Sequential Non-Rigid Structure from Motion using Physical Priors

Antonio Agudo, Francesc Moreno-Noguer, Begoña Calvo, and J.M.M. Montiel, *Member, IEEE*

Abstract—We propose a new approach to simultaneously recover camera pose and 3D shape of non-rigid and potentially extensible surfaces from a monocular image sequence. For this purpose, we make use of the EKF-SLAM (Extended Kalman Filter based Simultaneous Localization And Mapping) formulation, a Bayesian optimization framework traditionally used in mobile robotics for estimating camera pose and reconstructing rigid scenarios. In order to extend the problem to a deformable domain we represent the object's surface mechanics by means of Navier's equations, which are solved using a FEM (Finite Element Method). With these main ingredients, we can further model the material's stretching, allowing us to go a step further than most of current techniques, typically constrained to surfaces undergoing isometric deformations. We extensively validate our approach in both real and synthetic experiments, and demonstrate its advantages with respect to competing methods. More specifically, we show that besides simultaneously retrieving camera pose and non-rigid shape, our approach is adequate for both isometric and extensible surfaces, does not require neither batch processing all the frames nor tracking points over the whole sequence and runs at several frames per second.

Index Terms—Non-Rigid Structure from Motion, Extended Kalman Filter, Finite Element Method, Tracking.

1 Introduction

Simultaneously reconstructing a 3D rigid scene while estimating the trajectory of a monocular camera is a well studied problem in computer vision. Global optimization techniques based on Bundle Adjustment (BA) [23] or solutions using the Extended Kalman Filter (EKF) [13] have proven successful in a wide variety of applications, ranging from image-based modeling to autonomous robot navigation. These methods, though, cannot be applied to scenes undergoing *non-rigid* deformations. In these situations, the fact that many different 3D shape configurations can have very similar image projections produces severe ambiguities that can only be resolved by introducing smoothing priors about the camera trajectory and scene deformation.

The earliest *Non-Rigid Structure from Motion* (NRS*f*M) approaches modeled deformations as linear combinations of basis shapes [8], [9], [43], which in conjunction with the Tomasi and Kanade's factorization algorithm [42], allowed to simultaneously solve for non-rigid shape and camera motion. An additional assumption made by these and subsequent techniques [3], [14], [17], [19], [45], is that input images are acquired using an orthographic camera. This has been extended to full-perspective cameras in [4], [21]. In any event, all these approaches batch process all frames of the sequence at once, preventing them from being used on-line and in real-time applications. An interesting exception are the recent works [34], [41], which propose sequential solutions to the NRS*f*M problem. While these works offer promising directions, they share a fundamental limitation with previous methods, in that they rely on image points that can be observed and tracked over the sequence. This assumption cannot be guaranteed in practical situations where shape deformations may produce severe changes in the appearance of the object.

In this paper we propose a solution to simultaneously recover camera pose and 3D non-rigid shape that overcomes most of the aforementioned limitations: 1) it handles full-perspective calibrated cameras, 2) it is sequential, 3) it automatically establishes correspondences between consecutive frames, allowing feature points to appear or disappear along the sequence, and 4) both rigid and non-rigid points can be uniformly processed under the same formalism. Our approach draws inspiration on the probabilistic EKF-SLAM methodology used in mobile robotics for reconstructing rigid environments and estimating sensor poses. To bring these tools from a rigid to a deformable domain we consider the Finite Element Method (FEM), used to solve the mechanics of deformable solids. More specifically, the surface to be estimated is modeled as a set of *finite elements*, whose displacement is ruled by the Navier's equations [47], and jointly embedded with a smooth 3D camera motion into the EKF. As a result, we are then able to both estimate the state of the moving camera and the shape of the deforming surface, while guiding the feature matching between consecutive frames.

Additionally, and in contrast to traditional approaches using FEM for non-rigid shape modeling [27], [44], we propose a FEM formulation in which most of the physical parameters, such as the Young's modulus, can be factorized out from of the deformation model and

- *Antonio Agudo, Begoña Calvo and J.M.M. Montiel are with the Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, 50018, Spain. Email: {aagudo, bcalvo, josemari}@unizar.es.*
- *Francesc Moreno-Noguer is with the Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, 08028, Spain. Email: fmoreno@iri.upc.edu.*

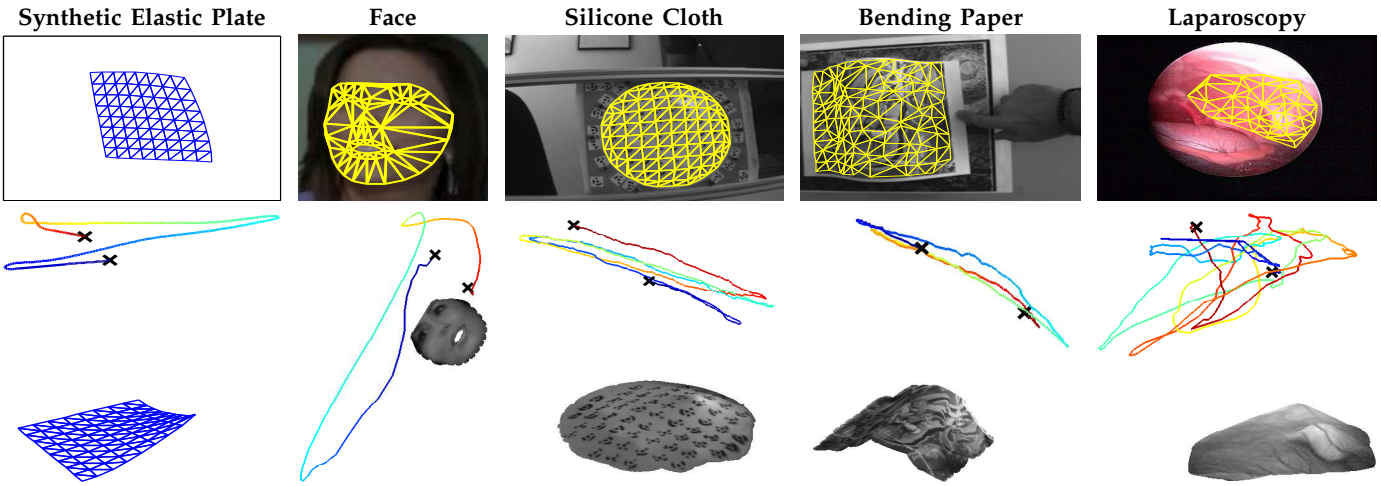

Fig. 1. **Simultaneous estimation of 3D non-rigid structure and camera trajectory.** Summary of the experiments. **Top**: Reconstructed 3D mesh overlaid on a specific frame of the input sequence. **Bottom**: 3D view of the same mesh and camera trajectory for the whole sequence. Black crosses indicate the initial and final camera positions. In contrast to standard EKF-SLAM methods where the scene is considered to be rigid, the mesh continuously deforms and we estimate its shape in each frame of the sequence. Our approach is suitable for both extensible surfaces (like the Silicone Cloth) and non-extensible materials (like the Bending Paper). The Synthetic Plate, the Face, and the Laparoscopic data, exhibit intermediate levels of extensibility.

incorporated within a Gaussian noise term, that can be naturally handled by the EKF. This gives us the additional benefit that we can deal with different materials by just roughly adjusting the magnitude of this noise term, which is much more intuitive than having to adjust the underlying physical parameters. As shown in Fig. 1, by doing this, we are able to reconstruct from isometric to highly extensible surfaces without using any learning method at all. This is a remarkable step-forward when compared to competing approaches, mostly constrained to inextensible surfaces or relying on relatively vast amounts of training data.

A preliminary version of this work was presented in [1]. Here, we have pushed the limits of our model to show that it can cope with large elastic deformations. New synthetic results demonstrate the advantage of our approach compared to current state of the art. In addition, further real results in challenging laparoscopy images are included in this version.

2 Related Work

Recovering non-rigid 3D shape from a single camera has been an active research area in the past two decades. It is an inherently ambiguous problem in which very different shapes can virtually produce the same projection. In order to limit the possible range of solutions, prior knowledge of either the nature of the deformations or the camera motion needs to be introduced.

Early approaches constrained the 3D deformation using physically-inspired models based on superquadrics [28], Fourier harmonics [33], balloons [11] or spring models [22]. These approaches, though, were only effective to capture relatively small and non-realistic deformations. More accurate representations were achieved with the FEM [26], [27], [39], [44], [46]. Yet, their applicability was limited to very specific materials for which geometric and mechanical parameters were known in advance. In a recent approach, the Poisson's ratio is jointly optimized with the shape using an energy minimization scheme [24]. Yet, this work is focused to the shape-from-template problem, which is out of the scope of this paper.

Statistical methods that learn deformation modes from training data, can capture the true shape variability without the need to resort to sophisticated optimization procedures to tune physical parameters. Active appearance and shape models [12], [25] and 3D morphable models [7] are examples of such approaches, in which deformations are represented as linear combinations of modes. Retrieving shape entails minimizing an image-based objective function, that generally requires to be accurately initialized to converge.

A new family of solutions propose ways to avoid falling into local minima by either reformulating the deformable shape estimation as a convex optimization problem [38] or as a linearized system with closed form solution [32], [37]. Other approaches use global optimization techniques based on semidefinite programming [15] or evolutionary computing strategies [30] that avoid the need for an initialization. [5] introduced an analytical solution which was applicable to several types of deformation, such as developable, isometric and conformal.

However, despite all this tremendous progress, none of the previous approaches explicitly computes the camera pose, and either assume that it is already known in advance and the modes are aligned with the camera coordinate frame, or provide a shape estimation for which its pose with respect to the camera is unknown. This is addressed by NRS*f*M methods, which build upon the rigid factorization algorithm [42] to simultaneously recover deformable shape and relative camera motion from a sequence of images. NRS*f*M algorithms typically

represent the varying 3D shape as a linear combination of basis, being estimated in conjunction with the shape and motion parameters [9], [43]. Yet, while these methods do not require learning deformation modes off-line, they often rely on a set of assumptions which are difficult to hold in practice. Orthographic cameras, continuous point tracks along the whole sequence, batch processing and relative (not absolute) camera pose, are common requirements which need to be satisfied [3], [17], [19], [36]. There have been attempts to alleviate these assumptions, such as expanding orthographic to perspective cameras [4], [21], handling outlier correspondences or discontinuous point tracks [31], and sequentially processing input frames [34], [41]. In [14], [24], subsets of rigid points are initially used to estimate the absolute camera pose, prior retrieving the shape. In any event, and to the best of our knowledge, there is no current method able to simultaneously handle all these issues.

In this paper we will show that expanding the EKF-SLAM formulation from a rigid to a deformable domain, will let to cope with most of the fundamental limitations of previous approaches. In particular we will demonstrate that our solution is valid for full-perspective calibrated cameras, does not need continuous tracks of feature points, performs automatic data association, does not require training data, and works in sequential mode, potentially in real-time. Moreover, we estimate absolute camera pose by combining rigid and non-rigid points, but in contrast to [14], [24], we manage both kind of points in a single framework. The core of our approach relies on the Bayesian formulation of the EKF monocular SLAM [10], [13]. As in EKF-SLAM we will use the prior that the camera moves smoothly according to a constant velocity model. However, and as a main contribution of our work, we will introduce a non-rigid mechanical model of the deformable object by means of the FEM. As mentioned previously, physically-based methods usually require fine tuning of the material parameters and they normally have a high computational complexity limiting their real-time applicability. In our approach, the use of the EKF framework allows for a very loose tuning of these parameters.

In addition, although the FEM formulation we propose is a linear methodology only valid for small deformations [47], its combination with an EKF that digests every image of the video sequence, results in a solution able to accurately estimate large and potentially elastic scene deformations with a low computational overhead. This circumvents the need of expensive and non-linear computational methods [44].

Finally, as said above, this paper is an extended version of [1]. Our recent work [2], also holds on [1], but only handles sets of non-rigid points, and not a combination of rigid and non-rigid point as we do here. [2] results in a more general solution, but at the expense of lower estimation accuracy and most importantly, the inability to recover full camera trajectory. Exploring this direction is part of our future work.

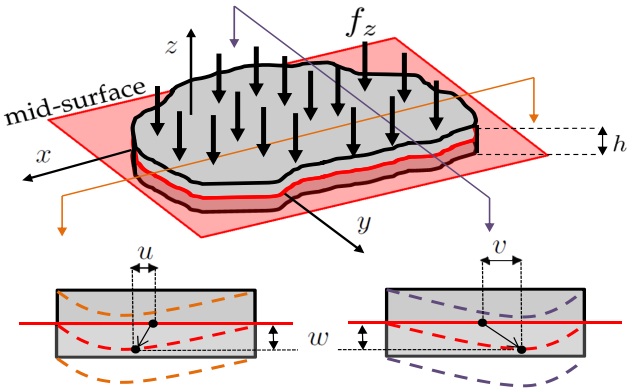

Fig. 2. **Thin-plate geometry. Top**: Structure at rest. The *mid-surface plane* (in red) is used to describe the geometry of the deformation. **Bottom**: Cross sectional views of the deformed structure. Displacements u, v and w of the middle-plane.

3 Overview of the Method

Our approach to simultaneously retrieve non-rigid shape and camera pose cross fertilizes Bayesian estimation with physically-based modeling. More specifically, we model the non-rigid displacement solving a 2D version of Navier's equations, combining the plane-stress typology with Kirchhoff-Love theory for thin-plates. In order to solve the resulting partial differential equations we carry out a spatial discretization by means of FEM, yielding a formulation where displacement and forces are linearly related through a stiffness matrix, re-computed at each iteration. This mechanical model, in conjunction with a smooth camera motion prior are fed into a EKF-SLAM formulation, which jointly estimates the geometry of the shape and the 3D camera trajectory.

In the following sections, we discuss each one of these ingredients in more detail.

4 Physical Modeling of the Surface Deformation

The simplest approach to model the physical behavior of a deformable surface is using a combination between a plane-stress model (*membrane*) and one that accounts for out-of-plane *bending*. We will represent the former using the Navier's equations, and the latter through the Kirchhoff-Love theory, an extension of the Euler-Bernoulli beam theory to thin-plates. To keep the paper self-contained, we next provide a quick overview of these equations and how FEM applies to resolve them. For further details, the reader is referred to [47], [48].

4.1 Basics from Continuum Mechanics

Let us consider the thin-plate Ω depicted in Fig. 2, which bends under the action of an external distributed load f_z. From the cross-sectional view, it can be seen that the out-of-plane deformation induces a stretching of the bottom of the plate and a compression of the top. Assuming that the stresses vary monotonically between the top and bottom of the plate, we can further define a zero-stress surface, called the mid-surface plane. This allows

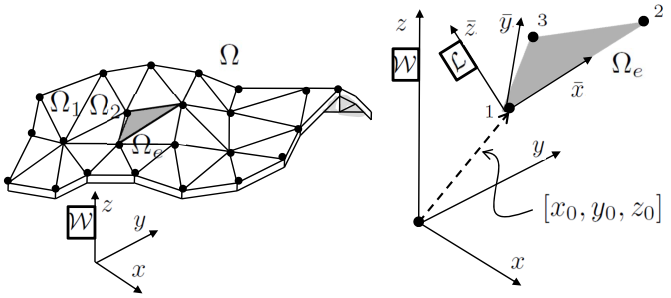

Fig. 3. Triangular discretization of the scene. $\mathcal{W}$ and $\mathcal{L}$ represent the global and the local reference systems, respectively.

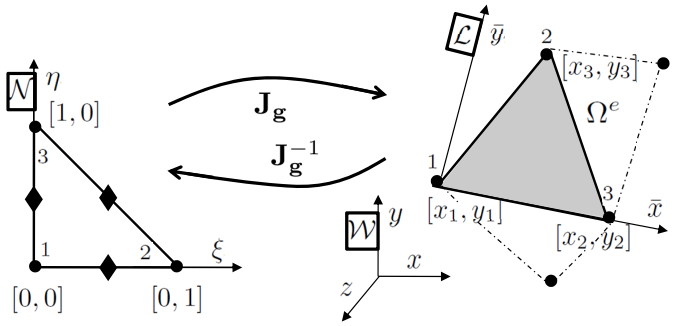

Fig. 4. Normalized triangle in natural coordinates $\mathcal{N}$ and real triangle element in local coordinates $\mathcal{L}$. They are related by the $\mathbf{J_g}$ transformation. Nodes are represented as black dots (•) and Hammer's integration points as black diamonds (♦).

to parameterize the 3D deformation as a 2D function, that assigns a vertical displacement $w(x, y, z) = w(x, y)$ at any point (x, y) on the mid-surface plane.

In order to describe the deformation, the Kirchhoff-Love theory makes the additional assumptions that 1) lines normal to the middle-plane remain straight, normal and unstretched after deformation, and 2) that the thickness of the plate does not change during a deformation. The governing equations for the bending (out-of-plane displacement) of the plate can then be established by setting equilibrium conditions of the external and internal forces and moments. This leads to the equilibrium equation of the Kirchhoff-Love thin-plate:

$$\frac{Eh^3}{12(1-\nu^2)}\nabla^4 w = -f_z \tag{1}$$

where E is the Young's modulus, ν the Poisson's ratio, h the thickness of the plate and $\nabla = [\partial/\partial x, \partial/\partial y]^\top$ is the gradient operator.

We can similarly represent membrane (the in-plane) displacements $\mathbf{u} = [u, v]^\top$ at any point (x, y) on the mid-surface plane using the Navier's equations:

$$\frac{\nu E}{1-\nu^2}\nabla(\nabla^\top \mathbf{u}) + \frac{E}{2(1+\nu)}\nabla^2 \mathbf{u} = -\mathbf{f}_{xy} \tag{2}$$

where $\mathbf{f}_{xy} = [f_x, f_y]^\top$ are the in-plane volumetric forces. Both equations need boundary conditions. That is, we impose a known component, for instance, $\mathbf{u} = \bar{\mathbf{u}}$ on the boundary $\delta\Omega^u$ where $\bar{\mathbf{u}}$ is a known displacement. Similar constraints can be imposed on the stress components.

4.2 Finite Element Method Solution

The previous partial differential equations (1) and (2) do not generally have an analytical solution, and one has to resort to approximate numerical optimization techniques such as FEM. These methods discretize the continuum surface Ω into a finite number of parts Ω_e defined by its nodes (see Fig. 3). The derivation of the finite element equations can then be obtained by the application of the principle of virtual work, which states that in stable elastic equilibrium the virtual work done by externally applied forces equals the virtual strain energy [47]. This gives rise to the classical FEM formulation for the complete system in global coordinates:

$$\mathbf{K}\mathbf{a} = \mathbf{f} \tag{3}$$

where $\mathbf{a}$ and $\mathbf{f}$ are the global vector of nodal displacements and forces, respectively. $\mathbf{K}$ is the system stiffness matrix which can be assembled from the stiffness matrices $\mathbf{K}^e$ associated to individual elements.

In Sect. 5 we will introduce Eq. (3) into the EKF formulation to constrain the non-rigid displacement of the surface. In the remaining of this subsection we will focus on one single surface element, and describe in detail how the nodal displacements and forces are represented and how the elementary stiffness matrix is computed.

We approximate the surface through a set of flat and thin triangular elements. Let us consider one such element, defined by three nodes and straight line boundaries. As shown in Fig. 4 we consider three coordinate systems: one global system $\mathcal{W}$ common to all triangles, a local reference $\mathcal{L}$ defined on the plane of each triangle, and one natural reference $\mathcal{N}$ in which local coordinates are normalized within the $[0, 1]$ interval. Given the i-th node of the triangle, we then denote by $\mathbf{g}_i = [x_i, y_i, z_i]^\top$ its coordinates in the global system, by $\bar{\mathbf{g}}_i = [\bar{x}_i, \bar{y}_i, \bar{z}_i]^\top$ the coordinates in the local reference, and by $\boldsymbol{\xi}_i = [\xi_i, \eta_i]^\top$ its natural coordinates.

In addition, in order to approximate the continuous solution in displacements within the triangle, we consider linear and quadratic shape functions for the membrane and bending effect respectively, both defined in the natural coordinate system. For instance, let $\bar{\mathbf{a}}_i$ be the displacement of the i-th node, expressed in local coordinates. The displacement $\bar{\mathbf{u}}$ at any point $(\bar{x}, \bar{y})$ within the planar triangle can then be linearly interpolated as:

$$\bar{\mathbf{u}}(\bar{x}, \bar{y}) = \sum_{i=1}^{3} N_i(\xi, \eta)\bar{\mathbf{a}}_i \tag{4}$$

where $N_i(\xi, \eta)$ are the continuous shape functions defined on the nodes of the triangle. See Appendix for the exact expressions of these shape functions.

From the principle of virtual work, it can be shown that the elemental stiffness matrix $\bar{\mathbf{K}}^e$ in local coordinates is obtained as:

$$\bar{\mathbf{K}}^e = \int_{\Omega_e} h\mathbf{B}^\top \mathbf{D}\mathbf{B} d\Omega_e \tag{5}$$

where Ω_e defines the element domain, $\mathbf{B}$ is the strain-displacement matrix derived in Appendix, and $\mathbf{D}$ is a behavior matrix containing the material properties. In practice, this integration is computed in the natural coordinate system, and numerically approximated using Hammer's point integration [20] as:

$$\begin{aligned}\bar{\mathbf{K}}^e &= \int_0^1 \int_0^{1-\xi} h\mathbf{B}(\xi,\eta)^\top \mathbf{D}\mathbf{B}(\xi,\eta)|\mathbf{J_g}|d\eta d\xi \\ &\approx h\sum_{p=1}^{r} \alpha_p \mathbf{B}(\xi_p,\eta_p)^\top \mathbf{D}\mathbf{B}(\xi_p,\eta_p)|\mathbf{J_g}| \end{aligned} \quad (6)$$

where $|\mathbf{J_g}|$ is the Jacobian of the mapping from natural to local coordinates, i.e., $\mathbf{J_g} = \partial\bar{\mathbf{g}}/\partial\boldsymbol{\xi}$. r is the number of integration points and α_p are fixed point weights.

4.3 Combining Bending and Membrane Strains

As mentioned above, we approximate the deformation of each surface element using a combination of bending (out-of-plane) and membrane (in-plane) displacements.

Following [48], the bending (b) of a triangular element, (we use a DKT, Discrete Kirchhoff Triangle), is defined by the displacement $\bar{w}_i$ in the z direction and two rotations $\theta_{\bar{x}i}$, $\theta_{\bar{y}i}$, for each of its three nodes $i = \{1,2,3\}$. The finite element representation of the bending is then written as:

$$\bar{\mathbf{f}}^{eb} = \bar{\mathbf{K}}^{eb}\bar{\mathbf{a}}^b \quad \text{with} \quad \bar{\mathbf{a}}_i^b = \begin{bmatrix} \bar{w}_i \\ \theta_{\bar{x}i} \\ \theta_{\bar{y}i} \end{bmatrix} \quad \bar{\mathbf{f}}_i^b = \begin{bmatrix} f_{\bar{z}i} \\ M_{\bar{x}i} \\ M_{\bar{y}i} \end{bmatrix} \quad (7)$$

where $M_{\bar{x}i}$ and $M_{\bar{y}i}$ are the bending moments along the $\bar{x}$ and $\bar{y}$ directions, $\bar{\mathbf{f}}^{eb} = [\bar{\mathbf{f}}_1^b, \bar{\mathbf{f}}_2^b, \bar{\mathbf{f}}_3^b]^\top$ and $\bar{\mathbf{a}}^b = [\bar{\mathbf{a}}_1^b, \bar{\mathbf{a}}_2^b, \bar{\mathbf{a}}_3^b]^\top$. The elemental stiffness matrix $\bar{\mathbf{K}}^{eb}$ is made up from submatrices $\bar{\mathbf{K}}_{ij}^b$ for every pair of nodes in the triangular element, each of them computed from Eq. (6). In the Appendix, we provide the particular expressions for the strain-displacement matrix $\mathbf{B}^b$ and deformation matrix $\mathbf{D}^b$ that we use to represent the bending.

Similarly, we describe the membrane (m) deformation in terms of the $\bar{u}$ and $\bar{v}$ displacements of each node i. The relation between the nodal forces and these displacements is expressed as:

$$\bar{\mathbf{f}}^{em} = \bar{\mathbf{K}}^{em}\bar{\mathbf{a}}^m \quad \text{with} \quad \bar{\mathbf{a}}_i^m = \begin{bmatrix} \bar{u}_i \\ \bar{v}_i \end{bmatrix} \quad \bar{\mathbf{f}}_i^m = \begin{bmatrix} f_{\bar{x}i} \\ f_{\bar{y}i} \end{bmatrix} \quad (8)$$

where $\bar{\mathbf{f}}^{em} = [\bar{\mathbf{f}}_1^m, \bar{\mathbf{f}}_2^m, \bar{\mathbf{f}}_3^m]^\top$ and $\bar{\mathbf{a}}^m = [\bar{\mathbf{a}}_1^m, \bar{\mathbf{a}}_2^m, \bar{\mathbf{a}}_3^m]^\top$. Again, the stiffness matrix $\bar{\mathbf{K}}^{em}$ is made up from the submatrices $\bar{\mathbf{K}}_{ij}^m$, which are computed from Eq. (6) and using the $\mathbf{B}^m$ and $\mathbf{D}^m$ matrices detailed in the Appendix.

In order to combine membrane and bending strains we consider an expanded nodal displacement vector and forces:

$$\bar{\mathbf{a}}_i = [\bar{u}_i, \bar{v}_i, \bar{w}_i, \theta_{\bar{x}i}, \theta_{\bar{y}i}]^\top \quad (9)$$

$$\bar{\mathbf{f}}_i = [f_{\bar{x}i}, f_{\bar{y}i}, f_{\bar{z}i}, M_{\bar{x}i}, M_{\bar{y}i}]^\top \ , \quad (10)$$

and an expanded stiffness matrix:

$$\bar{\mathbf{K}}_{ij} = \begin{bmatrix} \bar{\mathbf{K}}_{ij}^m & \mathbf{0}_{2\times 3} \\ \mathbf{0}_{3\times 2} & \bar{\mathbf{K}}_{ij}^b \end{bmatrix} \ . \quad (11)$$

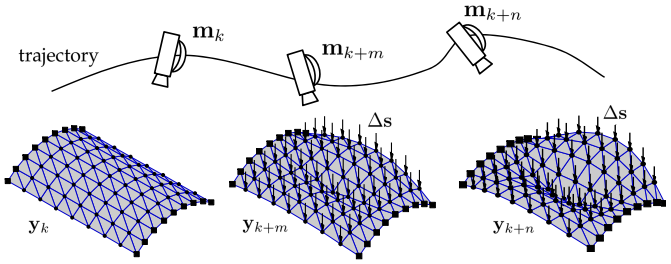

Fig. 5. **Problem Formulation.** A moving camera $\mathbf{m}$ observes a non-rigid structure $\mathbf{y}$. A set of Gaussian distributed forces $\Delta\mathbf{s}$ are acting on the structure, resulting in a non-rigid deformation. Our goal, is to estimate both the camera trajectory and surface deformation from only image observations. A set of rigid points on the surface (black squares ■) let us to disambiguate between rigid relative motions of the camera and the surface.

Note that the displacements, forces and stiffness matrix just derived use the system of local coordinates $\mathcal{L}$. In order to convert them to the global coordinate system $\mathcal{W}$ (see Fig. 3), we consider the 3×3 matrix $\mathbf{\Lambda}$ that transforms from the local to the global system:

$$\mathbf{a}_i = \mathbf{T}^\top\bar{\mathbf{a}}_i \qquad \mathbf{f}_i = \mathbf{T}^\top\bar{\mathbf{f}}_i \qquad \mathbf{K}_{ij} = \mathbf{T}^\top\bar{\mathbf{K}}_{ij}\mathbf{T} \quad (12)$$

where

$$\mathbf{T} = \begin{bmatrix} \mathbf{\Lambda} & \mathbf{0}_{3\times 2} \\ \mathbf{0}_{2\times 3} & \mathbf{\Lambda}[1,2;1,2] \end{bmatrix} \quad (13)$$

and $\mathbf{\Lambda}[1,2;1,2]$ are the first two rows and columns of $\mathbf{\Lambda}$.

Finally, we use the submatrices $\mathbf{K}_{ij}$ to assemble the elemental stiffness matrix $\mathbf{K}^e$ for each triangle, which in turn, are used to build the global stiffness matrix $\mathbf{K}$ of Eq. (3) for the whole surface.

5 Non-Rigid EKF

Our key contribution is to embed the FEM formulation that models the surface deformation within the Bayesian framework of an EKF. This combination will provide a mechanism to simultaneously estimate the shape of the deforming object and the pose of a moving camera.

5.1 Assumptions and Problem Formulation

We represent the surface as a triangulated mesh with n vertexes $\mathbf{g}_i$ concatenated in a $3n$ vector $\mathbf{y} = [\mathbf{g}_1, \ldots, \mathbf{g}_n]^\top$. We assume that $p << n$ of these points are rigid, i.e., they always remain steady, and are labeled as $\mathcal{B}_p$ (Fig. 5). In our experiments these points are manually chosen. Note that without this assumption, it would not be possible to disambiguate between camera motions and rigid displacements of the surface.

The state of the camera is represented by a 13-dimensional vector:

$$\mathbf{m} = [\mathbf{r}^\top, \mathbf{q}^\top, \mathbf{v}^\top, \boldsymbol{\omega}^{\mathcal{C}\top}]^\top \ , \quad (14)$$

where $\mathbf{r}$ and $\mathbf{q}$ are the position vector and orientation quaternion that express the pose of the camera, relative to the world coordinate system $\mathcal{W}$. $\mathbf{v}$ is the velocity vector, also relative to $\mathcal{W}$, and $\boldsymbol{\omega}^{\mathcal{C}}$ is the angular velocity relative to a frame $\mathcal{C}$ fixed to the camera.

Let us denote by $\mathbf{y}_k$, $\mathbf{m}_k$ and $\mathcal{I}_k$ the 3D mesh configuration, camera state and input image at time k. Our problem consists in using this information and the input image $\mathcal{I}_{k+1}$ at time $k+1$, to estimate $\mathbf{y}_{k+1}$ and $\mathbf{m}_{k+1}$.

We address this problem using a full covariance EKF formulation, in which the map is mathematically represented by a state vector $\mathbf{x} = [\mathbf{m}^\top, \mathbf{y}^\top]^\top$ composed of the camera configuration and mesh vertexes. Upon the arrival of a new input image, the state vector estimate $\hat{\mathbf{x}} = [\hat{\mathbf{m}}^\top, \hat{\mathbf{y}}^\top]^\top$ and its covariance matrix $\mathbf{P}$ are iteratively updated following the standard approach detailed in Algorithm 1. We next describe the main elements of this process, namely the dynamic models that predict the next state of the camera and surface, and the observation model that performs the correction.

5.2 Camera and Surface Motion Models

5.2.1 Camera Motion Model

Following [13], the camera motion is represented using a constant velocity model, which, at each time step Δt, introduces an impulse of linear and angular velocities:

$$\begin{aligned} \Delta\mathbf{v} &= \dot{\mathbf{v}}\Delta t \\ \Delta\boldsymbol{\omega}^{\mathcal{C}} &= \dot{\boldsymbol{\omega}}^{\mathcal{C}}\Delta t \end{aligned}$$

where $\dot{\mathbf{v}}$ and $\dot{\boldsymbol{\omega}}^{\mathcal{C}}$ are unknown linear and angular acceleration variables, with zero mean and Gaussian distribution and covariance matrix $\mathbf{Q}_{\mathbf{m}}$. The camera transition state function $\mathbf{m}_{k+1} \equiv \mathbf{m}_{k+1}(\mathbf{m}_k, \Delta\mathbf{v}, \Delta\boldsymbol{\omega}^{\mathcal{C}})$ is:

$$\mathbf{m}_{k+1} = \begin{bmatrix} \mathbf{r}_{k+1} \\ \mathbf{q}_{k+1} \\ \mathbf{v}_{k+1} \\ \boldsymbol{\omega}^{\mathcal{C}}_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_k + (\mathbf{v}_k + \Delta\mathbf{v})\Delta t \\ \mathbf{q}_k \times \mathbf{q}((\boldsymbol{\omega}^{\mathcal{C}}_k + \Delta\boldsymbol{\omega}^{\mathcal{C}})\Delta t) \\ \mathbf{v}_k + \Delta\mathbf{v} \\ \boldsymbol{\omega}^{\mathcal{C}}_k + \Delta\boldsymbol{\omega}^{\mathcal{C}} \end{bmatrix} \tag{15}$$

where $\mathbf{q}((\boldsymbol{\omega}^{\mathcal{C}}_k + \Delta\boldsymbol{\omega}^{\mathcal{C}})\Delta t)$ denotes the quaternion defined by the rotation vector $(\boldsymbol{\omega}^{\mathcal{C}}_k + \Delta\boldsymbol{\omega}^{\mathcal{C}})\Delta t$.

5.2.2 Surface Motion Model

In order to introduce the non-linear dynamics of the surface deformation into the EKF, we use the FEM formulation of Eq. (3), and consider a dynamic model which assumes that unknown force impulses $\Delta\mathbf{f}$ cause the following increment of the nodal displacements between consecutive frames:

$$\Delta\mathbf{a} = \mathbf{K}^{-1}\Delta\mathbf{f}\ . \tag{16}$$

We enforce the boundary conditions that constrain the displacement of the p rigid points to be zero, providing the necessary additional constraints to solve the linear system and the camera absolute location. Note that the vector $\Delta\mathbf{a}$ is made of both the nodal displacements and rotation increments. As for representing the surface we are only interested in the displacement components of the middle-plane, where the rotation effect is null, we consider a cropped inverse stiffness matrix $(\mathbf{K}^{-1})^*$ that results from removing the rows and columns corresponding the nodal rotations. This is formulated by considering the compliance matrix:

$$\mathbf{C} = (\mathbf{K}^{-1})^*, \tag{17}$$

in which we have sorted the columns such that the last p of them correspond to the rigid nodes. With this arrangement, we can finally rewrite Eq. (16) in terms of the nodal displacements:

$$\Delta\mathbf{y} = \mathbf{C}\Delta\mathbf{f}^* \tag{18}$$

where $\Delta\mathbf{f}^* = [\Delta\mathbf{f}_1, \ldots, \Delta\mathbf{f}_{n-p}, \mathbf{0}_{1\times 3p}]^\top$, and each $\Delta\mathbf{f}_i$ is assumed to be a random variable with zero mean and Gaussian distribution.

One of the main limitations of using FEM in practical situations is that they require knowing the material parameters (ν and E in Eq. (6)) and h in advance. In this work we will assume nearly incompressible materials such as soft biological tissues, rubbers or papers, and hence $\nu \approx 0.5$ is a reasonable approximation. No assumption will be made about the Young's modulus E. Instead, E can be factorized out of the compliance matrix, and h can be partially factorized out by normalizing the vector of forces as:

$$\Delta\mathbf{s} = \frac{\Delta\mathbf{f}^*}{Eh}\ . \tag{19}$$

With this normalization all unknown magnitudes are concentrated in the noise component of the surface state vector, and can be simultaneously processed by the EKF.

Algorithm 1 Online Non-Rigid EKF (EKF-FEM).

Input: Input sequence and rigid point labels $\mathcal{B}_p$
Output: 3D non-rigid shape and camera pose trajectory

while $\mathcal{I}_k$ **do**

I. EKF prediction
  if Computing_Rigid_Structure_at_Rest **then**
    $\mathbf{C}_k = \mathbf{0}$; $\mathbf{Q}_{\mathbf{y}} = \mathbf{0}$; Initialization (See Sect. 5.4)
  else
    $[\mathbf{C}_k] = \text{FEM}\big[\hat{\mathbf{x}}_{k-1|k-1}, \nu, h, \mathcal{B}_p\big]$ (Eq. 17)
    $\mathbf{Q}_{\mathbf{y}}$; Null for boundary points only
  end if
  $\hat{\mathbf{m}}_{k|k-1} = \mathbf{m}_{k+1}\left(\hat{\mathbf{m}}_{k-1|k-1}, 0, 0\right)$ (Eq. 15)
  $\hat{\mathbf{y}}_{k|k-1} = \mathbf{y}_{k+1}\left(\hat{\mathbf{y}}_{k-1|k-1}, 0\right)$ (Eq. 20)
  $\hat{\mathbf{x}}_{k|k-1} = \left[\hat{\mathbf{m}}^\top_{k|k-1}, \hat{\mathbf{y}}^\top_{k|k-1}\right]^\top$
  $\mathbf{P}_{k|k-1} = \mathbf{F}_k\mathbf{P}_{k-1|k-1}\mathbf{F}_k^\top + \mathbf{G}_k\mathbf{Q}_k\mathbf{G}_k^\top$ (Eqs. 24,25)

II. EKF data association
  $\hat{\mathbf{h}}_{k|k-1} = \mathbf{h}_k(\hat{\mathbf{x}}_{k|k-1})$ (Eq. 23)
  $\mathbf{S}_{k|k-1} = \mathbf{H}_{k|k-1}\mathbf{P}_{k|k-1}\mathbf{H}^\top_{k|k-1} + \mathbf{R}_k$ (Eq. 26)

III. EKF update
  $\mathcal{K}_{k|k-1} = \mathbf{P}_{k|k-1}\mathbf{H}^\top_{k|k-1}\mathbf{S}^{-1}_{k|k-1}$
  $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathcal{K}_{k|k-1}(\mathbf{z}_k - \hat{\mathbf{h}}_{k|k-1})$
  $\mathbf{P}_{k|k} = (\mathbf{I} - \mathcal{K}_{k|k-1}\mathbf{H}_{k|k-1})\mathbf{P}_{k|k-1}$
end while

Note however, that as the behavior matrix of the bending component $\mathbf{D}^b$ (see Appendix) depends on a h^2 factor, the material width h can not be completely factorized out of $\mathbf{C}$. In any event, this normalization is a remarkable contribution of our paper, as it lets us to easily handle both extensible and isometric materials.

If we now consider the surface configuration $\mathbf{y}_k$ at a time step k, and its associated compliance matrix $\mathbf{C}_k$, the new state estimate is computed via a simple additive transition state function:

$$\mathbf{y}_{k+1} \equiv \mathbf{y}_{k+1}\left(\mathbf{y}_k, \Delta\mathbf{s}\right) = \mathbf{y}_k + \mathbf{C}_k \Delta\mathbf{s}\ . \tag{20}$$

Note that with this equation we make it possible to bring the EKF-based formulation from a rigid to a non-rigid domain. Note also that while the stiffness matrix is quite sparse, its inverse, the compliance matrix, is dense. This provides a connection between all surface points and correlates all their displacements, i.e. the deformation of the scene in response to an applied force in a node, affects all nodes of the non-rigid scene. Another interesting point is that the compliance matrix $\mathbf{C}_k$ is re-estimated at each iteration, thus being adapted to the changing geometry of the shape. This allows correcting accumulation errors produced by the inherent linearization of the EKF, that might cause drifting problems.

Again, it will be necessary to associate a covariance matrix to this dynamic model. Since the normalized forces are expressed in length units[1], we represent the covariance matrix $\mathbf{Q_y}$ as a diagonal matrix whose elements encode deformation variances. These variances will be set to a constant value for all non-rigid nodes and to zero for the rigid ones. Yet, and as it will be made clear in the results, it is worth pointing that our dynamic model naturally codes anisotropic deformations.

5.3 Measurement Model

We next describe how the process of observing the mesh vertexes is modeled. Given the 3D coordinates of a vertex expressed in the world coordinate system $\mathcal{W}$, $\mathbf{g}_i = [x_i, y_i, z_i]^\top$, we initially use the pose components ($\mathbf{q}$ and $\mathbf{r}$) of the camera state vector to compute $\mathbf{g}_i^{\mathcal{C}}$, the expected position of the feature in the local coordinate system of the camera $\mathcal{C}$:

$$\mathbf{g}_i^{\mathcal{C}} = [x_i^{\mathcal{C}}, y_i^{\mathcal{C}}, z_i^{\mathcal{C}}]^\top = \underline{\mathrm{R}}\mathrm{ot}^\top(\mathbf{g}_i - \mathbf{r}), \tag{21}$$

where $\underline{\mathrm{R}}\mathrm{ot}$ is the rotation matrix corresponding to the quaternion $\mathbf{q}$. The measurement function $\boldsymbol{\pi}_i\left(\mathbf{m}, \mathbf{g}_i\right)$ returns the 2D projection (we assume a perspective calibrated camera) of $\mathbf{g}_i^{\mathcal{C}}$ onto the image, given the pose and shape values in the current state vector:

$$\boldsymbol{\pi}_i \equiv \boldsymbol{\pi}_i\left(\mathbf{m}, \mathbf{g}_i\right) = \begin{bmatrix} \beta_x - \alpha_x \frac{x_i^{\mathcal{C}}}{z_i^{\mathcal{C}}} \\ \beta_y - \alpha_y \frac{y_i^{\mathcal{C}}}{z_i^{\mathcal{C}}} \end{bmatrix}, \tag{22}$$

where (β_x, β_y) are the coordinates of the principal point of the camera and (α_x, α_y) its focal length values. In addition, in order to handle radial distortion, we further warp the perspective-projected coordinates according to a first order radial distortion model [29].

The measurement equations for the q mesh vertexes are stacked together into a unique non-linear measurement function of the state vector:

$$\mathbf{h}_k \equiv \mathbf{h}_k(\mathbf{x}) = \begin{bmatrix} \boldsymbol{\pi}_1 & \dots & \boldsymbol{\pi}_i & \dots & \boldsymbol{\pi}_q \end{bmatrix}^\top. \tag{23}$$

Each measurement is assigned a zero-mean Gaussian error with diagonal 2×2 covariance matrix $\Sigma_{\boldsymbol{\pi}_i}$. The global measurement noise covariance $\mathbf{R}_k$ is built by assembling these covariances into a block diagonal matrix.

5.4 Initialization

We initially assume that no forces are acting on the surface, which therefore behaves as a rigid object. In order to compute this initial shape, which we call *structure at rest*, we use the same non-rigid EKF framework we propose in this paper, but we set the covariance matrix $\mathbf{Q_y} = \mathbf{0}$. Note from Eq. (20) that setting this covariance to zero, the normalized forces vanish, and thus there is no deformation, i.e., $\mathbf{y}_{k+1} = \mathbf{y}_k$. $\mathbf{Q_m}$ is set to a constant diagonal matrix, as done in rigid SLAM [13].

Since depth cannot be computed from one single image, we proceed by first detecting Fast interest points [35] on the input image and creating a map of Inverse Depths [10] with them. We then move the camera around the object to capture several measurements of every feature from different viewpoints and turn the inverse depths to actual depths measurements. The structure at rest is finally computed by applying a Delaunay's tessellation of these points, yielding a triangular, yet not uniform, 3D mesh. Once the structure at rest is estimated, we switch back $\mathbf{Q_y}$ to its original value to model the non-rigid scene (Lines 3-8 in Algorithm 1).

Note that we could have chosen to define a mesh with uniformly distributed vertexes parameterized by the barycentric coordinates of the non-uniformly detected features, as in [32], [37], among others. While this would yield smoother results, we found it not to be realistic to handle practical situations in which non-textured areas require defining larger triangles than highly textured ones. Yet, both alternatives are technically equivalent.

5.5 Data Association

To perform data association we proceed as follows. During initialization, we associate a small rectangular image template to each feature. This template is defined as the rectangular patch surrounding the detected points in the first frame of the sequence. Then, at runtime, we predict the image coordinates $\hat{\boldsymbol{\pi}}_i$ of every keypoint feeding the current prediction estimate $\hat{\mathbf{x}}_{k|k-1}$ into the measurement model Eq. (23). In addition, the Jacobians of this function, $\mathbf{H}_k$ Eq. (26), are used to compute the uncertainty of the prediction, represented by the innovation covariance $\mathbf{S}_i$ (Line 15 in Alg. 1). This defines an ellipse on the input image (centered on $\hat{\boldsymbol{\pi}}_i$ and with size proportional to

1. Units of $[\Delta\mathbf{s}] = \frac{[\text{Force}]}{[\text{Pressure}][\text{Length}]} = [\text{Length}]$.

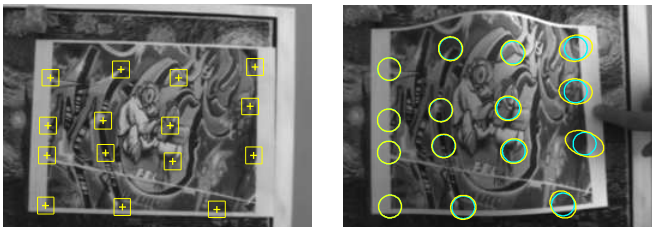

Fig. 6. **Data association. Left:** Some of the Fast interest points detected in the first frame of the sequence and their rectangular patches. **Right:** Search regions. Cyan ellipses correspond to the search regions if only a rigid model was considered. Yellow ellipses represent the actual predicted search areas, with an additional degree of uncertainty produced by the non-rigid component of the model. Note that for the left-most points –rigid boundary points– yellow and cyan ellipses coincide.

the standard deviation of $\mathbf{S}_i$), where to search for the matching observation. The pixel in the region yielding the highest template correlation, if over a threshold, is selected as the match. If none of the pixels scores over the threshold, the point is considered non-observed. The matched observations are stacked in the $\mathbf{z}_k$ vector (Line 18 in Alg. 1). This lets us to handle situations of self-occlusion or extreme deformation where points either disappear or can not be detected.

This guided search, is a key element of all EKF-SLAM methods, as allows for fast data association while minimizing the risk of mismatches due to image aliasing. As shown in Fig. 6, the search area is slightly larger in our formulation than in standard SLAM approaches for rigid objects, as it also contemplates an additional uncertainty term due to the surface deformation.

5.6 EKF Formulation

The proposed sequential non-rigid and monocular EKF-FEM is summarized in Algorithm 1. It is composed of the three main steps of an EKF: prediction, data association and updating (Lines 2, 13 and 16 of Alg. 1, respectively). Since this process is standard, we will skip details, and we just provide information about how the Jacobian matrices $\mathbf{F}_k$, $\mathbf{G}_k$ and $\mathbf{H}_k$, are assembled.

For the non-rigid case, the prediction stage is different to that of the rigid case, as besides the dynamic model of the camera, it includes one for the dynamic structure. Given an input image $\mathcal{I}_k$, $\mathbf{F}_k$ and $\mathbf{G}_k$ are the Jacobian matrices of the dynamic model with respect to the state vector and state noise respectively, and are defined as:

$$\mathbf{F}_k = \begin{bmatrix} \frac{\partial \mathbf{x}}{\partial \mathbf{m}} \\ \frac{\partial \mathbf{x}}{\partial \mathbf{y}} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{I}\Delta t & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathbf{q}_{k+1}}{\partial \mathbf{q}_k} & \mathbf{0} & \frac{\partial \mathbf{q}_{k+1}}{\partial \boldsymbol{\omega}_k^c} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad (24)$$

$$\mathbf{G}_k = \begin{bmatrix} \frac{\partial \mathbf{m}}{\partial \mathbf{n}} \\ \frac{\partial \mathbf{y}}{\partial \mathbf{n}} \end{bmatrix} = \begin{bmatrix} \mathbf{I}\Delta t & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathbf{q}_{k+1}}{\partial \Delta \boldsymbol{\omega}^c} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_k \end{bmatrix}, \quad (25)$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{n} = [\Delta \mathbf{v}^\top, \Delta \boldsymbol{\omega}^{c\top}, \Delta \mathbf{s}^\top]^\top$ is the state vector noise whose covariance matrix $\mathbf{Q}$, is block diagonal, composed of $\mathbf{Q_m}$ and $\mathbf{Q_y}$.

Given the set of q measurements of Eq. (23), the Jacobian matrix $\mathbf{H}_k$ is written as:

$$\mathbf{H}_k = \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right] = \begin{bmatrix} \frac{\partial \boldsymbol{\pi}_1}{\partial \mathbf{m}} & \frac{\partial \boldsymbol{\pi}_1}{\partial \mathbf{y}_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial \boldsymbol{\pi}_i}{\partial \mathbf{m}} & \mathbf{0} & \mathbf{0} & \frac{\partial \boldsymbol{\pi}_i}{\partial \mathbf{y}_i} & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial \boldsymbol{\pi}_q}{\partial \mathbf{m}} & \mathbf{0} & \cdots & \mathbf{0} & \frac{\partial \boldsymbol{\pi}_q}{\partial \mathbf{y}_n} \end{bmatrix}. \quad (26)$$

5.7 Computational Cost

One of the main virtues of the sequential formulation we propose is that it has a small computational load. We next provide details this complexity, and compare it against rigid versions of EKF and BA.

Let us first consider the cost of processing one single frame using EKF. In the rigid case the cost is $\mathcal{O}(n^3)$, when all features are measured, being n the state vector size. This is mainly due to the EKF update stage, as the cost of the prediction is negligible. Yet, for the non-rigid case, the complexity of the prediction significantly increases because it requires assembling and inverting the stiffness matrix, in Eq. (17), which is $\mathcal{O}(n^3)$ [18]. In any event, the total cost per frame remains $\mathcal{O}(n^3)$.

When considering an n-size rigid map observed by m cameras, the number of unknowns becomes $m+n$. Thus, when using EKF, we need to solve m steps $\mathcal{O}(n^3)$ resulting in $\mathcal{O}(mn^3)$ complexity. This can be done in real time for moderate values of n, preventing though, from a dense map computation. On the other hand, rigid BA approaches simultaneously solve for the map and all camera poses, which, theoretically would yield an $\mathcal{O}((n+m)^3)$ problem. However, since for the rigid case there are no constraints between camera poses or between map points, sparsity patterns can be exploited to reduce the complexity to $\mathcal{O}(nm^2 + m^3)$ [16], [40]. In addition, BA does not require processing the whole sequence, but just a small set $m_{BA} << m$ of keyframes, reducing even more the cost to $\mathcal{O}(nm_{BA}^2 + m_{BA}^3)$. This complexity, allows for much denser and more accurate estimations of rigid maps than when using EKF [40].

Yet, when dealing with non-rigid maps the advantages of the BA are lost. The map points change every frame, thus having to estimate mn map points. Additionally, the sparsity is lost due to interframe smoothing constraints –absent in the rigid case–. Hence, BA approaches need to solve an $\mathcal{O}((m+nm)^3) \approx \mathcal{O}((nm)^3)$ problem. For this scenario, EKF sequential methods are more adequate, because they still keep the $\mathcal{O}(mn^3)$ cost for m images. This holds on the fact that scene and cameras at previous time steps are marginalized-out and their effect in the current time step is coded in the $\mathcal{O}(n^2)$ covariance matrix.

6 Experimental Results

We now present the results obtained on synthetic and real image sequences, providing both qualitative and

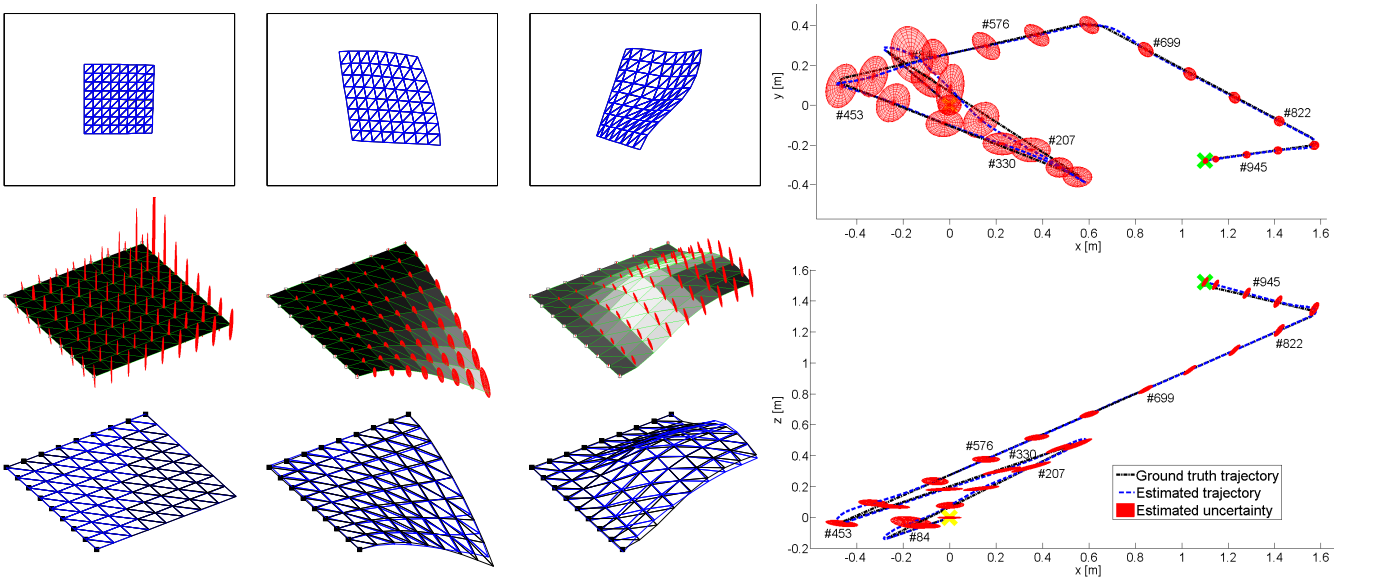

Fig. 7. **Results on the Synthetic Elastic Plate. Left:** 3D reconstruction results for selected frames #50, 650 and 950. Top: The reconstructed mesh (blue) is projected onto the input mesh (black), which is hardly visible as the projection is almost perfect. Middle: 3D view of the estimated mesh, where the red ellipsoids are the associated nodal uncertainties represented as a 95% confidence region. The real elasticity of the plate is coded with the white-black pattern, where white indicates larger elasticity, reaching levels where the patch area is increased by a factor $2\times$. Bottom: Ground truth 3D shape (black) and our estimate (blue), computed from the mean of the Gaussian distributions at each nodal position. **Right:** Estimated camera trajectory, seen from two viewpoints (X-Y and X-Z). Although our approach provides the whole 6-dof camera pose for each frame, for clarity we just plot the position its center, and the associated uncertainty at specific frames. Note that this path is consistent with the ground truth camera trajectory.

quantitative evaluation where we compare our EKF-FEM against other state-of-the-art approaches (*Please, see accompanying videos submitted as supplemental material*).

6.1 Parameter Tuning

The datasets we consider include deformable objects with different levels of extensibility, going from materials deforming isometrically, such as a sheet of paper, to very elastic materials such as a silicon cloth. We will show that our approach can naturally handle all this kind of deformations without explicitly knowing the true material properties. We just need a very simple tuning of the magnitude of the material width h and the covariance of the normalized forces $\Delta\mathbf{s}$, that model the expected standard deviation of the nodal displacement between consecutive frames. The values we have used for each experiment are summarized in Table 1.

In all cases we set the Poisson's ratio to $\nu = 0.499$ under the general assumption of a quasi incompressible material. h is roughly approximated to the width of the scenes (skin, silicone, paper and abdominal tissue). Note that small values of h allow rendering isometric behavior because, as we consider incompressibility, the element volume can only be maintained by keeping constant the element area. On the other hand, if h is larger, the volume may be maintained even if the element area is changed by either increasing or reducing the width of each element of the model. The amount of elastic deformation is further controlled by the magnitude of $\Delta\mathbf{s}$. As the actual acting forces on the surface do not follow a Gaussian model we set the parameter $\Delta\mathbf{s}$ to values

	h (mm)	ν	$\Delta\mathbf{s}$ (m)
Synthetic Elastic	1.5	0.499	$1.5 \cdot 10^{-4}$
Actress Face	1.5	0.499	$4.0 \cdot 10^{-6}$
Silicone Cloth	1.5	0.499	$2.5 \cdot 10^{-5}$
Bending Paper	0.1	0.499	$1.0 \cdot 10^{-9}$
Laparoscopic	3.5	0.499	$4.0 \cdot 10^{-6}$

TABLE 1
Parameter selection: h is the surface thickness, ν is the Poisson's ratio and $\Delta\mathbf{s} = \frac{\Delta\mathbf{f}^*}{Eh}$ is the normalized force.

larger than those expected theoretically. By doing this, we will search for the points of interest in larger image regions than those strictly necessary, slightly increasing the computation time, but ensuring an accurate data association for the non-rigidly deforming scene. Recall that this is one of the main advantages of the approach we propose: deviations in the values of the physical parameters ($\Delta\mathbf{s}$, h, and E), can be naturally bypassed by the EKF formulation, by just increasing the value of the covariance $\mathbf{Q_y}$ of the underlying surface motion model.

Regarding the camera motion, $\mathbf{Q_m}$ is tuned using the diagonal covariance matrix values proposed in [13]. We have verified that this tuning produces accurate data association during the initialization stage, and for the rigid points while observing the deforming non-rigid scene. Since the actual camera does not follow a simple smooth motion model, we again set the acceleration magnitudes to larger values than the expected camera accelerations in order to ensure a correct data association.

Finally, it is worth to mention that in all our experiments we choose between $n = [50 - 100]$ features per frame, yielding a similar amount of triangles. Although

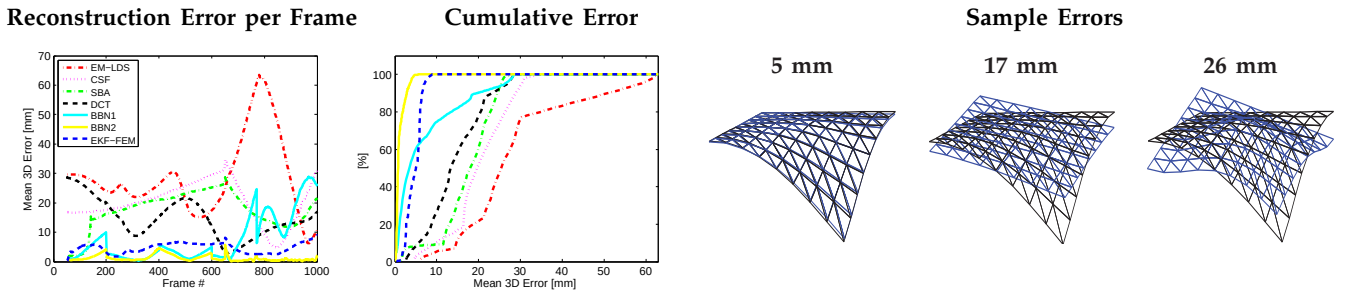

Fig. 8. **Synthetic Elastic Plate: Comparing EKF-FEM against SBA [34], EM-LDS [43], CSF [19], DCT [3] and BBN1-BBN2 [31]. Left:** 3D reconstruction error for every method at each frame of the sequence. **Middle:** Cumulative histogram of the reconstruction error. **Right:** Significance of the reconstruction error values. Black: ground truth; Blue: reconstructed mesh. Observe that the EKF-FEM error is around 5 mm, and thus it provides very accurate reconstructions. Similar results are obtained with the BBN2, but at the expense requiring good training data, representative of all deformations undergone by the plate.

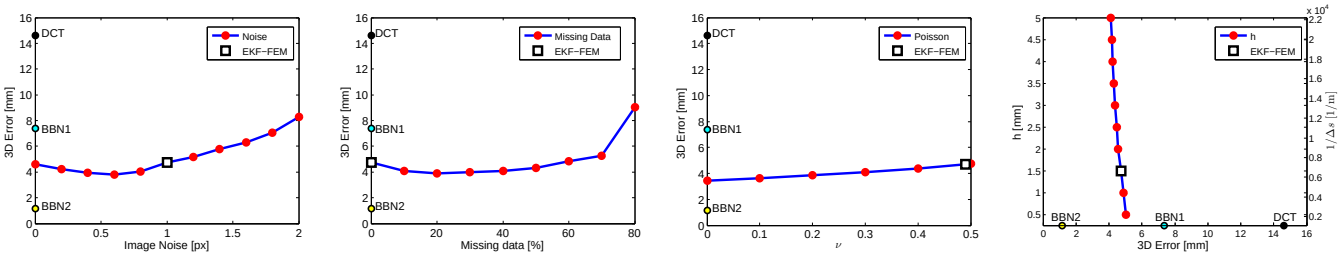

Fig. 9. **Synthetic Elastic Plate: Robustness against image noise, missing data and elastic parameter tuning.** Mean 3D reconstruction error of all 1000 frames as a function of the image noise, percentage of missing data, Poisson's ratio ν, thickness surface h and normalized forces Δs. In the right-most graph we simultaneously plot h and the inverse of the normalized forces Δs. In all cases, the white-filled square shows the parameters used in our original EKF-FEM formulation. Additionally, as a baseline reference, we show the mean error of the DCT [3] and BBN1-BBN2 [31] from Fig. 8. Note that the results we obtain remain within reasonable bounds for a wide range the tuned parameters, demonstrating that a fine tuning of parameters is not required.

stronger deformations might be better modeled by a larger number of triangles, we decided not doing so, in order to keep the computation time within reasonable bounds. Recall that at each frame we need to invert the $5n \times 5n$ stiffness matrix $\mathbf{K}$. Overall, we obtain computation times between 3 fps (Laparoscopic Sequence) and 8 fps (Synthetic Elastic Plate), using unoptimized Matlab code.

6.2 Synthetic Elastic Plate

In the first experiment we applied our approach to a 1000 frames sequence of a deforming elastic plate, generated with the general-purpose simulation tool Abaqus. During the first 50 frames, the plate remained rigid, in a flat 500×500 mm^2 configuration. Then it started to deform elastically, combining both membrane and bending effects, being the membrane component dominant between frames $51 - 650$, and the bending component dominant between frames $651-1000$. In order to generate the sequence we fed the simulator with all elastic parameters, the nodal forces, and as a boundary conditions we fixed the nodal position along two edges of the mesh. Figure 7 shows a few sample frames under different levels of deformation and elasticity. For some of the elements, the extent was increased by an order of $2\times$.

The sequence was observed by a monocular moving camera, following the trajectory shown in Fig. 1-left. The nodal points of the plate were projected on a 320×240 image, considering a perspective camera model with radial distortion and adding 1 pixel std 2D noise.

Figure 7 also shows the estimated shape for three different frames, and the complete camera path recovered by our approach. Note that in both cases, an uncertainty ellipsoid is associated to the estimated values, and proves the consistency of the results we obtain, as both the ground truth 3D shape and camera path, are within those ellipsoids. Indeed, if we compute the expected nodal positions (blue meshes in Fig. 7-bottom) and camera pose (blue paths in Fig. 7-right), we can observe that the results we obtain are very accurate and close from the ground truth.

In this experiment we also compare the performance of our approach (denoted EKF-FEM) against state-of-the-art methods, both sequential and batch algorithms. As a representative of the sequential methods, we evaluate the Sequential Bundle Adjustment optimization SBA proposed in [34], which, as our approach also requires having a static plate at the beginning of the sequence. Among the batch methods we consider: EM-LDS [43], that models object deformations using a Linear Dynamic System, learned using EM; DCT [3], that uses the Discrete Cosine Transform to express the 3D structure in an object independent basis; the Column Space Fitting (CSF) [19], which is similar in spirit to DCT, but uses higher-frequency components; and BBN [31], which formulates the problem using a Bayesian Belief Network, and uses the assumption that the shape is represented as a weighted sum of *known* PCA-learned modes. Note that this is indeed a strong assumption that heavily constrains the solution space and simplifies the problem.

In order to make a fair comparison, we considered two configurations of this approach, the BBN1, in which the PCA basis is trained with 20 sample plates taken between the frames $200-350$; and the BBN2 in which the 20 training sample plates are evenly distributed along the whole test sequence.

Figure 8-left plots the 3D reconstruction error of all methods. Note that our approach consistently outperforms all other techniques, except the BBN2. This was in fact expected, as this approach used an accurate deformation model learned from the ground truth data of the whole sequence. Yet, when this approach is just trained with the initial part of the sequence (BBN1), it fails in the final part of the sequence, where the plate undergoes more elastic deformations. Our approach, naturally handles all this kind of deformations, without explicitly using training data, which may be difficult to obtain in real applications. In addition, even when BBN2 performs better, our EKF-FEM yields reconstruction errors of about 5 mm, which are very small as seen from Fig. 8-right.

6.2.1 Robustness to Image Noise and Missing Data

We have also used the synthetic data to assess the robustness of our approach against increasing levels of noise and missing data in the image observations. The results are summarized in the two left most plots of Fig. 9. In regard to image noise, it can be observed a graceful degradation of the reconstruction error with increasing levels of noise. Yet, even with a noise of 2 pixels std, the results remain within reasonable bounds. The system also shows a nice performance with respect to random missing data, without significantly degrading until a breaking point in 70%. This is a consequence of the Bayesian formulation and the ability of the EKF to code the correlation among each pair of feature points in the covariance matrix. That is, when a feature point is observed in a new image, it is updated not only the corresponding state of that point but also the state of all other points. Thus, only a few observations are enough to produce most of the map update, allowing to cope with significant amounts of missing data.

6.2.2 Influence of the Elastic Parameter Tuning

We have evaluated the stability of the results against different settings of the elastic parameters discussed in Sec. 6.1. The two right-most plots in Fig. 9, depict the effect of changing the Poisson ratio ν, and the material thickness h from their true values. The variations are significant, ν varies within the range $[0, 0.5]$, and h is swept within $[0.5, 5]$ mm, (a factor $0.3\times$ to $3\times$ w.r.t the ground truth of 1.5 mm). Nevertheless, the effect in the mean error over the 1000 frames is very limited, lower than a 20% change, still being consistently more accurate than the closer competitors BBN1 [31] and DCT [3]. Indeed, for values different from ground truth, it can be observed a slight reduction of the overall reconstruction error. Yet, this is not surprising in a Bayesian formulation as ours, where uncertainties (of the camera, shape and motion) are modeled as Gaussian distributions, that are just rough approximations of reality. These uncertainties may better model situations where physical parameters do not exactly match the true ones.

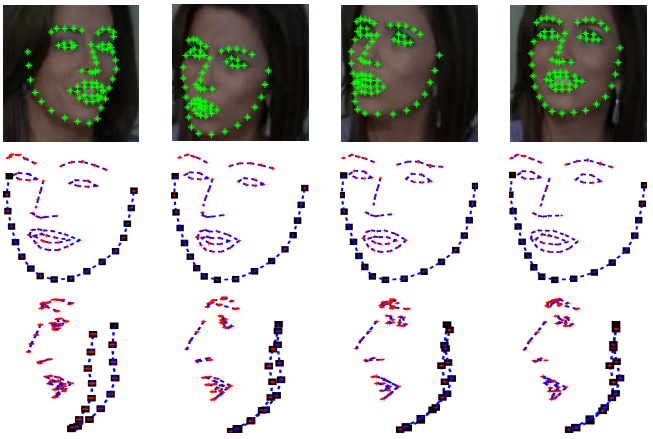

Fig. 10. **Results on the *Actress* sequence. Top**: Selected frames #24, 44, 64 and 84 with the 2D tracked features. **Middle and Bottom rows:** Two different views of the 3D reconstruction. Note, specially on the bottom row, that each of the 3D features has the associated uncertainty ellipsoid. In this case, though, the ellipsoids are very small as the inter-frame deformation is relatively small compared to the camera motion.

6.3 Real Images

We evaluated our approach on four experiments involving surfaces with very distinct behaviors: a human face, an elastic silicone cloth, a bending paper under isometric deformation, and a laparoscopy sequence of a rabbit abdominal cavity. We will see that our EKF-FEM can tackle all these situations, by just intuitively setting the parameters of Table 1, such that we assign higher values of the normalized force Δs to materials which are expected to undergo larger deformations. Due to a lack of 3D ground truth or long point tracks, other methods were not applicable in these experiments, except for the case of the *Actress Sequence*, where we could perform a qualitative comparison of the methods.

6.3.1 Actress Sequence

We tested our approach on a 102 frames sequence of an actress talking and moving her head. As 2D measured features, we used the sequence tracks provided by [4], and shown in Fig. 10-top. As a boundary points, we chose the points of the jaw. Since for this experiment the ground truth is not available, we only report qualitative results, by plotting the 3D reconstruction from a frontal and side view (two bottom rows of Fig. 10). Observe, from the side view, that the uncertainty associated with the EKF-FEM estimation is in this case very small. This is because the deformation of the face is relatively small, compared to the motion of the camera between consecutive frames. In Fig. 1 (second column) we depict the camera trajectory we have estimated.

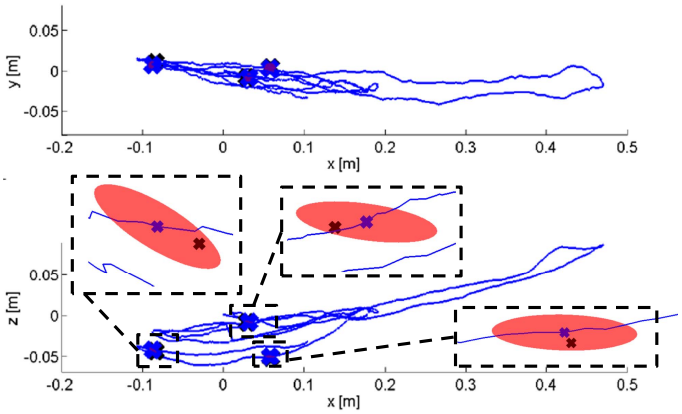

Fig. 11. **Estimated camera trajectory for the *Silicone Cloth* experiment.** (X-Y) and (X-Z) views of the estimated trajectory. Close-ups for three selected camera locations: Black crosses indicate the ground truth camera position, and blue crosses correspond to the center of the Gaussian that defines the estimated pose. The 95% confidence level of these estimations is represented by red ellipses. Note that the ground truth position lies in all cases inside these uncertainty regions. The associated reconstructed shape for these three frames is shown in Fig. 12.

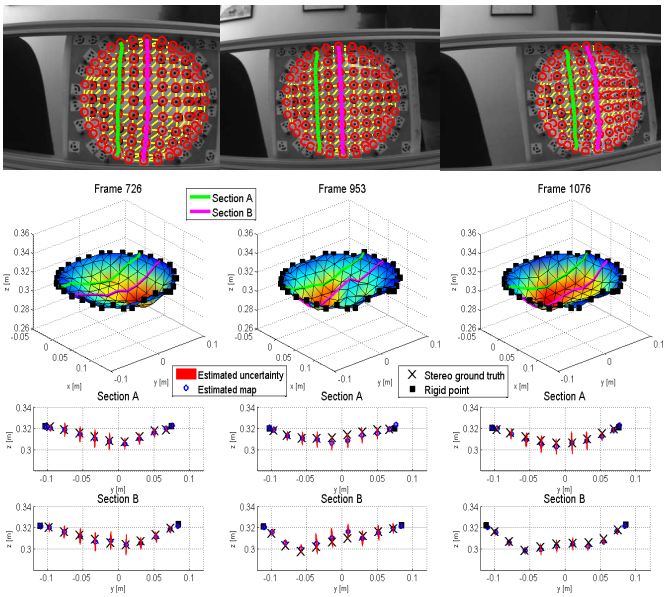

Fig. 12. **Reconstruction results for three frames of the *Silicone Cloth* experiment. Top**: Input images and estimated location of the points of interest, with their associated uncertainty (red ellipse). The color lines represent the cross-sectional views shown below. **Middle**: General view of the 3D reconstructed shape. The degree of extensibility of the mesh, compared to the structure at rest, is color-coded. Bluish regions are isometrically deformed, while reddish areas have undergone larger elastic deformations. **Bottom**: Two cross sections of the reconstructed surface, in which we represent, for each feature point, both the ground truth position computed using stereo (black crosses), and the estimated position (blue circles) with their 95% confidence regions (red ellipses). Note that all ground truth points fall inside these regions.

Since this scene is quasi-orthogonal (the object depth is small compared to the distance from the camera), the results are comparatively very similar to those obtained by [34], which makes the assumption of orthographic camera model. We provide a comparison with EM-LDS [43] and DCT [3] methods in appendix. BBN [31] was not applicable, as there was no 3D training data to build the deformation modes this method requires.

6.3.2 *Silicone Cloth Sequence*

To quantitatively evaluate the performance of the EKF-FEM with real extensible data, we used a hand-held waving stereo-rig to capture a sequence of a deforming elastic silicone cloth fixed to a circular stretcher to enforce the boundary conditions (Fig. 1-third column). Artificial circular markers were painted onto the surface to facilitate the feature detection both when computing the stereo ground truth (just for a few frames) and when applying the proposed method.

The 320×240 sequence of one of the stereo cameras was then used to test our approach, with the parameter setting given in Table 1. The data association was resolved using normalized cross correlation, as detailed in Sect. 5.5. Note that all map points have a similar texture patch. In this situation, the guided search constrained to the elliptical regions is essential to avoid mismatches.

Figure 11 shows the camera trajectory estimated by our approach. For three of the frames we plot the 95% confidence level ellipsoids, and validate that our approach is consistent in all cases, i.e., these confidence regions include the true camera positions, represented with black crosses. In Fig. 12 we show, for the same three frames, the reconstruction results. We depict two cross sectional views and the corresponding 95% confidence uncertainty ellipsoids. Note, as expected, that the maximum extend of these ellipsoids is along the viewing direction (the Z-axis). But in any event, these ellipsoids are rather small, and include the ground truth, confirming again the consistency of the estimated values.

If we compare the mean of these uncertainty distributions with the ground truth values, we obtain mean reconstruction errors below 2.5 mm, which is indeed fairly small considering that the silicone cloth has an approximate diameter of 200 mm, and the camera is located at more than 700 mm.

6.3.3 *Bending Paper Sequence*

We also tested our approach on a textured paper being smoothly bent (Fig 1-fourth column). The interest of this experiment is twofold: First, we analyze the behavior of our approach on a quasi-isometrically deforming material. And second, we demonstrate that we can handle frames where not all the features can be observed.

At initialization, well spread Fast interest points [35] are detected in the first frame of the sequence, and used to build a triangular mesh structure. Since the points of interest are not uniformly distributed, the mesh will not be regular. The leftmost points of the surface, are fixed and chosen as rigid points. Then, at runtime, we search for all these features in every input frame. When the correlation coefficient of the match is below a certain threshold, we consider the feature not to be correctly

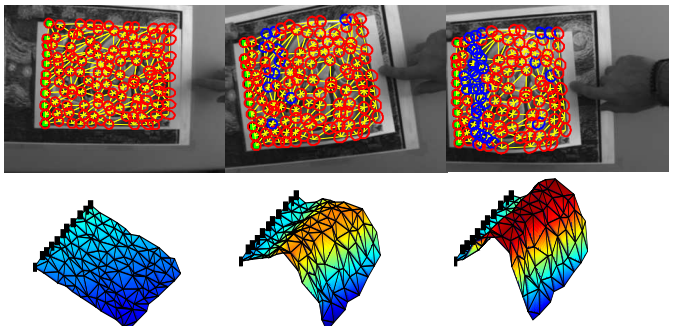

Fig. 13. **Bending Paper sequence. Top:** Reconstructed mesh overlaid on the image. Observed feature points are shown with their uncertainty ellipse in red. Unobserved keypoints, are highlighted with the ellipse in blue. **Bottom**: Side view of the mesh. The degree of deformation is represented in color, going from low (blue) to large deformation (red) levels.

matched and do not consider its measurement. This may happen due to appearance changes produced when the surface deforms or due to lack of visibility. In Fig. 13-top we highlight these features with blue ellipses. Yet, when the visibility improves the system is able to re-observe them again. The plots at the bottom of the figure show the reconstruction error, for increasing levels of bending.

6.3.4 Laparoscopic Sequence

As a final experiment, we present an in-vivo 3D reconstruction of a rabbit abdominal cavity from a 400 frames laparoscopic monocular sequence. This is a very challenging scenario as the laparoscope produces occlusions, the images are low resolution (288×384), and sudden camera motions often wash out the features making their observations unreliable. The rest shape was computed by moving the camera inside the cavity during an initial and passive exploration stage. As rigid points, we chose points located far apart from the region that was going to be deformed, and were assumed to be fixed. Once this was done, the surgeon performed an external tactile exam of the abdomen that produced the internal deformations. Figure 1 represents the estimated camera trajectory and Fig. 14 shows the reconstructed shapes for two frames.

7 Conclusion and Future Work

In this paper, Navier's equations, solved by means of FEM, have been embedded into a visual EKF-SLAM framework to provide a Bayesian estimation for non-rigid scenes. The proposed method can deal with a mixture of rigid and non-rigid scene points, to simultaneously estimate full 3D camera pose and 3D deformable structure, from the sole input of the image sequence.

Furthermore, we have shown that the proposed EKF-FEM can handle both isometric and elastic deformations without the need to accurately know material parameters. Yet, if these mechanical properties were available, the method could take advantage of it. This could be the case in medical images, as they are acquired under

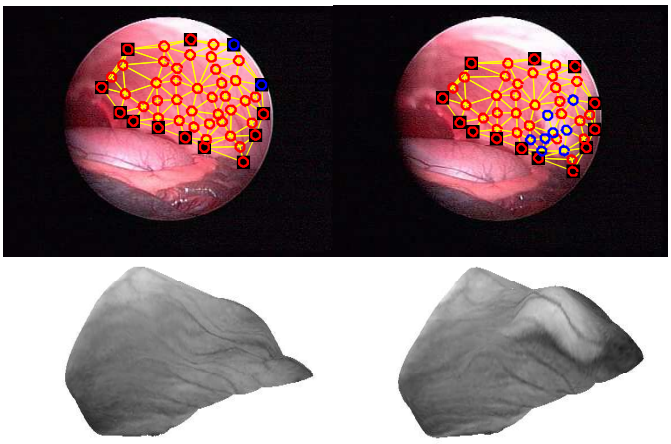

Fig. 14. **Laparoscopic sequence. Top:** Two selected frames, in which we overlay the estimated mesh that models the abdominal cavity. Some of the features are not detected (shown in blue). **Bottom**: 3D reconstructed shapes. Note the deformation in the upper-right side of the mesh, produced by the external tactile exam performed by the surgeon.

controlled conditions where accurate deformation models are readily available (we did not exploit this in the Laparoscopy experiment). Similarly, the high efficiency of our approach, makes it appropriate for tasks involving manipulation of deformable objects in real time (for instance, for automatic laundry handling), which we seek to explore in the future. In addition, we also intend to integrate additional sources of information into our observation model that could be useful for textureless materials. The use of silhouettes and shading cues will be explored for this purpose.

Acknowledgments

This work was partly funded by the MINECO projects Abdomesh DPI2011-27939-C02-01, RobInstruct TIN2014-58178-R and SVMap DPI2012-32168; by the ERA-net CHISTERA project VISEN PCIN-2013-047; and by a scholarship FPU12/04886 from the Spanish MECD. The authors thank P. Gotardo for fruitful discussions, and J. M. Bellon for the laparoscopy sequence.

References

[1] A. Agudo, B. Calvo, and J.M.M. Montiel. FEM models to code non-rigid EKF monocular SLAM. In *ICCVW*, 2011.
[2] A. Agudo, B. Calvo, and J.M.M. Montiel. Finite element based sequential bayesian non-rigid structure from motion. In *CVPR*, 2012.
[3] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Trajectory space: A dual representation for nonrigid structure from motion. *PAMI*, 33(7):1442–1456, 2011.
[4] A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S. Olsen, and P. Sayd. Coarse-to-fine low-rank structure-from-motion. In *CVPR*, 2008.
[5] A. Bartoli, Y. Gerard, F. Chadebecq, and T. Collins. On template-based reconstruction from a single view: Analytical solutions and proofs of well-posedness for developable, isometric and conformal surfaces. In *CVPR*, 2012.
[6] J. L. Batoz, K. J. Bathe, and L. W. Ho. A study of three-node triangular plate bending elements. *IJNME*, 15:1771–1812, 1980.
[7] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, 1999.
[8] M. Brand. A direct method of 3D factorization of nonrigid motion observed in 2D. In *CVPR*, 2005.

[9] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *CVPR*, 2000.
[10] J. Civera, A. Davison, and J.M.M. Montiel. Inverse depth parametrization for monocular SLAM. *TRO*, 24(5):932–945, 2008.
[11] L. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2D and 3D images. *PAMI*, 15(11):1131–1147, 1993.
[12] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *ECCV*, 1998.
[13] A. Davison. Real-time simultaneous localisation and mapping with a single camera. In *ICCV*, 2003.
[14] A. Del Bue, X. Llado, and L. Agapito. Non-rigid metric shape and motion recovery from uncalibrated images using priors. In *CVPR*, 2006.
[15] A. Ecker, A. Jepson, and K. Kutulakos. Semidefinite programming heuristics for surface reconstruction ambiguities. In *ECCV*, 2008.
[16] C. Engels, H. Stewénius, and D. Nistér. Bundle adjustment rules. In *Photogrammetric Computer Vision*, 2006.
[17] R. Garg, A.Roussos, and L.Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *CVPR*, 2013.
[18] G. Golub and C. Van Loan. *Matrix computations*. Johns Hopkins Univ Pr, 1996.
[19] P. Gotardo and A. Martinez. Computing smooth time-trajectories for camera and deformable shape in structure from motion with occlusion. *PAMI*, 33(10):2051–2065, 2011.
[20] P. Hammer, O. Marlowe, and A. Stroud. Numerical integration over simplexes and cones. *MTAC*, 10:130–137, 1956.
[21] R. Hartley and R. Vidal. Perspective nonrigid shape and motion recovery. In *ECCV*, 2008.
[22] Y. Kita. Elastic-model driven analysis of several views of a deformable cylindrical object. *PAMI*, 18(12):1150–1162, 1996.
[23] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *ISMAR*, 2007.
[24] A. Malti, R. Hartley, A. Bartoli, and J. H. Kim. Monocular template-based 3D reconstruction of extensible surfaces with local linear elasticity. In *CVPR*, 2013.
[25] I. Matthews and S. Baker. Active appearance models revisited. *IJCV*, 60(2):135–164, 2004.
[26] T. McInerney and D. Terzopoulos. A finite element model for 3D shape recognition and nonrigid motion tracking. In *ICCV*, 1993.
[27] T. McInerney and D. Terzopoulos. A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis. *Computational Medical Imaging and Graphics*, 19(1):69–83, 1995.
[28] D. Metaxas and D.Terzopoulos. Constrained deformable superquadrics and nonrigid motion tracking. *PAMI*, 15(6):580–591, 1993.
[29] E. Mikhail, J. Bethel, and J. McGlone. *Introduction to Modern Photogrammetry*. John Wiley & Sons, 2001.
[30] F. Moreno-Noguer and P. Fua. Stochastic exploration of ambiguities for non-rigid shape recovery. *PAMI*, 35(2):463–475, 2013.
[31] F. Moreno-Noguer and J. M. Porta. Probabilistic simultaneous pose and non-rigid shape recovery. In *CVPR*, 2011.
[32] F. Moreno-Noguer, M. Salzmann, V. Lepetit, and P. Fua. Capturing 3D stretchable surfaces from single images in closed from. In *CVPR*, 2009.
[33] C. Nastar and N. Ayache. Frequency-based nonrigid motion analysis: application to four dimensional medical images. *PAMI*, 18(11):1067–1079, 1996.
[34] M. Paladini, A. Bartoli, and L. Agapito. Sequential non rigid structure from motion with the 3D implicit low rank shape model. In *ECCV*, 2010.
[35] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *ECCV*, 2006.
[36] C. Russell, J. Fayad, and L. Agapito. Energy based multiple model fitting for non-rigid structure from motion. In *CVPR*, 2011.
[37] M. Salzmann, F. Moreno-Noguer, V. Lepetit, and P. Fua. Closed-form solution to non-rigid 3D surface registration. In *ECCV*, 2008.
[38] M. Salzmann and P.Fua. Linear local models for monocular reconstruction of deformable surfaces. *PAMI*, 33(5):931–944, 2011.
[39] S. Sclaroff and A. Pentland. Physically-based combinations of views: Representing rigid and nonrigid motion. In *MNRAO*, 1994.
[40] H. Strasdat, J.M.M Montiel, and A. Davison. Visual SLAM: Why Filter? *IMAVIS*, 30(2):65–77, 2012.
[41] L. Tao, S. J. Mein, W. Quan, and B. J. Matuszewski. Recursive non-rigid structure from motion with online learned shape prior. *CVIU*, 117(10):1287–1298, 2013.
[42] C. Tomasi and T.Kanade. Shape and motion from image streams under orthography:a factorization approach. *IJCV*, 9(2):137–154, 1992.
[43] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid structure-from motion: estimating shape and motion with hierarchical priors. *PAMI*, 30(5):878–892, 2008.
[44] L. Tsap, D. Goldof, and S. Sarkar. Nonrigid motion analysis based on dynamic refinement of finite element models. *PAMI*, 22(5):526–543, 2000.
[45] J. Xiao, J. Chai, and T. Kanade. A closed-form solution to non-rigid shape and motion. *IJCV*, 67(2):233–246, 2006.
[46] A. Young and L. Axel. Non-rigid wall motion using MR tagging. In *CVPR*, 1992.
[47] O. Zienkiewicz and R. Taylor. *The finite element method: Basic formulation and linear problems*. McGraw-Hill, 1989.
[48] O. Zienkiewicz and R. Taylor. *The finite element method: Solid and fluid mechanics, dynamics and non-linearity*. McGraw-Hill, 1989.

Antonio Agudo received the M.Sc. degree in industrial engineering and electronics in 2010 and M.Sc. degree in Computer Science in 2011, both at the Universidad de Zaragoza (UZ) with Extraordinary Award. He is working towards the Ph.D. degree at the I3A Robotics, Perception and Real-Time Group in the UZ. He was a visiting student at vision group of Queen Mary University of London and with the vision and imaging science group of University College London. His research interests include non-rigid structure from motion, monocular SLAM and computational mechanics.

Francesc Moreno-Noguer received the MSc degrees in industrial engineering and electronics from the Technical University of Catalonia (UPC) and the Universitat de Barcelona in 2001 and 2002, respectively, and the PhD degree from UPC in 2005. From 2006 to 2008, he was a postdoctoral fellow at the computer vision departments of Columbia University and the École Polytechnique Fédérale de Lausanne. In 2009, he joined the Institut de Robòtica i Informàtica Industrial in Barcelona as an associate researcher of the Spanish Scientific Research Council. His research interests include retrieving rigid and nonrigid shape, motion, and camera pose from single images and video sequences. He received UPC's Doctoral Dissertation Extraordinary Award for his work.

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.

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 non-robotic 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.

APPENDIX

For the completeness of the paper, we next describe the shape functions and their derivatives necessary to compute the strain-displacement matrix $\mathbf{B}$, and the behavior matrix $\mathbf{D}$, both for the membrane and bending components of the deformation. These matrices are used in Section 4.3.

A.1 Linear Shape Functions

Displacements due to membrane effect $\bar{\mathbf{u}}^m(\bar{x},\bar{y})$ within a triangular element, are interpolated by the nodal displacements $\bar{\mathbf{a}}_i^m = [\bar{u}_i, \bar{v}_i]^\top$ as Eq. (4):

$$\bar{\mathbf{u}}^m(\bar{x},\bar{y}) = \sum_{i=1}^{3} N_i^l(\xi,\eta)\bar{\mathbf{a}}_i^m$$

where N_i^l denote linear (l) element shape functions [47] (see Fig. 15):

$$N_1^l = 1-\xi-\eta \qquad N_2^l = \xi \qquad N_2^l = \eta \tag{27}$$

Let us define the arrays $\mathbf{M}_1 = [N_1^l, 0, N_2^l, 0, N_3^l, 0]^\top$ and $\mathbf{M}_2 = [0, N_1^l, 0, N_2^l, 0, N_3^l]^\top$. We can then write the interpolation in matrix form as:

$$\bar{\mathbf{u}}^m(\bar{x},\bar{y}) = \begin{bmatrix}\mathbf{M}_1^\top \\ \mathbf{M}_2^\top\end{bmatrix}\begin{bmatrix}\bar{\mathbf{a}}_1^m \\ \bar{\mathbf{a}}_2^m \\ \bar{\mathbf{a}}_3^m\end{bmatrix}.$$

For computing the strain-displacement matrix $\mathbf{B}^m$, we will need the following derivatives of $\mathbf{M}_1$ and $\mathbf{M}_2$ with respect to the natural coordinates (ξ,η):

$$\begin{aligned}
\frac{\partial \mathbf{M}_1}{\partial \xi} &= \mathbf{M}_{1,\xi} = \begin{bmatrix}-1 & 0 & 1 & 0 & 0 & 0\end{bmatrix}^\top \\
\frac{\partial \mathbf{M}_1}{\partial \eta} &= \mathbf{M}_{1,\eta} = \begin{bmatrix}-1 & 0 & 0 & 0 & 1 & 0\end{bmatrix}^\top \\
\frac{\partial \mathbf{M}_2}{\partial \xi} &= \mathbf{M}_{2,\xi} = \begin{bmatrix}0 & -1 & 0 & 1 & 0 & 0\end{bmatrix}^\top \\
\frac{\partial \mathbf{M}_2}{\partial \eta} &= \mathbf{M}_{2,\eta} = \begin{bmatrix}0 & -1 & 0 & 0 & 0 & 1\end{bmatrix}^\top
\end{aligned}$$

The same linear shape functions are also used to interpolate the geometry (i.e., the 3D position) of any point within the normalized triangle:

$$\bar{\mathbf{g}}(\bar{x},\bar{y}) = \sum_{i=1}^{3} N_i^l(\xi,\eta)\bar{\mathbf{g}}_i$$

A.2 Quadratic Shape Functions

Displacements due to bending effect $\bar{\mathbf{u}}^b(\bar{x},\bar{y})$ within the normalized triangle are interpolated by the bending components of the nodal displacements $\bar{\mathbf{a}}_i^b = [\bar{w}_i, \theta_{\bar{x}i}, \theta_{\bar{y}i}]^\top$. The displacement $\bar{w}$ is estimated using the linear shape of Eq. (27):

$$\bar{w} = \sum_{i=1}^{3} N_i^l(\xi,\eta)\bar{w}_i$$

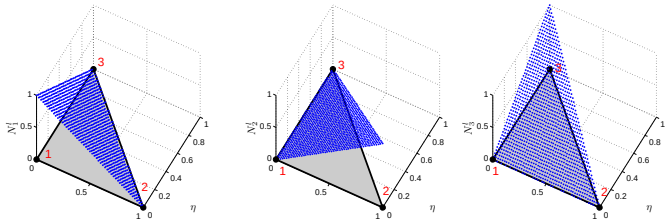

Fig. 15. Linear shape functions N^l (blue surfaces) used to interpolate data within the normalized triangle (gray) when describing geometry and membrane deformations.

The rotation components, on the other hand, are interpolated using quadratic shape functions:

$$\begin{bmatrix}\theta_{\bar{x}} \\ \theta_{\bar{y}}\end{bmatrix} = \begin{bmatrix}\mathbf{N}_1^\top \\ \mathbf{N}_2^\top\end{bmatrix}\begin{bmatrix}\bar{\mathbf{a}}_1^b \\ \bar{\mathbf{a}}_2^b \\ \bar{\mathbf{a}}_3^b\end{bmatrix}$$

where $\mathbf{N}_1 = [N_{x1},\ldots,N_{x9}]^\top$, $\mathbf{N}_2 = [N_{y_1},\ldots,N_{y9}]^\top$, and the shape functions are defined as [6]:

$$\begin{aligned}
d=\{1,4,7\}&\begin{cases} N_{xd} &= \frac{1}{4}(p_a N_a^q - p_b N_b^q) \\ N_{yd} &= \frac{1}{4}(t_a N_a^q - t_b N_b^q)\end{cases} \\
d=\{2,5,8\}&\begin{cases} N_{xd} &= \frac{1}{4}(q_a N_a^q + q_b N_b^q) \\ N_{yd} &= -N_m^q + e_a N_a^q + e_b N_b^q\end{cases} \\
d=\{3,6,9\}&\begin{cases} N_{xd} &= N_m^q - c_a N_a^q - c_b N_b^q \\ N_{yd} &= -\frac{1}{4}(q_a N_a^q + q_b N_b^q)\end{cases}
\end{aligned} \tag{28}$$

where for $d=\{1,2,3\}$ we use the triplet $\{m,a,b\} = \{1,6,5\}$; for $d=\{4,5,6\}$ we set $\{m,a,b\}=\{2,4,6\}$; and for $d=\{7,8,9\}$ we set $\{m,a,b\}=\{3,5,4\}$. The N_i^q are quadratic shape functions, defined in natural coordinates (ξ,η) as:

$$\begin{aligned}
N_1^q &= (1-\xi-\eta)(1-2(\xi+\eta)) & N_2^q &= \xi(2\xi-1) \\
N_3^q &= \eta(2\eta-1) & N_4^q &= 4\xi\eta \\
N_5^q &= 4\eta(1-\xi-\eta) & N_6^q &= 4\xi(1-\xi-\eta)
\end{aligned}$$

The shape of these six quadratic functions over the normalized triangle is shown in Fig. 16.

The coefficients $\{p_k, q_k, t_k, c_k, e_k\}$ in Eq. (28) are computed by:

$$\begin{aligned}
x_{ij} &= \bar{x}_i - \bar{x}_j & y_{ij} &= \bar{y}_i - \bar{y}_j & l_{ij}^2 &= x_{ij}^2 + y_{ij}^2 \\
p_k &= \frac{-6x_{ij}}{l_{ij}^2} & q_k &= \frac{3x_{ij}y_{ij}}{l_{ij}^2} & t_k &= \frac{-6y_{ij}}{l_{ij}^2} \\
c_k &= \frac{\frac{1}{4}x_{ij}^2 - \frac{1}{2}y_{ij}^2}{l_{ij}^2} & e_k &= \frac{\frac{1}{4}y_{ij}^2 - \frac{1}{2}x_{ij}^2}{l_{ij}^2}
\end{aligned}$$

for the triplets $\{k,i,j\} = \{4,2,3\},\{5,3,1\},\{6,1,2\}$.

To compute the strain-displacement matrix $\mathbf{B}^b$, we will need the following derivatives of $\mathbf{N}_1$ and $\mathbf{N}_2$ with

respect to the natural coordinates (ξ, η):

$$\mathbf{N}_{1,\xi} = \begin{bmatrix} p_6(1-2\xi)+(p_5-p_6)\eta \\ q_6(1-2\xi)-(q_5+q_6)\eta \\ -3+4(\xi+\eta)-4c_6(1-2\xi)+(c_5+c_6)4\eta \\ -p_6(1-2\xi)+(p_4+p_6)\eta \\ q_6(1-2\xi)+(q_4-q_6)\eta \\ -1+4\xi-4c_6(1-2\xi)-(c_4-c_6)4\eta \\ -(p_4+p_5)\eta \\ (q_4-q_5)\eta \\ -(c_4-c_5)4\eta \end{bmatrix}$$

$$\mathbf{N}_{1,\eta} = \begin{bmatrix} -p_5(1-2\eta)+(p_5-p_6)\xi \\ q_5(1-2\eta)-(q_5+q_6)\xi \\ -3+4(\xi+\eta)-4c_5(1-2\eta)+(c_5+c_6)4\xi \\ (p_4+p_6)\xi \\ (q_4-q_6)\xi \\ -(c_4-c_6)4\xi \\ p_5(1-2\eta)-(p_4+p_5)\xi \\ q_5(1-2\eta)+(q_4-q_5)\xi \\ -1+4\eta-4c_5(1-2\eta)-(c_4-c_5)4\xi \end{bmatrix}$$

$$\mathbf{N}_{2,\xi} = \begin{bmatrix} t_6(1-2\xi)+(t_5-t_6)\eta \\ 3-4(\xi+\eta)+4e_6(1-2\xi)-(e_5+e_6)4\eta \\ -q_6(1-2\xi)+(q_5+q_6)\eta \\ -t_6(1-2\xi)+(t_4+t_6)\eta \\ 1-4\xi+4e_6(1-2\xi)+(e_4-e_6)4\eta \\ -q_6(1-2\xi)-(q_4-q_6)\eta \\ -(t_4+t_5)\eta \\ (e_4-e_5)4\eta \\ -(q_4-q_5)\eta \end{bmatrix}$$

$$\mathbf{N}_{2,\eta} = \begin{bmatrix} -t_5(1-2\eta)+(t_5-t_6)\xi \\ 3-4(\xi+\eta)+4e_5(1-2\eta)-(e_5+e_6)4\xi \\ -q_5(1-2\eta)+(q_5+q_6)\xi \\ (t_4+t_6)\xi \\ (e_4-e_6)4\xi \\ -(q_4-q_6)\xi \\ t_5(1-2\eta)-(t_4+t_5)\xi \\ 1-4\eta+4e_5(1-2\eta)+(e_4-e_5)4\xi \\ -q_5(1-2\eta)-(q_4-q_5)\xi \end{bmatrix}$$

A.3 Strain-Displacement Matrices B

Using all previous elements, we can finally write the strain-displacement $\mathbf{B}$ matrix for membrane and bending effect as follows:

$$\mathbf{B}^m = \frac{1}{|\mathbf{J_g}|}\begin{bmatrix} J_{22}\mathbf{M}_{1,\xi}^\top - J_{12}\mathbf{M}_{1,\eta}^\top \\ J_{11}\mathbf{M}_{2,\eta}^\top - J_{21}\mathbf{M}_{2,\xi}^\top \\ J_{11}\mathbf{M}_{1,\eta}^\top - J_{21}\mathbf{M}_{1,\xi}^\top + J_{22}\mathbf{M}_{2,\xi}^\top - J_{12}\mathbf{M}_{2,\eta}^\top \end{bmatrix},$$

$$\mathbf{B}^b = \frac{1}{|\mathbf{J_g}|}\begin{bmatrix} J_{22}\mathbf{N}_{1,\xi}^\top - J_{12}\mathbf{N}_{1,\eta}^\top \\ J_{11}\mathbf{N}_{2,\eta}^\top - J_{21}\mathbf{N}_{2,\xi}^\top \\ J_{11}\mathbf{N}_{1,\eta}^\top - J_{21}\mathbf{N}_{1,\xi}^\top + J_{22}\mathbf{N}_{2,\xi}^\top - J_{12}\mathbf{N}_{2,\eta}^\top \end{bmatrix},$$

where J_{ij} are the entries of $\mathbf{J_g} = \partial\bar{\mathbf{g}}/\partial\boldsymbol{\xi}$, the Jacobian matrix of the transformation from natural to local coordinates.

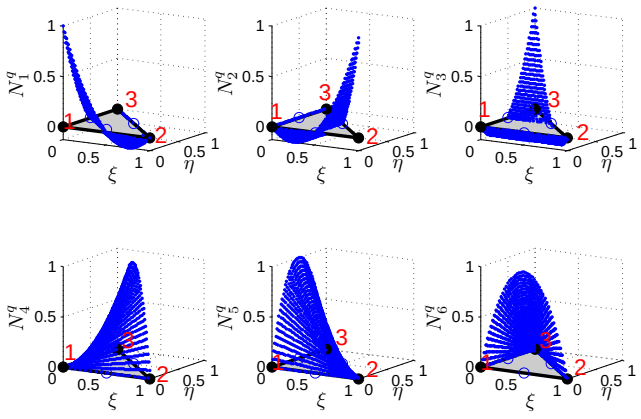

Fig. 16. Quadratic shape functions N^q (blue surfaces) used to interpolate data within the normalized triangle (in gray) when describing bending deformations.

A.4 Behavior Matrices D

Finally, the behavior matrix $\mathbf{D}$ for membrane and bending effect are:

$$\mathbf{D}^m = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}, \quad (29)$$

$$\mathbf{D}^b = \frac{Eh^2}{12(1-\nu^2)}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}.$$

A.5 Additional Results

We next report two additional results that complete the experimental section. The first demonstrates the advantage of combining rigid and non-rigid points in a pose estimation task. The second experiment complements the results of in Sec. 6.3.1 and Fig. 10.

Trajectory Estimation using Rigid and Non-Rigid Points

The EKF formulation we propose naturally handles both rigid and non-rigid points. Once the rigid points are identified, they are assigned null values in the dynamic covariance matrix $\mathbf{Q_y}$, and treated identically as non-rigid points, whose entries in $\mathbf{Q_y}$ are non-zero. This joint processing is advantageous. In order to proof this empirically we have performed a very simple experiment with the synthetic data of Sec. 6.2, and have computed camera pose using either the rigid points or these in combination with the non-rigid ones. In both cases we use exactly the same EKF formulation. As shown in Fig. 17 using all points (rigid and non-rigid) brings important accuracy improvements, reducing the relative error or the camera location from 10.12% to 4.58%.

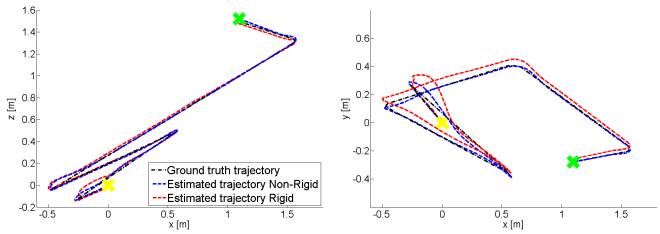

Fig. 17. **Trajectories comparison.** We display two viewpoints (X-Y and X-Z) for the ground truth trajectory in black, the estimated trajectory using both rigid and non-rigid points in blue, and finally the estimated trajectory using only rigid points in red. The mean relative Euclidean error is reduced from 10.12% to 4.58% when we use all rigid and non-rigid points.

Actress Sequence: Additional Results

For the *actress sequence*, ground truth shapes are not given. We run several other approaches using the provided point tracks and the visual differences of the reconstructed shapes between all methods are subtle but informative. See some frames in Fig. 18. Note that EM-LDS [43] (first row) yields a quasi-rigid solution, with almost no deformation adaption. On the other hand, DCT [3] does produce non-rigid changes of the face, but as seen from the side-views, the resulting 3D face seems to contain large errors, e.g. the two jaws do not have a symmetric depth. Furthermore, we found these approaches to be very sensitive to the number of modes in the basis, and in particular we did not manage to make CSF [19] work with this sequence.

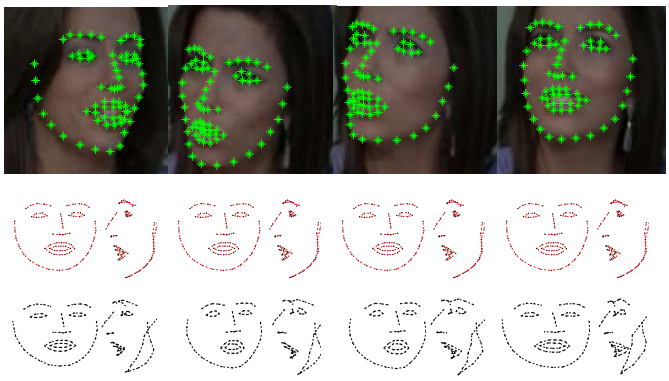

Fig. 18. **Actress sequence comparison. Top:** Selected frames #24, 44, 64 and 84 with the 2D tracked features. Two views of the 3D reconstruction corresponding to frames. **Top:** Using EM-LDS [43]. **Bottom:** Using DCT [3].