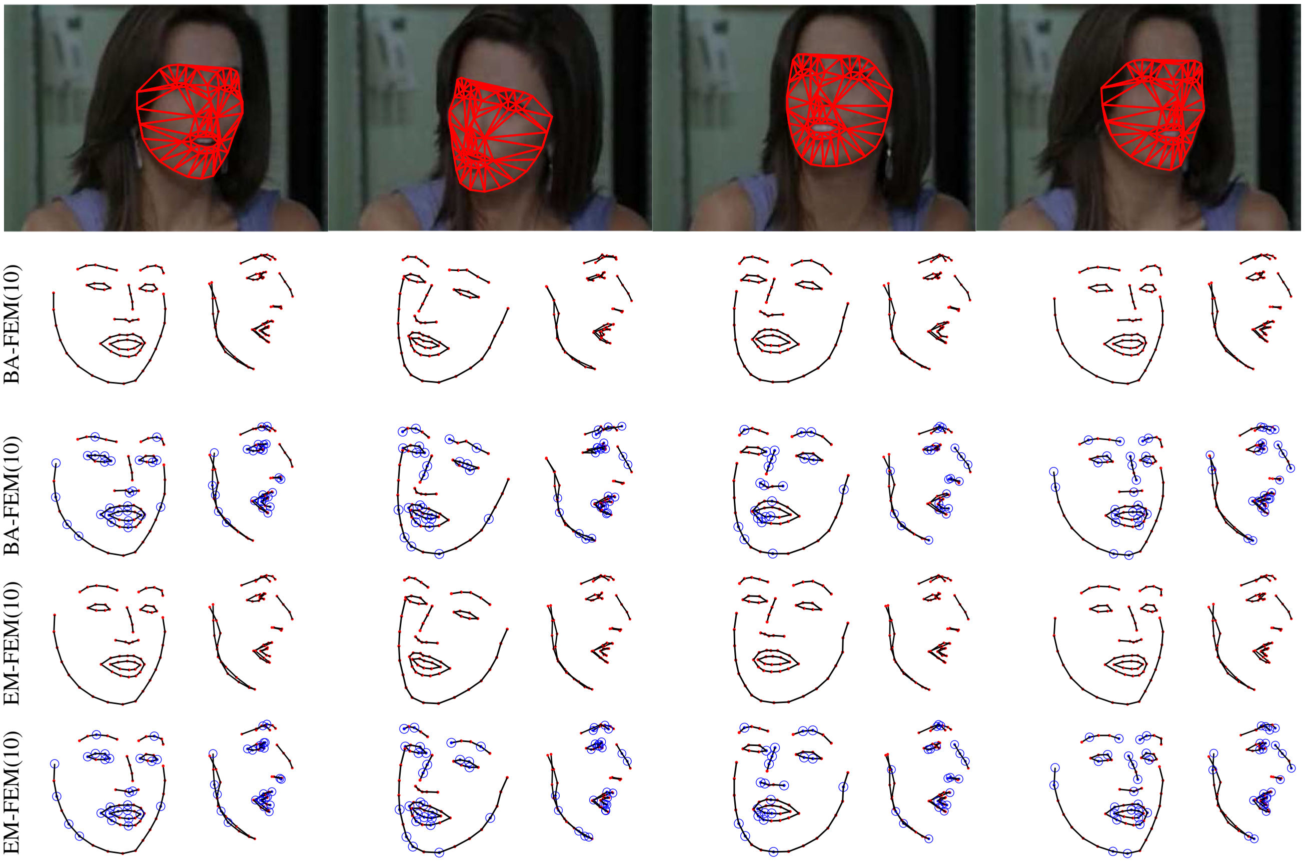


Fig. 14 Actress sequence. Selected frames #31, #48, #84 and #102. *First row* Image frame sequence with 3D reprojected mesh. *Second and third rows* Original viewpoint and side views of the 3D reconstruction in the cases of full data and 40 % missing data, respectively, using the BA-FEM algorithm. *Fourth and fifth row* The same views applying the EM-FEM algorithm in the cases of full data and 40 % missing data, respectively. Visible points are displayed in *red dots*, and occluded points in *blue circles* (Colorfigure online)

Fig. 13. Both the BA-FEM and EM-FEM algorithms consistently outperform the SBA [40] method in terms of accuracy and efficiency from sparse to dense case and when applying noise. Although SBA [40] estimates the mode shapes on-the-fly—with shorter initialization computation time only for shape at rest—it is not able to overcome the mode shapes for this sequence. Note that the three methods use exactly the same initialization, although both BA-FEM and EM-FEM exploit the shape at rest to compute a mode shape basis. Our sequential methods are also more accurate than the batch methods MP [39], PTA [8], EM-PND [28], and KSTA [25] which for the sparse case obtain an error $e_{3D}[\%]$ of 16.02(2), 14.11(2), 8.65(2), and 8.61, respectively. Regarding the dense case, we obtain an error $e_{3D}[\%]$ of 21.94(2), 14.83(2), and 9.64(4) for the methods MP [39], PTA [8], and KSTA [25], respectively. SPM [15] and EM-PND [28] did not manage to process this sequence due to their nonscalable nature.

In both cases, the error $e_{3\mathcal{D}}$ when applying EM-FEM is smaller than that of the BA-FEM algorithm. For this sequence, the Gaussian prior is more accurate and produces better solutions. Note that the initialization computation time for the sparse case is the same using both approaches since we do not use the coarse-to-fine approach. When we apply this algorithm for the dense case, the increase in the computation time is negligible—0.03 sec—compared to the sparse case, and it is dominated by the rigid factorization step with 25.67sec, similar to SBA [40]. However, the computation basis for the dense case without applying the coarse-to-fine approach is more expensive. It is worth noting that

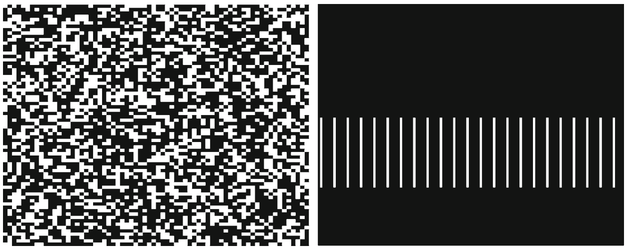


Fig. 15 Missing data pattern. Patterns used in the actress and paper bending sequences, respectively. Each row is a nonrigid frame and each column is a point track. Points in *black* and in *white* are marked as visible and occluded, respectively. *Left* 40 % random missing data. *Right* 22 % structured missing data

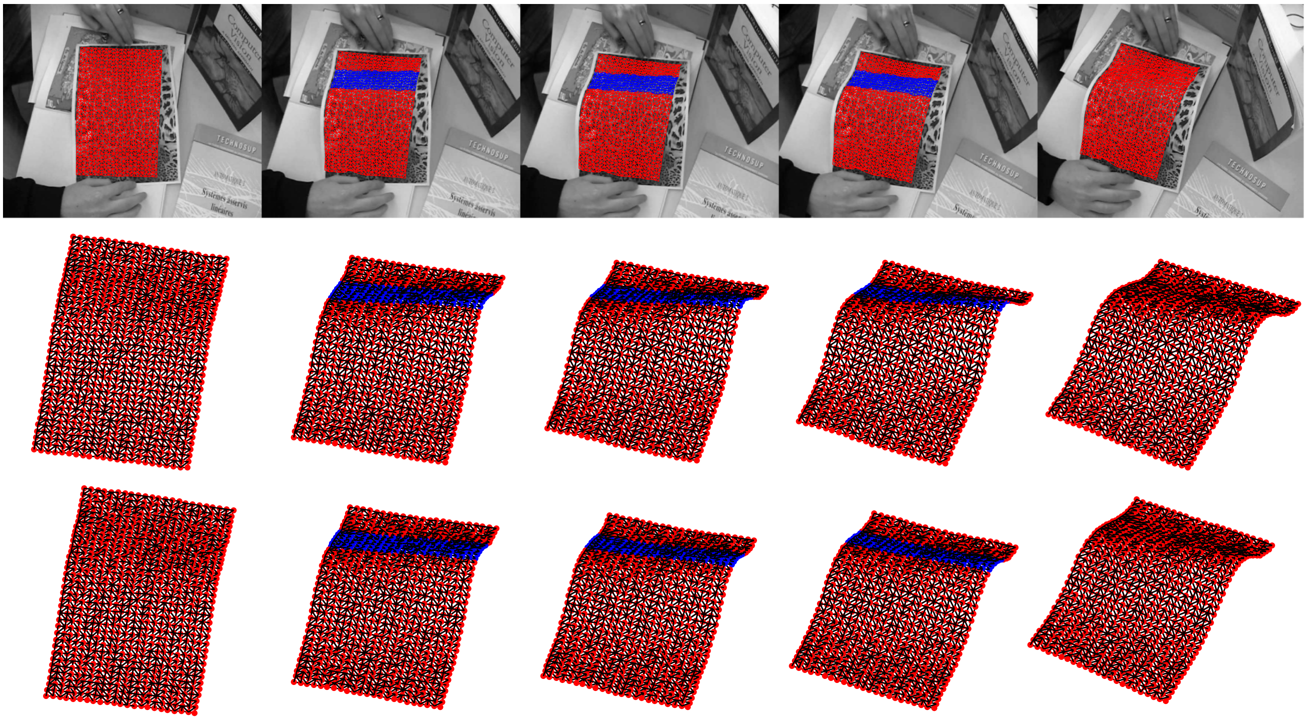


Fig. 16 Paper Bending Sequence. We show the 3D reconstruction of a deformed scene for a few selected frames: #25, #50, #60, #75, and #100. Up to 22 % structured occlusion is introduced (*blue points*) to show the robustness with respect to self-occlusion. *Top* Frames with 2D tracks. *Middle* General view using BA-FEM. *Bottom* General view using EM-FEM. (Color figure online)

the reconstructions applying the coarse-to-fine approximation are less accurate than the standard method without significantly degrading the estimation, and the shape basis computation is much more efficient. For the sequential estimation, we obtain more efficient results using EM-FEM, with a better scalability in the number of modes. Concerning the sensitive when the rank is selected, we also observe a stability in our estimates with respect to this parameter, by obtaining more accurate solutions but at higher computational cost. Our conclusions can be extended to the noise case, showing the robustness of our methods.

We can conclude that our methods outperform the sequential state-of-the-art methods in terms of accuracy and efficiency. Although our method is implemented in unoptimized Matlab code without parallelization over a commodity computer Intel core i7@2.67 GHz, the results show a low computational cost per frame and the results can still be significantly speeded up.

6.3 Real Images

In this section, we evaluate our methods qualitatively using several existing datasets. We show the scalability of our methods for sparse to dense sequences of real images. Moreover, we show the robustness of our methods in the cases of structured and random missing data.

6.3.1 Actress Sequence

First, we use the sequence and tracks provided by [9] to report qualitative results for the actress sequence, which consists of 102 frames where an actress is talking and moving her head. We use a rigid model for the first 30 frames to compute the shape at rest for both approaches, similarly to [40]. In Fig. 14, we show our estimated 3D reconstruction applying both BA-FEM and EM-FEM algorithms with $r = 10$ stretching mode shapes. Our results are comparatively similar to those reported by [40]. Moreover, we provide results in the case of corrupted observations. The missing data were generated by randomly deleting entries in the 2D input tracks. We report comparatively similar results to those of the full data case applying 40 % of missing data for both the BA-FEM and EM-FEM algorithms, showing their robustness to artifacts. We show the 40 % missing data mask in Fig. 15(left).

6.3.2 Paper Bending Sequence

In this case, we use the first 100 frames of a paper bending sequence proposed in [9] to provide a qualitative evaluation of our methods with respect to a structured missing data pattern. We use the sparse tracking of 828 points obtained by dense tracking data reported in [22]. We process the sequence

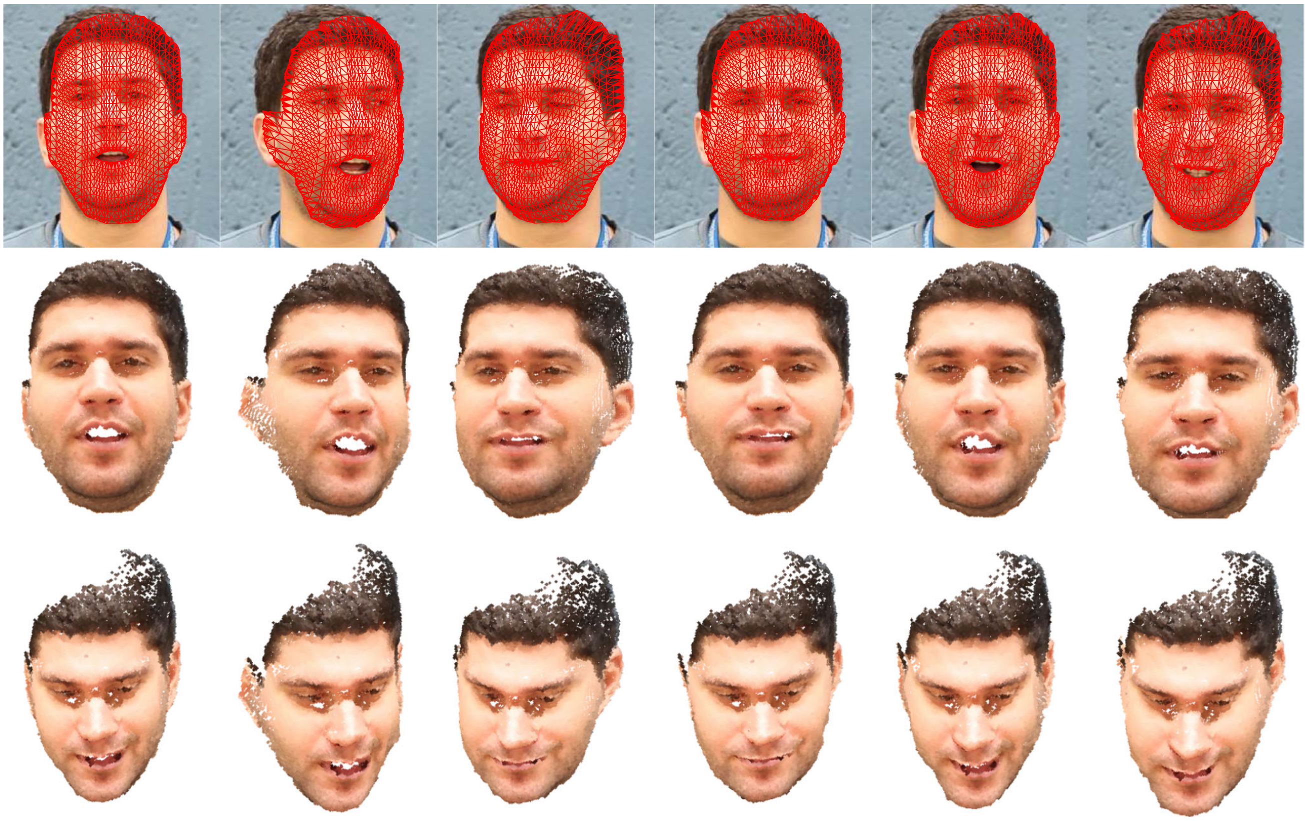


Fig. 17 Niko's sequence: dense 3D point cloud estimation. *Top* Selected frames #17, #43, #55, #75, #100, and #121 with 3D reprojected mesh. We display the mesh we use to obtain the mode shape basis. *Middle* Textured original viewpoint. *Bottom* Textured general view

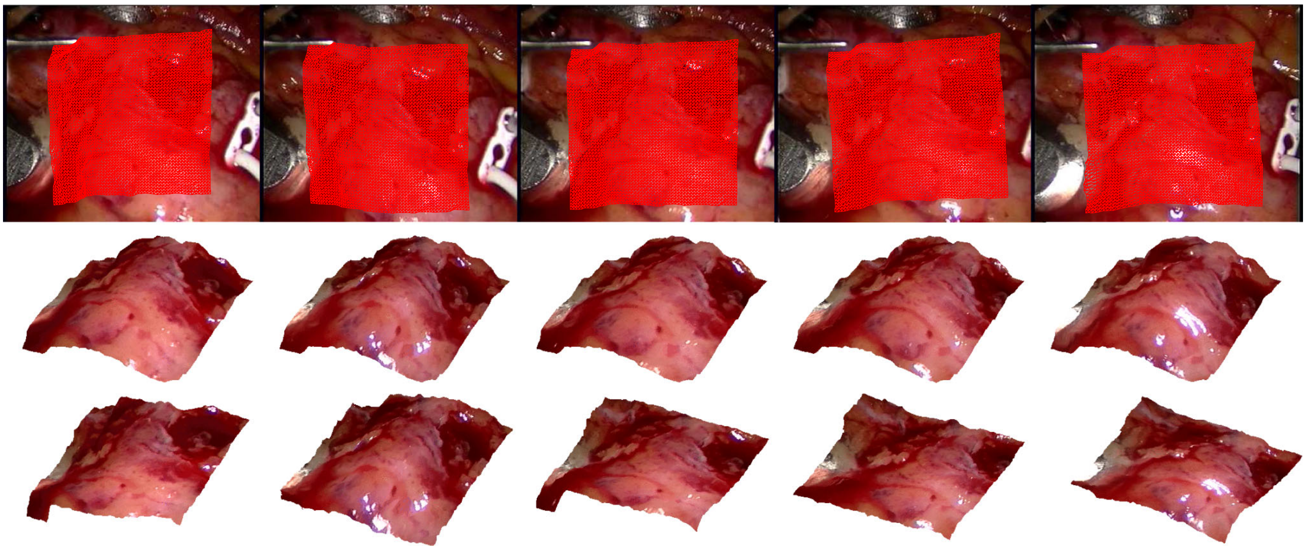


Fig. 18 Heart sequence. 3D reconstruction of a beating heart during bypass surgery. *Top* Selected frames #9, #24, #32, #58, and #74 with 3D reprojected mesh. *Middle* Textured general view subtracting the rigid motion for our BA-FEM method. *Bottom* Same views using KSTA [25]

using $r=$ 10 bending mode shapes and 5 frames to compute the shape at rest. Between frames #48 and #76, a 22 % band of missing data is introduced simulating strong self-occlusion. Our performance is close to the full data case for both BA-FEM and EM-FEM, and our 3D reconstruction does not degrade significantly. We show both 3D reconstructions with missing data for this sequence in Fig. 16, the 22 % structured missing data mask in Fig. 15(right).

6.3.3 Niko's Sequence

We now evaluate our approach using dense 28,332 tracks provided by [22] to show a qualitative evaluation for the face sequence compared to [21], where Niko is performing natural expressions and moving his head. Since we obtain similar results with both algorithms, in Fig. 17, we show a few frames with our 3D reconstruction using our EM-FEM algorithm with $r = 30$ stretching mode shapes. We compute mode shapes with $q = 1{,}442$ points and applying a coarse-to-fine approach to $p = 28{,}332$ points (see Fig. 5).

6.3.4 Heart Sequence

Finally, we process a challenging *Heart* sequence of a beating heart captured during bypass surgery.[3] This shows the generality of our approach to retrieve the 3D reconstruction of extensible objects. In this case, we track 3,024 points computed by [22] without any sub-sampling technique. Figure 18 shows some images and our 3D reconstruction using our BA-FEM algorithm with $r = 30$ stretching mode shapes (similar results are provided by EM-FEM). Since 3D ground truth is not available for this experiment, we provide a qualitative comparison with respect to KSTA [25] using a rank in the basis of 5, which obtained one of better performance in quantitative experiments of the previous subsection. Even though this method is known to be very accurate, the estimated 3D shapes do not seem very realistically plausible being almost planar for some frames.

7 Conclusion

We have proposed two sequential algorithms to estimate camera motion and nonrigid 3D shape from monocular sequences in close to real-time operation. Our systems work with significant missing data, with both elastic or isometric deformations and, crucially, without a training step. Our methods can be used to model either sparse or dense data and are robust to noisy measurements. We use a thin-plate model with unknown material properties to code the behavior of a nonrigid 3D object by applying a force balance equation. This equation is directly solved by FEM modal analysis for sparse meshes, and we also propose two efficient methods to solve the equation for dense meshes. The resulting physics-based shape basis is combined with temporal and spatial smoothness constraints without using restrictive additional distance constraints such as inextensibility, allowing us to model elastic deformations. Both approaches provide a competitive solution in terms of accuracy versus per-frame computation time. Our claims have been experimentally validated for both synthetic and real sequences showing a performance better than or comparable to state-of-the-art methods, with the additional advantage that our methods are sequential, accurate, and scalable.

Our future work is to pursue the goal of merging feature tracking and outliers detection to provide a unified framework. Additionally, we plan to apply our methods to the problem of 3D reconstruction from endoscopic videos, where accurate FEM biomechanical models for internal organ tissues are available. We expect that our method will be able to exploit the rich priors available and provide new avenues of research for the challenging use of NRS*f*M in minimally invasive surgery.

Acknowledgements This work was partly supported by the MINECO projects RT-SLAM DPI2015-67275-P, RobInstruct TIN2014-58178-R and Keratocono DPI2014-54981-R; by the SecondHands project funded by the EU Horizon 2020 Research and Innovation programme under grant agreement No 643950; by the ERA-net CHISTERA projects VISEN PCIN-2013-047 and I-DRESS PCIN-2015-147; and by a scholarship FPU12/04886 from the Spanish MECD. We thank R. Garg for his optical flow data and M. A. Ariza for the stretching data.

8 Rotation Update on the Stiefel Manifold

The camera rotation $\mathbf{Q}_i^t$ is subject to orthonormality constraints, and a closed-form update is not possible. The rotation matrix lies exactly on a smooth manifold based on the orthogonal group $SO(3)$, where it is possible to generalize a Riemannian-Newton algorithm [19,49]. We define the problem as that of minimizing the function $\mathcal{B}\left(\mathbf{Q}_i^t\right)$, where $\mathbf{Q}_i^t$ is constrained to the set of matrices such that $\mathbf{Q}_i^{t\top}\mathbf{Q}_i^t = \mathbf{I}$, i.e., a Stiefel matrix. In this work, we use the Riemannian manifold optimization to update the rotation matrices. First, we rewrite the expected negative log-likelihood function Eq. (21) dropping the dependence on σ^2 and $\mathcal{B}\left(\mathbf{Q}_i^t\right)$ can be expressed as:

$$\underset{\mathbf{Q}_i^t \in SO(3)}{\arg\min} \sum_{i=f-\mathcal{W}+1}^{f} \sum_{\varrho\in\mathcal{V}} \mathbb{E}\left[\|\mathbf{w}_{i\varrho} - \mathbf{\Pi}\mathbf{Q}_i^t \tilde{\mathcal{S}}_\varrho \tilde{\boldsymbol{\gamma}}_i - \mathbf{t}_i\|_{\mathcal{F}}^2\right] \tag{23}$$

where $\mathbf{Q}_i^t \in SO(3)$ and its tangent $\Delta_{Q_i^t} \in T_{Q^t}(SO(3))$ can be expressed as $\Delta_{Q_i^t} = \mathbf{Q}_i^t[\boldsymbol{\delta}]_\times$ with $[\boldsymbol{\delta}]_\times$ being the skew-symmetric matrix. On $SO(3)$, the geodesic at $\mathbf{Q}_i^t$ in the tangent direction can be expressed by means of the Rodrigues' rotation formula:

$$\mathbf{Q}^{t+1}(\hat{\boldsymbol{\delta}}, \alpha) = \mathbf{Q}^t\left(\mathbf{I}_3 + \left[\hat{\boldsymbol{\delta}}\right]_\times \sin(\alpha) + \left[\hat{\boldsymbol{\delta}}\right]_\times^2 (1 - \cos(\alpha))\right) \tag{24}$$

[3] Video available from: http://hamlyn.doc.ic.ac.uk/vision.

where $[\boldsymbol{\delta}]_\times \in \mathfrak{so}(3)$ is the Lie algebra of $SO(3)$ the group and $[\boldsymbol{\delta}]_\times = \alpha \left[\hat{\boldsymbol{\delta}}\right]_\times$. This explicit formula for geodesics is necessary for computing the gradient $d\mathcal{B}(\Delta_{Q_i^t})$ and the Hessian $Hess\,\mathcal{B}(\Delta_{Q_i^t}, \Delta_{Q_i^t})$ of the cost function Eq. (23) along the geodesics on the manifold. Given the previous definition, we can obtain both the gradient and Hessian in a tangent direction $\Delta_{Q_i^t}$ as:

$$\begin{aligned} d\,\mathcal{B}(\Delta_{Q_i^t}) &\equiv \left.\frac{d\,\mathcal{B}\left(\mathbf{Q}_i^t(\alpha)\right)}{d\alpha}\right|_{\alpha=0} \\ &= \left(\mathbf{R}_i^t \sum_{\varrho\in\mathcal{V}} \left(\tilde{\mathcal{S}}_\varrho \tilde{\boldsymbol{\phi}}_i \tilde{\mathcal{S}}_\varrho^\top\right) - \sum_{\varrho\in\mathcal{V}} \left(\left(\mathbf{w}_{i\varrho} - \mathbf{t}_i\right) \tilde{\boldsymbol{\mu}}_i^\top \tilde{\mathcal{S}}_\varrho^\top\right)\right) \Delta_{R_i^t}^\top \qquad (25) \\ Hess\,\mathcal{B}(\Delta_{Q_i^t}, \Delta_{Q_i^t}) &\equiv \left.\frac{d^2\,\mathcal{B}\left(\mathbf{Q}_i^t(\alpha)\right)}{d\alpha^2}\right|_{\alpha=0} \\ &= \left(\mathbf{R}_i^t \sum_{\varrho\in\mathcal{V}} \left(\tilde{\mathcal{S}}_\varrho \tilde{\boldsymbol{\phi}}_i \tilde{\mathcal{S}}_\varrho^\top\right) - \sum_{\varrho\in\mathcal{M}} \left(\left(\mathbf{w}_{i\varrho} - \mathbf{t}_i\right) \tilde{\boldsymbol{\mu}}_i^\top \tilde{\mathcal{S}}_\varrho^\top\right)\right) \Delta_{Q_i^t}^\top \mathbf{Q}_i^t \Delta_{R_i^t}^\top \\ &\quad + \Delta_{R_i^t} \sum_{\varrho\in\mathcal{V}} \left(\tilde{\mathcal{S}}_\varrho \tilde{\boldsymbol{\phi}}_i \tilde{\mathcal{S}}_\varrho^\top\right) \Delta_{R_i^t}^\top \qquad (26) \end{aligned}$$

where $\Delta_{R_i^t} = \boldsymbol{\Pi}\Delta_{Q_i^t}$ are the first two rows of a full tangent vector. The expectations are $\tilde{\boldsymbol{\mu}}_i = \mathbb{E}\left[\tilde{\boldsymbol{\gamma}}_i\right]$ and $\tilde{\boldsymbol{\phi}}_i = \mathbb{E}\left[\tilde{\boldsymbol{\gamma}}_i \tilde{\boldsymbol{\gamma}}_i^\top\right]$. The Hessian can be obtained by polarizing $Hess\,\mathcal{B}(\Delta_{Q_i^t}, \Delta_{Q_i^t})$ [29,49]. Assuming that the Hessian is nondegenerate, we compute the optimal updating vector for a generalized Newton method as $\Delta_{Q_i^t} = -Hess^{-1}G$, where G is the gradient on the manifold. To compute the Hessian, we use an orthonormal basis $\mathbf{E}_b$ of the tangent space on $SO(3)$. For simplicity, we can choose the standard basis $\mathbf{e}_b$ for $\mathbb{R}^3$. The Hessian matrix $\mathbf{H}$ and gradient vector $\mathbf{g}$ can be obtained as:

$$\mathbf{g}_b = d\,\mathcal{B}(\mathbf{E}_b), \qquad (27)$$

$$\mathbf{H}_{bc} = Hess\,\mathcal{B}(\mathbf{E}_b, \mathbf{E}_c). \qquad (28)$$

Finally, the optimal updating vector can be computed as $\Delta_{Q_i^t} = -Hess^{-1}G = \mathbf{Q}_i^t\,[\boldsymbol{\delta}]_\times$ and to move it in the tangent direction along the geodesic on $SO(3)$. The outline of the algorithm is shown in Algorithm 1.

Next, the noise variance and the translation vector can be updated in an on-line manner as:

$$\begin{aligned} \sigma^2 &= \frac{1}{2\varrho\mathcal{W}} \sum_{i=f-\mathcal{W}+1}^{f} \sum_{\varrho\in\mathcal{V}} \Big(-2\left(\mathbf{w}_{i\varrho} - \bar{\mathbf{t}}_i\right)^\top \mathbf{G}_i \tilde{\mathcal{S}}_\varrho \tilde{\boldsymbol{\mu}}_i \\ &\quad + \|\mathbf{w}_{i\varrho} - \bar{\mathbf{t}}_i\|^2 + \mathrm{tr}\left(\tilde{\mathcal{S}}_\varrho^\top \mathbf{G}_i^\top \mathbf{G}_i \tilde{\mathcal{S}}_\varrho \tilde{\boldsymbol{\phi}}_i\right)\Big), \\ \mathbf{t}_i &= \frac{1}{\varrho} \sum_{\varrho\in\mathcal{V}} \left(\mathbf{w}_{i\varrho} - \mathbf{R}_i \tilde{\mathcal{S}}_\varrho \tilde{\boldsymbol{\mu}}_i\right). \end{aligned}$$

Algorithm 1: Minimizing $\mathcal{B}\left(\mathbf{Q}_i^t\right)$ Eq. (23)

```
Output: Optimal updated rotation matrix $\mathbf{Q}_i^{t+1}$
$i = f - \mathcal{W} + 1$
while $\mathbf{Q}_i^t \leq \mathbf{Q}_f^t$ do
    I. Compute Optimal Updating Vector $\Delta_{Q_i^t}$
        $\mathbf{E}_b = \mathbf{Q}_i^t\,[\mathbf{e}_b]_\times \quad 1 \leq b, c \leq 3$
        $\mathbf{g}_b = d\,\mathcal{B}(\mathbf{E}_b)$                    (Eq. (25))
        $\mathbf{H}_{bc} = Hess\,\mathcal{B}(\mathbf{E}_b, \mathbf{E}_c)$       (Eq. (26))
        $\boldsymbol{\delta} = -\mathbf{H}^{-1}\mathbf{g}$
        $\Delta_{Q_i^t} = \mathbf{Q}_i^t\,[\boldsymbol{\delta}]_\times$
    II. Update the Rotation Matrix $\mathbf{Q}_i^{t+1}$
        $\mathbf{Q}_i^{t+1} = \mathbf{Q}_i^t \exp\left(\alpha\left[\hat{\boldsymbol{\delta}}\right]_\times\right)$        (Eq. (24))
        with $\alpha = \sqrt{\frac{1}{2}\mathrm{tr}\left(\Delta_{Q_i^t}^\top \Delta_{Q_i^t}\right)}$
        $i \longleftarrow i + 1$
end while
```

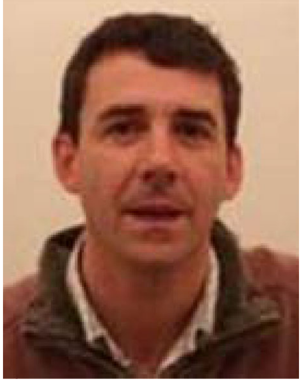
References

1. Agudo, A., Calvo, B., Montiel, J.M.M.: FEM models to code non-rigid EKF monocular SLAM. In: IEEE International Workshop on Dynamic Shape Capture and Analysis, pp. 1586–1593 (2011)
2. Agudo, A., Calvo, B., Montiel, J.M.M.: 3D reconstruction of non-rigid surfaces in real-time using wedge elements. In: Workshop on non-rigid shape analysis and deformable image alignment, pp 113–122 (2012a)
3. Agudo, A., Calvo, B., Montiel, J.M.M.: Finite element based sequential bayesian non-rigid structure from motion. In: IEEE conference on computer vision and pattern recognition, pp. 1418–1425 (2012b)
4. Agudo, A., Agapito, L., Calvo, B., Montiel, J.M.M.: Good vibrations: a modal analysis approach for sequential non-rigid structure from motion. In: IEEE conference on computer vision and pattern recognition, pp. 1558–1565 (2014a)
5. Agudo, A., Montiel, J.M.M., Agapito, L., Calvo, B.: Online dense non-rigid 3D shape and camera motion recovery. In: British Machine Vision Conference (2014b)
6. Agudo, A., Moreno-Noguer, F., Calvo, B., Montiel, J.M.M.: Sequential non-rigid structure from motion using physical priors. IEEE Trans. Pattern Anal. Mach. Intell. **38**(5), 979–994 (2016)
7. Akhter, I., Sheikh, Y., Khan, S.: In defense of orthonormality constraints for nonrigid structure from motion. In: IEEE conference on computer vision and pattern recognition, pp. 1534–1541 (2009)
8. Akhter, I., Sheikh, Y., Khan, S., Kanade, T.: Trajectory space: a dual representation for nonrigid structure from motion. IEEE Trans. Pattern Anal. Mach. Intell. **33**(7), 1442–1456 (2011)
9. Bartoli, A., Gay-Bellile, V., Castellani, U., Peyras, J., Olsen, S., Sayd, P.: Coarse-to-fine low-rank structure-from-motion. In: IEEE Conference on computer vision and pattern recognition, pp. 1–8 (2008)
10. Bathe, K.J.: Finite Element Procedures in Engineering Analysis. Prentice-Hall, Upper Saddle River (1982)
11. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. In: International conference and exhibition on computer graphics and interactive techniques, pp. 187–194 (1999)
12. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3D shape from image streams. In: IEEE conference on computer vision and pattern recognition, pp. 690–696 (2000)
13. Chhatkuli, A., Pizarro, D., Bartoli, A.: Non-rigid shape-from-motion for isometric surfaces using infinitesimal planarity. In: British Machine Vision Conference (2014)

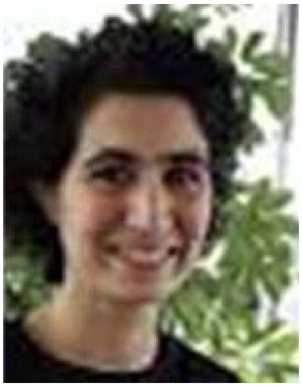
14. Cohen, L., Cohen, I.: Finite-element methods for active contour models and balloons for 2D and 3D images. IEEE Trans. Pattern Anal. Mach. Intell. **15**(11), 1131–1147 (1993)
15. Dai, Y., Li, H., He, M.: A simple prior-free method for non-rigid structure-from-motion factorization. Int. J. Comput. Vis. **107**(2), 101–122 (2014)
16. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: MonoSLAM: real-time single camera SLAM. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 1052–1067 (2007)
17. Del Bue, A., Llado, X., Agapito, L.: Non-rigid metric shape and motion recovery from uncalibrated images using priors. In: IEEE conference on computer vision and pattern recognition, pp 1191–1198 (2006)
18. Demmel, J.W., Philadelphia: Applied Numerical Linear Algebra. SIAM, Philadelphia (1997)
19. Edelman, A., Arias, T., Smith, S.: The geometry of algorithms with orthogonality constraints. SIAM J. Matrix Anal. Appl. **20**(20), 303–353 (1998)
20. Fayad, J., Agapito, L., Del Bue, A.: Piecewise quadratic reconstruction of non-rigid surfaces from monocular sequences. In: European conference on computer vision, pp. 297–310 (2010)
21. Garg, R., Roussos, A., Agapito, L.: Dense variational reconstruction of non-rigid surfaces from monocular video. In: IEEE conference on computer vision and pattern recognition, pp. 1272–1279 (2013a)
22. Garg, R., Roussos, A., Agapito, L.: A variational approach to video registration with subspace constraints. Int. J. Comput. Vis. **104**(3), 286–314 (2013b)
23. Ghahramani, Z., Hinton, G.E (1996) The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, University of Toronto (2013a)
24. Golub, G., Van Loa, C.: Matrix Computations. Johns Hopkins University Pess, Baltimore (1996)
25. Gotardo, P.F.U., Martinez, A.M.: Kernel non-rigid structure from motion. In: International conference on computer vision, pp. 802 – 809 (2011)
26. Kita, Y.: Elastic-model driven analysis of several views of a deformable cylindrical object. IEEE Trans. Pattern Anal. Mach. Intell. **18**(12), 1150–1162 (1996)
27. Klein, G., Murray, D.W.: Parallel tracking and mapping for small AR workspaces. In: International symposium on mixed and augmented reality, pp. 225–234 (2007)
28. Lee, M., Cho, J., Choi, C.H., Oh, S.: Procrustean normal distribution for non-rigid structure from motion. In: IEEE conference on computer vision and pattern recognition, pp. 1280–1287 (2013)
29. Ma, Y., Kosecka, J., Sastry, S.: Optimization criteria and geometric algorithms for motion and structure estimation. Int. J. Comput. Vis. **44**(3), 219–249 (1999)
30. Malti, A., Hartley, R., Bartoli, A., Kim, J.H.: Monocular template-based 3D reconstruction of extensible surfaces with local linear elasticity. In: IEEE conference on computer vision and pattern recognition, pp. 1522–1529 (2013)
31. Marques, M., Costeira, J.: Optimal shape from estimation with missing and degenerate data. In: IEEE workshop on motion and video computing, pp. 1–6 (2008)
32. Metaxas, D., Terzopoulos, D.: Constrained deformable superquadrics and nonrigid motion tracking. IEEE Trans. Pattern Anal. Mach. Intell. **15**(6), 580–591 (1993)
33. Moreno-Noguer, F., Porta, J.M.: Probabilistic simultaneous pose and non-rigid shape recovery. In: IEEE conference on computer vision and pattern recognition, pp. 1289–1296 (2011)
34. Mouragnon, E., Lhuillier, M., Dhome, M., Dekeyser, F., Sayd, P.: Generic and real-time structure from motion using local bundle adjustment. Image Vis. Comput. **27**(8), 1178–1193 (2009)
35. Nastar, C., Ayache, N.: Fast segmentation, tracking and analysis of deformable objects. In: IEEE international conference on computer vision, pp. 275–279 (1993)
36. Newcome, R., Davison, A.J.: Live dense reconstruction with a single moving camera. In: IEEE conference on computer vision and pattern recognition, pp. 1498–1505 (2010)
37. Newcome, R., Lovegrove, S., Davison, A.J.: DTAM: Dense tracking and mapping in real-time. In: IEEE international conference on computer vision, pp. 2320–2327 (2011)
38. Ostlund, J., Varol, A., Ngo, D.T., Fua, P.: Laplacian meshes for monocular 3D shape recovery. In: European conference on computer vision, pp. 412–425 (2012)
39. Paladini, M., Del Bue, A., Stosic, M., Dodig, M., Xavier, J., Agapito, L.: Factorization for non-rigid and articulated structure using metric projections. In: IEEE conference on computer vision and pattern recognition, pp. 2898–2905 (2009)
40. Paladini, M., Bartoli, A., Agapito, L.: Sequential non rigid structure from motion with the 3D implicit low rank shape model. In: European Conference on computer vision, pp. 15–28 (2010)
41. Park, H.S., Shiratori, T., Matthews, I., Sheikh, Y.: 3D reconstruction of a moving point from a series of 2D projections. In: European conference on computer vision, pp. 158–171 (2010)
42. Pentland, A., Horowitz, B.: Recovery of nonrigid motion and structure. IEEE Trans. Pattern Anal. Mach. Intell **13**(7), 730–742 (1991)
43. Pentland, A., Sclaroff, S.: Closed-form solutions for physically based shape modeling and recognition. IEEE Trans. Pattern Anal. Mach. Intell **13**(21), 715–729 (1991)
44. Roweis, S.: EM algorithms for PCA and SPCA. In: Neural information processing systems, pp. 626–632 (1998)
45. Russell, C., Fayad, J., Agapito, L.: Energy based multiple model fitting for non-rigid structure from motion. In: IEEE conference on computer vision and pattern recognition, pp. 3009–3016 (2011)
46. Salzmann, M., Fua, P.: Reconstructing sharply folding surfaces: A convex formulation. In: IEEE conference on computer vision and pattern recognition, pp. 1054–1061 (2009)
47. Salzmann, M., Urtasun, R., Fua, P.: Local deformation models for monocular 3D shape recovery. In: IEEE conference on computer vision and pattern recognition, pp. 1–8 (2008)
48. Sclaroff, S., Pentland, A.: Modal matching for correspondence and recognition. IEEE Trans. Pattern Anal. Mach. Intell. **17**(6), 545–561 (1995)
49. Shaji, A., Chandran, S.: Riemannian manifold optimisation for non-rigid structure from motion. In: Workshop on non-rigid shape analysis and deformable image alignment, pp. 1–6 (2008)
50. Tao, H., Huang, T.S.: Connected vibrations: a modal analysis approach for non-rigid motion tracking. In: IEEE conference on computer vision and pattern recognition, pp. 735–740 (1998)
51. Tao, L., Mein, S.J., Quan, W., Matuszewski, B.J.: Recursive non-rigid structure from motion with online learned shape prior. Computer Vision and Image Understanding **117**(10), 1287–1298 (2013)
52. Taylor, J., Jepson, A.D., Kutulakos, K.N.: Non-rigid structure from locally-rigid motion. In: IEEE conference on computer vision and pattern recognition, pp. 2761–2768 (2010)
53. Tipping, M.E., Bishop, C.M.: Mixtures of probabilistic principal component analysers. Neural Comput. **11**(2), 443–482 (1999)
54. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: a factorization approach. Int. J. Comput. Vis. **9**(2), 137–154 (1992)
55. Torresani, L., Yang, D.B., Alexander, E.J., Bregler, C.: Tracking and modelling non-rigid objects with rank constraints. In: IEEE conference on computer vision and pattern recognition, pp. 493–500 (2001)
56. Torresani, L., Hertzmann, A., Bregler, C.: Learning non-rigid 3D shape from 2D motion. In: Neural information processing systems, pp. 1555–1562 (2004)

57. Torresani, L., Hertzmann, A., Bregler, C.: Nonrigid structure-from-motion: estimating shape and motion with hierarchical priors. IEEE Trans. Pattern Anal. Mach. Intell. **30**(5), 878–892 (2008)
58. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment: a modern synthesis. Vis. Algorithms **1883**, 298–372 (2000)
59. Tsap, L.V., Goldof, D.B., Sarkar, S.: Nonrigid motion analysis based on dynamic refinement of finite element models. IEEE Trans. Pattern Anal. Mach. Intell. **22**(5), 526–543 (2000)
60. Varol, A., Salzmann, M., Tola, E., Fua, P.: Template-free monocular reconstruction of deformable surfaces. In: IEEE international conference on computer vision, pp. 1811–1818 (2009)
61. Vicente, S., Agapito, L.: Soft inextensibility constraints for template-free non-rigid reconstruction. In: European conference on computer vision, pp. 426–440 (2012)
62. Wendel, A., Maurer, M., Graber, G., Pock, T., Bischof, H.: Dense reconstruction on-the-fly. In: IEEE conference on computer vision and pattern recognition, pp. 1450–1457 (2012)
63. Woodbury, M.A.: Inverting modified matrices. Statistical Research Group, Memorandum Report No 42 (1950)
64. Yeoh, O.H.: Some forms of the strain energy function for rubber. Rubber Chem. Technol. **66**(5), 754–771 (1993)
65. Young, A., Axel, L.: Non-rigid wall motion using MR tagging. In: IEEE conference on computer vision and pattern recognition, pp. 399–404 (1993)
66. Zienkiewicz, O.C., Taylor, R.L.: The Finite Element Method. Vol. 1: Basic Formulation and Linear Problems. McGraw-Hill, London (1989)

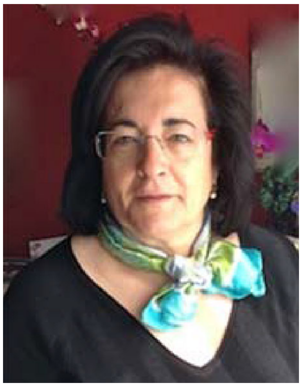
Antonio Agudo received the M.Sc. degree in industrial engineering and electronics in 2010, M.Sc. degree in computer science in 2011, and the Ph.D. degree in computer vision and robotics in 2015, from University of Zaragoza. He was a visiting student at vision group of Queen Mary University of London in 2013 and with the vision and imaging science group of University College London in 2014. He was also a visiting fellow at Harvard University in 2015. He is a postdoctoral fellow at the computer vision department of the Instituto de Robótica e Informática Industrial (CSIC-UPC) in Barcelona. His research interests include non-rigid structure from motion, machine learning, and deformation analysis to medical and robotics applications.



José M. Martínez Montiel received the M.Sc. and Ph.D. degrees in electrical engineering from the Universidad de Zaragoza (UZ), Spain, in 1991 and 1996, respectively. He is currently Full Professor with the Departamento de Informática, UZ, where he is in charge of Perception and Computer Vision research grants and courses. His current interests include real-time vision localization and mapping for rigid and non-rigid environments, and the transference of this technology to robotic and nonrobotic application domains. He is member of the I3A Robotics, Perception, and Real-Time Group. He has been awarded several Spanish MEC grants to fund research at the University of Oxford and at Imperial College London.



Lourdes Agapito obtained her BSc, MSc and PhD (1996) degrees from the Universidad Complutense de Madrid (Spain). She held an EU Marie Curie Postdoctora Fellowship at The University of Oxford's Robotics Research Group before being appointed as a Lecturer at Queen Mary, University of London in 2001. In 2008 she was awarded an ERC Starting Grant to carry out research on the estimation of 3D models of dynamic scenes from monocular video sequences. In July 2013 she joined the Department of Computer Science at University College London (UCL) where she leads a research team that focuses on developing algorithms for 3D understanding of the real world from video. Professor Agapito is Program Chair for CVPR 2016, the top annual conference in computer vision; in addition she was Programme Chair for 3DV'14 and Area Chair for CVPR'14, ECCV'14, ACCV'14. Lourdes is Associate Editor for the International Journal of Computer Vision (IJCV), a member of the Executive Committee of the British Machine Vision Association and a member of the EPSRC Peer Review College.



Begoña Calvo became Professor of the Department of Mechanical Engineering of the Universidad de Zaragoza (UZ), Spain, in 2010. From 1997 to 2010 she was Associate Professor of Structural Mechanics at the UZ. She got the degree of Mechanical Engineering in 1989 and the Ph.D. in Computational Mechanics in 1994, both from UZ. She is member of the Aragón Institute of Engineering Research and the National Networking Center on Bioengineering, Biomaterials and Nanomedicine. Her current research is related to Biomechanics, mainly in the field of mechanics of soft tissues, mechanical behavior of biomaterials and prostheses for clinical applications and experimental methods to characterize biological tissues.