Shape Basis Interpretation for Monocular Deformable 3-D Reconstruction

Antonio Agudo and Francesc Moreno-Noguer

Abstract—In this paper, we propose a novel interpretable shape model to encode object non-rigidity. We first use the initial frames of a monocular video to recover a rest shape, used later to compute a dissimilarity measure based on a distance matrix measurement. Spectral analysis is then applied to this matrix to obtain a reduced shape basis, that in contrast to existing approaches, can be physically interpreted. In turn, these pre-computed shape bases are used to linearly span the deformation of a wide variety of objects. We introduce the low-rank basis into a sequential approach to recover both camera motion and non-rigid shape from the monocular video, by simply optimizing the weights of the linear combination using bundle adjustment. Since the number of parameters to optimize per frame is relatively small, specially when physical priors are considered, our approach is fast and can potentially run in real time. Validation is done in a wide variety of real-world objects, undergoing both inextensible and extensible deformations. Our approach achieves remarkable robustness to artifacts such as noisy and missing measurements and shows an improved performance to competing methods.

Index Terms—Deformable Shape Analysis, Dynamic Modeling, Structure from Motion, Low-Rank Representation, Optimization.

1 Introduction

Digital images and videos are nowadays present in everyone's life and they can be accessed through the Internet, mainly thanks to the rapid development of recording devices. In this context, many efforts have been done in developing systems able to perceive in three dimensions. However, building algorithms that can emulate the human 3D perception has proven to be a much harder task than initially anticipated. While some degree of success has been achieved when the object observed by the camera is rigid, inferring the 3D geometry of the vivid moving real world is still in its infancy. In these cases, the problem is still open, since including deformation priors is substantially more difficult than using simple rigidity, and retrieving deformable shape is very weakly constrained compared to retrieving rigid structure. This problem represents an active research area, and can be exploited in many application domains including multimedia, human-computer interaction, computer graphics, augmented reality and medical imaging, to name just a few.

The joint estimation of non-rigid 3D shape and pose parameters normally results in a non-convex optimization problem, and the orthogonality constraints on the pose parameters make the problem even more complicated. This problem is known as Non-Rigid Structure from Motion (NRSfM), and in the last decade many efforts have been made [?], [?], [?], [?], [?], [?], [?] which formulate a number of assumptions and exploit deformation priors that allow to retrieve the time-varying 3D configuration of deformable objects. The main difficulty to resolve the problem is due to the fact that many different 3D configurations can produce similar image observations, and hence the reprojection constraints are not sufficient to achieve a single solution. To solve this, most works use additional priors about the deformation of the object and the motion of the camera [?], [?], [?], [?], [?], but only recently, the problem has been addressed in a sequential manner [?], [?], [?], [?], [?]. In this case, only the measurements until the current frame are considered. This represents an even more complex scenario compared to the batch case because of the intrinsic strong ill-posedness of the problem. However, this scenario is paramount for bringing such algorithms to real situations and recovering live motion (such as in operating rooms, where an on-line estimation is mandatory to achieve an interaction between the 3D virtual model and the medical team) that require fast and potentially real-time solutions at frame rate. The problem becomes even more challenging when neither a deformation model nor 3D training data can be considered. In fact, obtaining adequate and large set of deformable training data could become a complex and arduous task in these scenarios.

In this work, we introduce a new shape basis interpretation to encode the deformation of time-varying shapes. To achieve this, we only need a shape at rest estimation of the non-rigid object, which can be obtained from an initial exploration by using the first few images of the monocular video. From this 3D configuration, we compute a matrix encoding the distances between every pair of points of the structure, with the purpose of obtaining a reduced shape basis through spectral decomposition. We present different alternatives that exploit the intrinsic information of the 3D shape to model the distance matrix. Once the reduced shape basis is estimated, we propose a novel method to physically interpret it in the 3D space. Later, it will be used to encompass the time-varying configuration of the object in a low-rank shape subspace in which the weight coefficients need to be recovered. Compared to competing algorithms, our method obtains the shape basis at a lower computational cost, thanks to the eigenvalue problem we

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

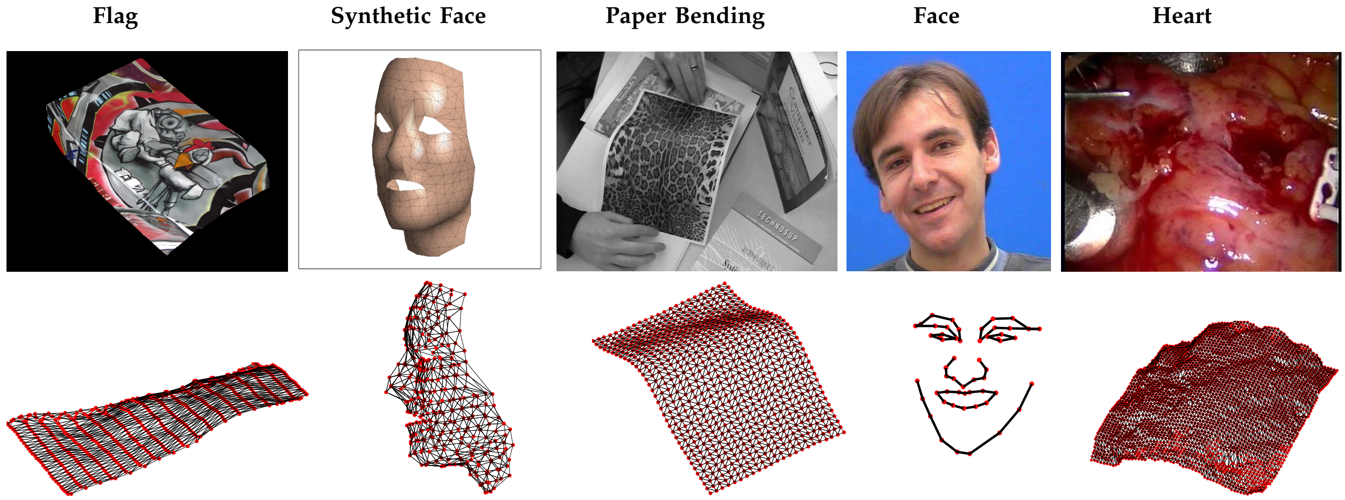


Fig. 1. **3D Reconstruction of non-rigid objects using our interpretable shape model.** We apply our method to retrieve different types of shapes, deformations, and materials; such as a flag waving in the wind, a smiling face, a sheet of paper under bending or a beating heart. **Top:** A specific image of the 2D input monocular video. **Bottom:** 3D reconstruction of the corresponding image. The reader is referred to the experimental section for more details. Best viewed in color.

solve is fairly simple. Moreover, our approach is adequate to encode both in- and extensible continuous deformations, can be applied on planar and non-planar shapes, and on a wide variety of objects and materials, without requiring any 3D training data at all (some examples are shown in Fig. **??**). As a limitation, our shape basis is not available to encode articulated motion.

Observe that although our shape basis is computed considering only the shape at rest, it has proven experimentally to be able to encode subsequent shape deformations without the need for additional 3D pre-learned data. It is also relevant that the proposed shape basis is able to describe different types of future deformations for the same rest shape, by fitting the coefficients of the linear shape subspace.

In order to obtain the 3D reconstruction from 2D motion, we incorporate the low-rank constraint into an on-line Bundle Adjustment (BA) framework. Our method is fast and may run in real time since the number of parameters per frame to optimize (i.e., the time-varying shape coefficients and the camera parameters) is relatively small. The complexity of our sequential approach is linear with the number of points, so it can handle a wide variety of scenarios, going from sparse to semi-dense or dense objects. Further, our method is robust to corrupted measurements such as missing data and noise, as it is shown in the experimental section.

The part of this work regarding the use of an interpretable shape model based on distance matrices was already presented in [?]. Here, we extend our method by proposing different alternatives to compute the distance matrix and include more theoretical discussions and comparisons with respect to competing techniques. Additional experimental results to demonstrate the wide range of scenarios where our method can be applicable are included in this version. Firstly, we present experimental results to show the suitability of our shape basis to code real-world deformations in 3D. Finally, we validate our approach to recover the 3D configuration of deformable objects from 2D data.

The remainder of this paper is organized as follows. Section II discusses the related work in this field and emphasizes our contributions to the shape basis interpretation and computation that we use to retrieve the non-rigid 3D reconstruction from a monocular video in a sequential manner. In Section III we introduce the notations and present different ways to model a distance matrix. After that, in Section IV we present the novel deformation model based on the proposed shape basis and its physical interpretation. In Section V we provide both qualitative and quantitative evaluation with respect to competing techniques and show the ability to code 3D real-world deformations by using different dissimilarity measures. This is followed in Section VI by a description of a sequential algorithm we use to jointly recover motion and time-varying shape from image streams. In Section VII we present the experimental results and provide a comparison with respect to state-of-the-art techniques. Conclusions are described in Section VIII.

2 Related Work

Reconstructing a time-varying 3D shape while estimating camera pose from solely the observation of 2D point trajectories is a severely under-constrained problem that requires additional prior knowledge. The most popular prior consists of constraining the surface to lie in a low-rank shape [?], [?], [?], [?], [?], trajectory [?], [?], [?], shape-trajectory [?], [?] or force [?] model. Firstly, low-rank shape models were proposed to encode the time-varying shape by means of a linear combination of rigid deformation modes. These models, combined with the orthonormality constraints on the camera motion, have proven successful in the 3D reconstruction of many real-world non-rigid objects. Both unknown shape basis and weight coefficients were estimated along with camera parameters by factorization-based algorithms [?], [?], or adding additional priors such as temporal and spatial smoothness by means of optimization techniques [?], [?], [?]. Later, the low-rank constraint was applied to the temporal evolution of each 3D point instead of applying it to the spatial configuration of the shape basis [?]. To this end, each 3D point evolution was independently coded by means of a linear combination of trajectory vectors based on the Discrete Cosine Transform (DCT). The problem was even further simplified in [?] where additional static points were used to independently solve for the camera motion, resulting finally in a linear problem. The compact DCT representation was also used to approximate the time-evolving shape basis coefficients in the shape-trajectory model proposed in [?], [?]. In this case, temporal smoothness was implicitly imposed on each 3D point trajectory. More recently, a low-rank force subspace was proposed in [?] to give a physical interpretation of previous subspaces, since they were linked with a physical model.

Many efforts have been made to recover the shape basis on the fly [?], [?], [?], [?], but the problem quickly becomes under-constrained when complex deformations, requiring larger rank values, need to be acquired. In general, it is not possible to assume that small rank values can represent the variation of real-world objects. This ambiguity can be reduced using a pre-defined basis in terms of shape or trajectory which acts as a representative basis while reducing the amount of parameters to be learned. For pre-defined shape basis, dimensionality reduction techniques such as Principal Component Analysis (PCA) [?], [?], [?], have been proposed in order to reduce the problem complexity and they assume a relatively large set of training data. Similarly, for 3D face reconstruction, an active appearance model could be used to obtain a 3D shape basis from trained 2D shapes [?], [?]. However, the accuracy of these techniques relies on the appropriateness of the learning data, but this information is not always available in advance, requiring alternative methods to obtain pre-defined bases. Modal Analysis (MA) was also presented to get a physics-based modal family of a known object [?], or of a rest shape which can be estimated from an initial exploration [?]. While these methods do not require training data, they rely on physical deformation models that have to be defined a priori. Other dimensionality reduction techniques include 3D warps [?] or free form deformation models [?]. Finally, since the trajectory model needs the full temporal sequence to obtain the pre-defined trajectory basis [?], this method cannot be applied to sequential estimation and we discard this subspace.

On the other hand, invariant transformations for isometric deformations were proposed applying Multi-Dimensional Scaling (MDS) on distance matrices over a known template [?], [?], [?], [?] for 3D shape recognition purposes. In this case, a new configuration is obtained enforcing the point-wise euclidean distances by means of the original point-wise geodesic ones for both 2D and 3D. Similar formulations were presented by [?] in order to encode quasi-isometric deformations for 3D face recognition. The alternative group of methods known as template-based [?], [?], [?], [?], [?] infers a deformed 3D surface from its image 2D projection and a known reference 3D shape. In [?], [?] the unknown surface was modeled as a linear combination of rigid deformation modes learned in advance from non-rigid 3D training data. The need of training data was circumvented in [?] by introducing Laplacian meshes. In order to avoid inherent ambiguities, in-extensibility constraints [?], [?], [?], [?] have been extensively used in the literature to perform non-rigid reconstruction but only for isometric deformations, thus limiting their applicability.

Despite all this tremendous advance, none of the previous approaches to NRSfM process the monocular sequence in a sequential manner. While sequential real-time SfM [?], [?], [?] solutions exist for rigid scenes, sequential estimation of non-rigid shape from a single camera remains a challenging problem. This is mainly due to the fact that most techniques remain batch and process all frames in the video at once, after video capture. Recently, this has been addressed by several sequential formulations [?], [?], [?], [?] that process the monocular video frame by frame as the data arrives. However, these methods were only demonstrated for a small number of landmarks [?], [?], [?], or relied on a known deformation model [?]. It is worth pointing out that sequential NRSfM methods are related to template-based ones since a 3D initial exploration is required for initialization. However, the estimated rest shape is normally less accurate than a 3D template. Furthermore, most template-based approaches do not compute the camera motion since they assume that the deformation modes are aligned with the camera referential or yield a solution shape for which the pose is unknown.

In this paper, we exploit the available information from an initial exploration of the dynamic shape acquired by a monocular camera, in order to estimate a pre-defined shape basis that we will use to model its deformation. Our method employs this exploration to recover a 3D rest shape that is then exploited to obtain a dissimilarity measure based on a representation of the shape. We present different alternatives to code the distance matrix employed to obtain a reduced shape basis from spectral analysis. Once the shape basis is estimated, it will be interpreted and used to code both in- and elastic 3D deformations. To show the effectiveness of our shape basis, first of all, we provide experimental validation by fitting 3D real objects. After that, we provide experimental validation to reconstruct them from 2D trajectories. Even though we also incorporate a shape at rest estimation in our formulation, in contrast to MA [?] and PCA-based formulations [?], [?], neither a deformation model nor non-rigid 3D training data are required in our case. Our model may be seen as a simplification of the standard MA, that exploits the geometric properties without assuming extra prior information to predict future deformations of the objects. This yields a reduction of the computational complexity while still being valid for a wide variety of materials and objects.

3 Preliminaries and Distance Matrices

Before proceeding and describing the problem of computing distance matrices, we define some notations about a 3D shape configuration.

3.1 Preliminaries

Let us define a 3D rest configuration of a deformable object made of p 3D points by means of the matrix $\bar{\mathbf{S}} = [\bar{\mathbf{s}}_1, \bar{\mathbf{s}}_2, \ldots, \bar{\mathbf{s}}_j, \ldots, \bar{\mathbf{s}}_p]$, where the columns contain the 3D locations for every single point $\bar{\mathbf{s}}_j \in \mathbb{R}^3$. The object is also represented through a triangular mesh, where every vertex corresponds to a 3D object point, and the list of vertexes is defined as $\mathcal{S} := \{\bar{\mathbf{s}}_j \in \mathbb{R}^3\}_{j=1}^{p}$, with the index set of $\mathcal{S}$ as $N_p := \{1, \ldots, p\}$. We also introduce edges, that represent line segments connecting two different vertexes of $\mathcal{S}$, and can be expressed by a tuple of indexes (j, h), $j, h \in N_p, j \neq h$. Let $\mathcal{E} \subset N_p \times N_p$ be the list of edges with $\mathcal{E} := \{(j,h)_e\}_{e=1}^{n}$ and n the number of edges. Eventually, we compute the m-triangular mesh by means of a Delaunay's tessellation [?], where the list of triangles is represented as $\mathcal{T} \subset N_p \times N_p \times N_p$ with $\mathcal{T} := \{(j,h,l)_t\}_{t=1}^{m}$ and $j, h, l \in N_p, j \neq h \neq l \neq j$. In the general case, Delaunay triangulation is normally computed in the last frame of the initial frames we use for initialization. For the dense case, its computation could become trivial, since a reference frame is required to compute optical flow, where a regular grid is known as every pixel correspond to a nodal point. Moreover, it is worth noting that we could take advantage of having an estimation of the 3D rest shape and applying alternative connectivity algorithms [?]. Once the triangulation $\mathcal{T}$ is available, we obtain the set of edges $\mathcal{E}$. For later computations, we also define a path between two generic points $\bar{\mathbf{s}}_j$ and $\bar{\mathbf{s}}_h$, as the sequence $\Theta(j, h) = \{\bar{\mathbf{s}}_j\}_{j=1}^{p}$, following the piecewise non-directed edges denoted by the set $\mathcal{E}$ that connect the points j and h in the set $\mathcal{S}$.

3.2 Computing Distance Matrix

We now show how to exploit an initial shape configuration $\bar{\mathbf{S}}$ –denoted as the shape at rest– in order to obtain a symmetric $p \times p$ distance matrix $\mathbf{D}$. To show the generality of our approach, we present different alternatives to encode the inherent geometric properties of a 3D shape and their corresponding distance matrices.

Euclidean Distance Matrix: We first define the Euclidean distance matrix $\mathbf{D}_E$ that includes the Euclidean distances between pairs of points on $\bar{\mathbf{S}}$ as:

$$\mathbf{D}_E = \left[\mathbf{b}\mathbf{1}_p^\top + \mathbf{1}_p\mathbf{b}^\top - 2\bar{\mathbf{S}}^\top\bar{\mathbf{S}}\right]^{\frac{1}{2}} \odot \left[\mathbf{1}_p\mathbf{1}_p^\top - \mathbf{I}_p\right], \qquad (1)$$

where $\mathbf{b} = \sum[\bar{\mathbf{S}} \odot \bar{\mathbf{S}}]^\top$ is a $p \times 1$ vector. $\mathbf{1}_p$ and $\mathbf{I}_p$ indicate a $p \times 1$ vector of ones and a $p \times p$ identity matrix, respectively. $\odot$ represents the Hadamard product, i.e., element-wise product, and $\frac{1}{2}$ indicates a element-wise square root. It is worth pointing that the second product lets us to set a null diagonal, avoiding numerical errors. This matrix is the same as a geodesic distance matrix for perfectly planar shapes, and represents a good approximation for quasi-planar objects (see Fig. **??**).

Manhattan Distance Matrix: We next define the generalized Manhattan distance matrix $\mathbf{D}_M$ for 3D irregular domains, that is made of Euclidean distances between pairs of points following the path of minimal cost from j to h by means of the Dijkstra's shortest path:

$$\mathbf{D}_M = \mathbb{D}_{n=1}^{p(p-1)/2} d_{m_n}(j, h), \qquad (2)$$

with:

$$d_m(j, h) = \min_{\Theta} \sum_{j=1}^{p-1} d_e(j, j+1), \qquad (3)$$

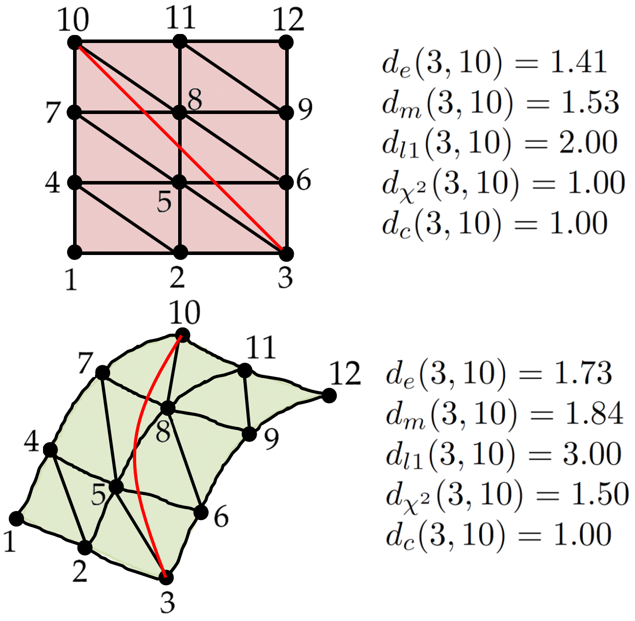


Fig. 2. **Distance comparison.** We display a unit square shape by means of 12 points and a 12-triangle regular mesh, by considering both planar (red shape) and non-planar (green shape) cases. In both cases, we represent Euclidean $d_e(j,h)$, Manhattan $d_m(j,h)$, L1 $d_{l1}(j,h)$, chi-squared $d_{\chi^2}(j,h)$ and cosine $d_c(j,h)$ distances between the points (j,h). To establish a comparison, we compute the distance between the points 3 and 10, being the geodesic distance 1.41 in both cases (see red line). For the planar case, this distance is well approximated by Euclidean distance. Manhattan distance can provide good results, but this solution depends on the nodal connectivity. For non-planar objects, this effect can be minimized when the number of points is high enough (in practice, one hundred of them). The rest of analyzed distances show an intermediate level of estimation, and they can become useful in real applications as we show later. Best viewed in color.

where $\mathbb{D}$ represents the distance assembly operator, i.e., this matrix is assembled from distances between points $d_m(j,h)$. As the matrix is symmetric with null diagonal, only $p(p-1)/2$ terms need to be considered. The estimation of this matrix depends on the nodal connectivity, even though this effect can be reduced when the number of points is not very reduced. In this case, the matrix is well suited when a small neighborhood –where the distances between points are small– is considered, such as into dense structures.

Geodesic Distance Matrix: We also define the geodesic distance matrix $\mathbf{D}_G$ over pairs of points on $\bar{\mathbf{S}}$. To this end, we use the robust and efficient approach based on heat flow, as was proposed by [?]. For further details, we refer the reader to this paper.

L1 Distance Matrix: A distance matrix can also be modeled by L1 distances. For this purpose, we represent it by $\mathbf{D}_{L1} \equiv [\mathbf{d}_{*1}, \ldots, \mathbf{d}_{*j}, \ldots, \mathbf{d}_{*p}]$, where the subindex $_{*j}$ represents the j-th column of a matrix and is computed for this case as:

$$\mathbf{d}_{*j} = \sum_{i=1}^{3} \left[\left|\bar{\mathbf{S}}^\top - \left[\mathbf{1}_p \otimes \bar{\mathbf{s}}_j^\top\right]\right|\right]_{*i}. \qquad (4)$$

where $\otimes$ denotes the Kronecker product.

Chi-squared Distance Matrix: In this case, the chi-squared distance between vectors (3D points) is used to model the distance matrix $\mathbf{D}_{\chi^2}$. Again, the j-th column of this matrix is computed as:

$$\mathbf{d}_{*j} = \sum_{i=1}^{3} \left[\frac{\left[[\mathbf{1}_p \otimes \bar{\mathbf{s}}_j^\top] - \bar{\mathbf{S}}^\top\right] \odot \left[[\mathbf{1}_p \otimes \bar{\mathbf{s}}_j^\top] - \bar{\mathbf{S}}^\top\right]}{[\mathbf{1}_p \otimes \bar{\mathbf{s}}_j^\top] + \bar{\mathbf{S}}^\top} \right]_{*i}, \quad (5)$$

where the division operator defines an element-wise operation. This matrix represents an intermediate level towards to previous matrices.

Cosine Distance Matrix: This distance is defined as the cosine of the included angle between two points, treated as vectors. The global matrix $\mathbf{D}_C$ can be obtained as:

$$\mathbf{D}_C = \mathbf{1}_p \mathbf{1}_p^\top - \left[\frac{\bar{\mathbf{S}}^\top}{\mathbf{b}^{\frac{1}{2}} \otimes \mathbf{1}_3^\top} \right] \left[\frac{\bar{\mathbf{S}}^\top}{\mathbf{b}^{\frac{1}{2}} \otimes \mathbf{1}_3^\top} \right]^\top, \quad (6)$$

where the division operator defines again an element-wise operation. This matrix uses a similar information to methods that impose conformal constraints to penalize changes in angles [?].

Laplace-Beltrami Matrix: We also define the discrete Laplace-Beltrami matrix $\mathbf{D}_{LB}$ that can be constructed from the well-known cotangent edge weight [?], on a Euclidean triangular surface. We refer the reader to this paper for further details.

Some examples of previous distances are showed in Fig. **??**. In this work, we will model the distance matrix $\mathbf{D}$ by a single matrix in $\{\mathbf{D}_E, \mathbf{D}_M, \mathbf{D}_G, \mathbf{D}_{L1}, \mathbf{D}_{\chi^2}, \mathbf{D}_C, \mathbf{D}_{LB}\}$. Note that it could also be modeled as a combination of these matrices. However, we discard this alternative as it is more computationally demanding.

4 Proposed Deformation Model

Euclidean and geodesic distance constraints were presented in [?] to recover isometric transformations on non-rigid objects over time for monocular 3D reconstruction purposes. Even though these constraints are very restrictive, they have proven to be a powerful prior solving the inherent ambiguities of both template-based [?], [?], [?] and template-free [?], [?] approaches. Despite their popularity, these constraints cannot be applied to encode elastic deformations, such as stretching and shearing. To solve this, we depart from the traditional shape-basis-based techniques and embrace a different formulation to obtain a shape basis family with a physical interpretation, without requiring neither training data nor a deformation model. We just exploit some types of dissimilarity measures based on representations of the 3D rest shape (see section **??**) to compute a shape basis that is valid to encode both in- and extensible deformations.

4.1 Shape Basis Computation

First of all, we describe how the non-rigid shape basis is computed. Following classical MDS [?], [?], a normalization and double centering is enforced to the distance matrix $\mathbf{D}$ through a centering matrix $\mathbf{C} = \mathbf{I}_p - \frac{1}{p}\mathbf{1}_p\mathbf{1}_p^\top$. We then obtain a spectral decomposition of $\bar{\mathbf{D}} \equiv -\frac{1}{2}\mathbf{CDC}$ by sorting out the following eigenvalue problem:

$$\bar{\mathbf{D}}\boldsymbol{\psi}_j = \omega_j^2 \boldsymbol{\psi}_j, \quad (7)$$

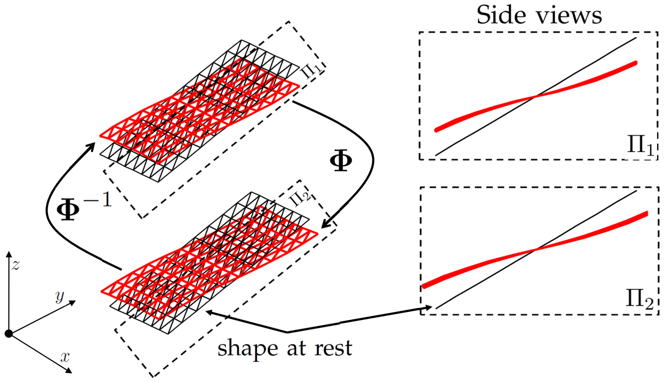


Fig. 3. **Transformation matrix $\mathbf{\Phi}$.** We display an arbitrary mode shape in red, and the corresponding shape at rest $\bar{\mathbf{S}}$ in black. Before adding this mode shape over the rest shape (top graph), we have to apply a rigid transformation $\mathbf{\Phi}$ to interpret the proposed mode shape as 3D displacements over $\bar{\mathbf{S}}$ in the global axis system. To perform a fair comparison, we also include two side views (defined by the planes Π_1 and Π_2). The mode shape is correctly interpreted as displacement in the bottom graph, as it can be observed in the plane Π_2, after applying the rigid transformation.

where $(\boldsymbol{\psi}_j, \omega_j^2)$, $j \in N_p$ are the tuple of $p \times 1$ eigenvectors and eigenvalues of $\bar{\mathbf{D}}$, respectively. The eigenvectors are normalized to enforce the orthonormality conditions $\boldsymbol{\psi}_j^\top \bar{\mathbf{D}} \boldsymbol{\psi}_h = \omega_j^2 \boldsymbol{\psi}_j^\top \boldsymbol{\psi}_h$ and $\boldsymbol{\psi}_j^\top \boldsymbol{\psi}_h = \delta_{jh}$ with δ_{jh} the Kronecker's delta, such that $\|\boldsymbol{\psi}_j\|_2 = 1$.

It is worth noting that MDS is normally applied to a distance matrix to recover new configurations where pointwise distances remain almost constant [?], [?], i.e., it is employed to encode non-extensible deformations. In contrast, in this work, we just use the normalization of previous approaches before computing the eigenvectors, obtaining a *reduced shape basis* (p-order vectors). As we show in next subsection, we propose a physical interpretation of this basis to implicitly obtain a *full shape basis* ($3p$-order vectors) that is suitable to code both in- and extensible deformations.

4.2 Shape Basis Interpretation

Coding the deformation of an object by means of a linear combination of shapes is a common practice in many fields such as computer graphics, animation and computer vision [?], [?], [?], [?], [?]. While most techniques use a full shape basis, we propose using the reduced shape basis, i.e., the eigenvectors computed in the previous subsection. To this end, the dynamic 3D displacements $\mathbf{U}$ over a rest shape are represented by a physically interpretable linear combination of r reduced modes:

$$\mathbf{U} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Upsilon}, \quad (8)$$

where $\mathbf{\Phi} \in \mathbb{R}^{3\times 3}$ is a transformation matrix to align the shape basis with the rest shape. $\mathbf{\Upsilon} \in \mathbb{R}^{r\times p}$ includes the r reduced mode shapes associated to a p-points object $\bar{\mathbf{S}}$ as:

$$\mathbf{\Upsilon} = \begin{bmatrix} \boldsymbol{\psi}_1^\top \\ \boldsymbol{\psi}_2^\top \\ \vdots \\ \boldsymbol{\psi}_r^\top \end{bmatrix} = \begin{bmatrix} \psi_{11} & \psi_{12} & \dots & \psi_{1p} \\ \psi_{21} & \psi_{22} & \dots & \psi_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \psi_{r1} & \psi_{r2} & \dots & \psi_{rp} \end{bmatrix}. \quad (9)$$

Finally, $\mathbf{\Lambda} \in \mathbb{R}^{3\times r}$ is a deformation transformation matrix that includes the time-varying coefficients in order to interpret each reduced mode shape as:

$$\mathbf{\Lambda} = \begin{bmatrix} \boldsymbol{\gamma}_{\alpha}^{\top} \\ \boldsymbol{\gamma}_{\beta}^{\top} \\ \boldsymbol{\gamma}_{\tau}^{\top} \end{bmatrix} = \begin{bmatrix} \gamma_{\alpha 1} & \gamma_{\alpha 2} & \cdots & \gamma_{\alpha r} \\ \gamma_{\beta 1} & \gamma_{\beta 2} & \cdots & \gamma_{\beta r} \\ \gamma_{\tau 1} & \gamma_{\tau 2} & \cdots & \gamma_{\tau r} \end{bmatrix}, \qquad (10)$$

where three different mode shapes –in a 3D space– per eigenvector in the reduced basis can be obtained. Recall that each $\boldsymbol{\gamma}$ component is a $r \times 1$ vector that corresponds to a different interpretation of the reduced basis.

As the principal directions in which our data varies are not usually aligned with the global axis system (see Fig. **??**), the computed eigenvectors have to be transformed before applying them to the rest shape. To do this, we first obtain a 3×3 covariance matrix $\mathbf{\Xi}$ as:

$$\mathbf{\Xi} = \left[\bar{\mathbf{S}} - \left[\bar{\mathbf{s}}_{*} \otimes \mathbf{1}_{p}^{\top}\right]\right]\left[\bar{\mathbf{S}} - \left[\bar{\mathbf{s}}_{*} \otimes \mathbf{1}_{p}^{\top}\right]\right]^{\top}, \qquad (11)$$

where $\bar{\mathbf{s}}_{*}$ is the mean values vector of all the data points in the rest shape. Once the covariance matrix is computed, we obtain the transformation matrix $\mathbf{\Phi}$ by stacking the three eigenvectors of $\mathbf{\Xi}$ together as columns. The properties of the orthogonal matrices, i.e., $\mathbf{\Phi} \equiv \mathbf{\Phi}^{-\top}$, will be considered to obtain $\mathbf{\Phi}$.

It is worth mentioning that the shape basis was also coded by p-dimensional vectors in [**?**], where a 3D-implicit low-rank shape model was proposed. However, our approach just employs a distance matrix to obtain the pre-defined reduced shape basis, using exactly the same initialization (a rest shape estimation), i.e., it is able to make the most of the available information. This means that our approach only needs to estimate the time-varying coefficients, in contrast to [**?**] that has to recover the full vectors. This yields a simplification of the problem (from trilinear to bilinear) by reducing the number of parameters to be estimated.

4.3 Spectral Analysis of the Shape Basis

In order to analyze the shape basis, i.e., the computed eigenvectors of $\bar{\mathbf{D}}$, we arrange them in a frequency spectrum from higher to lower frequency. As it can be seen in Fig. **??**(left), the eigenvectors with higher frequency dominate the global and smooth deformation since the most of deformation energy is included in these eigenvectors. This means that the largest eigenvalues of $\bar{\mathbf{D}}$ contribute the most to the variance in deformation, justifying our low-rank shape representation by means of the first eigenvectors (see Fig. **??**(right)). Consequently, in practice, solving the full eigenvalue problem in Eq. (**??**) is not required, and only the first r eigenvectors of $\bar{\mathbf{D}}$ need to be obtained, leading to a lower computational cost. The three interpretations of some mode shapes are displayed in Fig. **??**, for a rest shape corresponding to a cylinder with two holes. It is worth noting that while the competing methods need three eigenvectors to produce three mode shapes, thanks to our physical interpretation, we only need one.

4.4 Including Physical Priors

An interesting feature of our model is that we can associate the entries of the deformation transformation matrix $\mathbf{\Lambda}$ with physical behaviors. Without loss of generality, physical knowledge about the deformation of an object can be easily included to pre-define some of the entries in $\mathbf{\Lambda}$. Note that this observation is a direct consequence of the deformations we can handle in real applications. For instance, when we handle deformable objects that cannot follow bending deformations –like an elastic hair ribbon with planar forces– the entries in $\boldsymbol{\gamma}_{\tau}$ may be directly set to zero, i.e., $\mathbf{\Lambda}_{3*} = \mathbf{0}$. On the other hand, if the object cannot follow stretching deformations –like a sheet of paper or a flag waving in the wind–, the entries in $\boldsymbol{\gamma}_{\alpha}$ and $\boldsymbol{\gamma}_{\beta}$ should be directly set to zero (i.e., $\mathbf{\Lambda}_{1*} = \mathbf{\Lambda}_{2*} = \mathbf{0}$), because the object surface cannot undergo in-plane deformations. Every shape interpretation in our model could be considered as an example of deformation that can be achieved by setting to zero the rest of entries in $\mathbf{\Lambda}$ (see some examples in Fig. **??**). When no physical knowledge is known, all entries in $\mathbf{\Lambda}$ can be considered. However, we have observed that while the high-order bending mode shapes can code better shape deformation since local components are better approximated, high-order stretching modes are very restrictive and they could include artifacts and unrealistic shape deformations.

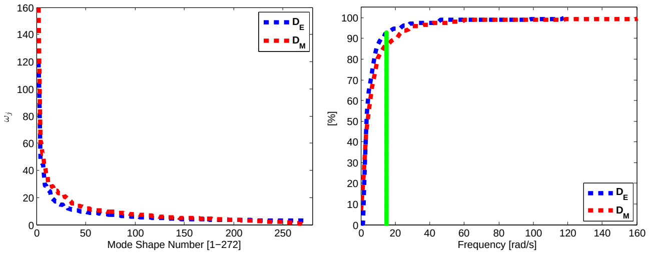


Fig. 4. **Left:** We represent the frequency-major spectrum for a synthetic cylinder with two holes, by considering Euclidean $\mathbf{D}_E$ and Manhattan $\mathbf{D}_M$ distances. Since the object consists of 272 points, we obtain a reduced shape basis of 272 eigenvectors. **Right:** Cumulative histogram of eigenvalues ω_j. For this particular object, the 93.38% and the 92.28% of the variance is modeled by considering 15 eigenvectors of $\mathbf{D}_E$ and $\mathbf{D}_M$, respectively.

5 Shape-Basis Techniques Comparison

In this section, we present a qualitative comparison against other techniques in the literature that make use of shape bases and show the ability of our proposed method to code real deformations in several types of shapes.

We first consider the type of data these approaches require to code the shape basis. Similar to MA-based techniques [**?**], [**?**], [**?**], we only require to estimate the resting shape rather than use deformable 3D training data like PCA-based methods [**?**], [**?**], [**?**]. As a positive point, our approach reduces the amount of physical prior knowledge since it does not need to know a deformation model a priori. In previous literature [**?**], [**?**], [**?**], the deformation models are normally used to compute physics-based matrices (such as stiffness and mass) where some material properties are known in advance, as it is the case of the Poisson's ratio.

On the other hand, another relevant point is to analyze the efficiency in terms of computational complexity. While our approach solves a p-order eigenvalue problem, a $3p$-order is required in PCA or MA. This means the memory requirements are much smaller in our approach, an important

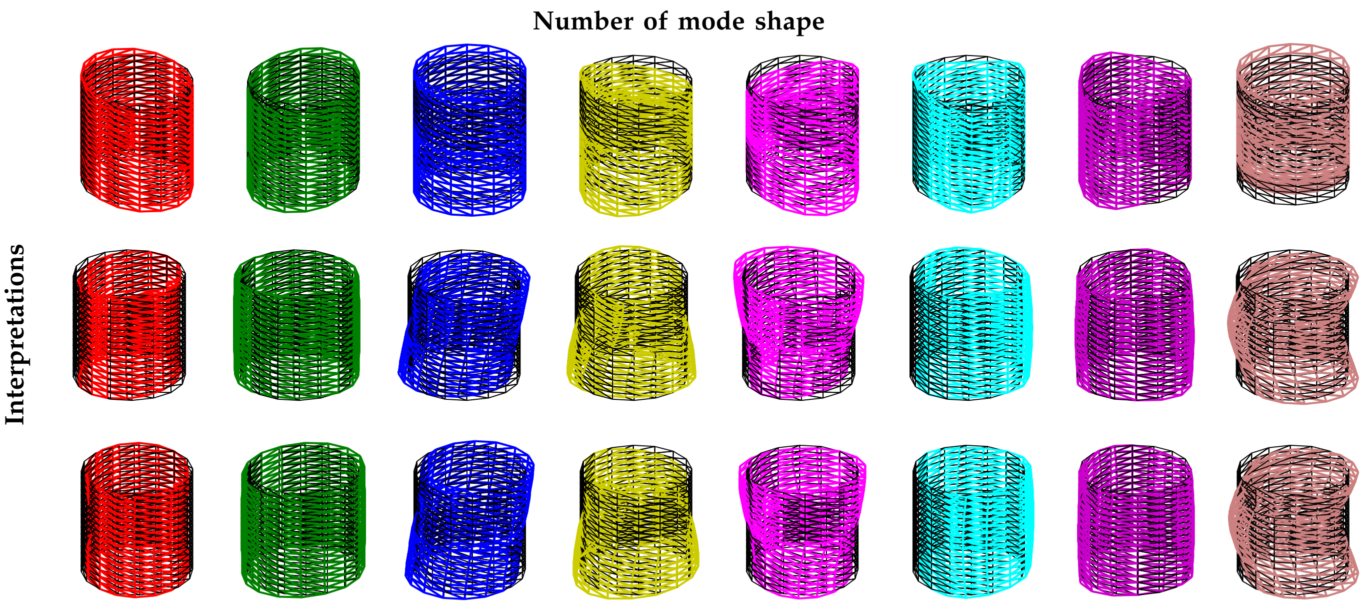


Fig. 5. **Mode-shape interpretation.** We display eight mode shapes of the proposed reduced shape basis $\boldsymbol{\Upsilon}$ using Euclidean distances over a cylinder with two holes that we consider as the rest shape (black mesh). As a result to our physical interpretation, we use each eigenvector to obtain three different interpreted 3D displacements (coded by the same color). Every column corresponds to an eigenvector with different frequency, and every row, to its specific interpretation: γ_α, γ_β, and γ_τ, respectively. For visualization purposes, we use a constant and arbitrary positive weight by adding the corresponding mode shape to the rest shape. Recall that the effect of subtracting could be obtained using the opposite weight. We represent the first eight mode shapes in the frequency-major spectrum. The figure is best viewed in color.

advantage for real applications with limited computational resources [?], and a key factor for dense reconstruction. As a result of this, our approach also reduces the computational complexity from $f(p,r) = 9p^2r$ to $f(p,r) = p^2r$ [?] when the eigenvalue problem in Eq. (**??**) is solved. Table **??** provides a qualitative comparison of our approach with respect to the most relevant state-of-the-art approaches to obtain pre-defined shape basis.

Quality / *Method*	Training	Model	Accurate	Complexity
PCA	$\mathcal{X}$	✓	✓	$(3p)^2r$
MA	✓	$\mathcal{X}$	✓	$(3p)^2r$
Ours	✓	✓	✓	p^2r

TABLE 1
Shape-basis techniques comparison. We present a qualitative comparison with respect to other methods to obtain a pre-defined shape basis, by considering learning methods such as PCA [?], [?], [?] and physics-based ones such as MA [?], [?], [?]. Complexity is represented by a function $f(p,r)$ with p and r the number of points and modes, respectively. We indicate strong (✓) and weak ($\mathcal{X}$) qualities.

5.1 Shape Basis Duality

In the literature, it is standard to represent a shape basis by means of $3p$-order vectors. They can be obtained using either PCA over a set of training data [?], [?] or exploiting a physical model over a rest configuration [?], [?]. To perform a fair comparison, we can compute an equivalent full shape basis by using our reduced representation as:

$$\boldsymbol{\Psi} = [\boldsymbol{\Upsilon} \otimes \mathbf{I}_3]^\top, \tag{12}$$

where $\boldsymbol{\Psi} \in \mathbb{R}^{3p \times 3r}$ includes the full mode shapes. As a consequence of our physical-interpretation model, we obtain $3r$ vectors from r eigenvectors. The 3D displacement can be then expressed as:

$$\mathbf{U} = \boldsymbol{\Phi}\, \mathcal{R}\left(\boldsymbol{\Psi}\boldsymbol{\zeta}\right), \tag{13}$$

where $\boldsymbol{\zeta} \in \mathbb{R}^{3r \times 1}$ contains the $3r$ weight coefficients of the linear shape subspace, and $\mathcal{R}(\cdot)$ is a permutation operator to rearrange the entries of a $3p \times 1$ vector into a $3 \times p$ matrix, where the j-th column contains the locations of the point j.

5.2 Fitting Real-World Deformations

To empirically show the suitability of our proposed shape basis to capture real-world deformations, we first use our technique to fit 3D time-varying objects. Recall that our interpretable model exclusively codes the non-rigid contribution, so any rigid contribution is included in these objects. To this end, we use three datasets with 3D ground truth acquired from the motion capture systems[1]. We denote these datasets as serviette, carton and face [?]; and they consist of 102 shapes with 63 nodal points, 53 shapes with 81 nodal points and 100 shapes with 313 nodal points, respectively.

Figure **??** represents the consistent reduction of the 3D errors as more mode shapes are included in the subspace, i.e., as more rank r is considered –a few mode shapes reduce the error by half–. Regarding our dissimilarity measures, we obtain, on average, the best 3D reconstructions using Euclidean, geodesic, and L1 distances; as well as the Laplace-Beltrami matrix. An intermediate solution is achieved by using the Manhattan distance, depending on its solution of the nodal connectivity.

To establish a fair comparison, we also include MA-based solutions; and two configurations in order to train

[1] This data was acquired with a Vicon motion capture system; and it contains one sequence of a deforming piece of cloth (*serviette* data) and one sequence of a deforming piece of cardboard (*carton* data). Both are available from: *http://cvlab.epfl.ch/data/dsr.*

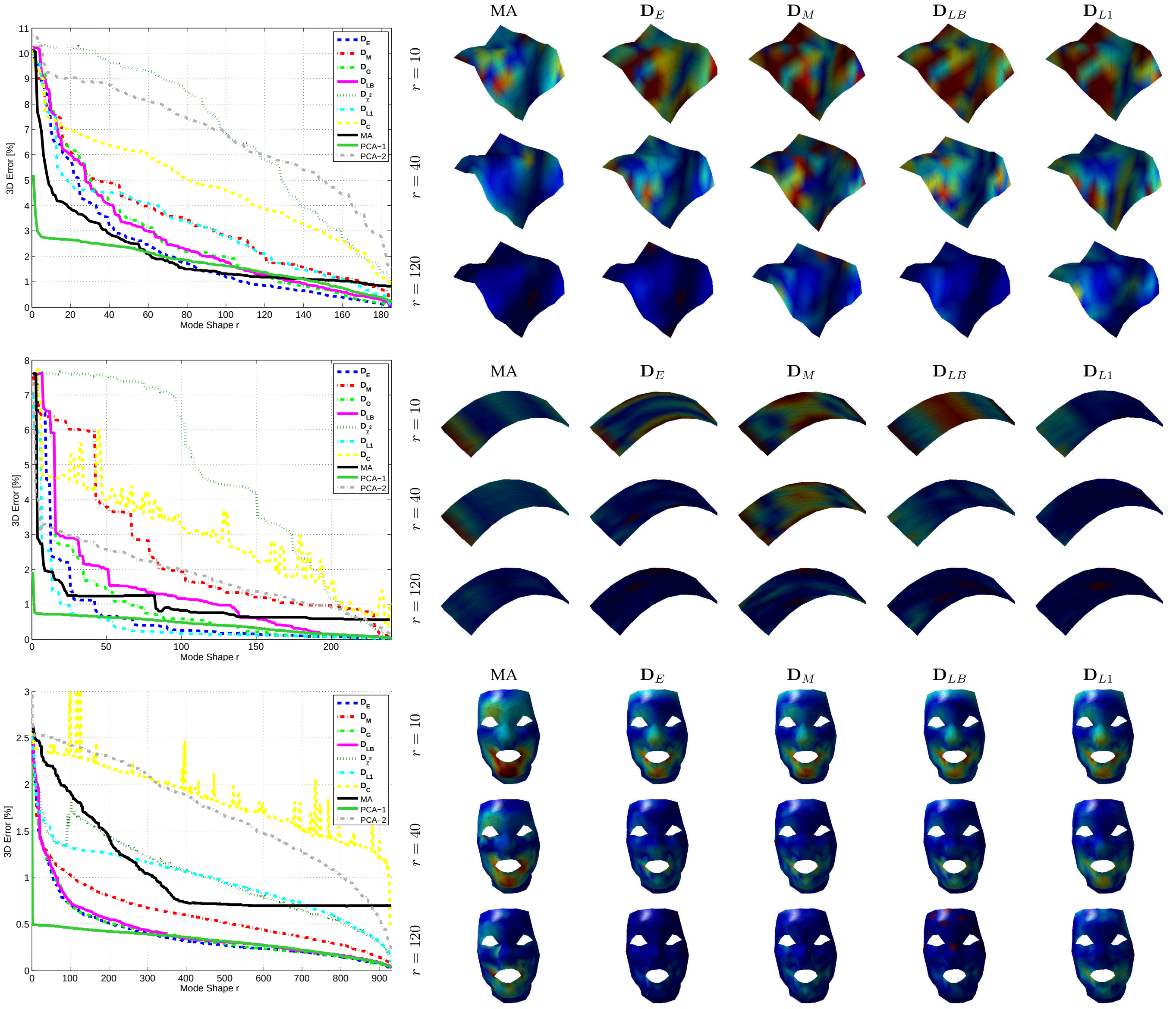


Fig. 6. **Fitting 3D real-world models: *serviette*, *carton* and *face* datasets. Left:** Evolution of the 3D error [%] as a function of the number of mode shapes r included in the shape subspace. We represent the solution of our method based on $\mathbf{D}_E$, $\mathbf{D}_M$, $\mathbf{D}_G$, $\mathbf{D}_{LB}$, $\mathbf{D}_{\chi^2}$, $\mathbf{D}_{L1}$ and $\mathbf{D}_C$ distance matrices; and the baselines MA [?], PCA-1 and PCA-2 [?]. **Right:** We represent a specific shape for some cases by using a color code, such that reddish areas indicate larger errors. For all cases, we display different values of rank $r = \{10, 40, 120\}$ in the subspace. Best viewed in color.

a PCA-based approach that we denote as PCA-1 and PCA-2. In both cases, we use the 3D ground truth data to learn the shape basis, considering the whole data in PCA-1 and the first 10 samples in PCA-2. It is worth pointing out that our method consistently outperforms PCA-2, while performing comparable to MA but without knowing the deformation model. Particularly, it is relevant our solution for the face dataset compared to the MA-based solution, that requires a much smaller number of modes to span the deformations. PCA-1 shows a well-known result, since PCA-based methods become very accurate if appropriate training data are available. However, this requirement may be hard to obtain in many real scenarios. This limitation is outperformed by our method, which in contrast just needs a rest shape estimation without assuming any other prior. In the same figure, we also display some examples by applying our dissimilarity measures.

6 Sequential NRSfM with the Proposed Shape Basis

We propose using the shape basis resulting from our interpretable algorithm to code the non-rigidity deforming scene. This section is devoted to describe the details of our sequential approach to NRSfM, i.e., to solve the inverse problem of estimating the 3D shape from 2D trajectories.

6.1 Problem Formulation

To simplify the problem, the orthographic camera model is typically used [?], [?], [?], [?], which is a good approximation when the object depth is much smaller than the distance from the camera. Let us assume a 3D shape $\mathbf{S}^f$ with p points, its 2D coordinates onto image frame f can be written as:

$$\mathbf{W}^f = \begin{bmatrix} u_1^f & u_2^f & \dots & u_p^f \\ v_1^f & v_2^f & \dots & v_p^f \end{bmatrix} = \mathbf{R}^f \mathbf{S}^f + \mathbf{T}^f, \qquad (14)$$

where $\mathbf{R}^f$ represents a truncated 2×3 rotation matrix (i.e., $\mathbf{R}^f {\mathbf{R}^f}^\top = \mathbf{I}_2$) and $\mathbf{T}^f$ stacks p copies of a 2×1 translation vector $\mathbf{t}^f$. For the special case of rigid objects, the 3D shape $\mathbf{S}$ remains constant for every frame f, i.e., $\mathbf{U}^f = \mathbf{0}$. However, our problem consists in incrementally recovering the 3D reconstruction of a time-varying object $\mathbf{S}^f$ along with the camera pose $(\mathbf{R}^f, \mathbf{t}^f)$ from 2D incomplete point trajectories $\mathbf{W}^f$ in a monocular video. The measurement matrix can be obtained by feature tracking algorithms for the sparse case, or by means of optical flow for dense correspondences. To cope with lost tracks due to outliers or occlusions, a binary vector $\mathbf{h}^f \in \{0,1\}^{p \times 1}$ is also introduced, where the non-null values indicate the presence of entries in $\mathbf{W}^f$. To this end, we define the matrix $\mathbf{M}^f = \left[\mathbf{1}_2 \otimes {\mathbf{h}^f}^\top\right]$. In the next subsection, we encode the non-rigidity of the structure over time by incorporating our proposed non-rigid model.

6.2 Interpreted Deformation Model

We now represent the non-rigid shape using a linear subspace with pre-defined mode shapes. Recovering the 3D locations of the time-varying object at every image frame f boils down to recovering the deformation matrix $\mathbf{\Lambda}^f$ in Eq. (**??**). Consequently, we only need to retrieve $3r$ coefficients for every frame to model the current configuration of the shape. However, note that we can get off this amount of coefficients including physical constraints, as it was discussed in section **??**. Finally, we express the projection in Eq. (**??**) at frame f as a function of the matrix $\mathbf{\Lambda}^f$ as:

$$\mathbf{W}^f = \mathbf{R}^f \left[\bar{\mathbf{S}} + \mathbf{\Phi}\mathbf{\Lambda}^f\mathbf{\Upsilon}\right] + \mathbf{T}^f. \qquad (15)$$

6.3 Non-linear Optimization

We now present our approach to jointly recover the camera pose and the 3D reconstruction of non-rigid objects. The outline of the algorithm is shown in Algorithm **??**. First of all, we perform an initial exploration of the object by using a few frames –a dominant rigid motion is assumed– so as to initialize and estimate the rest shape $\bar{\mathbf{S}}$ by rigid factorization [**?**], that we will use to obtain a dissimilarity measure. Note that when the initial frames include strong non-rigid motion, a bigger camera motion is required for initialization. In other cases, non-rigid factorization strategies could also be used for initialization. Once the distance matrix $\mathbf{D}$ is computed as was described in section **??**, the matrices $\mathbf{\Phi}$ and $\mathbf{\Upsilon}$ are obtained following section **??**. Accordingly, our problem is simplified to the estimation of the shape coefficients in $\mathbf{\Lambda}^i$, and the camera parameters $(\mathbf{R}^i, \mathbf{t}^i)$ per image frame. This implies the estimation of just a few parameters per image, which leads to a low computational cost method that can run in real time. To obtain a sequential estimation while the data is available –frame by frame–, we run sparse bundle adjustment on a sliding temporal window of the last $\mathcal{W}$ frames, as was done in [**?**], [**?**], [**?**]. Particularly, shape and pose parameters are recovered by minimizing a data term that penalizes deviations of the image measurements combined with smoothness terms. Considering all observed points over all frames in the corresponding temporal window, our loss function $\mathcal{A}\left(\mathbf{R}^i, \mathbf{t}^i, \mathbf{\Lambda}^i\right)$ is defined as:

$$\begin{aligned} \underset{\mathbf{R}^i, \mathbf{t}^i, \mathbf{\Lambda}^i}{\arg\min} \sum_{i=f-\mathcal{W}+1}^{f} \|\mathbf{M}^i \odot \left[\mathbf{W}^i - \mathbf{R}^i \left[\bar{\mathbf{S}} + \mathbf{\Phi}\mathbf{\Lambda}^i\mathbf{\Upsilon}\right] - \mathbf{T}^i\right]\|_{\mathcal{F}}^2 \\ + \lambda_q \sum_{i=f-\mathcal{W}+2}^{f} \|\nabla^i \mathbf{q}\|_{\mathcal{F}}^2 + \lambda_t \sum_{i=f-\mathcal{W}+2}^{f} \|\nabla^i \mathbf{t}\|_{\mathcal{F}}^2 + \lambda_\gamma \sum_{i=f-\mathcal{W}+2}^{f} \|\nabla^i \mathbf{\Lambda}\|_{\mathcal{F}}^2 \end{aligned}$$

where $\|\cdot\|_{\mathcal{F}}$ indicates the Frobenius norm. ∇^i represents the discrete temporal derivative operator. To guarantee orthonormality, we internally parameterize the rotation matrices by means of quaternions $\mathbf{R}^i(\mathbf{q}^i)$.

Algorithm 1 On-line Bundle Adjustment with a Mode-Shape-Interpretation model (BA-MSI).

```
Require: Incomplete 2D trajectories in a monocular video
Ensure: Time-varying 3D shape and camera pose
I. Initialization
  I.I Rigid Factorization
    $\mathbf{U} = \mathbf{0}$ (Eq. ??)
    $\bar{\mathbf{S}} = f(\mathbf{W})$; First few frames
  I.II Distance Matrix
    $\mathbf{D} \equiv \{\mathbf{D}_E, \mathbf{D}_M, \mathbf{D}_G, \mathbf{D}_{L1}, \mathbf{D}_{\chi^2}, \mathbf{D}_C, \mathbf{D}_{LB}\}$ (Eqs. ??-??)
  I.III Shape Basis Computation
    $\bar{\mathbf{D}} \equiv -\frac{1}{2}\mathbf{CDC}$
    $\bar{\mathbf{D}}\boldsymbol{\psi}_j = \omega_j^2 \boldsymbol{\psi}_j, \;\; 1 \le j \le r$ (Eq. ??)
    $\mathbf{\Upsilon} = [\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_r]^\top$
    $\mathbf{\Phi} = f(\bar{\mathbf{S}})$ (Eq. ??)
II. On-line Estimation
    $\mathbf{U} = \mathbf{\Phi}\mathbf{\Lambda}\mathbf{\Upsilon}$ (Eq. ??)
    $\mathcal{B} \equiv \{\mathbf{R}^i, \mathbf{t}^i, \mathbf{\Lambda}^i, \mathbf{W}^i, \mathbf{M}^i\}, \;\; f - \mathcal{W} + 1 \le i \le f - 1$
    $\left[\mathbf{R}^f, \mathbf{t}^f, \mathbf{\Lambda}^f\right] \equiv \arg\min \mathcal{A}\left(\mathbf{W}^f, \mathbf{M}^f, \bar{\mathbf{S}}, \mathbf{\Upsilon}, \mathbf{\Phi}, \mathcal{B}\right)$
    $\mathbf{S}^f = \bar{\mathbf{S}} + \mathbf{\Phi}\mathbf{\Lambda}^f\mathbf{\Upsilon}$ (Eq. ??)
```

To prevent ambiguities, first-order temporal smoothness on both camera and shape parameters are included, which influence is regulated by λ_q, λ_t, and λ_γ, respectively. In practice, these regularization weights are empirically determined and unchanged for all experiments. Recall that our formulation does not implicitly impose in-extensibility constraints, allowing us to model both in- and extensible deformations. We minimize the energy $\mathcal{A}\left(\mathbf{R}^i, \mathbf{t}^i, \mathbf{\Lambda}^i\right)$ using sparse Levenberg-Marquardt. To initialize the model parameters for a new incoming frame, we simply apply temporal smoothness, such as $\mathbf{R}^i \equiv \mathbf{R}^{i-1}$, $\mathbf{t}^i \equiv \mathbf{t}^{i-1}$ and $\mathbf{\Lambda}^i \equiv \mathbf{\Lambda}^{i-1}$.

7 Experimental Results

We now introduce our experimental evaluation on real monocular videos, providing both qualitative and quantitative results for a wide variety of objects and shapes, including planar and non-planar objects. We also compare our estimation with respect to state-of-the-art techniques based on low-rank models, when the 3D ground truth is available.

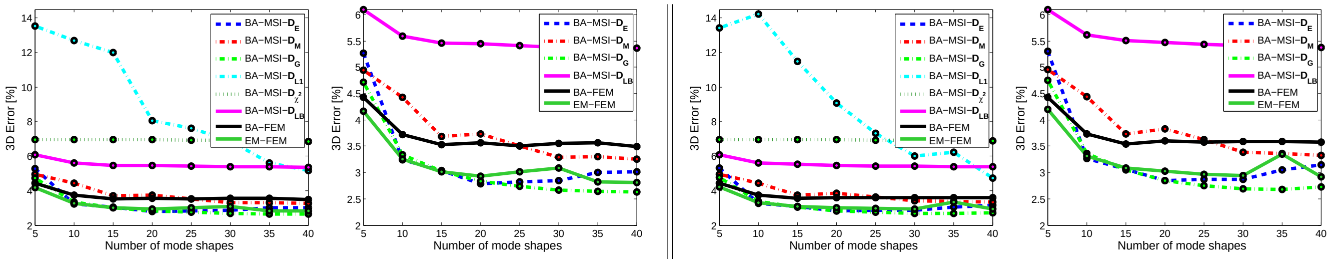


Fig. 7. **Quantitative evaluation and comparison on flag sequence.** Evolution of the 3D error $e_{3\mathcal{D}}$ as a function of the number of mode shapes r. We display our solution considering different dissimilarity measures, as well as the sequential baselines BA-FEM [?] and EM-FEM [?]. **Leftmost two columns:** Performance for noise-free measurements, and the corresponding zooming view. **Rightmost two columns:** Performance for noisy measurements, and the corresponding zooming view.

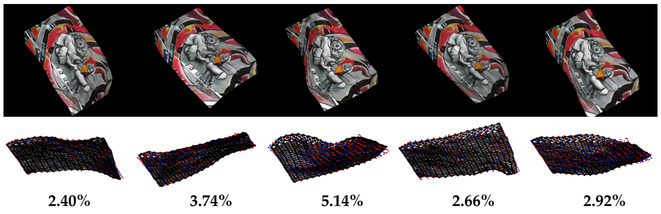


Fig. 8. **Flag sequence. Top:** Images #9, #20, #24, #35 and #50 of a flag waving in a wind. **Bottom:** 3D reconstruction from a different viewpoint considering a 40% of missing points. We represent our 3D estimation with red and blue dots for observed and unobserved points, respectively. Black circles correspond to the 3D ground truth. We also show the corresponding 3D reconstruction error $e_{3\mathcal{D}}$[%] for each shape. Best viewed in color.

To achieve this, we report the RMS error across all non-rigid images n_f, which is defined as: $e_{3\mathcal{D}} = \frac{1}{n_f}\sum_{i=1}^{n_f}\frac{\|\hat{\mathbf{S}}^i-\mathbf{S}^i_{GT}\|_{\mathcal{F}}}{\|\mathbf{S}^i_{GT}\|_{\mathcal{F}}}$, where $\hat{\mathbf{S}}^i$ and $\mathbf{S}^i_{GT}$ represents the estimated 3D reconstruction and its 3D ground truth, respectively. Videos of our experimental validation are provided in the supplemental material. For all cases, we denote our algorithms as BA-MSI-AA from bundle adjustment with our mode-shape-interpretation model, where "AA" codes the distance matrix employed. For instance, and following section **??**, when Euclidean distances are used our method is denoted as BA-MSI-DE.

7.1 Motion Capture Data

Firstly, we evaluate our approach on a 594-point sequence of a flag waving in the wind, provided by [?]. Since this deformation has little stretching, we can easily apply the physical constraints discussed in subsection **??** and set to zero the first two rows of the matrix $\mathbf{\Lambda}$. Note that this deformation was also modeled using in-extensibility constraints in [?] that restates our observation.

We process this video considering the seven dissimilarity measures that were presented in section **??**. In addition, we also compare our estimation to other sequential methods based on low-rank models: BA-FEM [?] and EM-FEM [?]. For all cases, we exactly use the same initial exploration and strategy for initialization, i.e., the rest shape we use is equal for all evaluated techniques. Furthermore, we include results adding a zero-mean Gaussian noise to every point in the object to model noisy measurements, with standard deviation $\sigma = 0.01\max_j\{|d_e(j,\kappa)|\}$, where the κ-index corresponds to the centroid of all the points.

Method / *Data*	SBA	PTA	CSF2	EM-PND	KSTA	BA-MSI-DG
Flag	7.10(38)	14.11(2)	8.80(2)	8.65	8.61(2)	**2.63(40)**

TABLE 2
Quantitative comparison on flag sequence. We provide $e_{3\mathcal{D}}[\%]$ for shape basis methods SBA [?] and EM-PND [?]; for the trajectory-based method PTA [?]; and for the shape-trajectory methods CSF2 [?] and KSTA [?]. For low-rank methods, we show the basis rank (in brackets) that yielded the lowest error. Recall that to increase the rank in the subspace, not always to produce a more accurate solution.

As shown in Fig. **??**, our methods produce a consistent reduction of the error as more mode shapes are considered. We observe that BA-MSI-DE, BA-MSI-DM and BA-MSI-DG yield better results than the rest of dissimilarity measures for this sequence. Particularly, BA-MSI-DE beats BA-MSI-DM since the rest shape is quasi-planar and the points are sparsely distributed, a situation that favors the modal shapes computed by Euclidean distances. However, even though BA-MSI-DE and BA-MSI-DG produce similar solutions, BA-MSI-DG outperforms the rest of the evaluated dissimilarity measures, showing its superiority to capture the inherent geometric properties of the 3D shape. BA-MSI-DC is not included in this experiment since the results are not accurate enough. Our BA-MSI-DG algorithm consistently outperforms BA-FEM [?] and EM-FEM [?] for both noise-free and noisy measurements, with the additional advantage of not requiring a deformation model. Since both BA-FEM [?] and EM-FEM [?] use the same shape basis, we attribute this deviation to the optimization framework which may be also combined with our basis, producing more accurate solutions.

We now present a quantitative comparison with state-of-the-art methods that learn the low-rank shape subspace [?], [?], that use a pre-defined trajectory basis [?]; or the shape-trajectory alternatives CSF2 [?] and KSTA [?]. The parameters of these methods were set in accordance to their original papers. A summary of these results are provided in table **??**. It can be seen that our approach consistently outperforms the other batch baselines [?], [?], [?], [?], even being sequential. Most of the distance matrices provide more accurate solutions compared to previous methods.

Finally, we also test our method with respect to random missing data, annotating the 40% of the points as missing. Our method is quite robust, with a 3D error of $e_{3\mathcal{D}} = 2.92\%$ when 20 shapes are used. In fact, our 3D reconstruction does not significantly degrade until a breaking point around 80% of missing data in the measurement matrix. Some instances

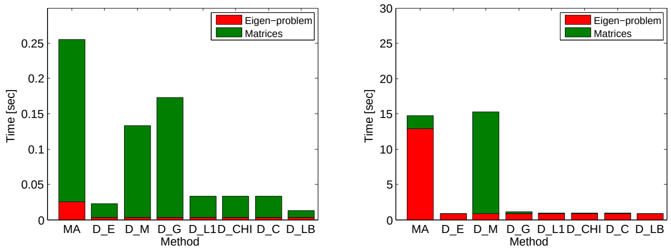


Fig. 9. **Run-time comparison.** We show run-time to compute the shape basis for our methods based on $\mathbf{D}_E$, $\mathbf{D}_M$, $\mathbf{D}_G$, $\mathbf{D}_{L1}$, $\mathbf{D}_{\chi^2}$, $\mathbf{D}_C$ and $\mathbf{D}_{LB}$ matrices, respectively. We also include MA-based techniques, such as was used in [?], [?]. For each case, we display the computational cost to compute the matrices (in green) and to resolve the eigenvalue problem (in red). **Left:** Talking face sequence of 56 points. **Right:** Flag sequence of 594 points.

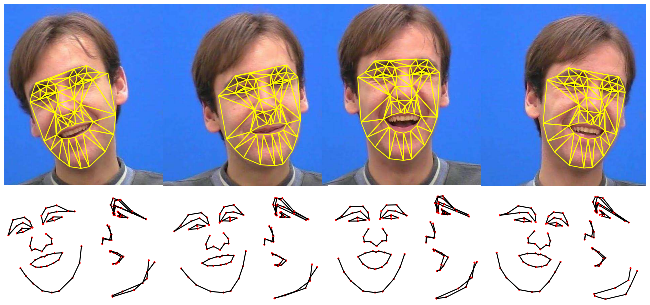


Fig. 10. **Talking face sequence. Top:** Images #51, #70, #142, and #170 of a smiling face with reconstructed mesh. **Bottom:** Original viewpoint and side views of our 3D reconstruction.

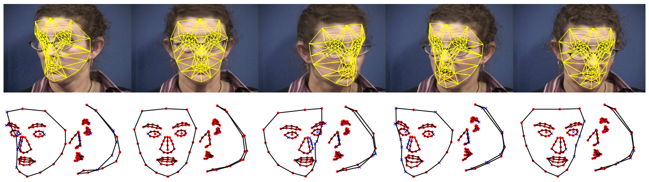


Fig. 11. **American sign language sequence. Top:** Images #41, #54, #69, #79 and #105 of gesturing face with reconstructed mesh. **Bottom:** Original viewpoint and side views of our 3D reconstruction with red dots and blue squares for observed and unobserved points, respectively. Best viewed in color.

of our 3D reconstruction and the corresponding input image are showed in Fig. **??** for this case.

Regarding computational cost, we analyze the run-time using non-optimized Matlab code to compute the shape basis, showing the matrices-computation complexity and the solution of the eigenvalue problem. For our approaches, the complexity is defined by the distance matrices, and for MA-based methods by the stiffness/mass matrices. Figure **??**(right) summarizes these results for two sequences with 56 and 594 points, respectively. While the computational complexity to obtain the physical matrices can be approximated by $\mathcal{O}(p)$ (plus the cost of some fixed operations), where p represents the number of points, solving the eigenvalue problem has a computational complexity of at most of $\mathcal{O}(p^3)$. This means the fixed operations domain the computational cost whether p is low, but become negligible for large values of p. Additionally, it can be seen that our methods have significantly lower computational cost than MA-based methods to solve the eigenvalue problem. Yet, while the time for computing the distance matrices is almost negligible, the computation of the Manhattan matrix can become more expensive when the number of points increases. This may be reduced using an optimized Dijkstra's algorithm in order to compute the corresponding distance matrix. In any event, comparing to existing approaches and $\mathbf{D}_M$, the reduction on complexity using the matrices $\mathbf{D}_E$, $\mathbf{D}_G$, $\mathbf{D}_{L1}$, $\mathbf{D}_{\chi^2}$, $\mathbf{D}_C$ and $\mathbf{D}_{LB}$ is remarkable. Note that an optimized implementation to sort out the eigenvalue problem may result in similar efficiency boosts for every algorithm, including MA-based algorithms with the corresponding scale factor.

7.2 Real Monocular Video

In this section, we qualitatively evaluate our approach on several real-world sequences, going from inextensible, quasi-inextensible and extensible deformations.

We first test a 249-frame real video where a man simultaneously talks and moves his head while engaged in conversation. The sequence has been tracked with an active appearance model using a 56 point model. With the purpose of preventing pure-bending deformations, we use our BA-MSI-DM method with physical constraints. Since this scenario uses few points, similar results could be obtained by using BA-MSI-DE. Figure **??** shows the reprojection of the deforming 3D mesh into the image plane and the corresponding 3D reconstruction for several views when $r = 30$ mode shapes are used. For this experiment, we also show the run-time to compute the shape basis in Fig. **??**(left), observing how our methods have significantly lower computational cost than competing methods.

We also process a 115-frame real video where a woman moves her head while talks and hand gesturing. In this case, we use the incomplete 77 feature annotations provided by [?], with a 17.4% of structured missing tracks. In Fig. **??** we display our estimation by using our BA-MSI-DM method with $r = 30$ mode shapes. Recall that our method can handle the structured occlusions on the fly, in contrast to other state-of-the-art approaches [?], [?], [?] which cannot handle these artifacts.

To evaluate our approach on human motion, we process a back sequence, which consists of 150 frames and 250 feature points [?] where a human back is deforming sideways and flexing. In this case, we use our BA-MSI-DG method with $r = 30$ mode shapes (similar solutions are obtained by using Euclidean distances), showing some examples of our 3D reconstruction in Fig. **??**. Despite the very fast deformations, our approach can estimate easily the time-varying 3D reconstruction.

We next process a 100-frame real video where a sheet of paper is deformed under bending, relying on the semi-dense 828-point tracks from [?]. Again, this type of material cannot undergo extensible deformations and physical constraints on the deformation matrix $\mathbf{\Lambda}$ are imposed. In this case, we use our method BA-MSI-DE considering a shape basis with $r = 30$ mode shapes. Our qualitative results are shown in Fig. **??**, including the 2D reprojection of the deforming mesh into the image plane and the corresponding 3D reconstruction re-texturing the paper surface with a logo. It is worth noting that the augmentation is performed in a sequential

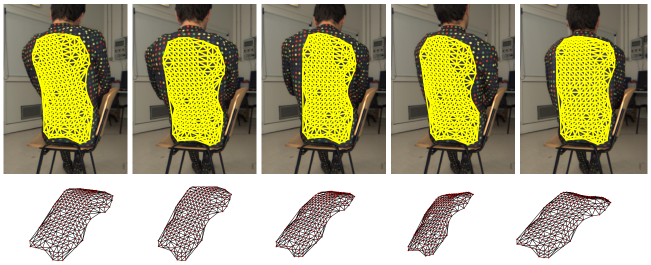


Fig. 12. **Back sequence. Top:** Images #31, #52, #85, #105 and #139 of a human back deformation with reconstructed mesh. **Bottom:** General viewpoint of our 3D reconstruction with red dots and back mesh. Best viewed in color.

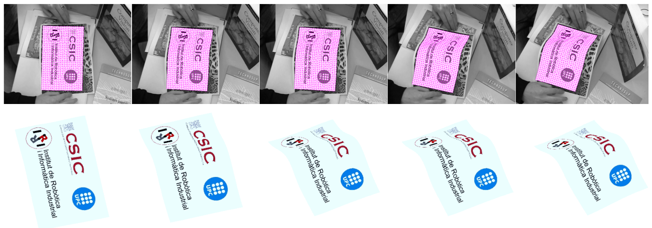


Fig. 13. **Paper bending sequence. Top:** Images #20, #40, #60, #80 and #100 of a piece of paper under bending deformations with reconstructed mesh. Notice how the 3D mesh is correctly projected and bent into the image. We also show our automatic re-texturing of the paper sequence that is sequentially executed. **Bottom:** General view of the textured 3D reconstruction seen from a different viewpoint. Best viewed in color.

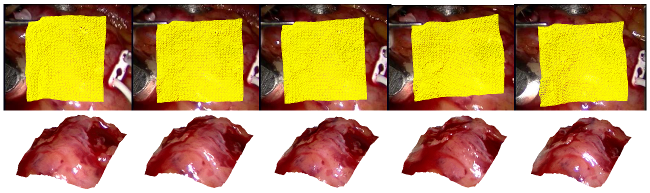


Fig. 14. **Heart sequence. Top:** Images #12, #36, #49, #66 and #74 of a beating heart with reconstructed mesh. **Bottom:** Textured rendering of the recovered 3D reconstruction from a different viewpoint. In spite of the very small camera motion, our approach can retrieve accurately the rhythmic deformations of the heart.

fashion, upon the arrival of new frames.

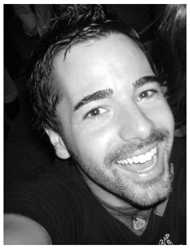
Finally, we test a challenging 79-frame real video where a beating heart is captured during bypass surgery. We track 3024 points using [?]. By processing this sequence, we show the generality of our method to handle extensible objects. In this case, since obtaining a priori knowledge of the type of deformation may become very difficult, we do not impose physical constraints. As the number of points is high enough, we use our BA-MSI-DM method to optimize the model parameters with $3r(r = 10)$ weight coefficients on the linear subspace. Our semi-dense 3D reconstruction is displayed in Fig. ??.

8 Conclusions

In this paper, we have proposed a new shape basis interpretation to model both in- and extensible deformations of time-varying objects. To this end, we have exploited the distance information of a rest shape estimated from initial frames on the video, presenting several alternatives to code it. The dissimilarity measure of the 3D configuration is then used to compute a reduced shape basis at low computational cost by means of spectral analysis and without assuming any additional model or training data. Thanks to our 3D physical interpretation, we obtain a shape basis that is used as a low-rank constraint and that in combination with simple regularization priors, it provides an effective and efficient solution to sequentially retrieve non-rigid 3D shape and motion from monocular video. Our claims have been experimentally validated on challenging real-world deformations for a wide variety of objects and materials, showing accurate results obtained on the fly. Regarding the real-time capability, our method is fast and scalable and we consider that it could be a suitable groundwork for augmented-reality applications in real time. Further exploring this is part of our future work, as well as adapting our formulation to handle articulated motion. Other fields, such as computer graphics animation or medical imaging, could also benefit from this approach by modeling dynamic objects as well as transferring real deformations to virtual ones.

Acknowledgments

This work is supported in part by a Google Faculty Research Award, by the MINECO projects HuMoUR TIN2017-90086-R and María de Maeztu Seal of Excellence MDM-2016-0656.



Antonio Agudo received the M.Sc. degree in industrial engineering and electronics in 2010, M.Sc. degree in computer science in 2011, and the Ph.D. degree in computer vision and robotics in 2015, from University of Zaragoza. He was a visiting student at vision group of Queen Mary University of London in 2013 and with the vision and imaging science group of University College London in 2014. He was also a visiting fellow at Harvard University in 2015. After two years as a postdoctoral fellow at Institut de Robòtica i Informàtica Industrial, CSIC-UPC, in Barcelona, he joined as an associate researcher of the Spanish Scientific Research Council in 2017. His research interests include non-rigid structure from motion, machine learning, and deformation analysis to medical and robotics applications.

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