



### **Science Arts & Métiers (SAM)**

is an open access repository that collects the work of Arts et Métiers Institute of Technology researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <https://sam.ensam.eu>  
Handle ID: <http://hdl.handle.net/10985/23641>

#### **To cite this version :**

Alberto BADIAS, Icíar ALFARO, David GONZALEZ, Francisco CHINESTA SORIA, Elias CUETO  
- MORPH-DSLAM: Model Order Reduction for Physics-Based Deformable SLAM - IEEE  
Transactions on Pattern Analysis and Machine Intelligence - Vol. 44, n°11, p.7764-7777 - 2022

Any correspondence concerning this service should be sent to the repository

Administrator : [scienceouverte@ensam.eu](mailto:scienceouverte@ensam.eu)



# MORPH-DSLAM: Model Order Reduction for Physics-based Deformable SLAM

A. Badias, I. Alfaro, D. Gonzalez, F. Chinesta, E. Cueto

**Abstract**—We propose a new methodology to estimate the 3D displacement field of deformable objects from video sequences using standard monocular cameras. We solve in real time the complete (possibly visco-)hyperelasticity problem to properly describe the strain and stress fields that are consistent with the displacements captured by the images, constrained by real physics. We do not impose any ad-hoc prior or energy minimization in the external surface, since the real and complete mechanics problem is solved. This means that we can also estimate the internal state of the objects, even in occluded areas, just by observing the external surface and the knowledge of material properties and geometry. Solving this problem in real time using a realistic constitutive law, usually non-linear, is out of reach for current systems. To overcome this difficulty, we solve off-line a parametrized problem that considers each source of variability in the problem as a new parameter and, consequently, as a new dimension in the formulation. Model Order Reduction methods allow us to reduce the dimensionality of the problem, and therefore, its computational cost, while preserving the visualization of the solution in the high-dimensionality space. This allows an accurate estimation of the object deformations, improving also the robustness in the 3D points estimation.



## 1 INTRODUCTION

HUMAN visual sense is able to extract and process an enormous quantity of information, simply by observing a few images. A similar behavior is present in standard cameras, where an image can provide a high quantity of information. A strong development in computer vision has been carried out along last decades to extract relevant information from images. In this work, we want to focus in 3D reconstruction methods since our goal is to provide valuable information about the physical changes that happen in the surrounding space of an agent (a navigation robot or a user, employing augmented reality tools for the latter).

Visual augmented reality is based in the addition of graphical information to interact with reality, so it requires great efforts to fix virtual objects within the real world. This means that we need to know the map of the real world to locate the virtual objects. This process is known as Structure-from-Motion (SfM), and it consists in the estimation of rigid 3D structures from planar images using different camera view points.

Using the same tools, but with a different goal, we may want a robot to understand the surrounding spatial environment, with the ability of understanding the scene where it is moving. Sometimes not only geometric measurements from a sensor are needed, in some cases we should provide the agent with the ability of semantic perception and understanding of the physical changes that happen. We live in a dynamic world, where changes happen at any time, so robots need to have this comprehension ability to really understand the surrounding space.

Some authors suggest the use of the term *Spatial AI* [1].

- A. Badias, I. Alfaro, D. Gonzalez, and E. Cueto are with the Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Spain. E-mail: {abadias, icar, gonzal, ecueto}@umizar.es
- F. Chinesta is with the ESI Group Chair at the PIMM Lab, Arts et Métiers Institute of Technology, Paris, France. E-mail: francisco.chinesta@ensam.eu

It consists of a 3-layer hierarchical system for the spatial understanding of the environment [2], namely *Robust Localization*, *Dense Mapping* and *Semantic Understanding*. In our work we suggest to add a fourth layer to understand the physical changes that occur around the agent (see Fig. 1). The physical understanding may be gathered within the *semantic understanding*, but we consider that both are very relevant features and must be considered separately.

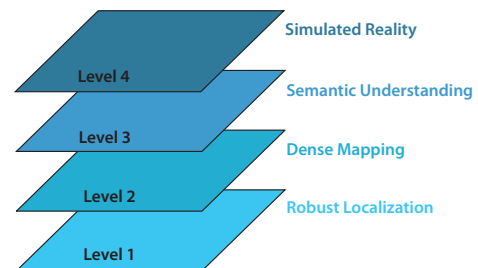


Fig. 1: Levels in the Spatial AI approach. Image inspired in the CVPR presentation of A. Davison [2].

The first level of the Spatial AI approach is related to the ability of the agent to locate itself in space and navigate through it in a robust manner. The great development in the last decades has brought new algorithms of *Simultaneous Localization and Mapping* (SLAM) performing localization tasks very efficiently and robustly [3]. Level 2 refers to the dense mapping that reconstructs the environment with a high level of detail [4]. Level 3 requires the system to identify and recognize the objects around it and to be able to establish semantic relationships [5]. And finally, at level 4, we place the term defined as *Simulated Reality* (SR) [6], where the system needs to know the physics of the surrounding environment in order to have a deeper understanding of the world. SR is based on continuously simulating the physical phenomena of the environment with which the agent interacts. Learning this kind of concepts is a long and

complex process, since we must teach a machine how to interpret the physics.

In the field of computer graphics, recent investigations have focused in the development of simplified physics (see, to name but a few [7], [8]) in order to overcome obvious computational limitations. However, these simplified approaches provide results with no known accuracy bounds. Dealing with deformable solids, there are some physics engines to approximate the dynamics, but they hardly comply with basic conservation laws. We consider there is no better approach than really solving the (in)elasticity equations, so we suggest to solve the high-fidelity equations. However, standard solvers are not able to work in real time, and even less on portable devices. Since we use visual cameras, the video frequencies are usually fixed in 30 or 60 frames per second, so this is why we suggest the use of Model Order Reduction (MOR) techniques to work in this range of frequencies. MOR methods are based on the compression of the data to work in a reduced space, but at the same time they must recover the original high-fidelity data in a fast way, fulfilling visualization frequencies, with a minimal loss in accuracy. Moreover, this loss is limited by well established error bounds for the vast majority of MOR techniques. The power of these techniques allows us to simulate realistic behaviors that describe and adapt perfectly to the changes that occur in the environment.

In this paper we propose the use of reduced-order numerical methods to estimate the behavior of real deformable solids. Certainly it is usually thought that solving the equations of solid mechanics can be a heavy process, not suitable for video frequencies. But in recent years some tools have been developed to reduce the complexity of these mechanical models. They allow to approximate the true physics of solids at real-time rates.

Including this introduction section, we have divided this paper into 8 parts. Section 2 includes other relevant works in this field from a computer vision perspective. Section 3 explains our method in a few words. Section 4 shows the mechanical formulation of the problem. Section 5 shows the dimensionality reduction problem and the formulation of some MOR methods. Section 6 explains the inverse problem assimilating the data with a deformable implementation of ORB-SLAM2. Section 7 covers the experimentation phase. And finally, section 8 contains the conclusions of the work.

## 2 RELATED WORK

The standard approach to compute rigid SfM is based in bundle adjustment (BA). This technique tries to minimize the reprojection error between observed objects and projected objects (observed in more than one image), allowing the estimation of the 3D points and camera locations [9]. SfM method is normally a post-processing technique giving, in general words, more accurate results as much information is extracted from images. But some applications, such as automatic navigation, need an on-line mapping in real time. This means that the agent needs to compute SfM method in real time to be able to navigate in an unknown world, receiving the name of Simultaneous Localization And Mapping (SLAM), involving tasks of estimating the 3D environment structure at the same time the camera is located in the map.

Many works have been developed in this area, allowing to use different sensors (monocular [10], stereo [11], RGB-D [12] or laser technology [13], among others). We can make a division in visual SLAM between direct [14] and feature-based [3]. Direct approaches use directly the image pixels minimizing the photometric error, while feature-based systems extract features from the images and use them as keypoints to apply BA techniques. Another classification can be made between filtered [15] or keyframe-based [16] techniques, resulting the first idea in a filtering technique based in the propagation of probability functions and the second idea in a sparse implementation requiring graph optimizations. The work carried out for the last decades has allowed these techniques to move from only research to the development stage, appearing some companies that offer consumer tools based in SLAM techniques.

Another classification, with special interest for us, can be made regarding the position of the objects in the scene. Standard implementations of SfM (and SLAM) assume a rigid world, but as we said, sometimes this assumption cannot be made. The term non-rigid structure from motion (NRSfM) appeared to take into account this kind of movements, and it assumes the camera can move inside the world but also scene objects can move (or deform), as we can see in the right side of Fig. 2. This is an ill-posed problem as we are not able to build the 3D objects from 2D points triangulation. In fact, a different configuration (non-rigid object deformation) can be observed in each image, making the direct estimation of the geometries difficult.

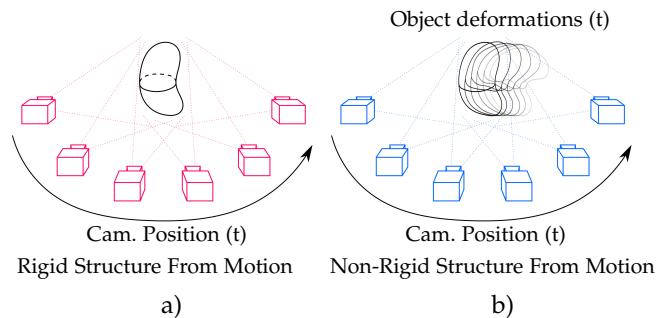


Fig. 2: a) Rigid SfM allowing a reconstruction of the world from different views. b) Non-rigid SfM implies that both the camera and the scene may take different positions and shapes (time-dependent).

During the last years, many works regarding NRSfM have been developed trying to solve this problem. Some of them used factorization approaches to obtain low-rank representations of the objects from image streams [17], where the 3D shape in each frame is a linear combination of a set of basis shapes (previously applied to rigid shapes by [18]). Also the non-rigid movements have been computed as a union of subspaces to model complex motions with clustering local subspaces [19]. Some other works computed the trajectory space by a linear combination of basis trajectories instead of basis shapes using *generic* bases [20]. Non-linear dimensionality reduction methods have also been used to model the 3D shape as a non-linear combination of basis shapes, using kernel functions [21], allowing a reduced number of bases if the problem is non-linear and the kernel

is well defined. Also bayesian implementations have been used [22].

Priors were introduced to narrow down the search for an optimal solution [23], and some other works allow a prior less estimation of the structure from motion assuming compressible 3d objects as a block sparse dictionary learning problem [24], and also applying dense variational reconstructions requiring GPU acceleration [25].

Other works reconstruct the shape of a deformable object using templates previously created [26], obtaining accurate results, but requiring 2D parameterizations as they work only with the external surface, also with curved geometries [27]. It means some energy constraints need to be imposed to assure boundary continuity and isometric deformations. Other works estimate the surface normals to help in the deformation tracking [28]. And also simple physics-based implementations have tried to solve the motion of the external surface using physical priors and Kalman filter formulations [29].

There is also another approach that involves the use of the classical theory of finite element methods to estimate and track deformable objects with some advantages like robustness and real forces estimation in large deformations, applied to 1D objects [30] but also to more complex surfaces [31], [32].

To the best of our knowledge, there is no alternative approach, however, in which the physics of the system under consideration is solved rigorously. This is due, undoubtedly, to the high computational cost of solving a high-fidelity model by employing state-of-the-art techniques such as finite elements or finite differences. This motivates our choice of reduced-order models, whose accuracy as a function of the complexity of the model is nowadays well-known and can be controlled by the analyst.

### 3 OVERVIEW OF THE PROPOSED METHOD

To overcome the just mentioned difficulties, we propose to track deformable solids from video sequences by solving the (possibly non-linear) elasticity problem. The goal is to create an agent able to understand the physical deformations of the objects to finally show some information to a user (e.g. stress or strain fields, or deformations of internal and hidden areas). We make use of reduced-order and parametric models, with which we solve the inverse problem to estimate the kinematic (displacements and strain fields) and dynamic (forces, stress) states, along with the rest of parameter values. Equivalently, the problem can be viewed as an example of data assimilation, in which we obtain parameter values by measuring displacements in the object external surface. Our approach is based on a two-stage problem. First, in an off-line stage we precompute a reduced-order, multiparametric approximation of the displacement field of the solids. This can be viewed as the estimation of a response surface. Second, an on-line stage is carried out to optimize the parameter values that better approximate the measured displacements in the video sequence under real time constraints.

We do not impose any artificial spatio-temporal restriction or prior, nor *ad hoc* energy laws, since we are solving the actual physics in a finite element framework, guaranteeing

energy conservation—dissipative behaviors do not impose any additional conceptual difficulty to this method—and elastic (or hyperelastic) constitutive laws. Of course, initial and boundary conditions need to be applied in the computation step, but they can also be considered as parameters in the parameterized problem. The proposed method can be applied both to articulated and continuous solids, beating any other existing method in terms of frequency rates (since it only requires an *evaluation* of the parametric solution, rather than a true simulation), accuracy and robustness (noisy observations are minimized without the need of Bayesian implementations). Our method has no dependency on the acquisition system used (monocular, stereo or RGB-D systems), but we think, however, that one of the biggest advantages appears with the use of monocular sensors, since the introduction of parametric models helps in the minimization process, converging on an admissible solution.

Strictly speaking, we can say that our procedure is not a structure-from-motion method, since we are not building a 3D structure from the motion data. The proposed method is related to the *Shape from Template* family of methods [26], in which we register an off-line (reduced-order) parametric model to the camera observations.

## 4 PROBLEM FORMULATION

Very often the problem of NRSfM has been formulated respect to the motion of the visible portion of the deforming solid (the domain for which some information is captured by the camera). However, it is well-known that the motion of the visible part of the solid is heavily influenced by the whole geometry, including internal details, as described by the Navier equations of solid mechanics. Therefore, in this paper we formulate the problem as a complete, three-dimensional description of the solid [33], thus avoiding other NRSfM formulations focused only in the visible part of the solid whose motion was modeled by shell finite elements [29], [26]. We are interested in some problems like, for example, augmented reality for mechanical design or surgery guidance, some problems that allow us to assume that a full three-dimensional model is available off-line.

### 4.1 Kinematic description of deforming solids

Neglecting inertial effects and assuming, for simplicity, that there is no change in temperature, we consider a continuous solid  $\Omega$ , where we identify a point  $Q$ , with coordinates  $\mathbf{X} = (X, Y, Z)$  respect to *World* ( $W$  in Fig. 3). These coordinates are defined in the *undeformed configuration*  $\varphi_0 : \Omega \rightarrow \mathbb{R}^3$ . The object undergoes a deformation so that point  $Q$  moves to the new position  $Q'$  with coordinates  $\mathbf{X}' = (X', Y', Z')$ , respect to  $W$ , in the deformed configuration  $\varphi_t : \Omega \rightarrow \mathbb{R}^3$ . The function that explains the movement of the object is  $\phi_{\text{Obj}} : \Omega \times \mathcal{I} \rightarrow \mathbb{R}^3$ , where  $\mathcal{I} \in \mathbb{R}$  is the time interval where the movement is produced. Therefore, making use of a Lagrangian description of the amount of movement (material, always following the same infinitesimal particle), we can express the deformed configuration of point  $Q$  as

$$\mathbf{X}' = \phi_{\text{Obj}}(\mathbf{X}, t),$$

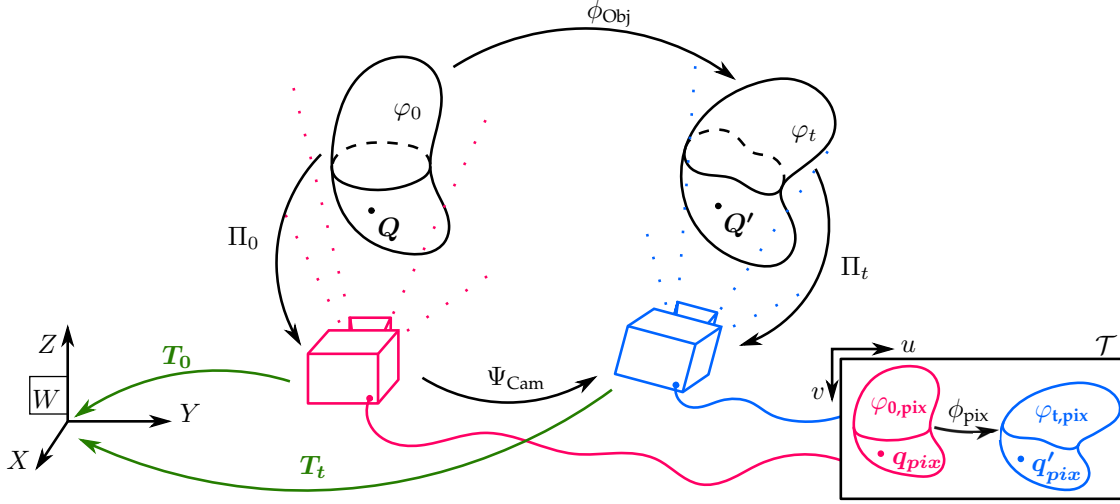


Fig. 3: Kinematic modeling of the proposed formulation.

where  $t$  is the instant of time elapsed between both states. The deformation gradient tensor  $\mathbf{F}$  (a two-point tensor) is the fundamental quantity that explains the deformation produced between two neighboring particles between the initial and the deformed state (assuming continuity in the mapping function). It is defined, with respect to the initial configuration, as

$$\mathbf{F} = \frac{\partial \mathbf{X}'}{\partial \mathbf{X}} = \frac{\partial \phi_{\text{Obj}}(\mathbf{X}, t)}{\partial \mathbf{X}} = \nabla \phi_{\text{Obj}}.$$

It is convenient to define a strain measurement independent of the type of movement (rigid-solid translation or rotation, relative deformations or the combination of all of them). For independence with respect to rotations, we use the *right Cauchy-Green deformation tensor*, defined as

$$\mathbf{C} = \mathbf{F}^T \mathbf{F}.$$

And finally, using the *Green-Lagrange strain tensor* (material frame of reference) we can measure the deformations independently of rigid-body motions

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I}) = \frac{1}{2}((\nabla \mathbf{U})^T + \nabla \mathbf{U} + (\nabla \mathbf{U})^T \nabla \mathbf{U}), \quad (1)$$

with  $\mathbf{U}(\mathbf{X}) = \mathbf{X}' - \mathbf{X}$  the material displacement.

## 4.2 Camera-centered kinematic description

Let us assume that we obtain an image of the boundary of an opaque solid at the undeformed configuration,  $\partial\Omega_0$ . We consider a standard perspective projection camera [34] (after intrinsic calibration and lens distortion correction) applying a transformation  $\Pi_0 : \partial\Omega_0 \rightarrow \mathcal{T}$ , where  $\mathcal{T} \in \mathbb{R}^2$  represents the image space. This camera operator  $\Pi_0$  maps a point  $\mathbf{Q}$  from *World* coordinates to  $\mathbf{q}_{\text{pix}}$  in *pixel* coordinates  $\mathbf{x}_{\text{pix}} = (u, v)$  making use of the camera intrinsic and extrinsic parameters. The same process occurs with projection  $\Pi_t : \partial\Omega_t \rightarrow \mathcal{T}$ , where points in the deformed configuration  $\varphi_t$  are projected to the image  $\varphi_{t,\text{pix}} \subset \mathbb{R}^2$ .

## 4.3 Solving the unknown kinematics

Using a monocular implementation, the unknown mapping functions in our problem for any time instant  $t \in \mathcal{I}$  are

- The displacements of the current configuration in a material description (i.e., with respect to the reference configuration)  $\phi_{\text{Obj}}(\mathbf{X}, t)$ .
- The camera projection  $\Pi_t$ , but only the extrinsic component  $\mathbf{T}_t$ , since it is the part that can change with the camera movement (and also the redundant function  $\Psi_{\text{Cam}} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ , that maps the movement of the camera with respect to its initial position).

The estimation of the mapping  $\Pi_t$  for each frame capturing rigid scenes is known, as we said, as the Structure-from-Motion (SfM) problem and it involves the estimation of the extrinsic parameters  $\mathbf{T}_t$  (and the three-dimensional position of the points of the object,  $\mathbf{Q}_t$ ), by minimizing a reprojection error in  $L_2$ -norm, commonly. If we also add the estimation of  $\phi_{\text{Obj}}$ , where objects can deform at any time instant, the problem becomes more difficult (ill-posed). It is possible that a different deformed configuration appears at every instant, with different camera positions. In this paper we assume that the camera captures both static and deformed points. Rigid or static points are related to the general scene (static objects, blue points in Fig. 4) and are used to estimate the pose of the camera, and non-static points are used to determine the deformations of the non-rigid objects in the scene (red points in Fig. 4). We assume all deformable objects have fixed boundary conditions, so at least one point of the boundary is fixed in the scene. However, we could also assume that the objects, in addition to be deformed, move or rotate around the scene, in which case it would be necessary to estimate the relative position of the camera with respect to the object solving a similar but double problem (PnP problem [35] and displacements).

The position and orientation (pose) of the camera  $\mathbf{T}_t$  is estimated by using fixed points (image features in our case) along with consensus techniques such as RANSAC [36] to identify the subset of fixed points in the scene. To estimate the camera pose, the reprojection error  $d(\mathbf{x}_{\text{pix}}, \hat{\mathbf{x}}_{\text{pix}})$  is minimized, with  $d$  the euclidean distance between the

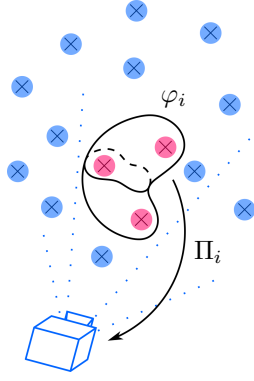


Fig. 4: In general, we consider applications where an important part of the scene is static (blue points), while some regions (red points) are non-rigid or deformable.

pixel coordinates of a point observed in an image and the pixel coordinates of the point projected on the image (from the 3D point  $\mathbf{x}$  and the camera projection  $\Pi_t$ , parameters that are also optimized). We use ORB-SLAM2 [37] for the estimation of the camera pose and the static mapping.

The next task is to estimate the object deformations from its projection in the images. We use the red points in Fig. 4, or in other words, the *outliers*. RANSAC techniques applied to SLAM usually neglect those points that do not remain fixed in the scene. But our true interest lies in the information associated with the deformation of the objects. It is located precisely there, in the outliers. Therefore, these points provide the most valuable information about the displacements suffered by the objects, which we have to identify after carrying out a registration of the objects.

#### 4.4 Equilibrium in the deformable solid

In previous sections we included a brief summary of the kinematic analysis of the deformable solid, but to fully resolve the mechanical problem it is necessary to take into account the stress analysis to guarantee equilibrium. If we split the solid by an arbitrary plane of unit normal  $\mathbf{n}$ , a force per unit area appears in any point of the newly created surface,

$$\mathbf{t}(\mathbf{n}) = \boldsymbol{\sigma} \mathbf{n}; \quad \boldsymbol{\sigma} = \sum_{i,j=1}^3 \sigma_{ij} \mathbf{e}_i \otimes \mathbf{e}_j$$

where  $\mathbf{t}$  is the stress vector,  $\mathbf{n}$  is the vector normal to the plane and  $\boldsymbol{\sigma}$  is the *Cauchy stress tensor*.

The stress magnitude that is *conjugate* to the gradient of deformation tensor  $\mathbf{F}$  (their product gives an energy measure) is the so-called *first Piola-Kirchhoff stress tensor*  $\mathbf{P}$ , defined as

$$\mathbf{P} = J \boldsymbol{\sigma} \mathbf{F}^{-T},$$

where  $J = \det(\mathbf{F})$  is the Jacobian of the transformation  $\phi$ . The stress measure conjugate to tensor  $\mathbf{E}$  is the *second Piola-Kirchhoff stress tensor*  $\mathbf{S}$  defined as

$$\mathbf{S} = J \mathbf{F}^{-1} \boldsymbol{\sigma} \mathbf{F}^{-T},$$

so we can write equivalently

$$\mathbf{P} = \mathbf{F} \mathbf{S}.$$

In order to guarantee equilibrium in the solid, we must satisfy the following equilibrium equation, in the absence of inertia terms (assuming a quasi-static process):

$$\nabla \cdot \mathbf{P} + \mathbf{B} = \mathbf{0} \text{ in } \Omega_t, \quad (2)$$

where  $\mathbf{B}$  are the applied body forces. These equations need to be supplemented with appropriate boundary conditions,

$$\begin{cases} \boldsymbol{\sigma} \mathbf{n} = \bar{\mathbf{t}} & \text{on } \Gamma_t, \\ \mathbf{u} = \bar{\mathbf{u}} & \text{on } \Gamma_u \end{cases}$$

where  $\bar{\mathbf{t}}$  are the traction forces acting along the boundary region  $\Gamma_t$  in terms of the unit normal to the boundary,  $\mathbf{n}$ , and finally  $\bar{\mathbf{u}}$  are the prescribed displacements defined in  $\Gamma_u$ . The *weak form* of Eq. (2) is created to relax continuity and derivability restrictions on the functions that approximate the final solution, and is obtained after multiplying by an arbitrary function (admissible variation) and integrating by parts,

$$\int_{\Omega} \mathbf{S} : \delta \mathbf{E} \, d\Omega = \int_{\Gamma_t} \bar{\mathbf{t}} \delta \mathbf{u} \, d\Gamma. \quad (3)$$

#### 4.5 Constitutive equations

The last ingredient to describe the whole mechanical process is to define its constitutive nature. In this work we consider hyperelastic material laws instead of linear elasticity, as linear materials may introduce artificial increments in volume when large deformations occur. Hyperelastic materials are those whose stress-strain relationship derives from a strain energy density function  $W$ . For these materials, the second Piola-Kirchhoff tensor  $\mathbf{S}$  can be obtained from the strain energy density function  $W$  as

$$\mathbf{S} = \frac{\partial W(\mathbf{E})}{\partial \mathbf{E}}. \quad (4)$$

Adding Eq. (4) into Eq. (3), the problem to solve by using finite element methods [38] is

$$\int_{\Omega} \delta \mathbf{E} : \mathbf{C} : \mathbf{E} \, d\Omega = \int_{\Gamma_t} \bar{\mathbf{t}} \delta \mathbf{u} \, d\Gamma, \quad (5)$$

where  $\mathbf{C}$  is the fourth-order constitutive tensor. Section 7 shows a practical example of hyperelastic materials.

## 5 MODEL ORDER REDUCTION (MOR) METHODS

One of the most important aspects of our work is that we are assuming that the movement  $\phi_{\text{Obj}}$  is bounded and can be parameterized and projected onto a low-dimensional manifold. This means that the objects we simulate do not have the infinite number of degrees of freedom typical of continuum mechanics. Due to the laws of continuum mechanics, strong correlations exist between the displacement suffered by each material point. Actually, we can express the solution as a function of a much smaller number of degrees of freedom, what we call dimensions in our reduced space.

## 5.1 Dimensionality Reduction

The computational complexity of the continuum mechanics problem prevents its solution under real-time constraints. Note the non-linear character imposed by the strain measures introduced in Eq. (1), and also the common non-linear terms of hyperelastic laws. To guarantee that we comply with video frequencies, a reduction of the complexity of the problem is mandatory. In addition, the parameter dependency (material properties, geometry parameters or boundary conditions, among others) increases the complexity of the problem.

Let us assume that the governing equation depends on a vector of parameters  $\boldsymbol{\mu} \in \mathcal{P}$ , where  $\mathcal{P}$  is the set of all possible values of the parameters, and a compact subset of  $\mathbb{R}^{\text{param}}$ . The manifold or solution set is  $\mathcal{M}$ , where all solutions  $\boldsymbol{U}(\boldsymbol{\mu})$  remain,

$$\mathcal{M} = \varphi(\mathcal{P}) = \{\boldsymbol{U}(\boldsymbol{\mu}) \in V : \boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^{\text{param}}\},$$

where  $\varphi$  is again the solution map and  $V$  is a suitable Hilbert space. The solution map  $\varphi$  is defined as

$$\varphi : \mathcal{P} \rightarrow V, \quad \boldsymbol{\mu} \mapsto \boldsymbol{U}(\boldsymbol{\mu}).$$

It is not possible to work with the original solution set due to the finite memory of computers, so we have to make a simplification by choosing a suitable discretization technique to obtain the high-fidelity solution set  $\mathcal{M}_h$ , also known as the discrete manifold, usually obtained by finite element techniques. We assume that  $\mathcal{M}_h$  is so close to  $\mathcal{M}$  that very small differences appear at the discretization points. The discretized solution of the equation is  $\boldsymbol{U}_h(\boldsymbol{\mu})$  and it belongs to a finite-dimensional subspace  $V_h$  of dimension  $N_h$ ,

$$\mathcal{M}_h = \varphi_h(\mathcal{P}) = \{\boldsymbol{U}_h(\boldsymbol{\mu}) \in V_h : \boldsymbol{\mu} \in \mathcal{P}\} \subset V_h.$$

After applying the approximation to  $\boldsymbol{U}(\boldsymbol{\mu})$ , we obtain also the discrete solution map

$$\varphi_h : \mathcal{P} \rightarrow V_h, \quad \boldsymbol{\mu} \mapsto \boldsymbol{U}_h(\boldsymbol{\mu})$$

There are many MOR methods to reduce the dimensionality of the problem with the goal of obtaining a reduced basis with an optimal number of components. The reduced solution thus belongs to a low-dimensional subspace  $V_r \subset V_h$  of dimension  $N_r \ll N_h$  where the construction of the basis of  $V_r$  is generated from a set of  $r$  functions. The particular form of constructing these functions can be separated between *a posteriori* and *a priori* methods. The classical methods are *a posteriori*, where some data is coming from an unknown source and statistical tools like Principal Component Analysis [39], [40] are applied to project the data in a reduced and more efficient space. The Proper Orthogonal Decomposition (POD) method [41] is a common example. But, if a multiparametric problem has to be solved, the curse of dimensionality appears, so *a priori* methods can be applied to solve the mechanical problem directly in the reduced space, without evaluating the whole set of solutions in the high-dimensionality space.

To better understand the procedure, we detail next the type of model order reduction methods commonly applied to continuum mechanics and how they can be of great help in our work.

## 5.2 MOR in Computational Mechanics

We are looking for a suitable approximation of a space, time and parameter-dependent field  $\boldsymbol{U}(\boldsymbol{X}, \boldsymbol{\mu}, t)$ . Under the term *model order reduction* we encompass a family of methods that look for an approximation of the type

$$\boldsymbol{U}(\boldsymbol{X}, t, \boldsymbol{\mu}) \approx \sum_{i=1}^{\text{nmOR}} \boldsymbol{F}_i(\boldsymbol{X}) \circ \boldsymbol{G}_i(t) \circ \boldsymbol{H}_i(\boldsymbol{\mu}), \quad (6)$$

where  $\boldsymbol{F}_i(\boldsymbol{X})$ ,  $\boldsymbol{G}_i(t)$  and  $\boldsymbol{H}_i(\boldsymbol{\mu})$  represent the so-called *modes* of the approximation, i.e.,  $\text{nmOR}$  vector-valued functions approximated in a finite element sense that best approximate the unknown field  $\boldsymbol{U}$ . The symbol “ $\circ$ ” represents the Hadamard or Schur (component-wise) product of vectors.

The simplest way to determine an optimal approximation of the type given by Eq. (6), sometimes referred to as *affine* or *separate*, is by applying Principal Component Analysis of a set of snapshots of the evolution of the system, called Proper Orthogonal Decomposition in computational mechanics [42]. Once the modes have been identified, by injecting the approximation given by Eq. (6) into the weak form of the differential equation governing the problem, the reduced system can be solved.

Other methods like Reduced Basis (RB) employ directly a set of snapshots of the system, whose parameter value is chosen under well-defined error criterion [43].

On the other hand, Proper Generalized decompositions (PGD) [44] [45] do not employ any snapshot of the system. This approach presents several advantages. First of all, it is not necessary to solve the problem for all of the parameter combinations in the high-fidelity space. The problem is solved in the separated space, Eq. (6), and it can be computed off-line and stored advantageously in memory in the form of a collection of vectors. These vectors include only the nodal values, at mesh vertices, of the functions  $\boldsymbol{F}_i$ ,  $\boldsymbol{G}_i$  and  $\boldsymbol{H}_i$ . At any other point of the model, the value of the field  $\boldsymbol{U}$  is obtained by finite element interpolation. Secondly, this separated representation presents important advantages for the computation of inverse problems. The sensitivities (parameter influence in the solution) can be computed straightforward thanks to this precise form of the approximation, very important to boost the optimization step. A detailed formulation is given in Section 6.

## 5.3 Proper Generalized Decompositions (PGD)

As we said, PGD method does not need any observation of the system to construct the low-rank approximation to the unknown field. The method is based in two stages: an off-line step where the computation of the low-rank approximation given by Eq. (6) is performed; and an on-line phase where the parametric solution is evaluated under real-time constraints. To better explain the basics of PGD, we consider an introductory example. As a toy model problem we consider linear elasticity: the case of a cantilever beam in which the position of the applied load is considered as a parameter of the system.

### 5.3.1 How PGD works: linear elasticity

Let us assume a vertical and constant force  $F$ , applied over the upper surface of a cantilever beam. The unknown field

of the problem is the displacement  $U(\mathbf{X}, s)$ , which depends on the particular material position  $\mathbf{X}$  and on the position of the load,  $s$  (see Fig. 5). The load can be applied at any point of the upper part  $s \in \bar{\Gamma} \subset \partial\Omega$ , with  $\bar{\Gamma}$  the portion of the boundary  $\Gamma_t$  with non-vanishing Neumann conditions.

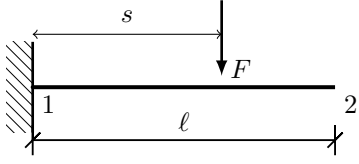


Fig. 5: Cantilever beam problem. A moving load  $F$  is parameterized by its position coordinate  $s$ .

The strong form of the governing equation (linear elasticity) under infinitesimal strain theory [38] is

$$U(\mathbf{X}, s) = \begin{cases} \nabla \cdot \boldsymbol{\sigma} + \mathbf{B} = \mathbf{0} & \text{in } \Omega, \\ \boldsymbol{\sigma} \mathbf{n} = \bar{\mathbf{t}} & \text{on } \Gamma_t, \\ \mathbf{U} = \bar{\mathbf{U}} & \text{on } \Gamma_U, \end{cases} \quad (7)$$

where  $\boldsymbol{\sigma}$  is the stress tensor,  $\mathbf{B}$  is the body force per unit volume (assumed vanishing in this example, for the sake of simplicity),  $\bar{\mathbf{t}}(s)$  are the traction forces applied to the solid and  $\bar{\mathbf{U}}$  are the prescribed displacements (clamped support at point 1). The constitutive equations for elastic materials, also known as *Hooke's law*, represents the material behavior and relates stress and strain as

$$\boldsymbol{\sigma} = \mathbf{C}:\boldsymbol{\varepsilon},$$

with  $\mathbf{C}$  the fourth-order constitutive tensor and  $\boldsymbol{\varepsilon}$  the (small) strain tensor. Assuming a hyperelastic framework (of which linear elasticity is a particular example), this constitutive tensor can be obtained by differentiating the strain energy density function  $W$ , Eq. (4), that in this case takes the form

$$W(\boldsymbol{\varepsilon}) = \frac{1}{2}E\boldsymbol{\varepsilon}^2,$$

where parameter  $E$  is the Young's modulus. Finally, the strain-displacement (kinematic) equations are

$$\boldsymbol{\varepsilon} = \frac{1}{2}[\nabla U + (\nabla U)^T].$$

Keeping in mind that  $U$  depends on  $\mathbf{X} = (X, Y)$  and  $s$ , the weak form of the governing equation under the PGD formalism can be described as

$$\int_{\bar{\Gamma}} \int_{\Omega} \nabla_s U^* : \boldsymbol{\sigma} \, d\Omega \, d\bar{\Gamma} = \int_{\bar{\Gamma}} \int_{\Gamma_t} U^* \mathbf{t} \, d\Gamma \, d\bar{\Gamma} \quad (8)$$

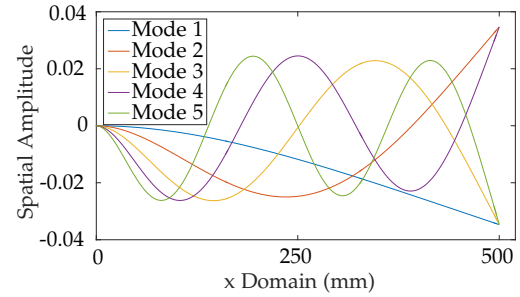
where  $U^* \in H_0^1$  is an arbitrary test function in the appropriate Sobolev space of functions vanishing along the Dirichlet boundary of the solid and  $\nabla_s = \frac{1}{2}[\nabla + (\nabla)^T]$  is the symmetric gradient operator. The solution  $U(\mathbf{X}, s)$  is assumed to be expressed in separate form,

$$U(\mathbf{X}, s) \approx \sum_{i=1}^{n_{\text{MOR}}} \mathbf{F}_i(\mathbf{X}) \circ \mathbf{G}_i(s), \quad (9)$$

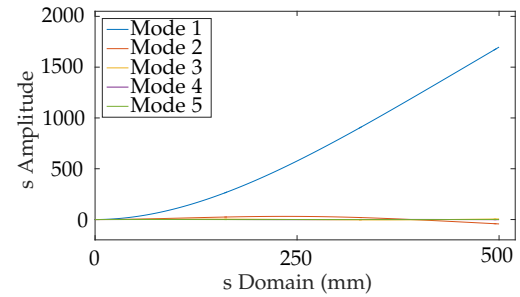
where  $n_{\text{MOR}}$  is the number of sums (rank of the tensor decomposition of the solution),  $\mathbf{F}_i(\mathbf{X})$  are the space modes in separate variables and  $\mathbf{G}_i(s)$  are the modes referred to the position of the load. An alternating direction scheme is

used to obtain the  $n_{\text{MOR}}$  functions  $\mathbf{F}_i(\mathbf{X})$  and  $\mathbf{G}_i(s)$  so that, at iteration  $p$  we search the  $p + 1$  functions  $\mathbf{R}(\mathbf{X})$  and  $\mathbf{S}(s)$  that enrich the solution

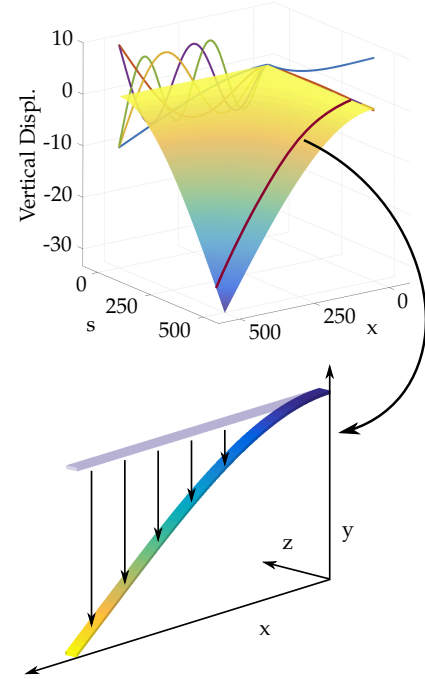
$$U_{p+1}(\mathbf{X}, s) = U_p(\mathbf{X}, s) + \mathbf{R}(\mathbf{X}) \circ \mathbf{S}(s).$$



(a)



(b)



(c)

Fig. 6: Separated solution of the vertical displacement. (a) First five modes depending on space. (b) First five modes depending on the position of the load. (c) Built solution using the sum of the product of modes. Note how the result of the PGD approximation provides with a sort of response surface, without the need of any parameter space sampling strategy. Finally, a slice of the surface with the prescribed load position  $s_0$  builds the solution.

Therefore, the test function is

$$U^*(\mathbf{X}, s) = \mathbf{R}^*(\mathbf{X}) \circ \mathbf{S}(s) + \mathbf{R}(\mathbf{X}) \circ \mathbf{S}^*(s).$$

The load  $F$  has been designed as a unitary force acting along the vertical axis  $Y$ ,

$$\mathbf{t} = F \delta(X - s) \mathbf{e}_Y, \quad X \in \Gamma_t,$$

where  $\delta$  is the Dirac delta function. Function  $\mathbf{t}$  needs also to be expressed in separate form to comply with the PGD formalism, i.e.,

$$t_Y \approx \sum_{j=1}^m h_j(\mathbf{X}) k_j(s); \quad t_X = 0,$$

where  $m$  is the number of sums to approximate function  $t_Y$  and  $h_j(\mathbf{X})$ ,  $k_j(s)$  are the functions in space and load position, respectively. To solve the non-linear product of functions  $\mathbf{R}(\mathbf{X})$  and  $\mathbf{S}(s)$ , the algorithm uses an alternating direction scheme based on the fixed point method at every enrichment iteration. For a detailed description of the problem and a Matlab implementation, the interested reader can consult [45].

The first modes of  $F(\mathbf{X})$  (only its vertical component, since it is a vector-valued displacement field) and  $G(s)$  are shown in Fig. 6.(a) and 6.(b), respectively. Finally, the sum of the products of pairs of modes builds the approximate solution, as it is shown in Fig. 6.(c).

It is important to highlight the fact that the accuracy of the proposed reduced-order method can be controlled by the user. The number of terms in Eq. (9),  $n_{\text{MOR}}$  controls this level of accuracy, with the help of a suitable error estimator, as the ones proposed in [46] [47] [48], for instance.

## 6 A DEFORMABLE IMPLEMENTATION OF ORB-SLAM2

We propose in this section a deformable implementation of the ORB-SLAM2 system [37]. ORB-SLAM2 is a sparse and feature-based SLAM system with outstanding results of accuracy [3]. It is used to build a rigid scene and locate the camera in any frame. In our work, after a few seconds of video capturing the static scene, we get the 3D point cloud and apply an automatic and on-line registration against a CAD model of the object, based in RANSAC methods and the ICP (Iterative Closest Point) algorithm [49].

The registration step allows us to know which points are deformable and, subsequently, to estimate the displacements of the 3D SLAM points ( $\mathbf{U}_{\text{SLAM}}(\mathbf{X}, \boldsymbol{\mu})$ ) using the precomputed data from the CAD object

$$U_{\text{SLAM}}(\mathbf{X}, \boldsymbol{\mu}) = f\left(\mathbf{U}_{\text{Neighs,CAD}}(\mathbf{X}, \boldsymbol{\mu})\right). \quad (10)$$

To do that, a mapping function between our undeformed 3D model and the 2D image features need to be established. It means we can afterwards deform the real object, extract image features in any frame and estimate the parameter values  $\boldsymbol{\mu}$  with respect to the precomputed set of solutions. The goal is to obtain the displacements  $\mathbf{U}_{\text{SLAM}}(\mathbf{X}, \boldsymbol{\mu})$  to add to the static set of deformable 3D points initially estimated using the SLAM technique. This solution must maintain the separate multiparametric structure to obtain a specific solution for each possible value of our set of parameters.

Therefore, the displacements of the SLAM points are expressed as a function of the closest registered CAD points. The procedure can be seen in the general overview schema (Fig. 7).

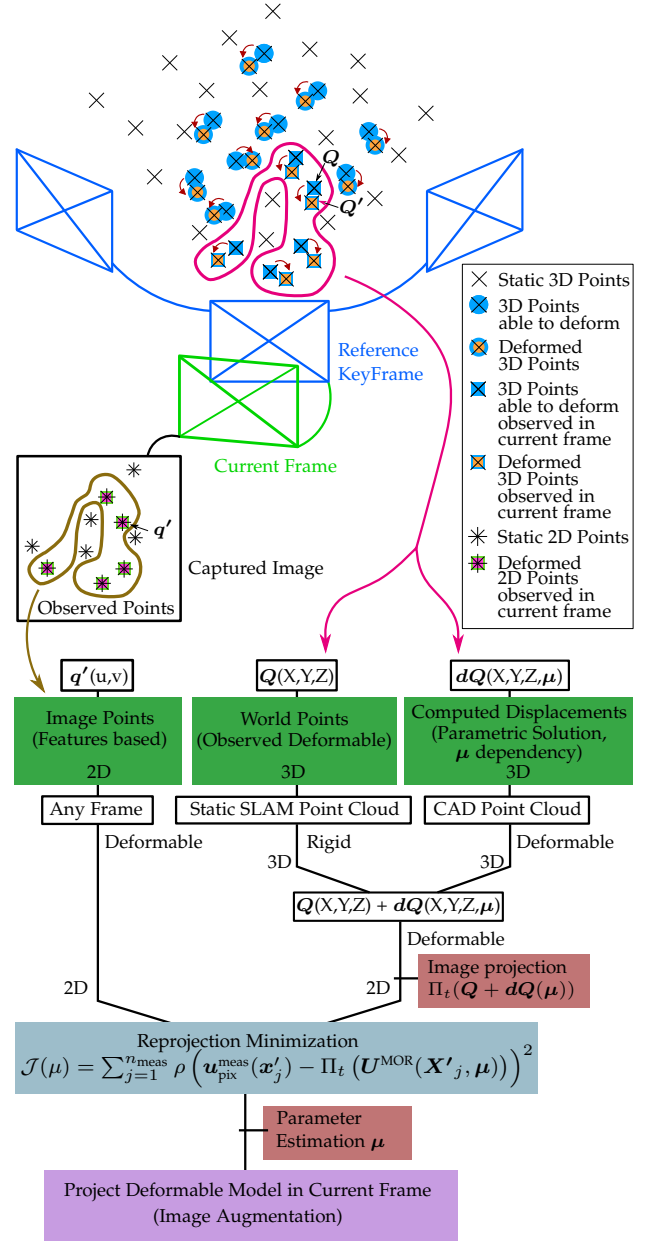


Fig. 7: Deformable object tracking and augmentation procedure based in ORB features matching.

In the visual scanned point cloud there are 3D points that remain static (used to fix camera position and orientation) and 3D points that are assigned to the deformable part of the domain in the current observation. Let us focus on the points belonging to the deformable part,  $Q$ , with coordinates  $(X, Y, Z)$ . The position in its deformed configuration is  $Q'$ , where  $\mathbf{U} = (\mathbf{X}' - \mathbf{X}) = (X', Y', Z') - (X, Y, Z)$ , but this displacement  $\mathbf{U}$  is actually estimated from the deformations of the CAD object, which depend on the multiparametric solution projected in separate variables. The point in its deformed configuration ( $Q'$ ) is observed from the image as  $q'$  with pixel coordinates  $(u', v')$ , which must match

the projection  $\mathbf{q}'_e = \Pi_t(\mathbf{Q} + \mathbf{U}(\boldsymbol{\mu}))$  for the optimal set of parameters  $\boldsymbol{\mu}$ . The reprojection error in the image is therefore defined as  $d(\mathbf{q}', \mathbf{q}'_e)$  and the optimal parameter values of our parametric solution that minimize this error for the whole set of observed points are obtained from the next functional

$$\mathcal{J}(\boldsymbol{\mu}) = \sum_{j=1}^{n_{\text{meas}}} \rho(\mathbf{u}_{\text{pix}}^{\text{meas}}(\mathbf{x}'_j) - \Pi_t(\mathbf{U}^{\text{MOR}}(\mathbf{X}'_j, \boldsymbol{\mu})))^2 \quad (11)$$

where  $n_{\text{meas}}$  are the number of measurements,  $\mathbf{u}_{\text{pix}}^{\text{meas}}$  the pixel coordinates measured in the image,  $\mathbf{U}^{\text{MOR}}(\mathbf{X}'_j, \boldsymbol{\mu})$  the 3D displacement predicted by the reduced-order parametric solution,  $\Pi_t$  the projection to the image plane at any instant  $t$  (frame), and  $\rho$  the Huber robust function. It is important to highlight that  $\mathbf{U}^{\text{MOR}}(\mathbf{X}'_j, \boldsymbol{\mu})$  represents the whole solution of our hyperelastic problem in question by storing the position, deformations and stresses of each point, with parametric dependence and represented in the reduced space. However, to simplify the process,  $\mathbf{U}$  in the functional defined in Eq. (11) only represents the deformed position of each point. Using Eq. (11) we can estimate the value of the  $\boldsymbol{\mu}$  parameters that minimize the reprojection error

$$\hat{\boldsymbol{\mu}} = \underset{\boldsymbol{\mu}}{\text{argmin}} \mathcal{J}(\boldsymbol{\mu}).$$

This minimization can be advantageously accomplished by employing the Levenberg-Marquardt algorithm [50][51], which takes the form

$$[\mathbf{J}^T \mathbf{J} + \lambda \text{diag}(\mathbf{J}^T \mathbf{J})] \delta = \mathbf{J}^T [\mathbf{U}_{\text{pix}}^{\text{meas}}(\mathbf{x}'_j) - \Pi_t(\mathbf{U}^{\text{MOR}}(\mathbf{X}'_j, \boldsymbol{\mu}))],$$

where  $\delta$  is the perturbation parameter that moves the parameters in the direction of steepest descent and  $\lambda$  an algorithmic parameter that adaptively varies the behavior between the Gradient Descent Method and the Gauss-Newton Method. Finally,  $\mathbf{J}$  is the Jacobian matrix, defined as

$$\mathbf{J} = \frac{\partial \mathbf{U}(\mathbf{x}, \boldsymbol{\mu})}{\partial \boldsymbol{\mu}}. \quad (12)$$

Here lies precisely the key point that makes the separate representation of the solution so powerful in the parameter optimization process. Recall the general form of PGD, Eq. (6), where we can notice that due to its separate form, a separate differentiation can also be performed. We can precompute the derivative function of each parameter, Eq.( 12), offline, since the finite element approximation computes the derivative as a multiplication of matrices. These derivative functions are usually known as *sensitivities* of the solution [52] with respect to each parameter  $\mu_k$ . We can interpret them as the local variation of the function  $\mathbf{U}(\mathbf{X}, \boldsymbol{\mu})$  with respect to the changes in parameter  $\mu_k$

$$\mathbf{J}_k = \sum_{i=1}^N \mathbf{F}_i^1(\mathbf{X}) \circ \dots \circ \frac{\partial \mathbf{F}_i^k(\mu_k)}{\mu_k} \circ \dots \circ \mathbf{F}_i^{n_{\text{param}}}(\mu_{n_{\text{param}}}).$$

It allows us to avoid the exploration of all of the parametric space in search of the derivatives, and directly apply the differentiation on the separated variable vectors, involving a great reduction in the computational task. In sum, we need the partial components of the derivative terms of the

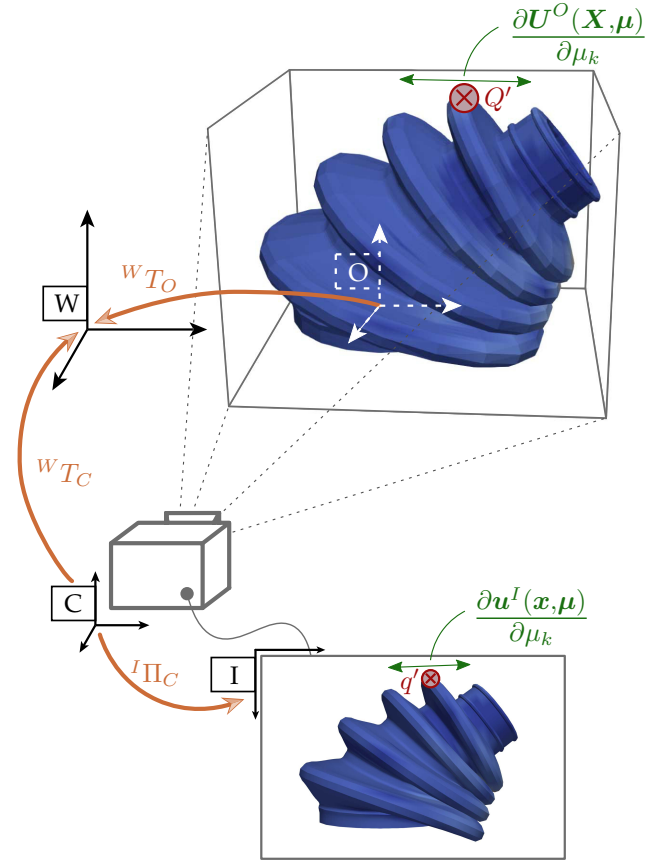


Fig. 8: Transformations and projection of the precomputed sensitivities from the original system of coordinates to the image reference.

multiparametric solution projected on the image, since we are minimizing the reprojection error (2D), but we precomputed the sensitivities in the original 3D space, so we have to project these sensitivities onto the image plane. A graphic example to understand this process appears in Fig. 8, where

$$\mathbf{J}_k = \frac{\partial \mathbf{U}^O(\mathbf{X}, \boldsymbol{\mu})}{\partial \mu_k} \quad (13)$$

refers to the partial derivative term of the multiparametric 3D solution of the CAD object with respect to the parameter  $\mu_k$ , evaluated at point  $\mathbf{Q}'$  and with the origin of coordinates defined on that object ( $O$ ), and

$$\mathbf{j}_k = \frac{\partial \mathbf{u}^I(\mathbf{x}, \boldsymbol{\mu})}{\partial \mu_k} \quad (14)$$

represents the same partial derivative projected on the image plane and evaluated on the point  $\mathbf{q}'$ . To obtain the value of this sensitivity in the image plane,  $\mathbf{j}_k$ , we use the chain rule

$$\mathbf{j}_k = \frac{\partial \mathbf{u}^I(\mathbf{x}, \boldsymbol{\mu})}{\partial \mathbf{U}^C(\mathbf{X}, \boldsymbol{\mu})} \frac{\partial \mathbf{U}^C(\mathbf{X}, \boldsymbol{\mu})}{\partial \mathbf{U}^W(\mathbf{X}, \boldsymbol{\mu})} \frac{\partial \mathbf{U}^W(\mathbf{X}, \boldsymbol{\mu})}{\partial \mathbf{U}^O(\mathbf{X}, \boldsymbol{\mu})} \frac{\partial \mathbf{U}^O(\mathbf{X}, \boldsymbol{\mu})}{\partial \mu_k},$$

where  $\mathbf{u}^I(\mathbf{x}, \boldsymbol{\mu})$  is the position of point  $\mathbf{q}'$  with respect to the image plane,  $\mathbf{U}^C(\mathbf{X}, \boldsymbol{\mu})$  with respect to the camera coordinates,  $\mathbf{U}^W(\mathbf{X}, \boldsymbol{\mu})$  with respect to the global coordinate axis (*World*) and  $\mathbf{U}^O(\mathbf{X}, \boldsymbol{\mu})$  with respect to the coordinate

axis defined in the object. The solution derivative terms are known in the object reference, Eq. (13), and their consecutive transformations in the Euclidean space  $\mathbb{R}^3$  are carried out by the matrices  ${}^W T_O$  and  ${}^W T_C$ , which are assumed to be known since they are estimated by the static part of ORB-SLAM2 and the registration step. However, the projection on the image plane  ${}^I \Pi_C$  does not allow such a simple treatment, since the camera conical projection used depends on the location of the 3D point with respect to the image plane. This means that the projection of the derivative terms  $\mathbb{R}^3 \mapsto \mathbb{R}^2$  must take into account the position of the point where the gradients are evaluated. This projection is known as the image Jacobian, and is expressed as

$${}^I J \Pi_C = \frac{\partial \mathbf{u}^I(\mathbf{x}, \boldsymbol{\mu})}{\partial \mathbf{U}^C(\mathbf{X}, \boldsymbol{\mu})} = \begin{bmatrix} \frac{1}{Z_C} f_x & 0 & -\frac{X_C}{Z_C^2} f_x \\ 0 & \frac{1}{Z_C} f_y & -\frac{Y_C}{Z_C^2} f_y \end{bmatrix},$$

where  $(X_C, Y_C, Z_C)$  are the 3D coordinates of point  $\mathbf{Q}$  with respect to the camera axes. Finally, the projection of the multiparametric derivative terms with respect to parameter  $\mu_k$  on the image plane is expressed as

$$\frac{\partial \mathbf{u}^I(\mathbf{x}, \boldsymbol{\mu})}{\partial \mu_k} = {}^I J \Pi_C \cdot {}^W T_C^{-1} \cdot {}^W T_O \cdot \frac{\partial \mathbf{U}^O(\mathbf{X}, \boldsymbol{\mu})}{\partial \mu_k}, \quad (15)$$

where we remember that the sensitivities defined with respect to the object axes are interpolated according to the neighbors in the separate variables precomputed solution. Regarding to the technical aspects, we use the feature search method developed in the ORB-SLAM2 code to track the deformations. Time requirements for the minimization algorithm are much smaller than the computation time of the SLAM tasks, so a simple parallelization of the whole code in a few threads allows us to work in a frequency of 30 fps using the CPU. We stop the *Mapping* tasks and work in *Localization* mode in ORB-SLAM2 once deformations are appearing.

## 7 EXPERIMENTS

We provide two examples to support our method. Both have been tested with different video sequences working in real time.

### 7.1 Rubber boot seal

This example shows the behavior of a rubber boot seal, the part that protects the gearshift of a vehicle. The object is deformed forced by the rotation of the gearshift in one direction, from  $-52^\circ$  to  $52^\circ$ , but the simulation has been only applied within the parametric range  $\theta \in [0^\circ, 52^\circ]$ , taking advantage of revolution symmetry.

The material behavior applied to the rubber is a non-linear law, called Neo-Hookean model, with the strain energy density

$$W = \frac{\mu}{2}(I_1 - 3) - \mu \ln(J) + \frac{\lambda}{2}(\ln(J))^2,$$

where  $I_1$  is the first invariant of the right Cauchy-Green deformation tensor

$$I_1 = \lambda_1^2 + \lambda_2^2 + \lambda_3^2,$$

and where  $\lambda_i$  are the eigenvalues. The mesh of the object has 25795 nodes and 18403 linear hexahedral elements.

Applying a sparse implementation of PGD [53], we obtain the parametric solution in the projected space, depending in space  $\mathbf{X} = (X, Y, Z)$  and one parameter,  $\theta$ , where the parameter optimization is a very fast process as we precompute the sensitivities. Due to the projection process of the data, a certain error in the approximation is made, as it can be seen in Fig. 9. With a total number of 29 modes, we get a cartesian mean error of 0.18 mm (the normalized error is 0.0041). For this solution, the compression factor is 92.58%, computed as

$$C(\%) = \left(1 - \frac{M_P}{M_O}\right) \cdot 100,$$

where  $M_P$  is the memory storage used by the projected solution and  $M_O$  is the memory cost necessary to store the original data. In other words, using a reduced space to express our solution we only need a 7.42% of the storage memory, allowing real time evaluations.

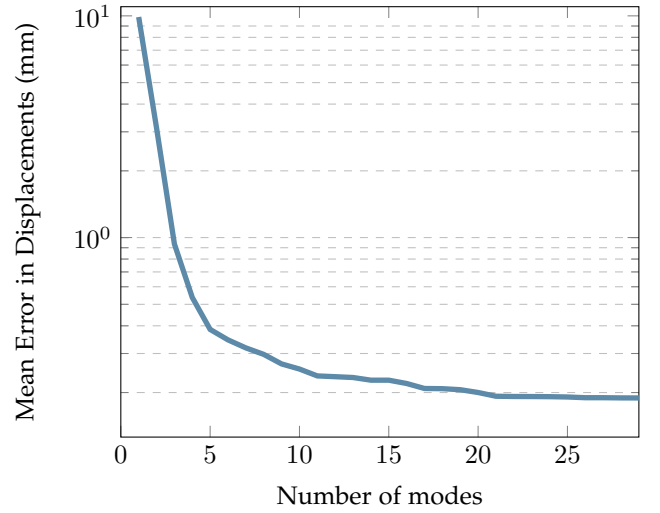


Fig. 9: Cartesian mean 3D error in displacements due to the projection of the solution in the reduced space respect to the number of modes.

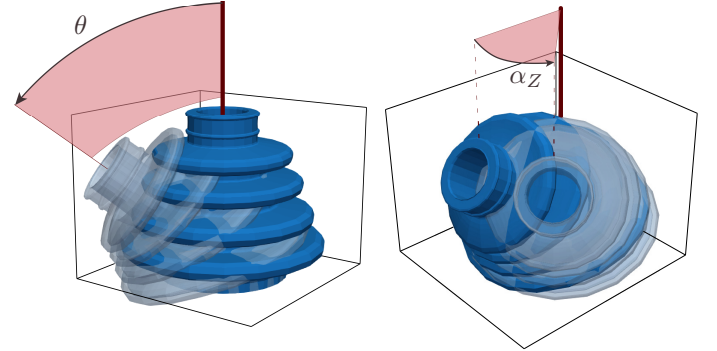


Fig. 10: Parameters to optimize in the video sequence.

Since we assume that the geometry of the deformable model is known, but not its external appearance (texture) or its location in the scene, we apply a registration process between the static point cloud and the CAD object.

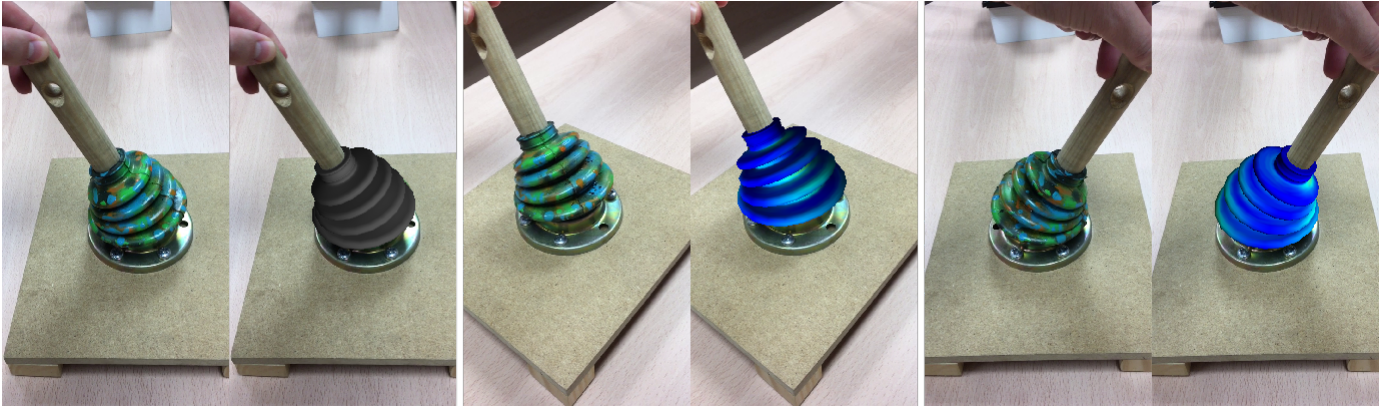


Fig. 11: Some images showing the original and the augmented frames. In the central and right images we can see the stress field that the object is suffering when the deformation appears.

The object to be deformed has geometry of revolution, so we must actually optimize the value of two parameters  $\mu = (\theta, \alpha_Z)$ , where parameter  $\theta$  measures the deformation and  $\alpha_Z$  the angle of orientation of the axis of rotation, a geometric parameter that we cannot estimate in the registration process (see Fig. 10). This is why, when the deformation occurs, we have to estimate the value of  $\theta$  in each frame. To favor the minimization process, the value of  $\alpha_Z$  is minimized throughout the entire sequence, so after a few frames observing the deformed object, the system itself is able to estimate with great degree of certainty the value of  $\alpha_Z$ , which involves also a better accuracy in the estimation of parameter  $\theta$ .

Fig. 11 shows some frames extracted from a video sequence with the original object, and the virtual object with the estimated displacements. The central and right frames show the stress field in colors, result of solving the whole mechanical problem. Finally, in Figs. 12 and 13 the 3D error is estimated. These error values have been calculated as the euclidean distances between the surface of the deformed virtual object and the object scanned with an RGB-D camera (Intel RealSense D415). It is important to note that using an accurate model of a complex geometry like the one in this example gives better results than the scanning of an RGB-D system, mainly due to the precision of the camera itself. For example, it can be seen in image 1 of Fig. 12 that the RGB-D camera is not able to accurately estimate the folds of the real object. In any case, the median error using a monocular camera is 1.33 mm (Fig. 13) and the mean error is 1.49 mm. These very accurate values are the result of using a detailed model of the geometry, an adequate description of the material law, and a measurement system as accurate as the non-deformable implementation of ORB-SLAM2.

## 7.2 Paper sheet buckling

The second example simulates the buckling phenomenon of a paper sheet. From a numerical simulation point of view, buckling is a complex problem that is usually solved using linear approximations to obtain the critical loads (buckling modes) that produce the beginning of buckling. For this, an eigenvalue analysis is used, looking for the loads for which the model stiffness matrix becomes singular [54]. However, if a complete analysis of the buckling problem is desired,

in order to know in detail the values of the stresses and deformations of the object in question, it is necessary to apply more complex methods such as the Riks method, introducing small imperfections on the geometry [54].

Again, the hypothesis of the use of a reduced model in this example is based on the fact that the number of degrees

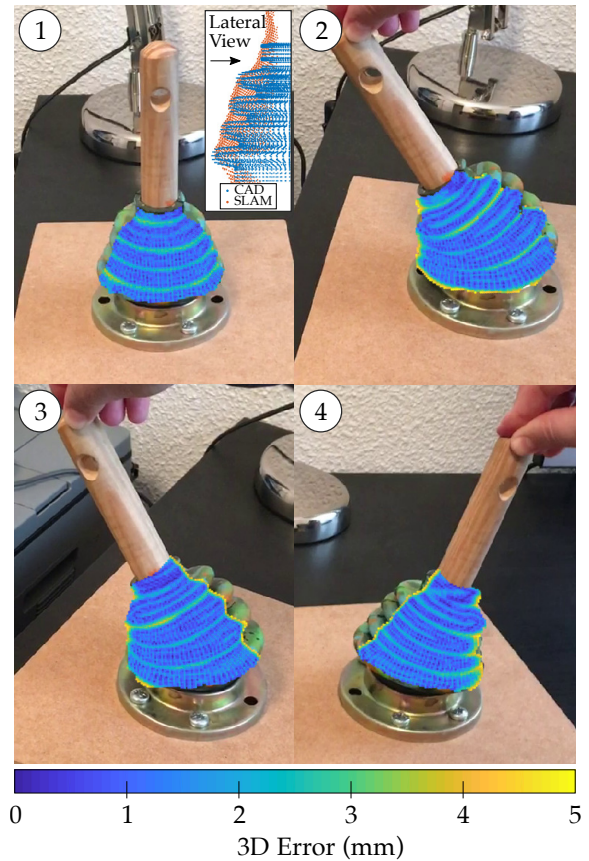


Fig. 12: Error of the monocular estimation, measured assuming the ground-truth is coming from a stereo camera. Due to the high complexity of the geometry, the stereo system can not be more accurate than the sum of model and monocular camera in some areas, producing a constant error that can be shown in the undeformed frame ①.

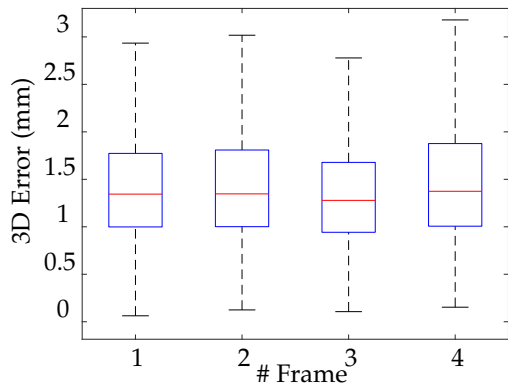


Fig. 13: Boxplot of the error distribution of all points of the external geometry, acquired in the frames of Fig. 12.

of freedom of the problem is not high. In other words, although it may appear that the object is arbitrarily deformed, there is a great restriction due to the geometry of the paper, so the number of degrees of freedom of the problem is very small, actually living in a reduced dimension manifold.

The paper sheet material has been considered anisotropic (reinforced with oriented fibers) to emulate the real behavior of the object [55], [56]. The model mesh has been discretized in 1333 nodes, 31 in the vertical direction and 43 in the horizontal direction. To model the buckling problem we use one single parameter ( $\mu_B$ ) discretized at 801 buckling positions. Applying again the sparse implementation of PGD [53] it is possible to obtain a projected solution  $\mathbf{u} = (X, Y, \mu_B)$ , reaching a compression factor of 98.36%.

The sequence was recorded with an RGB-D camera (StereoLabs Zed Mini) in order to estimate the error made by our monocular method, where Fig. 15 plots the mean error in millimeters between the model and the real object for each frame. Fig. 14 shows some frames extracted from the sequence, where colors represent the Von Mises stress field. Note that this error, that may seem big, is comparable to the best of the state-of-the-art, with one important difference: we are solving the physics of the problem, and therefore are subject to errors in the calibration of material parameters, for instance [28]. In addition, deformations in the considered sequence are much bigger than those in [57], with a maximum vertical displacement value (also corresponding to the maximum error peak) of 16.9 cm. Relative error is thus on the order of 10% at this maximal value.

The fact that both deformations and displacements (as a rigid solid) of the object are allowed makes it also necessary to estimate the deformation parameters and the pose of the object in each frame. This means that the robustness of ORB-SLAM2 in its rigid part cannot be exploited and the errors are slightly higher than those of the previous example, but our method is more general and robust to displacements. The 3D mean error of the whole sequence is 3.98 mm, measured respect to the ground truth data coming from the RGB-D camera and taking into account there is a relevant camera error of 1.2 mm (orange line in Fig. 15). This error is one order of magnitude less than the one reported in state of the art references, such as [28], for instance, that reported

errors up to 30–35 mm in the displacement prediction.

## 8 CONCLUSIONS

We propose in this paper the application of computational mechanics as a tool for estimating the deformations suffered by the objects perceived by a standard camera. This allows us to augment the video sequence with information regarding the displacements, forces and stresses suffered by the objects. To do this, we rely on the use of models to obtain a parametric solution (using finite element methods) that we use in the on-line phase. This parametric solution is stored in a compressed way to optimize both the cost of storage and its evaluation in real time.

One of the great differences of this work with respect to the bibliography is that we do not have to impose any type of spatio-temporal restriction, nor apply energy conservation laws implemented ad-hoc, as we are solving the real physics of the deformable solids. This also implies that the result is defined for the whole volume of the modeled objects, not only for the external surface, allowing us to capture the real behavior of the objects although partial occlusions appear. It also provides information about how the interior of the objects is changing, even though we are not able to capture them with the camera. It assumes a clear advantage for biomedical applications where it is necessary to know, for example, the location of the blood vessels or cancerous bodies inside the organs. Also different material properties can be applied in any point of the solid, giving different results depending on the stiffness, but assuming more expensive previous work.

As it is shown in the experiments section, our method has a great computational efficiency, is CPU-based and quite robust against noisy observations. It is not necessary to precompute sequences to extract trajectories or learning shapes, as we obtain the information directly from the models. But the solution must be obtained for a parametric set of loads, states and deformations enough to be able to work in any sequence.

Finally, this work can use any type of model order reduction method, and applied to any type of camera device, such as monocular, stereo or RGB-D systems.

## ACKNOWLEDGMENTS

We are grateful to our colleague Prof. J. M. Martínez Montiel, whose help with some technical aspects is much appreciated.

This work has been partially funded by the ESI Group through the ESI Chair at ENSAM Arts et Metiers Institute of Technology, and the project 2019-0060 “Simulated Reality” at the University of Zaragoza. We also thank the support of the Spanish Ministry of Economy and Competitiveness, grant CICYT-DPI2017-85139-C2-1-R and the Regional Government of Aragon and the European Social Fund.

## REFERENCES

- [1] A. J. Davison, “Futuremapping: The computational structure of spatial ai systems,” *arXiv preprint arXiv:1803.11288*, 2018.
- [2] —, “Novel hardware for spatial ai,” *Second International Workshop on Event-based Vision and Smart Cameras, CVPR*, 2019.

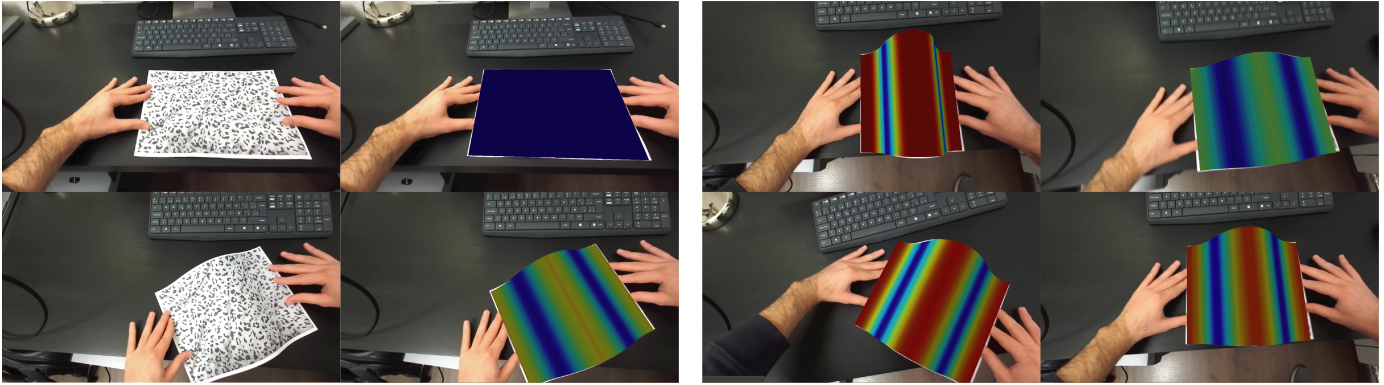


Fig. 14: Some images showing the original and the augmented frames in the second example. Colors in images represent the real-time stress field that the object is suffering when the deformation appears.

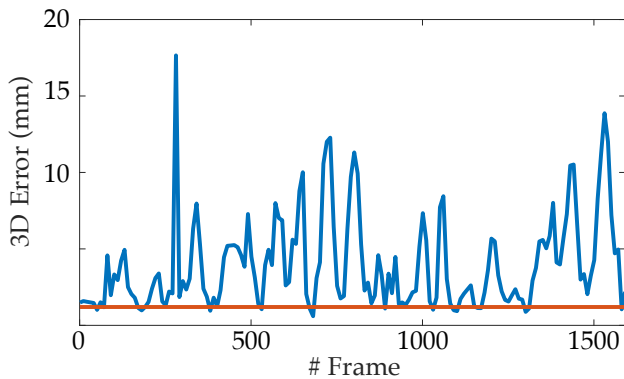


Fig. 15: Mean 3D error of our monocular method compared to ground truth (RGB-D camera) for the whole sequence. Orange line shows the camera mean error (base error) according to our experiments.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[4] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," *International conference on computer vision*, pp. 2320–2327, 2011.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 834–848, 2017.

[6] I. Stoica, D. Song, R. A. Popa, D. Patterson, M. W. Mahoney, R. Katz, A. D. Joseph, M. Jordan, J. M. Hellerstein, and J. E. Gonzalez, "A berkeley view of systems challenges for ai," *arXiv preprint arXiv:1712.05855*, 2017.

[7] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba, "Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids," *arXiv preprint arXiv:1810.01566*, 2018.

[8] M. Kennedy, K. Schmeckpeper, D. Thakur, C. Jiang, V. Kumar, and K. Daniilidis, "Autonomous precision pouring from unknown containers," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2317–2324, July 2019.

[9] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," *International workshop on vision algorithms*, pp. 298–372, 1999.

[10] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[11] L. M. Paz, P. Piniés, J. D. Tardós, and J. Neira, "Large-scale 6-dof slam with stereo-in-hand," *IEEE transactions on robotics*, vol. 24, no. 5, pp. 946–957, 2008.

[12] D. Gutierrez-Gomez, W. Mayol-Cuevas, and J. J. Guerrero, "Dense

rgb-d visual odometry using inverse depth," *Robotics and Autonomous Systems*, vol. 75, pp. 571–583, 2016.

[13] D. M. Cole and P. M. Newman, "Using laser range data for 3d slam in outdoor environments," *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 1556–1563, 2006.

[14] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," *European Conference on Computer Vision*, pp. 834–849, 2014.

[15] R. Smith, M. Self, and P. Cheeseman, *Estimating Uncertain Spatial Relationships in Robotics*. Springer, 1990.

[16] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pp. 225–234, 2007.

[17] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3d shape from image streams," *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2, pp. 690–696, 2000.

[18] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: a factorization method," *International Journal of Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

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

[20] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, "Nonrigid structure from motion in trajectory space," *Advances in neural information processing systems*, pp. 41–48, 2009.

[21] P. F. Gotardo and A. M. Martinez, "Kernel non-rigid structure from motion," *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 802–809, 2011.

[22] A. Agudo and F. Moreno-Noguer, "A scalable, efficient, and accurate solution to non-rigid structure from motion," *Computer Vision and Image Understanding*, vol. 167, pp. 121–133, 2018.

[23] L. Torresani, A. Hertzmann, and C. Bregler, "Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 5, pp. 878–892, 2008.

[24] C. Kong and S. Lucey, "Prior-less compressible structure from motion," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4123–4131, 2016.

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

[26] A. Bartoli, Y. Gérard, F. Chadebecq, T. Collins, and D. Pizarro, "Shape-from-template," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 10, pp. 2099–2118, 2015.

[27] M. Gallardo, D. Pizarro, T. Collins, and A. Bartoli, "Shape-from-template with curves," *International Journal of Computer Vision*, vol. 128, no. 1, pp. 121–165, 2020.

[28] J. Lamarca, S. Parashar, A. Bartoli, and J. M. M. Montiel, "Defslam: Tracking and mapping of deforming scenes from monocular sequences," *arXiv preprint arXiv:1908.08918*, 2019.

[29] A. Agudo, F. Moreno-Noguer, B. Calvo, and J. M. Montiel, "Sequential non-rigid structure from motion using physical priors,"

*IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 5, pp. 979–994, 2016.

- [30] S. Ilic and P. Fua, “Non-linear beam model for tracking large deformations,” *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, 2007.
- [31] D. Metaxas and D. Terzopoulos, “Constrained deformable superquadrics and nonrigid motion tracking,” *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 337–338, 1991.
- [32] T. McInerney and D. Terzopoulos, “A finite element model for 3d shape reconstruction and nonrigid motion tracking,” *1993 (4th) International Conference on Computer Vision*, pp. 518–523, 1993.
- [33] J. Bonet and R. D. Wood, *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, 2008.
- [34] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [35] V. Lepetit, F. Moreno-Noguer, and P. Fua, “Epnnp: An accurate o (n) solution to the pnp problem,” *International journal of computer vision*, vol. 81, no. 2, p. 155, 2009.
- [36] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [37] R. Mur-Artal and J. D. Tardós, “Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras,” *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.
- [38] J. Fish and T. Belytschko, *A first course in finite elements*. John Wiley, 2007.
- [39] K. Karhunen, *Über lineare Methoden in der Wahrscheinlichkeitsrechnung*. Universitat Helsinki, 1947, vol. 37.
- [40] M. M. Loeve, *Probability theory*. The University Series in Higher Mathematics, 3rd ed. Van Nostrand, Princeton, NJ, 1963.
- [41] G. Berkooz, P. Holmes, and J. L. Lumley, “The proper orthogonal decomposition in the analysis of turbulent flows,” *Annual review of fluid mechanics*, vol. 25, no. 1, pp. 539–575, 1993.
- [42] K. W. P. Benner, S. Gugercin, “A survey of projection-based model reduction methods for parametric dynamical systems,” *SIAM Review*, vol. in press, 2016.
- [43] A. Quarteroni, A. Manzoni, and F. Negri, *Reduced basis methods for partial differential equations: an introduction*. Springer, 2015, vol. 92.
- [44] F. Chinesta, P. Ladeveze, and E. Cueto, “A short review on model order reduction based on proper generalized decomposition,” *Archives of Computational Methods in Engineering*, vol. 18, pp. 395–404, 2011.
- [45] E. Cueto, D. González, and I. Alfaro, *Proper generalized decompositions: an introduction to computer implementation with Matlab*. Springer, 2016.
- [46] I. Alfaro, D. González, S. Zlotnik, P. Díez, E. Cueto, and F. Chinesta, “An error estimator for real-time simulators based on model order reduction,” *Advanced Modeling and Simulation in Engineering Sciences*, vol. 2, no. 1, p. 30, 2015.
- [47] J. P. Moitinho de Almeida, “A basis for bounding the errors of proper generalised decomposition solutions in solid mechanics,” *International Journal for Numerical Methods in Engineering*, vol. 94, no. 10, pp. 961–984, 2013. [Online]. Available: <http://dx.doi.org/10.1002/nme.4490>
- [48] A. Ammar, F. Chinesta, P. Díez, and A. Huerta, “An error estimator for separated representations of highly multidimensional models,” *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 25–28, pp. 1872 – 1880, 2010.
- [49] P. J. Besl and N. D. McKay, “Method for registration of 3-d shapes,” *Sensor Fusion IV: Control Paradigms and Data Structures*, vol. 1611, pp. 586–607, 1992.
- [50] K. Levenberg, “A method for the solution of certain non-linear problems in least squares,” *Quarterly of applied mathematics*, vol. 2, no. 2, pp. 164–168, 1944.
- [51] D. W. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.
- [52] F. Chinesta, A. Leygue, F. Bordeu, J. V. Aguado, E. Cueto, D. González, I. Alfaro, A. Ammar, and A. Huerta, “Pgd-based computational vademecum for efficient design, optimization and control,” *Archives of Computational Methods in Engineering*, vol. 20, no. 1, pp. 31–59, 2013.
- [53] R. Ibáñez, E. Abisset-Chavanne, A. Ammar, D. González, E. Cueto, A. Huerta, J. L. Duval, and F. Chinesta, “A multidimensional

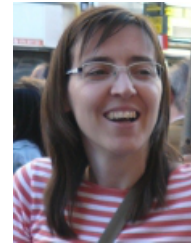
data-driven sparse identification technique: the sparse proper generalized decomposition,” *Complexity*, vol. 2018, 2018.

- [54] S. Novoselac, T. Ergić, and P. Baličević, “Linear and nonlinear buckling and post buckling analysis of a bar with the influence of imperfections,” *Tehnički vjesnik*, vol. 19, no. 3, pp. 695–701, 2012.
- [55] K. Schulgasser, “On the in-plane elastic constants of paper,” *Fibre Science and Technology*, vol. 15, no. 4, pp. 257–270, 1981.
- [56] T. Yokoyama and K. Nakai, “Evaluation of in-plane orthotropic elastic constants of paper and paperboard,” *Dimensions*, vol. 60, p. 50, 2007.
- [57] S. Parashar, D. Pizarro, and A. Bartoli, “Robust isometric non-rigidstructure-from-motion,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.



order reduction methods in applied mechanics and bioengineering.

**Alberto Badiás** is currently a postdoctoral researcher at the University of Zaragoza, Spain. He received the B.S. degree in mechanical engineering in 2011, the M.S. degree in industrial engineering (industrial automation and robotics) in 2014, the M.S. degree in biomedical engineering in 2016 and the Ph.D. degree in 2020, all from the University of Zaragoza. He has been working in the area of computer vision and robotics developing 3D reconstructions and new image processing tools, and is working currently in model



**Iciar Alfaro** is an associate professor of continuum mechanics at the university of Zaragoza. Her research has been devoted to the development of innovative numerical methods for complex problems in continuum mechanics. She developed natural element formulations for forming processes, and, more recently, she works in the development of advanced techniques for augmented reality with strong physical meaning.



**David González** is currently a professor at the university of Zaragoza. He obtained his Ph.D. degree in applied mathematics in the same university in 2004, with a thesis on the development of meshless methods for the Lagrangian simulation of free-surface flows. Since then, he has been working on the development of advanced model order reduction techniques, with an eye towards their application in real-time simulation, computational surgery and augmented reality.



**Francisco Chinesta** Francisco Chinesta is currently a professor of Computational Mechanics at the PIMM lab of ENSAM ParisTech, AIRBUS chair professor from 2008 to 2012 and ESI Chair professor since 2013. He is associate member of the University of Wales Institute of Non-Newtonian Fluid Mechanics, fellow of the “Institut Universitaire de France” (IUF) and fellow of the Spanish Royal Academy of Engineering. His main research contribution concerns the proposal and development of efficient model reduction strategies based on the Proper Generalized Decomposition and more particularly the development of separated representations based strategies. He is author of over 250 papers in peer reviewed journals and received many international scientific prizes and awards.



**Elías Cueto** is professor of continuum mechanics at the university of Zaragoza. His research is devoted to the development of advanced numerical strategies for complex phenomena. In particular, in the last years he has worked in model order reduction techniques and real-time simulation for computational surgery and augmented reality applications. His work has been recognized with the J. C. Simo award of the Spanish Society of Numerical Methods in Engineering, the European Scientific Association of

Material Forming (ESAFORM) Scientific Prize, and the O.C. Zienkiewicz prize of the European Community on Computational Methods in Applied Sciences, ECCOMAS, among others. He is fellow of the EAMBES society.