Real-Time 3D Reconstruction of Non-Rigid Shapes with a Single Moving Camera

Antonio Agudo [a,*], Francesc Moreno-Noguer [a], Begoña Calvo [b], J.M.M. Montiel [b]

[a] *Institut de Robòtica i Informàtica Industrial, Consejo Superior de Investigaciones Científicas (CSIC), 08028, Barcelona, Spain*

[b] *Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, 50018, Zaragoza, Spain*

Abstract

This paper describes a real-time sequential method to simultaneously recover the camera motion and the 3D shape of deformable objects from a calibrated monocular video. For this purpose, we consider the Navier-Cauchy equations used in 3D linear elasticity and solved by finite elements, to model the time-varying shape per frame. These equations are embedded in an extended Kalman filter, resulting in sequential Bayesian estimation approach. We represent the shape, with unknown material properties, as a combination of elastic elements whose nodal points correspond to salient points in the image. The global rigidity of the shape is encoded by a stiffness matrix, computed after assembling each of these elements. With this piecewise model, we can linearly relate the 3D displacements with the 3D acting forces that cause the object deformation, assumed to be normally distributed. While standard finite-element-method techniques require imposing boundary conditions to solve the resulting linear system, in this work we eliminate this requirement by modeling the compliance matrix with a generalized pseudoinverse that enforces a pre-fixed rank. Our framework also ensures surface continuity without the need for a post-processing step to stitch all the piecewise reconstructions into a global smooth shape. We present experimental results using both synthetic and real videos for different scenarios ranging from isometric to elastic deformations. We also show the consistency of the estimation with respect to 3D ground truth data, include several experiments assessing robustness against artifacts and finally, provide an experimental validation of our performance in real time at frame rate for small maps.

Key words: Deformable Shape and Camera Motion Recovery, Extended Kalman Filter, Finite Element Method, Tracking, Real-Time Vision.

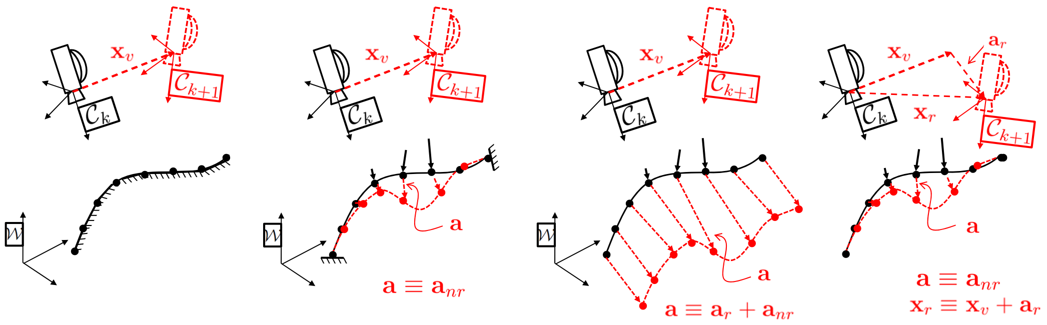
1 Introduction

Recovering live motion and the 3D reconstruction of a scene structure from a single moving video camera has potential applications in very different domains such as robotics, medical imaging, augmented reality, wearable computing and the automotive sector. This problem is known as Structure from Motion (SfM), and is one of the most active research areas in computer vision. In the last decade, significant progress has been made to sequentially solve this problem in real time for a sparse set of salient points by means of filtering techniques [12,16] based on the Extended Kalman Filter (EKF) or optimization techniques [26,28,37] based on Bundle Adjustment (BA). More recently, these approaches have been extended to produce per-pixel dense reconstruction in [38] for hand-held devices, or even for aerial vehicles [52]. While SfM is now considered to be a mature field, these methods depend heavily on a rigidity assumption of the structure and they can not be applied to scenes undergoing non-rigid deformations. In these cases, recovering the 3D shape of a non-rigid object is an inherently ill-posed problem since many different shapes can produce very similar image measurements. Additional a priori knowledge of the camera motion and shape deformation has to be considered to limit the possible range of solutions.

To overcome the limitations of rigid methods, Non-Rigid Structure from Motion (NRSfM) methods have been proposed. Most of them encode deformations by means of low-rank models, using the shape space [6,15,17,41,49] or the trajectory space [5,19,20] in addition to orthonormality constraints on camera motion. However, although these methods are now capable of obtaining accurate 3D reconstructions of deformable objects and camera motion, in contrast to their rigid counterparts, they remain behind when it comes to real-time performance. This is because they can only process the image sequence in a batch manner, after the complete acquisition, limiting their applicability in real-time and on-line applications. Only recently a few sequential approaches have been proposed [4,40,47]. Although this is a promising direction, these methods remain slow [40,47] or assume rigid priors that are not always available [4]. Furthermore, these methods use the 2D tracking data as input, *i.e.,* they do not normally compute the tracking and data association on-the-fly [40,47]. While non-rigid shapes have been previously registered in real time [39,42], these methods do not estimate the camera pose, assumed to be known in advance.

In contrast to all these previous approaches, in this work we propose a method

[*] Corresponding author. Present address: Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Llorens i Artigas, 4-6, E-08028, Barcelona, Spain.
Email address: aagudo@iri.upc.edu (Antonio Agudo).


$\mathbf{x}_v$
$\mathcal{C}_k$
$\mathcal{C}_{k+1}$
$\mathcal{W}$
$\mathbf{a}$
$\mathbf{a} \equiv \mathbf{a}_{nr}$
$\mathbf{a} \equiv \mathbf{a}_r + \mathbf{a}_{nr}$
$\mathbf{a}_r$
$\mathbf{x}_r$
$\mathbf{x}_r \equiv \mathbf{x}_v + \mathbf{a}_r$

(a) (b) (c) (d)

Fig. 1. **Marginalizing the non-rigid motion. (a):** A moving camera with rigid motion $\mathbf{x}_v$ observes a rigid structure. **(b):** A moving camera observes a structure with rigid and non-rigid points. The rigid points (boundary conditions) prevent rigid motions of the structure, which deforms non-rigidly $\mathbf{a}_{nr}$. **(c):** Removing rigid points allows rigid motions for both camera $\mathbf{x}_v$ and shape $\mathbf{a}_r$, while the shape also deforms non-rigidly $\mathbf{a}_{nr}$. **(d):** The two types of rigid motions are combined in a unique camera motion $\mathbf{x}_r$, allowing the non-rigid deformation $\mathbf{a}_{nr}$ to be recovered.

to sequentially estimate the camera pose and the 3D reconstruction of deformable objects directly from monocular video. Although our method uses a physically-inspired model, we do not assume any prior over rigid points like those typically required when exploiting finite-element approximations [4,23,29]. Our method uses a full perspective calibrated camera, automatically establishes correspondences between consecutive frames, and can run sequentially in real time at frame rate.

In order to relax the rigidity prior, let us consider a camera undergoing an unknown motion while observing an unknown rigid shape (see Fig. 1(a)). Both camera trajectory $\mathbf{x}_v$ and 3D structure can be estimated, up to scale factor, merely from the sole input of the image sequence gathered by the camera. Simultaneous-Localization-And-Mapping (SLAM) methods [16,26] have been proposed in robotics and augmented reality to address this problem in a sequential fashion. EKF-based approaches [12,16] were the first able to experimentally prove that visual SLAM in real-time is possible. More recently, the rigidity assumption was weakened in [1,4], allowing non-rigid deformations in the scene. To do this, the EKF estimator was combined with the Finite Element Method (FEM) to code the non-rigidity of the scene, resulting in a SfM system –denoted as EKF-FEM– able to deal with scenes containing both rigid and non-rigid map points. In the EKF-FEM formalism, the deformable object to be reconstructed is modeled as a finite set of patches –denoted as finite elements– whose displacement is ruled by the mechanics of deformable solids. In order to apply EKF-FEM, it is necessary to identify several points undergoing a rigid motion to establish the boundary conditions of the FEM problem. This method has the advantage of permitting to combine rigid and non-rigid points in the same framework, allowing the absolute camera trajectory to be recovered while estimating the displacement $\mathbf{a}$ of the scene due to the non-rigid motion $\mathbf{a}_{nr}$ (Fig. 1(b)). Unfortunately, a prior knowledge of which scene points undergo rigid motions may not be available in practice, preventing the applicability of these methods in purely non-rigid scenes. Our main contribution is therefore to propose a novel EKF-FEM formulation without the need for such rigid priors, that we will denote as EKF-FEM-FRP (from "Free Rigid Priors").

However, if rigid priors are not considered, both camera and scene can undergo an *interchangeable* rigid motion (Fig. 1(c)). Additionally, as the scene can suffer a non-rigid deformation, its motion $\mathbf{a}$ will be the combination of rigid $\mathbf{a}_r$ and non-rigid $\mathbf{a}_{nr}$ motions. In this work, since recovering both rigid motions without rigid priors is unconstrained, we propose to concentrate the total rigid motion in a single pose vector $\mathbf{x}_r \equiv \mathbf{x}_v + \mathbf{a}_r$, and the non-rigid displacements as $\mathbf{a} \equiv \mathbf{a}_{nr}$ (Fig. 1(d)). For this purpose, we propose encoding both camera pose and shape relative to the last camera location, following a robo-centric formulation [10], that we will denominate camera-centric. In this context, the solution of the linear FEM system has to be approximated by means of a generalization of the inverse FEM stiffness matrix. With these ingredients, we present the first approach –to the best of our knowledge– for simultaneously estimating camera pose and the 3D reconstruction of deformable objects from monocular images in real time without requiring any learning step.

This paper combines and extends two previous conference papers [2,3]. Here, we integrate these preliminary publications into a comprehensive presentation of our real-time sequential approach to recover non-rigid shapes. We analyze the rank deficiency of our stiffness matrices and present a linear estimation of the compliance matrix. In addition, we provide new experiments to show the performance of our approach studying the influence with respect to several parameters, such as the amount of noise or missing data, and the physical parameters of the model. We also provide a comparison with state-of-the-art algorithms and show an experimental validation of our performance in real time at frame rate.

The remainder of this paper is organized as follows. Related work is presented in Section 2. In Section 3 we define our new physics-based deformation model. In Section 4 we show how to solve the resulting linear system without knowing rigid points. This is followed in Section 5 by a description of our sequential algorithm to estimate camera motion and deformable shape from monocular video. In Section 6 we describe our experimental evaluation and present a comparison with respect to other approaches. Our conclusions are summarized in Section 7. Finally, to keep the paper self-contained, we provide an appendix with details of the FEM formulation.

2 Related Work

In the past two decades, physics-based models have been used for animation and simulation purposes in computer graphics [7,53]. At the same time, similar models have been proposed in computer vision to restrict the set of possible deformations of an object. For instance, in the medical imaging domain, early approaches proposed deformable superquadrics [33], balloons forces [13] or spring models [25]. In [31,32] were proposed more closely a FEM, modeling the surface as a thin-plate with acting forces. However, these approaches were only valid for capturing relatively small deformations, or to handle volumetric images such as in tomography. Similarly, the boundary element method was proposed to track non-rigid objects in 2D [21] and 3D [22]. Later, more accurate representations were achieved using non-linear FEM for large deformations of beam [24] and for 3D solid structures [23,50]. However, their applicability was limited to very specific geometries for which the material properties were known. While non-linear FEM models can be very accurate when the knowledge is rich, they are normally very complex and can only be adapted to very specific applications.

More recently, linear FEM models have been proposed to recover non-linear deformations [4]. The key insight was to approximate the non-linear problem by means of a linear problem per frame with strong feedback coming from the measurements at frame rate. Although the linear FEM formulation is only valid for small deformations [7,53], it is only used to predict the deformation between frames that it can be linearly approximated. For this purpose, a linear FEM problem is solved per image frame, recomputing a new stiffness matrix with the previous state, *i.e.* using a new shape at rest, and hence allowing sequences with large scene deformations. Simply by the adoption of the FEM formalism, the surface deformation continuity is naturally enforced without needing to resort to any additional post processing. In addition, most of the material properties can be factorized out and do not have to be known in advance. Another method using linear elasticity has recently been proposed in the related problem of shape-from-template [29]. In this case, the elastic model is used to code an extensible deformation optimizing a stretching energy expressed with Poisson's ratio. While [4] can produce very accurate solutions at low computational cost, it relies on several points in the shape having to undergo a rigid motion to establish the boundary conditions of the FEM system, a limitation shared with [29], that also needs this knowledge in advance.

As an alternative to physics-based models, statistical learning methods have been proposed to discover the laws governing the dynamics of a deformable object from training data. Active appearance shape models [14,30] and 3D morphable models [8] rely on a linear subspace of deformation modes to model the time-varying shape. The alternative strand of methods known as template-based methods [35,36,45] compute correspondences between the current image and a reference image in which the 3D shape is known. Similar to previous methods, the shape is coded as a linear combination of *known* deformation modes learnt in advance from a relatively large set of training data. These methods obtain accurate results if appropriate training data is available. However, this is not often the case in practical scenarios.

On the other hand, NRS*f*M methods have emerged to simultaneously recover deformable shape and relative camera motion from a sequence of monocular images. Most NRS*f*M methods typically model the non-rigid shape as a linear combination of *unknown* shape basis [6,9,15,17,41,49] under orthography, in addition to orthonormality constraints on camera motion. Although this prior has proven to be a powerful constraint, it is not sufficient to solve all inherent ambiguities in NRS*f*M. A dual formulation in the trajectory space was proposed in [5], where a pre-defined Discrete Cosine Transform (DCT) basis was used to express the 3D trajectory of each point. Later, a smoothness for the time-trajectory of each point was incorporated in [19]. In [27], a normal distribution over shapes in a Procrustes aligned space was proposed. Yet, despite these low-rank based methods can effectively encode global deformations, they cannot generally handle non-linear motion patterns and strong local deformations. An alternative to overcome this limitation is based on piecewise formulations [4,44,48,51]. Piecewise planar [51], locally rigid [48] or quadratic [44] models rely on common features shared between patches to enforce spatial consistency and create a continuous global surface. While [48] needs too many points to enforce the rigid local constraint, the quadratic model [44] can be considered as an extension of a low-rank model and its power to code non-linear motions might be limited. More recently, [4] encoded the piecewise model by means of elastic triangles, without requiring a post-processing step, in order to stitch all local reconstructions into a single smooth surface [44].

In this paper, we propose two piecewise physically-based models to approximate the non-rigid shape. First, we use the 3D Navier-Cauchy equations solved by FEM to model its behavior, and represent the shape by means of a mixture of 3D wedge elements (triangular prisms). Second, we also consider a simplified version of the FEM formulation based on 2D thin-plate using triangular planar elements, slightly modified to improve the computational efficiency. The classical FEM formulation needs to identify several points undergoing a rigid motion to establish the boundary conditions in displacements. Our main contribution is to remove the rigid prior for the FEM formulation. To do this, we propose to approximate the compliance matrix by means of a generalization of the inverse stiffness matrix, enforcing a 6 rank deficiency that corresponds to the 6 rigid body motions of an object in the 3D space. We embed this formulation along with a motion model for smooth camera motion within a Bayesian EKF estimator. The proposed method does not require fine tuning of the material properties since these are explicitly incorporated in an uncertainty component. This generality makes our approach adequate to handle inextensible or potentially extensible deformations. Both new linear FEM formulations have a lower computational cost as they provide matrices of smaller dimensions compared to [4], reducing the time cycle for the non-rigid prediction stage. This is a key factor for achieving real-time performance at frame rate, as we show in the experimental section. Similar to the EKF-FEM formulation, our approach uses a full perspective camera, does not need continuous tracks of feature points over the sequence and does not require any training data.

3 Continuum Mechanics Deformation Model

Let us consider the linearly elastic object Ω depicted in Fig. 2, with $\mathbf{y}$ being its rest geometry referred to a 3D rectangular Cartesian coordinate system $\mathcal{C} \equiv \{x, y, z\}$. The solid object deforms by volumetric forces $\mathbf{f}_c$ and becomes internally stressed due to prescribed loading conditions. This behavior is modeled using the governing equations of elastostatics [53], known as Navier-Cauchy equations, and can be expressed in vector notation in terms of displacements as:

$$\frac{E}{2(1+\nu)(1-2\nu)}\nabla(\nabla\cdot\mathbf{u}) + \frac{E}{2\,(1+\nu)}\nabla^2\mathbf{u} + \mathbf{f}_c = \mathbf{0} \quad in \quad \Omega, \tag{1}$$

where $\mathbf{u} \equiv [\mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_z]^\top$ is the unknown 3D displacement field. This expression includes the gradient operator $\nabla = [\delta/\delta x, \delta/\delta y, \delta/\delta z]^\top$, the divergence operator $\nabla\cdot\mathbf{u} = \delta\mathbf{u}_x/\delta x + \delta\mathbf{u}_y/\delta y + \delta\mathbf{u}_z/\delta z$, and the Laplacian operator $\nabla^2(\cdot)$ that represents the divergence of the gradient. Material properties for this isotropic elastic solid are introduced by means of the Young's modulus E and the Poisson's ratio ν. These equations require boundary conditions that can be expressed as a displacement vector (Dirichlet conditions) $\mathbf{u} = \bar{\mathbf{u}}$ on Γ^u, or a stress vector (Neumann conditions) $\mathbf{t} = \bar{\mathbf{t}}$ on Γ^t, the boundary being defined as $\Gamma = \Gamma^u + \Gamma^t$, with $\bar{\mathbf{u}}$ and $\bar{\mathbf{t}}$ a prescribed displacement and traction field, respectively.

3.1 Finite Element Method Solution

The partial differential equations in Eq. (1) do not usually have an analytical solution, and numerical methods such as FEM have to be applied. The FEM provides continuous surface approximations of the unknown vector $\mathbf{u}$ in terms of combinations of local basis. To apply FEM to Eq. (1), the continuum object Ω with boundary Γ is discretized into a finite set of patches, denominated

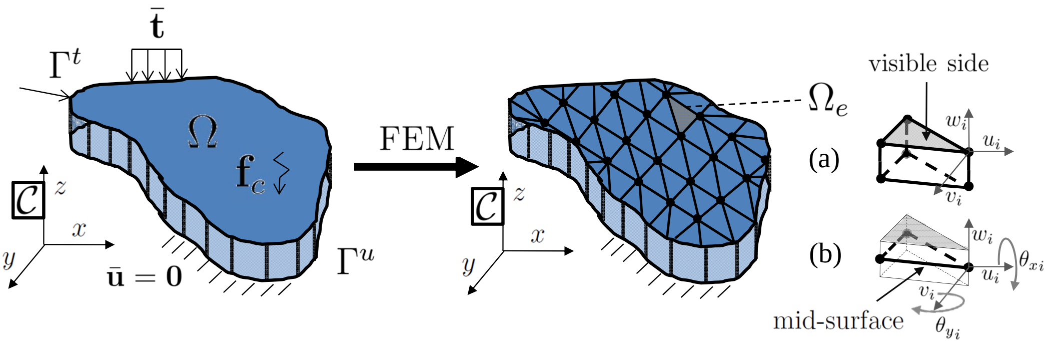


Fig. 2. **Linearly elastic solid formulation. Left:** An object Ω with body forces $\mathbf{f}_c$ and prescribed boundary conditions $(\bar{\mathbf{u}}, \bar{\mathbf{t}})$ on the boundary Γ. This problem can be solved using numerical methods such as FEM. **Right:** Discretization into finite elements Ω_e. The 3D deformation can be parameterized by (a) 3D wedge elements (6 nodes per element with 3 d.o.f. per node), or (b) using 2D thin-plate elements [4] (3 nodes per element with 5 d.o.f. per node) as a 2D function on the mid-surface plane. In both cases, element nodes are represented as black dots (●).

elements Ω_e, that are defined by its nodal points (Fig. 2). The displacement vector $\mathbf{u}$ at any point (x, y, z) is approximated as a weighted sum of piecewise shape basis functions N_i as:

$$\mathbf{u}(x, y, z) = \sum_i N_i(x, y, z)\, \mathbf{a}_i \tag{2}$$

where $\mathbf{a}_i$ is the nodal displacement vector for every node i in the discretization (see Fig. 2). Applying FEM to Eq. (1), we can obtain the classical global FEM linear system as:

$$\mathbf{Ka} = \mathbf{f} \tag{3}$$

where $\mathbf{a}$ and $\mathbf{f}$ are the 3D nodal displacement and force vector, respectively. The sparse and symmetric matrix $\mathbf{K}$ is the global stiffness matrix of the object. This matrix may be assembled from its associated element stiffness matrices as $\mathbf{K} = \mathbf{A}^e\, \mathbf{K}^e$, where $\mathbf{A}^e$ represents the assembly operator. The element stiffness matrix $\mathbf{K}^e$ in global coordinates can be defined as:

$$\mathbf{K}^e = \int_{\Omega_e} \mathbf{B}^\top \mathbf{D}\mathbf{B}\, d\Omega_e\,, \tag{4}$$

where $\mathbf{D}(E, \nu)$ is the behavior matrix for isotropic linear materials and $\mathbf{B}$ is the strain-displacement matrix that depends on the type of discretization [7,53]. This matrix relates the strain and the nodal displacements.

In this work, we propose a 3D FEM formulation based on 3D wedge elements to model the non-rigid shape. Additionally, we also propose a simplified version of the 2D thin-plate model proposed in [4]. In the following section, we describe each one of these formulations in more detail.

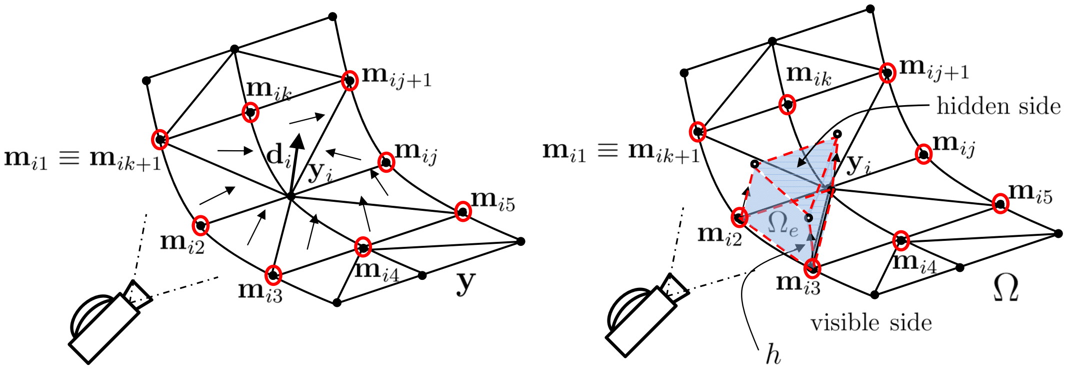


Fig. 3. **Generating a 3D wedge element. Left:** Normal unit vector estimation $\mathbf{d}_i$ per nodal point i as weighted average of normals on $\mathbf{y}_i$ neighborhood (red circles). **Right:** Extruded wedge element Ω_e.

3.2 3D FEM Formulation: Wedge Elements

In this case, we propose to employ wedge elements to model the scene object as a single layer of these elements. Without loss of generality, we employ just one layer to simplify the element definition, although more elements could be used. Nodal displacement and force vectors at the i-th point can be expressed as $\mathbf{a}_i = [u_i, v_i, w_i]^\top$ and $\mathbf{f}_i = [f_{x_i}, f_{y_i}, f_{z_i}]^\top$ for these elements (see Fig. 2). As the observed objects we will consider are opaque, the camera can only detect one surface side –denoted as the visible side– with n nodal points. We represent this surface $\mathbf{y}$ as a triangulated mesh computed by Delaunay triangulation. However, the 3D elements need both sides to compute the stiffness matrix of the solid object in Eq. (4). In this work, we generate the 3D mesh of wedge elements from visible side. For this purpose, we extrude each visible point $\mathbf{y}_i = [x_i, y_i, z_i]^\top$ on the visible side along its normal unit vector $\mathbf{d}_i$, up to a fixed thickness h (see Fig. 3(left)) that corresponds to the surface thickness. To obtain the normals for each nodal point i, we use a weighted average of normals on the $\mathbf{y}_i$ neighborhood (see Fig. 3(right)) to provide continuity C^1 within the object. Each normal vector is weighted according to the corresponding triangle area computed by means of the cross product:

$$\mathbf{d}_i = \frac{\sum_{j=1}^{k} (\mathbf{m}_{ij} - \mathbf{y}_i) \times (\mathbf{m}_{ij+1} - \mathbf{y}_i)}{\left\| \sum_{j=1}^{k} (\mathbf{m}_{ij} - \mathbf{y}_i) \times (\mathbf{m}_{ij+1} - \mathbf{y}_i) \right\|} \,, \tag{5}$$

where $\mathbf{m}_{ij} \in \{\mathbf{m}_{i1}, \mathbf{m}_{i2}, \ldots, \mathbf{m}_{ik}\}$ are the neighbor nodes defining the k Delaunay triangles to which the node $\mathbf{y}_i$ belongs.

With the solid geometry obtained, the 18-dimensional (3 d.o.f. per point) elemental stiffness matrix Eq. (4) for wedge elements can be expressed as:

$$\mathbf{K}^e = \int_{-1}^{1} \int_{0}^{1} \int_{0}^{1-\xi} \mathbf{B}^\top \mathbf{D} \mathbf{B} \, |\mathbf{J}| \, d\eta \, d\xi \, d\zeta \,, \tag{6}$$

where $\boldsymbol{\xi} = [\eta, \xi, \zeta]^\top$ are the natural coordinates which are used to express the piecewise shape functions [53]. The transformation from natural to global coordinates is expressed by means of the Jacobian $|\mathbf{J}|$, *i.e.,* $\mathbf{J} = \delta\mathbf{y}/\delta\boldsymbol{\xi}$. The integrals in Eq. (6) can be numerically approximated using integration points [46]. We select five points to approximate the elemental stiffness matrix. See the Appendix for the exact expressions of these matrices and shape functions.

Our focus is only to determine the geometry on the visible side of the surface since the hidden side is never observed. To do this, we can reorder the displacements and forces for each side of the object in order to marginalize out the hidden nodes. Let us denote as $\mathbf{a}^v$ and $\mathbf{f}^v$ the vectors for the visible side, and $\mathbf{a}^h$ and $\mathbf{f}^h$ the vectors for the hidden side. We can rewrite the linear system in Eq. (3) as:

$$\begin{bmatrix} \mathbf{K}^{vv} & \mathbf{K}^{vh} \\ \left(\mathbf{K}^{vh}\right)^\top & \mathbf{K}^{hh} \end{bmatrix} \begin{bmatrix} \mathbf{a}^v \\ \mathbf{a}^h \end{bmatrix} = \begin{bmatrix} \mathbf{f}^v \\ \mathbf{f}^h \end{bmatrix}, \tag{7}$$

where the resulting $\mathbf{K}$ stiffness matrix is $6n \times 6n$ since the n feature points have to be projected on the hidden side.

Although acting forces could be applied on both sides of the object, without loss of generality, we can reduce an acting force on the hidden side to an equivalent one on the visible side to produce similar deformations. Hence, assuming no acting forces on the hidden side $\mathbf{f}^h = \mathbf{0}$, Schur complement of $\mathbf{K}^{hh}$ in $\mathbf{K}$ can be applied to relate displacements $\mathbf{a}^v$ and forces $\mathbf{f}^v$ on the visible side as:

$$\left[\mathbf{K}^{vv} - \mathbf{K}^{vh}\left(\mathbf{K}^{hh}\right)^{-1}\left(\mathbf{K}^{vh}\right)^\top\right]\mathbf{a}^v = \mathbf{f}^v\ . \tag{8}$$

Note that the Schur complement is denoted as static condensation from the FEM analysis [53]. Finally, we obtain our $3n \times 3n$ stiffness matrix from static condensation of a wedge element as $\mathbf{K} = \mathbf{K}^{vv} - \mathbf{K}^{vh}\left(\mathbf{K}^{hh}\right)^{-1}\left(\mathbf{K}^{vh}\right)^\top$. Therefore, we are able to model the rigidity of just one visible side.

3.3 2D FEM Formulation: Triangular Elements

We also propose to employ the 2D thin-plate formulation to approximate Eq. (1) as a 2D function on the mid-surface plane. We follow the combination proposed in [4] except for the fact that rotations are removed from the formulation to reduce the computational overhead. Regarding this model, in-plane displacements are coded using a 2D plane stress model and out-of-plane displacements are coded with the Kirchhoff-Love plate model. Hence, the nodal displacement and force vectors are 5-dimensional for each node and can be expressed as $\mathbf{a}_i = \left[u_i, v_i, w_i, \theta_{xi}, \theta_{y_i}\right]^\top$ and $\mathbf{f}_i = \left[f_{xi}, f_{y_i}, f_{zi}, \Theta_{xi}, \Theta_{y_i}\right]^\top$, respectively (see Fig. 2). Since we only model the middle-plane in the solid, in this work we propose a novel additional simplification eliminating the rotation $\left[\theta_{xi}, \theta_{y_i}\right]$ and moment $\left[\Theta_{xi}, \Theta_{y_i}\right]$ components, *i.e.*, we only consider the translational displacements. To do this, we only have to eliminate the rows and columns of these degrees of freedom when the elemental stiffness matrix is computed. Note that only translational displacements can be detected by the camera, since the rotation effects are eventually also detected as translations. This allows us to obtain a $3n \times 3n$ stiffness matrix, *i.e.*, a model matrix to code only the 3D displacements of n map points. In addition, we also reduce the computational cost since the resulting linear system has a lower dimension.

4 FEM Linear System Solution

The linear system in Eq. (3) is under-constrained since $\mathbf{K}$ is rank deficient, and additional constraints are necessary to solve it. The classical FEM analysis uses boundary conditions to provide these additional constraints and is able to obtain a unique solution. These constraints are normally imposed as displacements, fixing nodal points in the space, *i.e.,* with a null displacement $\mathbf{a}_i = \mathbf{0}$.

While these constraints provide additional information about the type of deformation, they are not frequently available. In this work, it is our goal to remove the necessity of knowing several rigid points in order to solve Eq. (3). In this case, the full affine space solution can be computed as [18]:

$$\begin{cases} \breve{\mathbf{a}}_p + \breve{\mathbf{a}}_h = \mathbf{a} \\ \mathbf{K}\,\breve{\mathbf{a}}_h = \mathbf{0} \\ \mathbf{K}^\dagger\,\mathbf{f} = \breve{\mathbf{a}}_p \end{cases}$$

where the displacement vector $\mathbf{a}$ is a linear combination of the homogeneous solution $\breve{\mathbf{a}}_h$ and the particular solution $\breve{\mathbf{a}}_p$. The first represents the set of displacement vectors compatible with no acting forces and corresponds to rigid transformations with dimensions equal to the $\mathbf{K}$ rank deficiency. For the 3D case, this rank deficiency should be 6, corresponding to 6 possible 3D rigid body motions. The particular solution, $\breve{\mathbf{a}}_p$ is computed by means of a generalization of the Moore–Penrose pseudoinverse $\mathbf{K}^\dagger$. However, in practice the rank deficiency of $\mathbf{K}$ could not be 6, as we will see below. A rank deficiency lower than 6 will produce *phantom* deformations for a given set of forces. It is our aim to model the deformation by $\mathbf{a} = \breve{\mathbf{a}}_p$, *i.e.,* we have to subtract the rigid transformations. In these cases, the stiffness matrix rank has to be enforced to $r = 3n - 6$ by means of a Singular Value Decomposition (SVD), with n being the number of points. As $\mathbf{K}$ is a symmetric and square matrix, its factorization can be expressed as $\mathbf{K} = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{U}^\top$, and we can then compute the generalized pseudoinverse as:

$$\mathbf{K}^\dagger = \mathbf{U}_r\,\mathbf{\Sigma}^\dagger\,\mathbf{U}_r{}^\top, \tag{9}$$

where $\mathbf{U}_r$ contains the first r columns of $\mathbf{U}$ and $\mathbf{\Sigma}^\dagger$ is a diagonal matrix composed of the first r singular values σ inverses:

$$\mathbf{\Sigma}^\dagger = \mathbf{diag}\left[\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_r}\right]. \tag{10}$$

We analyze the rank deficiency of the proposed stiffness matrices by means of two synthetic examples, both with n=81 nodal points. In the first example the shape at rest is planar while in the second it is curved (see Fig. 4). First, we analyze our 3D FEM formulation using wedge elements (see section 3.2). In this case, if we consider the full stiffness matrix for the wedge element in Eq. (7), the $\mathbf{K}$ is a 486-dimensional matrix having 6 null singular values for both types of geometry, as theoretically expected. When we consider the static condensation in Eq. (8), where the stiffness matrix $\mathbf{K}$ is a 243-dimensional matrix, the resulting matrix also has 6 null singular values. However, in real cases the surface normally contains noise and we will not obtain a rank deficiency of 6 up to numerical accuracy. This deviation from the theoretical value is also produced by the matrix inversion in the Schur complement in Eq. (8), which introduces numerical round-off errors. We can conclude that the wedge formulation is able to code the 6 rank deficiency even in the case of the Schur complement which halves the stiffness matrix size with the corresponding computational overhead reduction.

Regarding the 2D FEM formulation using triangular elements (see section 3.3), this deviation is clearer. When we use the finite element proposed in [4], the stiffness matrix is 405-dimensional. In this case, the rank deficiency is only 5 up to numerical accuracy. This excess of rank is attributed to the thin-plate approximation used to model the 3D object on the mid-surface plane, and it is also included in our simplified version of the thin-plate model where the dimension of the stiffness matrix is also 243. Note that the rank deficiency is different depending on the type of geometry, *e.g.* planar or non-planar, showing that the rank deficiency depends on data quality. We can conclude this formulation provides an inferior coding of the 6 rank deficiency.

Figure 4 illustrates the singular value decomposition for each case, showing the rank deficiency obtained for these synthetic examples. We conclude that the 3D FEM formulation provides better conditioned matrices than the 2D thin-plate formulation. In any event, the 6 rank deficiency has to be enforced in both formulations to correctly subtract the rigid transformations, avoiding the artifacts such as noise when a standard pseudoinverse is computed. This rank deficiency enforcement has proven to be a key factor for proper experimental performance.

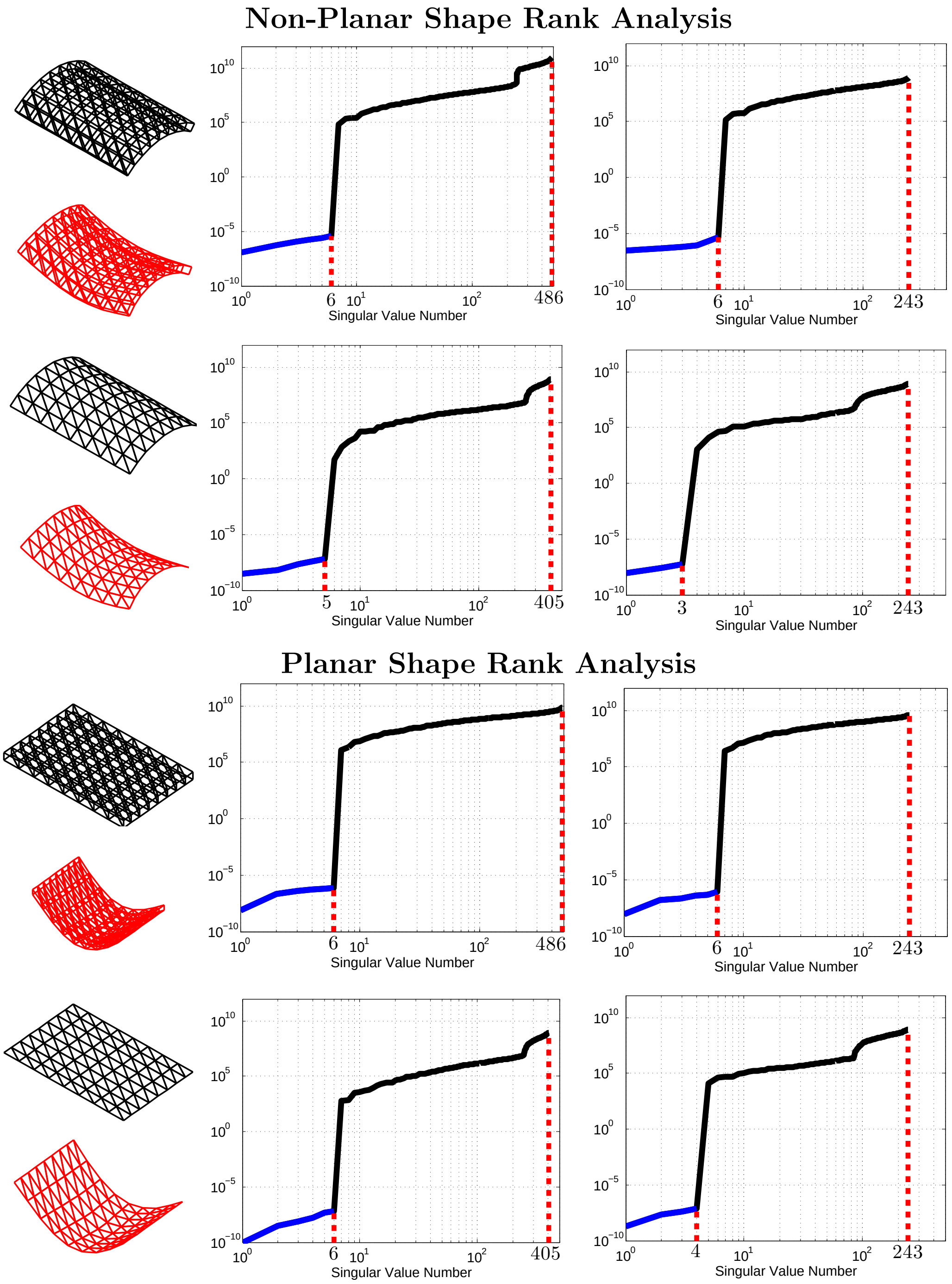


Fig. 4. **Rank deficiency analysis.** Singular values for a synthetic case: non-planar shape (above) and planar shape (below) in logarithmic scale. We show the shape at rest in black and a deformed shape in red for both discretizations. In all cases, the rank deficiency is displayed in blue. **Top:** 3D wedge elements using a full formulation (left) and static condensation (right). **Bottom:** 2D triangular elements using the 2D thin-plate model (left) and removing the rotations components (right). We code the deformation by means of the $3n$-dimensional stiffness matrices (represented on the right). In both cases, a 6 rank deficiency has to be enforced to avoid artifacts.

5 Camera-centric Non-Rigid EKF

This section is devoted to the embedding of our piecewise physics-based models within the sequential EKF estimator. Our EKF-FEM-FRP algorithm allows us to estimate simultaneously both camera motion and deformable shape for each frame from a monocular video. Like in other NRS*f*M approaches [5,41,49], as we do not assume any point to be rigid, only the relative pose of the non-rigid scene with respect to the camera can be estimated, instead of the full 3D camera trajectory when several rigid points are used [1,4]. For this purpose, we propose a camera-centric EKF-based formulation coding shape relative to the last camera location [10].

5.1 Problem Formulation

Let us consider a calibrated camera with unknown motion observing a non-rigid scene. Since no rigid points are assumed, we define the camera pose as a combination of the motion of the camera $\mathbf{x}_v$ and the six possible rigid motions of the structure $\mathbf{a}_r$ (see Fig. 1(d)). In this case, the state of the camera $\mathbf{x}_r^{\mathcal{C}_k}$ at frame k can be represented as:

$$\mathbf{x}_r^{\mathcal{C}_k} = \left[\mathbf{r}^{\mathcal{C}_k\top}, \mathbf{q}^{\mathcal{C}_k\top}, \mathbf{v}^{\mathcal{C}_k\top}, \boldsymbol{\omega}^{\mathcal{C}_k\top}\right]^\top, \tag{11}$$

where $\mathbf{r}^{\mathcal{C}_k}$ is the translation, $\mathbf{q}^{\mathcal{C}_k}$ is the quaternion to code the orientation enforcing the orthonormality constraints, and $\mathbf{v}^{\mathcal{C}_k}$ and $\boldsymbol{\omega}^{\mathcal{C}_k}$ are the linear and angular velocity, respectively. As we propose a camera-centric formulation, every component is related to a coordinate frame $\mathcal{C}_k$ fixed to the camera.

To code the non-rigid structure, we define it as a triangulated mesh with n 3D map points $\mathbf{y}_i$ –every feature point corresponds to a nodal FEM point– which can be concatenated in the state vector of the shape as:

$$\mathbf{y}^{\mathcal{C}_k} = \left[\mathbf{y}_1^{\mathcal{C}_k\top}, \ldots, \mathbf{y}_n^{\mathcal{C}_k\top}\right]^\top. \tag{12}$$

Our problem is to simultaneously estimate camera pose $\mathbf{x}_r^{\mathcal{C}_k}$ and deformable 3D shape $\mathbf{y}^{\mathcal{C}_k}$ in every frame k from a sequence of monocular images as the data arrives. For this purpose, we propose a camera-centric EKF-based formulation in which the state vector is represented as:

$$\mathbf{x}^{\mathcal{C}_k} = \left[\mathbf{x}_r^{\mathcal{C}_k\top}, \mathbf{y}^{\mathcal{C}_k\top}\right]^\top, \tag{13}$$

which has to be iteratively estimated with its corresponding covariance matrix $\mathbf{P}^{\mathcal{C}_k}$.

First, we assume that the camera motion follows a constant velocity model, similar to SLAM systems [4,16], and impulses of linear $\Delta\mathbf{v}^{\mathcal{C}} = \dot{\mathbf{v}}^{\mathcal{C}}\Delta t$ and angular $\Delta\boldsymbol{\omega}^{\mathcal{C}} = \dot{\boldsymbol{\omega}}^{\mathcal{C}}\Delta t$ velocities are introduced at every frame –with time step Δt–. Both velocities are coded with a zero-mean Gaussian distribution and covariance matrix $\mathbf{Q}_{\mathbf{x}_r}$. The state equation for the camera pose is:

$$\mathbf{x}_r^{\mathcal{C}_{k+1}} = \begin{bmatrix} \mathbf{r}^{\mathcal{C}_{k+1}} \\ \mathbf{q}^{\mathcal{C}_{k+1}} \\ \mathbf{v}^{\mathcal{C}_{k+1}} \\ \boldsymbol{\omega}^{\mathcal{C}_{k+1}} \end{bmatrix} = \begin{bmatrix} \mathbf{r}^{\mathcal{C}_k} + (\mathbf{v}^{\mathcal{C}_k} + \Delta\mathbf{v}^{\mathcal{C}})\Delta t \\ \mathbf{q}^{\mathcal{C}_k} \times \mathbf{q}\left((\boldsymbol{\omega}^{\mathcal{C}_k} + \Delta\boldsymbol{\omega}^{\mathcal{C}})\Delta t\right) \\ \mathbf{v}^{\mathcal{C}_k} + \Delta\mathbf{v}^{\mathcal{C}} \\ \boldsymbol{\omega}^{\mathcal{C}_k} + \Delta\boldsymbol{\omega}^{\mathcal{C}} \end{bmatrix}, \tag{14}$$

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

To code the non-rigid deformation of the scene, we use our FEM formulation in Eq. (3) that depends on two material parameters, the Poisson's ratio ν and the Young's modulus E. Without loss of generality, we can assume nearly incompressible materials setting $\nu \approx 0.5$, a reasonable approximation for a wide variety of materials such as rubbers, papers and human tissue. We can also factor out E so that it is not known a priori (see linear relation in Appendix). To do this, we define a vector of forces $\Delta\mathbf{s}^{\mathcal{C}}$ normalized by E, which allows to concentrate most of the tuning parameters in the state noise vector:

$$\Delta\mathbf{s}_i^{\mathcal{C}} = \frac{1}{E}\Delta\mathbf{f}_i^{\mathcal{C}} = \frac{1}{E}\left[\Delta f_{x_i}^{\mathcal{C}}, \Delta f_{y_i}^{\mathcal{C}}, \Delta f_{z_i}^{\mathcal{C}}\right]^{\top}. \tag{15}$$

This vector is modeled by means of a zero-mean Gaussian distribution with covariance matrix $\mathbf{Q}_{\mathbf{y}}$. In this vector, $\Delta\mathbf{f}_i^{\mathcal{C}}$ represents the incremental force in the current frame. Note that when we use the 2D thin-plate model (see section 3.3), this vector is normalized with the $\frac{1}{Eh}$ factor, since h can be partially factored out [4].

At each step, the applied force vector $\Delta\mathbf{s}^{\mathcal{C}}$ will recursively cause an incremental deformation $\Delta\mathbf{a}^{\mathcal{C}}$ with respect to the current state (see linear relation between displacements and forces in Eq. (3)). The state equation for the deformable scene can be expressed as:

$$\mathbf{y}^{\mathcal{C}_{k+1}} = \mathbf{y}^{\mathcal{C}_k} + \Delta\mathbf{a}^{\mathcal{C}_k} = \mathbf{y}^{\mathcal{C}_k} + \mathbf{K}^{\mathcal{C}_k \dagger}\Delta\mathbf{s}^{\mathcal{C}}, \tag{16}$$

where the rigidity of the scene is coded by the compliance matrix $\mathbf{K}^{\mathcal{C}_k \dagger}$ defined in Eq. (9), which is recomputed at each new frame using the current shape estimation $\mathbf{y}^{\mathcal{C}_k}$. Hence, the incremental non-rigid displacement $\mathbf{a}_{nr}$ (see Fig. 1(d)) caused by the Gaussian set of normalized forces is modeled as the particular solution of the FEM system without boundary conditions. Finally, we have to set a thickness surface h to extrude our 3D wedge element (see section 3.2). Note that the effect of this parameter is similar to E, since it gives rigidity to the structure. In practice, we could set it with a priori knowledge if this is available, or with a standard thickness of physical materials observed in computer vision. In the latter case, possible deviations will be compensated by the normalized force vector.

As a consequence of the generality of our approach, we can handle both inextensible and extensible deformations simply by tuning. While our FEM model favors elastic deformations with low energy, isometric deformations are a particular case of our model where assuming a thin solid and incompressibility $\nu \approx 0.5$ condition, the element area has to be almost fixed. Consequently, we do not have to use additional inextensibility constraints [11,36].

It is worth noting that both state transition equations (14) and (16) model all state parameters as dynamic, unlike rigid EKF [16] where only the camera parameters are dynamic or EKF-FEM [4] where, in addition, part of the map is dynamic. In any event, in both cases the dynamic state parameters represent a fraction of the state, while in our approach we fully exploit the estimation qualities of the Kalman filter since all the state vector is dynamic.

5.3 Measurement Equation

Let us consider a 3D point $\mathbf{y}_i^{\mathcal{C}_{k-1}} = [x_i, y_i, z_i]^\top$ expressed in the camera coordinate system $\mathcal{C}_{k-1}$. The observation of this point at frame k is defined by a ray coded by the directional vector $\mathbf{h}_i^{\mathcal{C}_k} = [h_{x_i}, h_{y_i}, h_{z_i}]^\top$ as:

$$\mathbf{h}_i^{\mathcal{C}_k} = \mathcal{R}^{\mathcal{C}_{k-1}} \left[\mathbf{y}_i^{\mathcal{C}_{k-1}} - \mathbf{r}^{\mathcal{C}_{k-1}} \right], \tag{17}$$

where $\mathcal{R}^{\mathcal{C}_{k-1}}$ represents the full rotation matrix depending on the quaternion $\mathbf{q}^{\mathcal{C}_{k-1}}$, and $\mathbf{r}^{\mathcal{C}_{k-1}}$ is the translation vector, and both are coded between references $\mathcal{C}_{k-1}$ and $\mathcal{C}_k$. The perspective projection of this point $\mathbf{y}_i^{\mathcal{C}_{k-1}}$ is modeled by means of a pinhole camera, being for a normalized retina:

$$\mathbf{h}_i = \begin{bmatrix} \beta_x - \alpha_x \frac{h_{x_i}}{h_{z_i}} \\ \beta_y - \alpha_y \frac{h_{y_i}}{h_{z_i}} \end{bmatrix}, \tag{18}$$

where (β_x, β_y) is the camera principal point and (α_x, α_y) its focal length. To deal with real camera lenses, a distortion model of two parameters has been applied [34]. Every feature point measurement is modeled with a zero-mean

Gaussian error to build a measurement noise covariance $\mathbf{R}^{\mathcal{C}_k}$. This is necessary in order to carry out the data association and to finally update the state vector.

5.4 EKF Formulation

We next briefly describe how to assemble the Jacobian matrices $\mathbf{F}^{\mathcal{C}_k}$ and $\mathbf{G}^{\mathcal{C}_k}$ of the state equations (14) and (16), respectively. Both matrices are required to perform the EKF prediction stage for the non-rigid case. They are defined as:

$$\mathbf{F}^{\mathcal{C}_k} = \begin{bmatrix} \frac{\partial \mathbf{x}^{\mathcal{C}_k}}{\partial \mathbf{x}_r^{\mathcal{C}_k}} \\ \frac{\partial \mathbf{x}^{\mathcal{C}_k}}{\partial \mathbf{y}^{\mathcal{C}_k}} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{I}\varDelta t & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathbf{q}^{\mathcal{C}_{k+1}}}{\partial \mathbf{q}^{\mathcal{C}_k}} & \mathbf{0} & \frac{\partial \mathbf{q}^{\mathcal{C}_{k+1}}}{\partial \boldsymbol{\omega}^{\mathcal{C}_k}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \tag{19}$$

$$\mathbf{G}^{\mathcal{C}_k} = \begin{bmatrix} \frac{\partial \mathbf{x}_r^{\mathcal{C}_k}}{\partial \mathbf{n}^{\mathcal{C}}} \\ \frac{\partial \mathbf{y}^{\mathcal{C}_k}}{\partial \mathbf{n}^{\mathcal{C}}} \end{bmatrix} = \begin{bmatrix} \mathbf{I}\varDelta t & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{\partial \mathbf{q}^{\mathcal{C}_{k+1}}}{\partial \Delta \boldsymbol{\omega}^{\mathcal{C}}} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{K}^{\mathcal{C}_k \dagger} \end{bmatrix}, \tag{20}$$

where $\mathbf{n}^{\mathcal{C}} = \begin{bmatrix} \Delta \mathbf{v}^{\mathcal{C}\top} & \Delta \boldsymbol{\omega}^{\mathcal{C}\top} & \varDelta \mathbf{s}^{\mathcal{C}\top} \end{bmatrix}^\top$ is the state noise vector with covariance matrix $\mathbf{Q}$. This block diagonal matrix is composed of $\mathbf{Q}_{\mathbf{x}_r}$ and $\mathbf{Q}_{\mathbf{y}}$. The covariance $\mathbf{Q}_{\mathbf{x}_r}$ is set as for the rigid EKF [16].

Finally, after the standard update stage, a rigid transformation is applied to change the reference system from the last camera $\mathcal{C}_{k-1}$ to the current camera $\mathcal{C}_k$ following a camera-centric formalism.

5.5 Initialization

Like other sequential approaches to recover non-rigid shapes [4,40,47], we assume that the monocular video contains some initial frames where the observed object is mostly rigid. We use this initial exploration to initialize our system, estimating a geometry at rest by applying the FEM model. To do this, we use our EKF-based method setting the covariance matrix $\mathbf{Q}_{\mathbf{y}} = \mathbf{0}$, *i.e.*, no acting forces are applied and the model is assumed to be rigid. It is worth noting

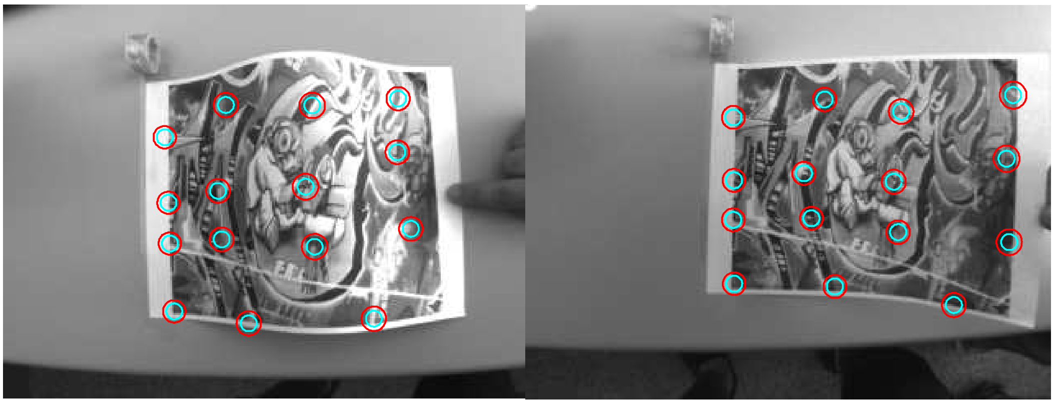


Fig. 5. **Data association for purely non-rigid shapes.** We display search regions for some feature points in several image frames. Cyan ellipses correspond to the search regions when only the camera motion is considered. Red ellipses represent the total predicted search areas, adding a degree of uncertainty produced by the deformable model (different for every nodal point). Note that as no rigid points are considered in our formulation, all search regions always include both contributions.

that we do not assume any training data [47] and our system can also be used for videos where this data is not available.

To estimate the geometry at rest, we use FAST interest points [43] on the first input frame and create a map of points encoded in inverse depth [12]. In the following frames, the camera moves around the object and features are observed with high parallax. We then switch from inverse depth to Euclidean parametrization. When the initial model is estimated, acting forces are permitted and the covariance matrix is $\mathbf{Q_y} \neq \mathbf{0}$ (see tuning in experimental Section 6).

We carry out the data association in a similar manner to [4,16], using a guided search. However, in this work all the feature points are non-rigid and hence all the predicted search areas have a contribution of uncertainty caused by the rigid camera motion and another cause by the non-rigid component of the model (see Fig. 5). We refer the reader to these papers for further details.

5.6 Computational Cost

Like other EKF-based methods [4,12,16], our approach has a complexity of $\mathcal{O}\left(n^3\right)$ when all the feature points are detected and measured in the input frame, where recall that n corresponds to the number of map points. In this work, we include a FEM model to code the non-rigid deformation of the structure, and in Eq. (9) we have to compute a per-frame SVD-based factorization of the stiffness matrix. Since the matrix $\mathbf{K}$ is symmetric, we propose to compute a symmetric SVD reducing the complexity from $f\left(n\right) = 21n^3$ to

$f(n) = 12n^3$ compared to the standard one [18]. Moreover, note that $\mathbf{K}$ is very sparse and, in practice, the previous complexity can be further reduced by exploiting the sparsity pattern.

When compared to [4], our formulations yield $3n \times 3n$ stiffness matrices instead of $5n \times 5n$, reducing the cost of computing the generalized compliance matrix. Additionally to this factorization, an inverse matrix is necessary when using 3D wedge elements, hence the computational cost is slightly more expensive compared to the thin-plate formulation where the Schur complement is not applied. In any event, both formulations reduce the time cycle for the non-rigid prediction stage, a key factor for achieving real-time performance. Similar to rigid EKF-based methods [12,16], the total complexity of our EKF-FEM-FRP algorithm remains $\mathcal{O}(n^3)$.

Finally, it is worth noting that our sequential approach only estimates the state vector at the current image thanks to the dynamic state estimation capabilities of the EKF, ensuring that the computation time per frame is bounded and does not increase with the number of images. We do not have to explicitly represent previous time step estimations, since the current state and its corresponding covariance have accumulated an estimation memory. The overall $\mathcal{O}(n^3)$ complexity, allows real-time performance at frame rate for small maps. For instance, in the experimental section we will show how our method can run in real time, between 30-35 Hz, for maps of 35 features.

6 Experimental Results

In this section we provide an extensive experimental validation for the proposed sequential method for both synthetic and real videos. Let us denote our methods as EKF-FEM-FRP1 and EKF-FEM-FRP2 when using 3D wedge elements and 2D triangular elements, respectively.

6.1 Synthetic Elastic Plate

We present a quantitative evaluation of our proposed method and a comparison with respect to other state-of-the-art approaches. For this purpose, we first use a synthetic sequence of 1000 frames where an elastic plate is deformed combining stretching and bending effects, provided by [4]. The deformation is particularly challenging since some elements increase their area by an order of 2×. This deformation was observed by a moving calibrated camera obtaining the synthetic projection on a 320×240 image. Figure 6 shows a few 3D reconstructions obtained with our free rigid priors algorithm using both

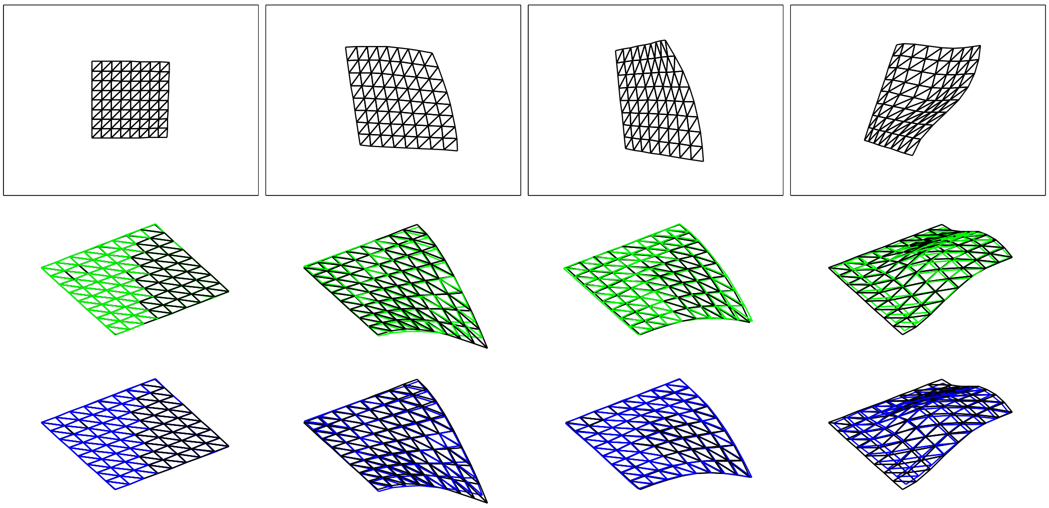


Fig. 6. **Synthetic sequence. Top:** Synthetic mesh observed by the camera at four selected frames. **Middle:** General view of our 3D reconstruction using EKF-FEM-FRP1. **Bottom:** General view of our 3D reconstruction using EKF-FEM-FRP2. In both cases the black mesh represents the ground truth. Best viewed in color.

FEM formulations. In both cases, the 3D reconstruction is very accurate and close to the ground truth. We also propose two novel synthetic sequences of a deforming elastic plate with different levels and types of deformation. In these cases, the deformation is concentrated in 200 frames and no rigid points are considered to constrain the solution, producing faster inter-frame changes compared to the previous sequence. Despite this difference, we can process the three sequences using the same physical tuning reported in Table 2. Figure 7 shows the 3D reconstruction we obtain for these experiments. Again, our 3D reconstruction is very accurate with respect to ground truth, even when the deformation does not include rigid points.

We compare our 3D reconstructions against state-of-the-art methods. We consider the batch methods EM-LDS [49], DCT [5], CSF [19], KSTA [20], SPM [15], EM-PND [27] and BBN [36]; and the sequential methods SBA [40] and EKF-FEM [4]. For the BBN [36] approach, we use the two configurations provided by [4], denoted as BBN1 when the training data set is small and BBN2 when a large set of training data is available. Every method was adjusted in accordance with its original paper. In all cases, we compute the 3D reconstruction error in *mm* as the average distance between the 3D ground truth and the 3D reconstruction over all the frames. Table 1 summarizes the results. For the sequence #1, our alternatives EKF-FEM-FRP1 and EKF-FEM-FRP2 yield a reconstruction error of 5.6mm and 5.3mm, respectively. This represents a remarkable improvement compared to most competing approaches, such as KSTA [20] (30.4mm), EM-LDS [49] (28.6mm), CSF [19] (20.2mm), SBA [40] (17.5mm), DCT [5] (16.6mm), EM-PND [27] (7.5mm)

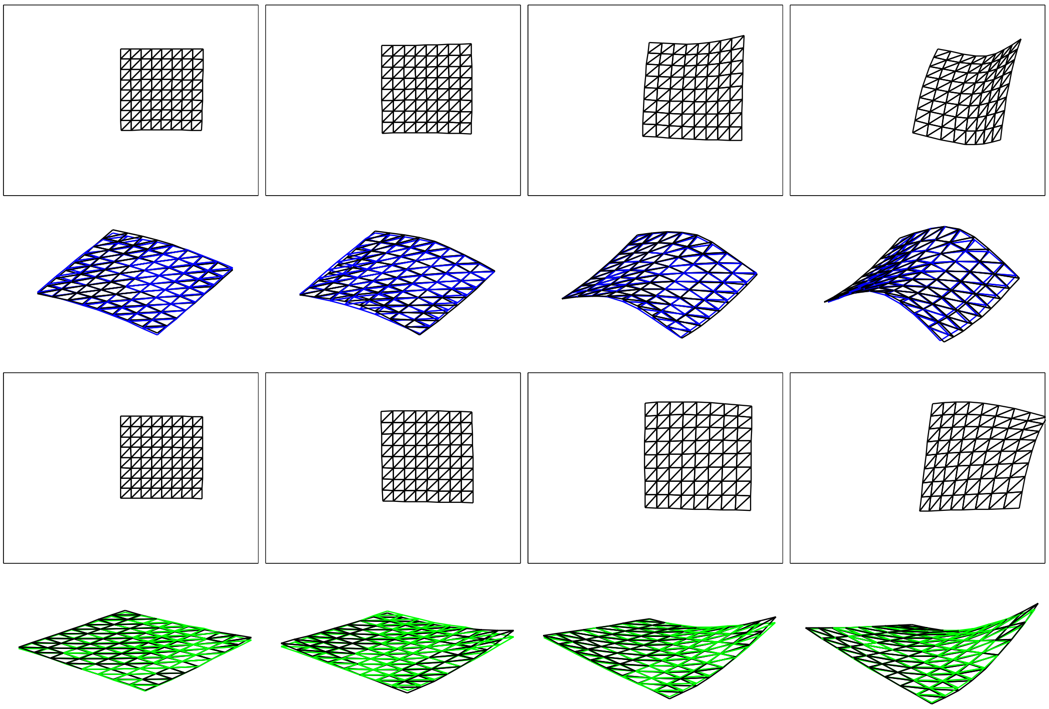


Fig. 7. **Synthetic sequences #2 and #3.** In both cases: **Top:** Mesh observed by the camera at four selected image frames. **Bottom:** General view of our 3D reconstruction. We represent the solution of the algorithm that gives the best performance according to Table 1 for each sequence, *i.e.*, EKF-FEM-FRP2 for #2 and EKF-FEM-FRP1 for #3. In both cases the black mesh represents the 3D ground truth. Best viewed in color.

and BBN1 [36] (7.4mm). While our methods outperform most of the evaluated techniques, our performance is slightly below EKF-FEM [4] (4.7mm) and BBN2 [36] (1.2mm) which, however, require rigid priors or a large set of training data, respectively (see Table 1). In fact, our algorithms show a reduction in accuracy compared to EKF-FEM [4], compatible with the removal of the rigid priors. For the sequences #2 and #3, neither training data nor rigid points are provided and hence BBN [36] and EKF-FEM [4] are not applicable. Regarding the rest of state-of-the-art techniques, we observe a similar pattern in the results as the one obtained for the previous experiment, with a consistent improvement of our proposed methods. It is worth to point that SPM [15] algorithm did not converge for these experiments.

Sequence #1 is also used to evaluate the sensitivity of our results against the tuning of the physical parameters that are required for our FEM models. We adjust the Poisson's ratio ν, the thickness surface h and the normalized forces $\Delta\mathbf{s}$. Note that the normalized force vector depends on the type of FEM formulation, being defined as $\Delta\mathbf{s} \equiv \frac{\Delta f}{E}$ for the 3D formulation and as $\Delta\mathbf{s} \equiv \frac{\Delta f}{Eh}$ for the 2D formulation. The results are summarized in Fig. 8.

First, we study the stability of our estimation as a function of the Poisson's

Table 1

Quantitative and qualitative evaluation. 3D reconstruction error for the synthetic sequences considering both batch and sequential state-of-the-art techniques and our two alternatives denoted as EKF-FEM-FRP1 and EKF-FEM-FRP2. The symbol – indicates the algorithm did not converge and ¬ that the algorithm did not manage to process the sequence. We also specify several characteristics of each of these techniques, considering: a specific rank of a deformation model (Rank), methods that use 3D training data (Training) to learn the deformation model, or rigid priors (Rigid) to constrain the solution. Additionally, we show when a method provides a strategy (Measure) to solve the data association and hence monocular videos can be directly handled. Strengths and weaknesses are denoted as (✓) and ($\mathcal{X}$), respectively.

3D error $[mm]$	*Batch Methods*								*Sequential Methods*			
Met. / Qua.	EM-LDS	DCT	CSF	KSTA	SPM	EM-PND	BBN1	BBN2	SBA	EKF-FEM	-FPR1	-FPR2
Seq. #1	28.6	16.6	20.2	30.4	–	7.5	7.4	1.2	17.5	4.7	5.6	5.3
Seq. #2	27.8	41.3	–	–	–	13.3	¬	¬	18.5	¬	5.0	4.9
Seq. #3	6.3	24.8	–	–	–	5.3	¬	¬	14.6	¬	3.0	3.2
Rank	$\mathcal{X}$	$\mathcal{X}$	$\mathcal{X}$	$\mathcal{X}$	$\mathcal{X}$		$\mathcal{X}$	$\mathcal{X}$				
Training							$\mathcal{X}$	$\mathcal{X}$				
Rigid										$\mathcal{X}$		
Measure							✓	✓		✓	✓	✓

ratio. We use values within the range $[0, 0.499]$, from highly compressible to incompressible materials. It is observed that this parameter has little influence in our algorithm, and hence we always set $\nu \approx 0.499$ for incompressible materials. Regarding the parameter tuning, we sweep a wide range of values for the thickness surface h and the normalized forces $\Delta\mathbf{s}$. We simultaneously set both parameters since they are related, and define the deformation of the solid object. We observe that the degradation of the reconstruction error is also very limited, being still more accurate than the closest competitors BBN1 [36], EM-PND [27], and DCT [5]. While EKF-FEM-FRP1 is more sensitive to h, since it is directly used within the FEM formulation –especially for very small values–, we can conclude that both algorithms remain within reasonable bounds for a wide range of tuned parameters. This means our methods do not need fine tuning of the physical parameters, although they may be used if available. Note that in both cases we obtain the smallest reconstruction error for $h = 1.5$mm, which is ground truth value for the surface thickness.

Finally, we have also evaluated the robustness of our algorithms against noise and missing data in the image measurements. We use the physical parameters that yielded the lowest 3D reconstruction error in the previous analysis, and show these results in Fig. 9. Regarding image noise, it can be observed that the accuracy for both methods gradually degrades for increasing levels of noise. We observe that EKF-FEM-FRP2 is more sensitive than EKF-FEM-FRP1, which even with a noise of 2 pixels std provides 3D errors below 7mm. Additionally, we add several random patterns of missing data per image frame. The accuracy with respect to the number of image correspondences for both meth-

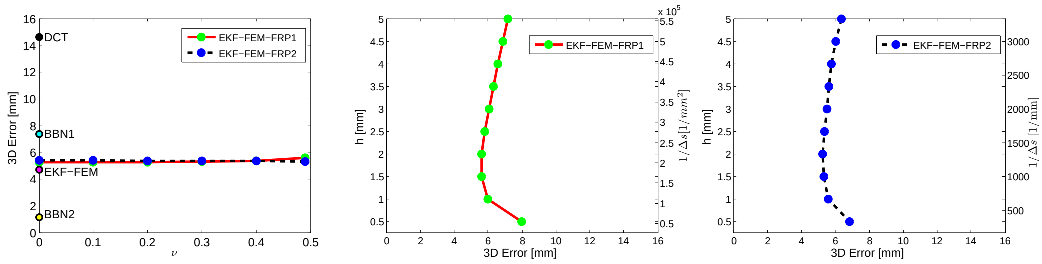


Fig. 8. **Sensitivity to the tuning of the physical parameters.** Mean 3D reconstruction error over all frames in the sequence with respect to Poisson's ratio ν, thickness surface h and normalized forces $\Delta\mathbf{s}$. We simultaneously display h along with the inverse of the normalized force $\Delta\mathbf{s}$. For the EKF-FEM-FRP1 method the normalized force is $\Delta\mathbf{s} \equiv \frac{\Delta f}{E}$, while it is $\Delta\mathbf{s} \equiv \frac{\Delta f}{Eh}$ for the EKF-FEM-FRP2. Note that the results remain within reasonable bounds for a wide range of tuned parameters.

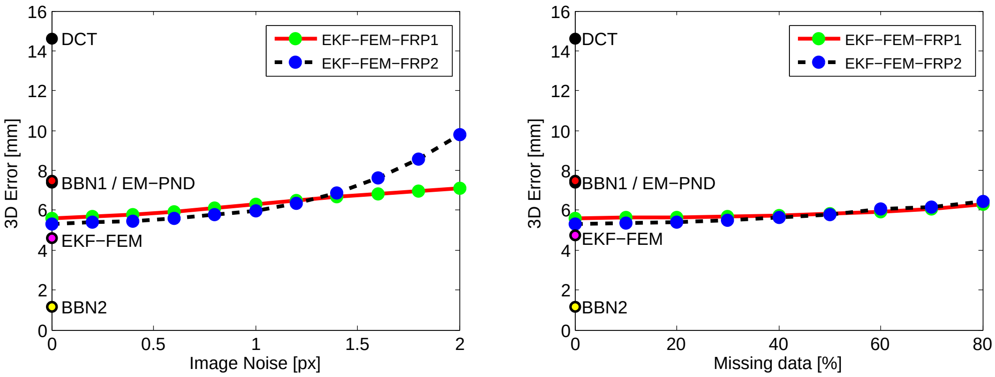


Fig. 9. **Robustness against image noise and missing data.** Mean 3D reconstruction error over all frames in the sequence with respect to artifacts. In addition, we also provide the 3D errors using the methods DCT [5], EM-PND [27], BBN1-BBN2 [36] and EKF-FEM [4] in absence of artifacts. **Left:** Reconstruction error as a function of the image noise. **Right:** Reconstruction error as a function of the percentage of missing data.

ods slightly decreases when the percentage of missing data increases, without any significant degrading until a breaking point at around 80%.

6.2 Real Videos

In this section, we evaluate our algorithms with real videos ranging from extensible materials –such as the silicone and laparoscopy sequence– to quasi-isometric deformations –such as the paper sequences–. Both types of deformations can be handled for our algorithms by simply tuning the parameters shown in Table 2. Since none of the competing methods considered in the

Table 2

Parameter selection. We tune the surface thickness h, the Poisson's ratio ν and the standard deviation of the normalized forces $\Delta\mathbf{s}$. The latter is different for each FEM formulation we propose. For 3D wedge elements it is $\Delta\mathbf{s} \equiv \frac{\Delta f}{E}$, while it is $\Delta\mathbf{s} \equiv \frac{\Delta f}{Eh}$ for the 2D thin-plate formulation [4].

	Synthetic Sequences		*Silicone Sequence*		*Paper Sequences*		*Face Sequence*		*Laparoscopy Sequence*	
Parameter	3D FEM	2D FEM	3D FEM	2D FEM	3D FEM	2D FEM	3D FEM	2D FEM	3D FEM	2D FEM
h $[mm]$	1.5	1.5	1.5	1.5	0.1	0.1	1.5	1.5	3.5	3.5
ν	0.499	0.499	0.499	0.499	0.499	0.499	0.499	0.499	0.499	0.499
$\frac{\Delta f}{E}$ $[mm^2]$	$6 \cdot 10^{-6}$	–	$2.25 \cdot 10^{-5}$	–	$4 \cdot 10^{-8}$	–	$1.5 \cdot 10^{-6}$	–	$7 \cdot 10^{-6}$	–
$\frac{\Delta f}{Eh}$ $[mm]$	–	$1 \cdot 10^{-3}$	–	$1.5 \cdot 10^{-2}$	–	$2 \cdot 10^{-6}$	–	$1 \cdot 10^{-3}$	–	$2 \cdot 10^{-3}$

previous subsection [5,15,19,20,27,40,49] provide a methodology to compute correspondences between frames –2D point tracks are used as input–, they are not directly applicable in these experiments (see Table 1). In addition, the real videos do not provide 3D training data, which would be necessary to apply BBN1-BBN2 [36]. We therefore just provide a comparison of our proposed alternatives, showing also that they can run in real-time.

6.2.1 Silicone Sequence

Firstly, we quantitatively evaluate our EKF-FEM-FRP algorithms with real extensible data. For this sequence, a moving hand-held camera observes a silicone cloth fixed to a stretcher while a person presses with the hand the back of the silicone surface. This results is an extensible deformation over time. Since the silicone surface is textureless, artificial markers were painted onto the visible surface to facilitate the feature detection. However, the coded markers are not used to compute the data association which is automatically solved using a guided search by our algorithm. It is worth noting that some of the points become *missing*, specially for skewed camera viewpoints. In this case, non-matched points will indirectly be updated from other point observations via the covariance matrix correlation terms. As shown in Fig. 10(top), the percentage of points annotated as non-matched can be relatively high –highlighted with blue circles–.

We process this sequence with the parameter setting detailed in Table 2. As no boundary conditions are known in advance, the normalized forces are smaller than those obtained using EKF-FEM [4] where rigid points are needed, in order to keep the size of the search regions within the image plane. For the EKF-FEM-FRP1, the normalized force does not depend on h, since the thickness effect is included in the wedge element extrusion. To perform a fair comparison, we set this force with an equivalent value just removing the h factor. Figure 10 shows our 3D reconstruction for a few selected frames with ground truth. Our estimation yields both the 3D location per point and its corre-

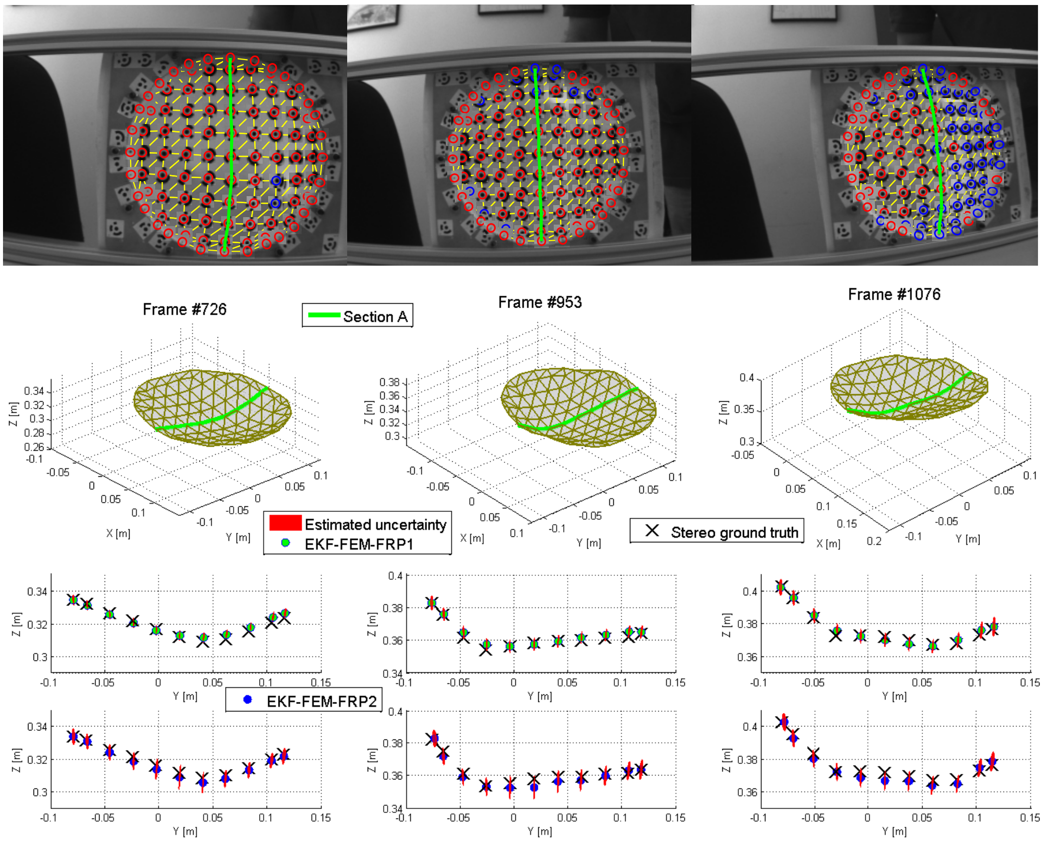


Fig. 10. **Silicone sequence. Top:** Images acquired by the camera at three selected frames with the estimated 3D mesh overlaid on it. The elliptical matching acceptance regions are also shown: red circles correspond to predicted and matched features; blue circles correspond to predicted but unmatched features. **Middle:** General view of the 3D reconstructed deformed scene. **Bottom:** A cross section of the reconstructed surface. We show for every point the 3D ground truth (black crosses), the 95% acceptance regions according to the estimated covariances (red ellipses) and the 3D estimated location using green circles for EKF-FEM-FRP1 and blue circles for EKF-FEM-FRP2. Note that both estimations are mostly consistent because the ground truth is within the acceptance region. Best viewed in color.

sponding covariance. We can observe after a qualitative comparison that our estimation is consistent with the ground truth since this is included in the estimated 95% acceptance region.

We also provide a quantitative comparison using a few frames with ground truth computed from stereo-vision. We obtain a mean 3D reconstruction error of 4.16mm for our EKF-FEM-FRP1 and 3.27mm for our EKF-FEM-FRP2. These are slightly greater than the 2.5mm reconstruction error reported by the EKF-FEM [4]. Again, these results are compatible with the removal of the rigid priors. Although both solutions are very similar in accuracy, we observe that EKF-FEM-FRP1 exhibits a larger bending rigidity. This effect can be detected by means of the size of the ellipses in 3D, which is smaller in the

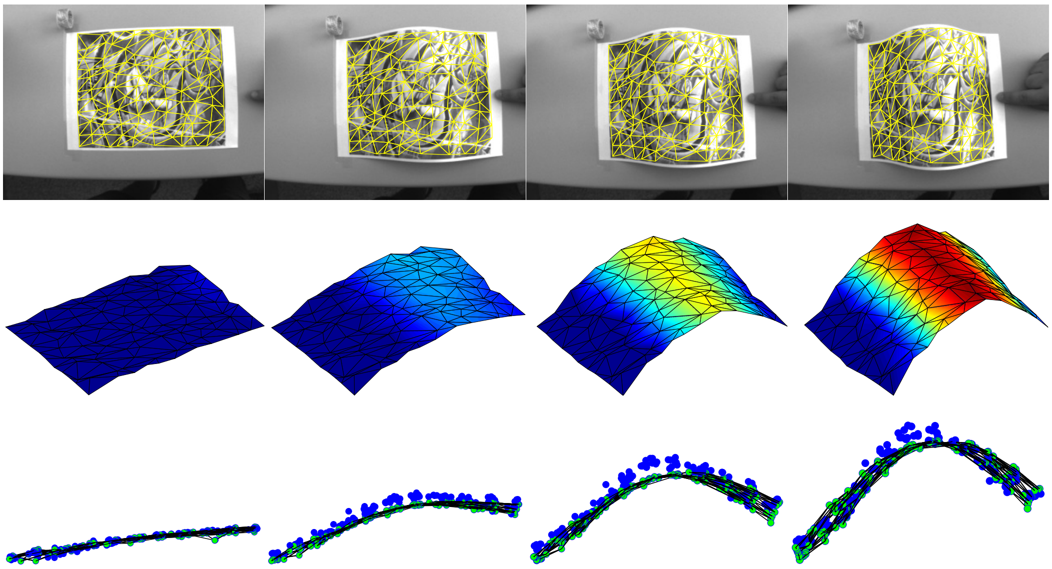


Fig. 11. **Paper bending sequence. Top:** Images acquired by the camera at four selected frames with the estimated 3D mesh overlaid. **Middle:** General view of our 3D reconstruction using EKF-FEM-FRP1. The 3D displacement field is encoded by a color pattern, where red indicates larger displacement. **Bottom:** Comparison of our FEM formulations: 3D FEM and 2D thin-plate reconstructions in green and blue points, respectively. Best viewed in color.

depth direction while remaining similar in the search areas within the image plane. Note that this effect is consistent with the FEM theory [53], since a single layer of 3D FEM elements magnifies the bending rigidity.

6.2.2 Paper Bending Sequences

We next evaluate our approaches on real videos acquired with a hand-held camera where a textured sheet of paper is being deformed. Although these sequences have static points, we never use this information to process the video. We consider two sequences with different levels of deformation. In these experiments, we use natural landmarks and the corresponding mesh is assembled by a soup of irregular triangles –with different stiffness–. To detect feature points, we use FAST interest points [43] in the first image of the sequence to build a triangular mesh structure. Note that our framework automatically computes the tracking and the data association throughout the sequence. Furthermore, it allows to handle this quasi-isometric deformation by simple parameter tuning, without using restrictive constraints to enforce inextensibility.

In the first experiment, a fixed camera observes a bending piece of paper. The reconstructed 3D shapes are plotted in Fig. 11, showing similar accuracies with both FEM formulations. We observe that the 3D FEM model exhibits

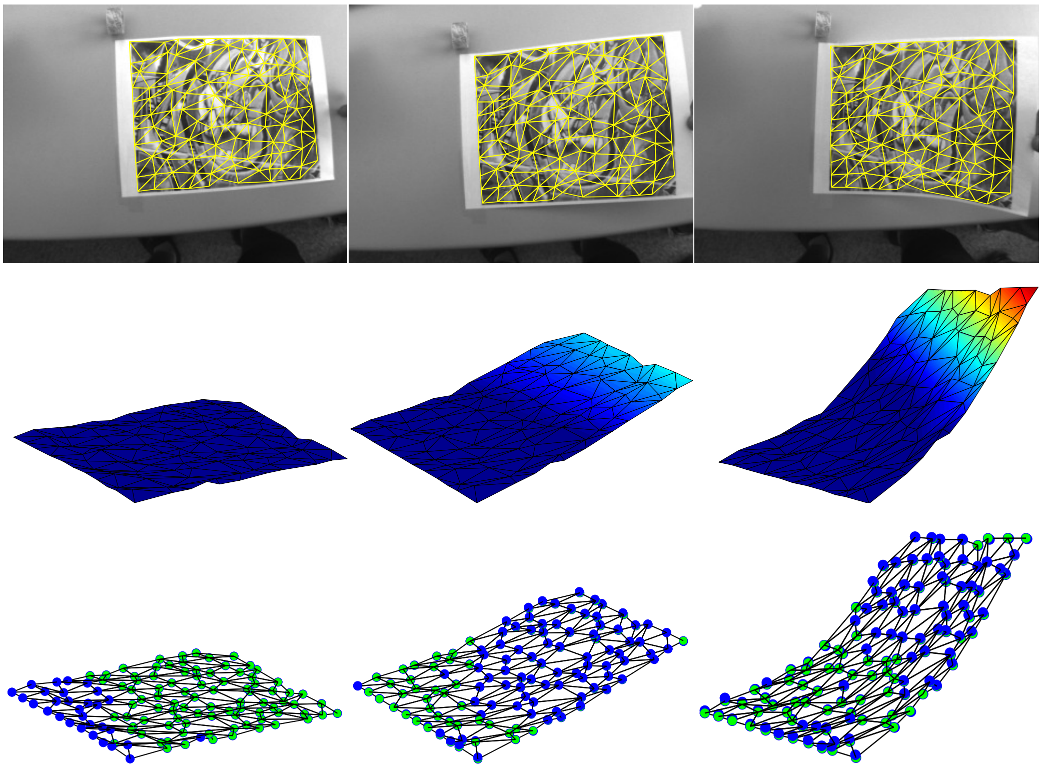


Fig. 12. **Flag paper sequence. Top:** Images acquired by the camera at four selected frames with the estimated 3D mesh overlaid. **Middle:** General view of the 3D shape reconstructed using EKF-FEM-FRP1. The 3D displacement field is encoded by a color pattern, where red indicates larger displacement. **Bottom:** Comparison of the two proposed FEM formulations: 3D FEM and 2D thin-plate reconstructions in green and blue points, respectively. Best viewed in color.

the expected magnification in bending rigidity compared to the 2D thin-plate model, especially because the thickness of this object surface is very small. In fact, as is shown in Table 2, to achieve a comparable performance we have to magnify the normalized force. Similar performances are obtained in a second experiment in which a paper is bended in a different manner. The 3D reconstructions of this experiment are plotted in Fig. 12.

6.2.3 Face Sequence

In order to demonstrate the generality of our approach, we also process a Franck's sequence, consisting of 248 frames and 56 feature tracks showing a man that simultaneously talks and moves his head. Figure 13 depicts four frames of the original images and several views of the resulting 3D reconstruction. Since ground truth is not available we can only qualitatively evaluate the results, which in seem to correlate with the actual deformation. Since both algorithms give similar results, we just represent the solution obtained by EKF-FEM-FRP2.

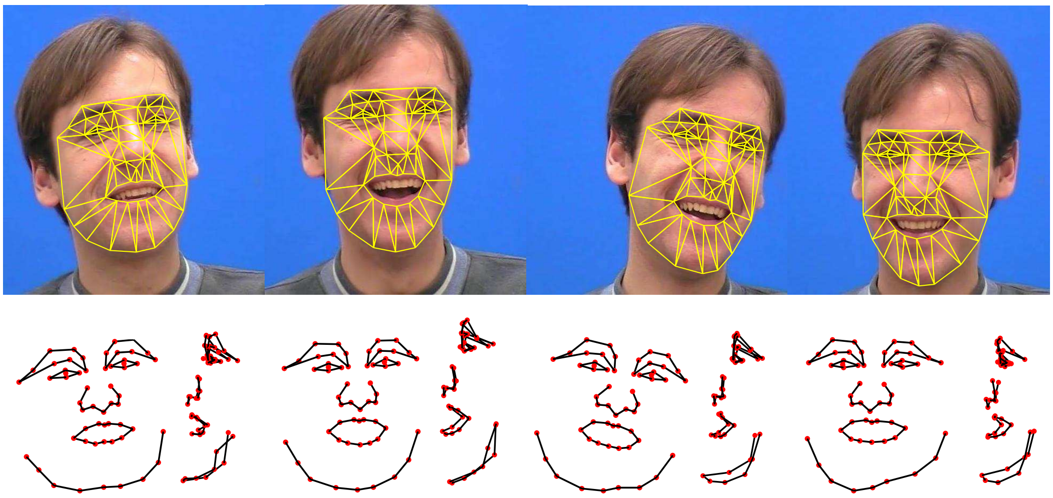


Fig. 13. **Franck's sequence. Top:** Images acquired by the camera at four selected frames on which the 3D estimated mesh has been reprojected. **Bottom:** Frontal and side viewpoints of the 3D shape estimated using EKF-FEM-FRP2.

6.2.4 Laparoscopy Sequence

We also consider a challenging real video made of 500 frames and 48 feature points acquired with a hand-held monocular laparoscope, in which a surgeon performs an external tactile exam on a rabbit abdominal cavity, producing different type of deformations. As a result of the camera motion and the shape deformation, several points are automatically annotated as occluded along sequence. Our methods can easily identify these points and still provide a 3D reconstruction of the whole surface. In Fig. 14 we present a few samples image frames and the corresponding 3D reconstruction.

6.2.5 Real-Time Performance

Finally, we experimentally demonstrate the efficiency of the proposed approach, permitting real-time performances for small maps. We implemented our monocular EKF-FEM-FRP method for both FEM formulations in C++. We exploit the fact that the stiffness matrix is symmetric in order to speed up the required SVD decomposition. To test our implementation, we use the paper bending sequence (see Fig. 11). Given that the computational complexity of the proposed formulations is $\mathcal{O}\left(n^3\right)$, real-time performance can be achieved for maps of up to 35 feature points, yielding 3D reconstructions at 30 Hz on an Intel Core i7 processor with a 2GHz based laptop. Figure 15 shows how the proposed approach produces accurate estimates even with small size maps, making it possible to handle low textured objects. Note that other piecewise methods in the literature [44,48] require overlapping features in neighboring patches, being only applicable to highly textured surfaces with hundreds of feature points.

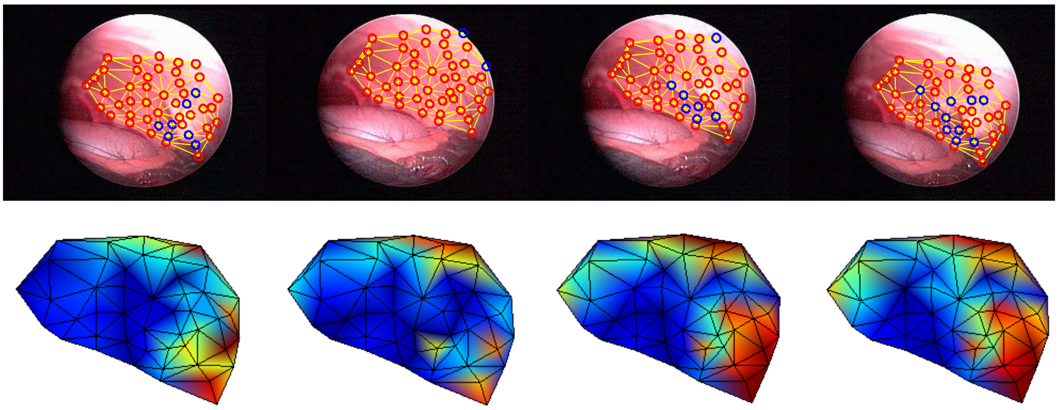


Fig. 14. **Laparoscopy sequence. Top:** Images gathered by the camera at four selected frames where 3D mesh and elliptical matching regions have been reprojected. **Bottom:** General view of our 3D reconstruction using EKF-FEM-FRP1. Color code represents the deformation field, where red areas are further from the rest shape and blue areas are closer. Best viewed in color.

Figure 16 shows a detail of the total cycle time budget, distinguishing two stages: rigid initialization and non-rigid estimation. For each stage, we identify three principal areas: 1) map management and matching, 2) EKF update and 3) EKF prediction. At the initialization stage, feature points are assumed to be rigid and thus the EKF prediction cost is negligible while the cubic dominant cost is related to the EKF update operation. Note that in this initialization stage, the computational cost for the update stage is larger than in the normal operation, since the feature points are coded in inverse depth [12] (6 parameters per point). When estimating the non-rigid shape, the map points are coded in Euclidean XYZ (3 parameters per point) with the corresponding computational load reduction in the EKF update stage. However, because of the FEM computation, the EKF prediction stage becomes more dominant. During the non-rigid shape estimation we also observe an increase in the matching cost, since the search regions –prediction ellipses– become larger.

We also display the state size per frame, that corresponds to $6n + 13$ d.o.f. for the rigid initialization and $3n + 13$ d.o.f. for the non-rigid estimation stage. In both cases, the camera state is represented by 13 parameters. If we compare both FEM formulations (Fig. 16), we observe that the EKF-FEM-FRP1 complexity is slightly large than that of the EKF-FEM-FRP2, specially in the prediction stage. This is due to the Schur complement computation. In any event, we can conclude that for small maps, both EKF-FEM-FRP formulations achieve real-time performance. In Fig. 16-right, we also display the cost per frame histogram for all frames in the video, showing that computation –for maps of up to 35 features– does not exceed 33.33 ms in any of the frames.

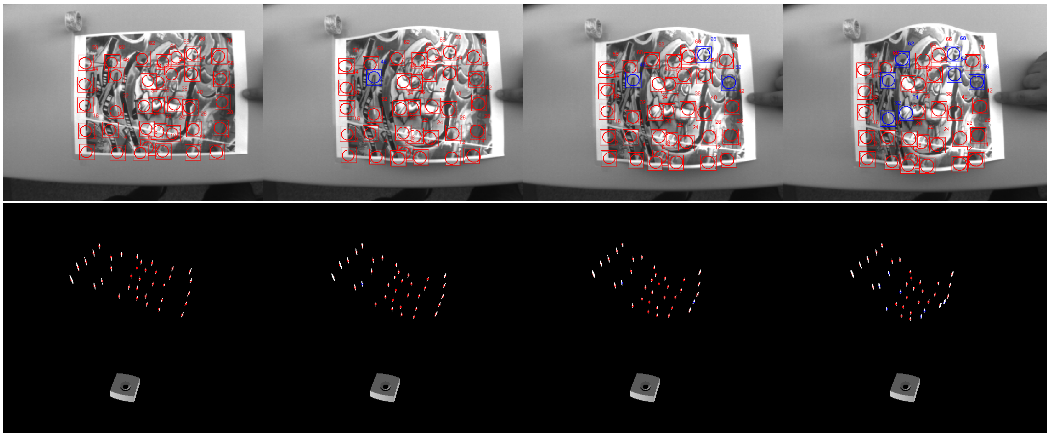


Fig. 15. **Recovering non-rigid shapes in real time. Top:** Input images with the elliptical search regions: predicted and matched features are represented in red, predicted and unmatched features in blue. **Bottom:** 3D reconstruction of a time–varying paper and estimated camera pose.

7 Conclusion

We have proposed a sequential algorithm for simultaneously estimating camera motion and time-varying shape from the sole input of the image sequence. Our system is able to handle different types of deformations, ranging from isometric to elastic, can cope with significant missing data and does not require training data. Additionally, the proposed approach performs data association over the whole sequence and can run in real time at frame rate for small maps. Regarding the mechanical model, we have proposed two piecewise FEM formulations to model the deformable shape. In both cases, we have eliminated the need to constrain several points to be rigid, a standard practice when physics-based models are used. This is achieved by modeling the compliance matrix through a generalization of the inverse stiffness matrix, and enforcing a six rank deficiency that allows us to marginalize out the six rigid body motions of the object. Although the 3D FEM formulation provides better conditioned matrices, it is slightly more expensive than the 2D FEM formulation. The accuracy of both is very similar.

Our claims have been experimentally validated both in synthetic and real videos, showing the results of our performance in real time at frame rate. We consider that our approach is particularly relevant for medical imaging, where rich priors and accurate models are often available, which could be readily incorporated into our EKF-FEM-FRP formulation. Additionally, our approach is also appropriate for robotics tasks involving the manipulation of non-rigid objects where an estimation in real time is mandatory. Our future work includes exploring medical data where rich biomechanical priors are available.

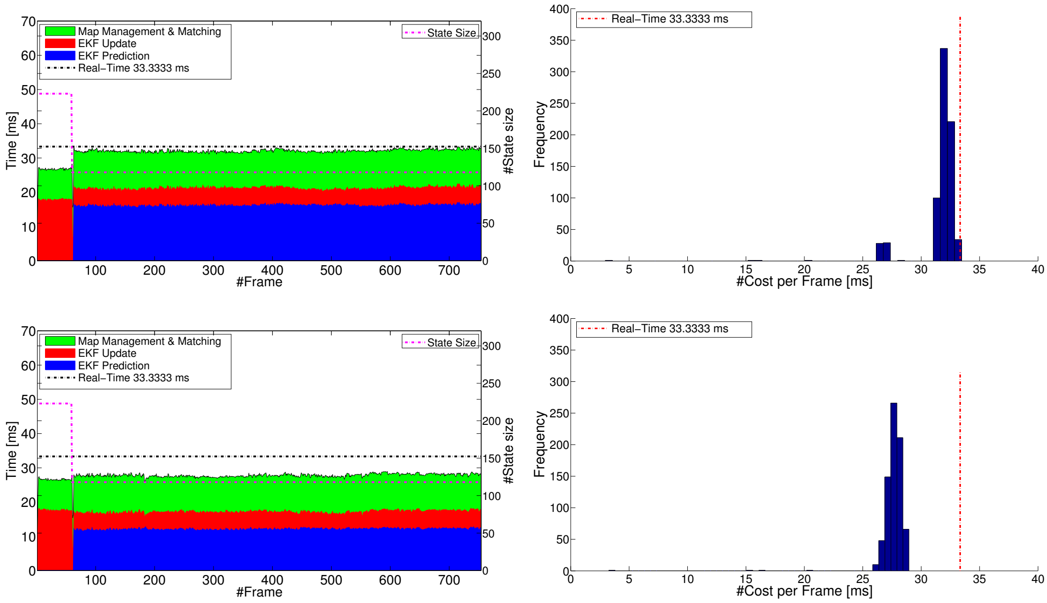


Fig. 16. **Real-time performance.** Real-time computation budget comparison for the *paper bending sequence*. The rigid initialization is performed during the first 60 frames, and then the non-rigid estimation begins. On the left, we show two scale plots: left-y axis time, right y-axis state size. The histogram on the right shows the distribution of execution times per frame, for all frames of the sequence. **Top:** EKF-FEM-FRP1. **Bottom:** EKF-FEM-FRP2.

Acknowledgments

This work has been partially supported by the Spanish Ministry of Science and Innovation under projects RT-SLAM DPI2015-67275-P, RobInstruct TIN2014-58178-R and Keratocono DPI2014-54981-R; by the FP7-SME-2013 606634 PopCorn project from the European Union FP7; by the ERA-net CHISTERA project VISEN PCIN-2013-047 and I-DRESS PCIN-2015-147; and by a scholarship FPU12/04886 awarded by the Spanish MECD. The authors wish to thank Imperial College London (Prof. A.J. Davison) and the University of Oxford (Prof. I. Reid and Dr. B. Williams) for the visual rigid SLAM software.

A Appendix

In this appendix, we show the bilinear (b) shape functions of the 3D wedge element used in section 3.2, and how to compute the strain-displacement $\mathbf{B}$ matrix in Eq. (6). These functions can be expressed in natural coordinates

(ξ, η, ζ) within the interval $[-1, 1]$ as:

$$\begin{cases} N_1^b = \frac{1}{2}\left(1-\xi-\eta\right)\left(1-\zeta\right) \\ N_2^b = \frac{1}{2}\xi\left(1-\zeta\right) \\ N_3^b = \frac{1}{2}\eta\left(1-\zeta\right) \\ N_4^b = \frac{1}{2}\left(1-\xi-\eta\right)\left(1+\zeta\right) \\ N_5^b = \frac{1}{2}\xi\left(1+\zeta\right) \\ N_6^b = \frac{1}{2}\eta\left(1+\zeta\right) \end{cases}$$

To compute the $\mathbf{B}$ matrix, it is necessary to have the derivatives of these shape functions with respect to the global coordinates (x, y, z). To achieve this, let us define the vector $\mathbf{N}_e = \left[N_1^b, N_2^b, N_3^b, N_4^b, N_5^b, N_6^b\right]^\top$. These derivatives can be computed using derivatives with respect to the natural coordinates (ξ, η, ζ) as:

$$\begin{bmatrix} \mathbf{N}_{e,x}^\top \\ \mathbf{N}_{e,y}^\top \\ \mathbf{N}_{e,z}^\top \end{bmatrix} = \mathbf{J}^{-1} \begin{bmatrix} \mathbf{N}_{e,\xi}^\top \\ \mathbf{N}_{e,\eta}^\top \\ \mathbf{N}_{e,\zeta}^\top \end{bmatrix}, \tag{A.1}$$

where $\mathbf{J}$ represents the Jacobian matrix of the transformation. Using the previous derivatives, we can finally express the strain-displacement $\mathbf{B}$ matrix as:

$$\mathbf{B} = \begin{bmatrix} \mathbf{N}_{e,x}^\top \otimes \mathbf{0}_1^\top \\ \mathbf{N}_{e,y}^\top \otimes \mathbf{0}_2^\top \\ \mathbf{N}_{e,z}^\top \otimes \mathbf{0}_3^\top \\ \mathbf{N}_{e,x}^\top \otimes \mathbf{0}_2^\top + \mathbf{N}_{e,y}^\top \otimes \mathbf{0}_1^\top \\ \mathbf{N}_{e,y}^\top \otimes \mathbf{0}_3^\top + \mathbf{N}_{e,z}^\top \otimes \mathbf{0}_2^\top \\ \mathbf{N}_{e,x}^\top \otimes \mathbf{0}_3^\top + \mathbf{N}_{e,z}^\top \otimes \mathbf{0}_1^\top \end{bmatrix} \tag{A.2}$$

where $\mathbf{0}_i$ is a 3×1 vector of zeros except the i-th position that is equal to 1. $\otimes$ denotes the Kronecker product.

Finally, the behavior $\mathbf{D}$ matrix for 3D linear elasticity [7,53] can be expressed as:

$$\mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}. \tag{A.3}$$

References

[1] A. Agudo, B. Calvo, J. M. M. Montiel, FEM models to code non-rigid EKF monocular SLAM, in: IEEE International Workshop on Dynamic Shape Capture and Analysis, 2011, pp. 1586–1593.

[2] A. Agudo, B. Calvo, J. M. M. Montiel, 3D reconstruction of non-rigid surfaces in real-time using wedge elements, in: Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment, 2012, pp. 113–122.

[3] A. Agudo, B. Calvo, J. M. M. Montiel, Finite element based sequential bayesian non-rigid structure from motion, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1418–1425.

[4] A. Agudo, F. Moreno-Noguer, B. Calvo, J. M. M. Montiel, Sequential non-rigid structure from motion using physical priors, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (5) (2016) 979–994.

[5] I. Akhter, Y. Sheikh, S. Khan, T. Kanade, Trajectory space: A dual representation for nonrigid structure from motion, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (7) (2011) 1442–1456.

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

[7] K. J. Bathe, Finite element procedures in Engineering Analysis, Prentice-Hall, 1982.

[8] V. Blanz, T. Vetter, A morphable model for the synthesis of 3D faces, in: ACM SIGGRAPH, 1999, pp. 187–194.

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

[10] J. A. Castellanos, J. Neira, J. D. Tardós, Limits to the consistency of EKF-based SLAM, in: Symposium on Intelligent Autonomous Vehicles, 2004.

[11] A. Chhatkuli, D. Pizarro, A. Bartoli, Stable template-based isometric 3D reconstruction in all imaging conditions by linear least-squares, in: IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 708–715.

[12] J. Civera, A. J. Davison, J. M. M. Montiel, Inverse depth parametrization for monocular SLAM, Transactions on Robotics 24 (5) (2008) 932–945.

[13] L. Cohen, I. Cohen, Finite-element methods for active contour models and balloons for 2D and 3D images, IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (11) (1993) 1131–1147.

[14] T. F. Cootes, G. J. Edwards, C. J. Taylor, Active appearance models, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (6) (2001) 681–685.

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

[16] A. J. Davison, I. D. Reid, N. D. Molton, O. Stasse, MonoSLAM: Real-time single camera SLAM, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (6) (2007) 1052–1067.

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

[18] G. Golub, C. Van Loan, Matrix computations, Johns Hopkins Univ Pr, 1996.

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

[20] P. F. U. Gotardo, A. M. Martinez, Kernel non-rigid structure from motion, in: IEEE International Conference on Computer Vision, 2011, pp. 802 – 809.

[21] M. A. Greminger, B. J. Nelson, Deformable object tracking using the boundary element method, in: IEEE Conference on Computer Vision and Pattern Recognition, 2003, pp. 289–294.

[22] M. A. Greminger, B. J. Nelson, A deformable object tracking algorithm based on the boundary element method that is robust to occlusions and spurious edges, International Journal of Computer Vision 78 (1) (2008) 29–45.

[23] N. Haouchine, J. Dequidt, M. O. Berger, S. Cotin, Single view augmentation of 3D elastic objects, in: International Symposium on Mixed and Augmented Reality, 2014, pp. 229–236.

[24] S. Ilić, P. Fua, Non-linear beam model for tracking large deformation, in: IEEE International Conference on Computer Vision, 2007, pp. 1–8.

[25] Y. Kita, Elastic-model driven analysis of several views of a deformable cylindrical object, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (12) (1996) 1150–1162.

[26] G. Klein, D. W. Murray, Parallel tracking and mapping for small AR workspaces, in: International Symposium on Mixed and Augmented Reality, 2007, pp. 225–234.

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

[28] J. Lim, J. Frahm, M. Pollefeys, Online environment mapping, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 3489–3496.

[29] A. Malti, R. Hartley, A. Bartoli, J. H. Kim, Monocular template-based 3D reconstruction of extensible surfaces with local linear elasticity, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 1522–1529.

[30] I. Matthews, S. Baker, Active appearance models revisited, International Journal of Computer Vision 60 (2) (2004) 135–164.

[31] T. McInerney, D. Terzopoulos, A finite element model for 3D shape recognition and nonrigid motion tracking, in: IEEE International Conference on Computer Vision, 1993, pp. 518–523.

[32] T. McInerney, 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) (1995) 69–83.

[33] D. Metaxas, D. Terzopoulos, Constrained deformable superquadrics and nonrigid motion tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (6) (1993) 337–343.

[34] E. M. Mikhail, J. S. Bethel, J. C. McGlone, Introduction to Modern Photogrammetry, John Wiley & Sons, 2001.

[35] F. Moreno-Noguer, P. Fua, Stochastic exploration of ambiguities for non-rigid shape recovery, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (2) (2013) 463–475.

[36] F. Moreno-Noguer, J. M. Porta, Probabilistic simultaneous pose and non-rigid shape recovery, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 1289–1296.

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

[38] R. Newcome, A. J. Davison, Live dense reconstruction with a single moving camera, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 1498–1505.

[39] J. Ostlund, A. Varol, P. Fua, Laplacian meshes for monocular 3D shape recovery, in: European Conference on Computer Vision, 2012, pp. 412–425.

[40] M. Paladini, A. Bartoli, L. Agapito, Sequential non rigid structure from motion with the 3D implicit low rank shape model, in: European Conference on Computer Vision, 2010, pp. 15–28.

[41] M. Paladini, A. Del Bue, M. Stosic, M. Dodig, J. Xavier, L. Agapito, Factorization for non-rigid and articulated structure using metric projections, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 2898–2905.

[42] J. Pilet, V. Lepetit, P. Fua, Fast non-rigid surface detection, registration and realistic augmentation, International Journal on Computer Vision 76 (2) (2008) 109–122.

[43] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, in: European Conference on Computer Vision, 2006, pp. 430–443.

[44] C. Russell, J. Fayad, L. Agapito, Energy based multiple model fitting for non-rigid structure from motion, in: IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 3009–3016.

[45] M. Salzmann, R. Urtasun, P. Fua, Local deformation models for monocular 3D shape recovery, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[46] A. H. Stroud, Approximate Calculation of Multiple Integrals, Prentice-Hall, 1971.

[47] L. Tao, B. J. Matuszewski, S. Mein, Non-rigid structure from motion with incremental shape prior, in: IEEE International Conference on Image Processing, 2012, pp. 1753–1756.

[48] J. Taylor, A. D. Jepson, K. N. Kutulakos, Non-rigid structure from locally-rigid motion, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2761–2768.

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

[50] L. V. Tsap, D. B. Goldof, S. Sarkar, Nonrigid motion analysis based on dynamic refinement of finite element models, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (5) (2000) 526–543.

[51] A. Varol, M. Salzmann, E. Tola, P. Fua, Template-free monocular reconstruction of deformable surfaces, in: IEEE International Conference on Computer Vision, 2009, pp. 1811–1818.

[52] A. Wendel, M. Maurer, G. Graber, T. Pock, H. Bischof, Dense reconstruction on-the-fly, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 1450–1457.

[53] O. C. Zienkiewicz, R. L. Taylor, The finite element method. Vol. 1: Basic formulation and linear problems, McGraw-Hill, London, 1989.