PIECEWISE BÉZIER SPACE: RECOVERING 3D DYNAMIC MOTION FROM VIDEO

Antonio Agudo

Institut de Robòtica i Informàtica Industrial, CSIC-UPC, 08028, Barcelona, Spain

ABSTRACT

In this paper we address the problem of jointly retrieving a 3D dynamic shape, camera motion, and deformation grouping from partial 2D point trajectories in a monocular video. To this end, we introduce a union of piecewise Bézier subspaces with enforcing continuities to model 3D motion. We show that formulating the problem in terms of piecewise curves, allows for a better physical interpretation of the resulting priors and a more accurate representation of the motion. An energy-based formulation is presented to solve the problem in an unsupervised, unified, accurate and efficient manner, by means of the use of augmented Lagrange multipliers. We thoroughly validate the approach on a wide variety of human video sequences, including those cases with noisy and missing observations, and providing more accurate joint estimations than state-of-the-art approaches.

***Index Terms*—** Non-Rigid Structure from Motion, Piecewise Bézier Space, Grouping, Optimization.

1. INTRODUCTION

The problem of Non-Rigid Structure from Motion (NRSfM) involves jointly retrieving 3D dynamic shape, and camera motion from partial 2D point trajectories in a RGB video. This represents a challenging problem in the community that nonetheless has a wide variety of real-world applications. Unfortunately, solving this problem without 3D supervision is an ill-posed problem that requires to explore the art of priors, being them more sophisticated than those utilized in the rigid case. Maybe, the most popular priors are based on low-rank subspaces constraining the solution space of either the entire shape [1, 2, 3, 4], the 3D point trajectories [5, 6], shape-trajectory combinations [7, 8, 9, 10] or the force patterns that induce the deformations [11]. Similar shape [12] and trajectory [13] subspaces have also been exploited in a deep learning context, where large amounts of training data are needed to learn the model. While shape and force bases need to be estimated from data, trajectory ones can be predefined a priori and are available for a wide range of deformations. Although several representations could be used, the discrete cosine transform is usually chosen to model the basis [5, 6, 13]. However, this representation is global and cannot be adapted properly for those scenarios with piece-varying motion. Piecewise methods [14, 15] based on shape basis showed better performance than their global counterparts, but this was never exploited in a trajectory space.

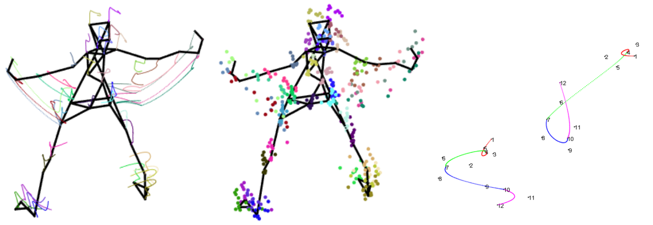

Fig. 1. **Human motion recovery by means of piecewise Bézier curves**. **Left:** Each color line represents a point trajectory detected by our algorithm. **Middle:** Control points estimated by our algorithm (the same color is used to represent every point set). **Right:** Two examples of our 3D trajectory estimation together with the corresponding spatial control-point distribution ($K = 12$). Every color represents a different piece. Best viewed in color.

More recently, the low-rank constraint has been imposed by directly minimizing the rank of a matrix representing the dynamic 3D shape, considering the data lie in a single [16, 17], in a union of temporal [18, 19], or in a dual union of spatio-temporal [20, 21] subspaces. Thanks to those representations, some segmentations into motion activities or spatial bodies were included in the estimation. In order to improve robustness, on top of those shape models additional spatial [3] and temporal [4, 22, 23] smoothness priors have also been considered. In this paper, we depart from previous work in that our approach exploits a union of piecewise Bézier subspaces with enforcing continuities to model 3D point trajectories that are observed in a RGB video. Our piecewise basis is predefined, and it is parametrized by a set of control points per point coordinate (some examples of full trajectories, control points and pieces are displayed in Fig. 1). With these ingredients, we present a novel unsupervised, unified, accurate and efficient approach that does not need any training data at all, and it outperforms state-of-the-art solutions. To the best of our knowledge, this is the first time a union of piecewise Bézier subspaces is introduced. Our approach can handle missing and noisy observations, from sparse to dense configurations, as well as articulated and continuous motions.

This work has been partially supported by the Spanish Ministry of Science and Innovation under project HuMoUR TIN2017-90086-R, by the ERA-Net Chistera project IPALM PCI2019-103386, and the María de Maeztu Seal of Excellence to IRI MDM-2016-0656.

2. PIECEWISE BÉZIER CURVES

A Bézier curve is a parametric curve defined by a set of control points that normally do not lie in the curve, excepting the first and the last. They have been widely used in many image-processing applications as image segmentation and compression [24]; vector fonts, camera paths and object modeling in the context of computer graphics; and in mechanical engineering for computer-aided design [25], just to name a few.

The Bézier curve $\mathbf{p}(s)$ can be defined by means of a linear combination of K Bernstein basis polynomials of degree $K-1$ restricted to the continuous interval $s = [0, \dots, 1]$ as:

$$\mathbf{p}(s) = \sum_{k=1}^{K} \binom{K-1}{k-1} s^{k-1}(1-s)^{K-k}\mathbf{c}_k = \sum_{k=1}^{K} b_k(s)\mathbf{c}_k \,, \quad (1)$$

where $\begin{pmatrix} K-1 \\ k-1 \end{pmatrix}$ is a binomial coefficient, and $b_k(s)$ and $\mathbf{c}_k$ represent the k-th Bernstein basis and control point, respectively. The control points, for any d-dimensional space, can be applied independently for each dimension, being they defined in a 3D space directly by its coordinates as $\mathbf{c}_k = [c_{xk}, c_{yk}, c_{zk}]^\top$.

While the curve in Eq. (1) is a good approximation for a wide variety of motions, high-degree (*i.e.*, for $K > 4$) Bernstein polynomials do not approximate very well in practice, providing over-smoothing curves. More specifically, when a high number of control points are used to define a path, the global approximation is very hard to be controlled. To solve this limitation, we can instead consider several low-order Bézier curves to code the global curve, by using quadratic and/or cubic ones, obtaining a piecewise representation. As it can be seen in table 1, the number of pieces (P) to be used depends on the number of control points. This type of approximation guarantees directly a C^0 continuity, since a unique control point acts as the last and the first in two consecutive pieces, respectively. Additionally, C^1 and C^2 continuities could be also imposed in order to share the same parametric first and second derivatives, respectively, by enforcing some geometrical constraints over the control points. These constraints would be applied every time a change of piece occurs, *i.e.*, T times, according to table 1. Particularly, to enforce C^1 and C^2 continuities between consecutive pieces we use first and second derivatives of the Bézier curves as $\mathbf{p}'_t(s)|_{s=1} = \mathbf{p}'_{t+1}(s)|_{s=0}$ and $\mathbf{p}''_t(s)|_{s=1} = \mathbf{p}''_{t+1}(s)|_{s=0}$, respectively, with $t = [1, \dots, T]$. It is worth noting that this can be only imposed for $K > 4$, as it can be seen in table 1.

In the next section, we introduce piecewise Bézier curves to model 3D motion, fitting it from 2D point trajectories in a monocular video.

K	P	T	Type	K	P	T	Type
3	1	0	1 qua.	10	3	2	3 cub.
4	1	0	1 cub.	11	4	3	2 qua. + 2 cub.
5	2	1	2 qua.	12	4	3	1 qua. + 3 cub.
6	2	1	1 qua. + 1 cub.	13	4	3	4 cub.
7	2	1	2 cub.	14	5	4	2 qua. + 3 cub.
8	3	2	2 qua. + 1 cub.	15	5	4	1 qua. + 4 cub.
9	3	2	1 qua. + 2 cub.	16	5	4	5 cub.

Table 1. Piecewise approximation by means of low-order Bézier curves. The table shows how a generic curve can be approximated by means of several Bézier curves as a function of the number of control points (K), the number of pieces (P) and transitions (T) to be used, and the type of polynomials. We consider both quadratic (qua.) and cubic (cub.) functions.

3. NON-RIGID STRUCTURE FROM MOTION

Let us consider a set of N 3D points observed along F images represented by $\mathbf{s}_n^f = [x_n^f, y_n^f, z_n^f]^\top$ for the n-th point at frame f. Considering that the point is observed by an orthographic camera, its 2D projection in the f-th image plane can be denoted as $\mathbf{w}_n^f = [u_n^f, v_n^f]^\top$. After acquiring all points in all pictures, and removing zero-mean measurements, *i.e.*, the 2D translations, the 3D-to-2D projection can be defined as:

$$\underbrace{\begin{bmatrix} \mathbf{w}_1^1 & \dots & \mathbf{w}_N^1 \\ \vdots & \ddots & \vdots \\ \mathbf{w}_1^F & \dots & \mathbf{w}_N^F \end{bmatrix}}_{\mathbf{W}} = \underbrace{\begin{bmatrix} \mathbf{R}^1 & & \\ & \ddots & \\ & & \mathbf{R}^F \end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix} \mathbf{s}_1^1 & \dots & \mathbf{s}_N^1 \\ \vdots & \ddots & \vdots \\ \mathbf{s}_1^F & \dots & \mathbf{s}_N^F \end{bmatrix}}_{\mathbf{S}},$$

where $\mathbf{W}$ is a $2F \times N$ measurement matrix to collect the 2D point trajectories, $\mathbf{G}$ is a $2F \times 3F$ block diagonal matrix, made of the F truncated 2×3 camera rotations $\mathbf{R}^f$, and $\mathbf{S}$ is a $3F \times N$ shape matrix with the 3D positions of the corresponding points. The NRSfM problem consists in factoring the measurement matrix $\mathbf{W}$ into the motion $\mathbf{G}$ and shape $\mathbf{S}$ factors, *i.e.*, inferring camera pose and non-rigid 3D reconstruction from 2D point trajectories in a monocular video.

Trajectory-based models [5, 13, 26, 27] approximate the position of every point coordinate over time by a linear combination of K low-frequency basis vectors. This representation is global and cannot be adapted properly for those scenarios with piece-varying motion. In other words, a specific point coordinate could have several local motions along time. To sort out this limitation, we introduce piecewise Bézier curves to approximate the evolution of every point coordinate over time. To this end, we define F K-dimensional vectors as $\mathbf{b}^f(s) = [b_1(s), \dots, b_K(s)]^\top$, where the entries $b_k(s)$ represent the k-th Bernstein basis at instant s. To find s, and considering the number of control points (and therefore, the number of pieces), we assign the corresponding number of frames to every piece, and assign a direct correspondence between those frames and the internal $[0, \dots, 1]$, since all pieces are modeled in the same interval. Moreover, we define N $3K$-dimensional vectors as $\boldsymbol{\kappa}^n = [c_{x1}^n, \dots, c_{xK}^n, c_{y1}^n, \dots, c_{yK}^n, c_{z1}^n, \dots, c_{zK}^n]^\top$, to collect the K control-point locations associated with the n-th point. Thanks to this representation, the 3D dynamic shape can then be written as $\mathbf{S} = \mathbf{BC}$, where $\mathbf{B} \in \mathbb{R}^{3F \times 3K}$ is a known matrix with the predefined piecewise trajectory basis, and $\mathbf{C} \in \mathbb{R}^{3K \times N}$ is a matrix of unknown control-point coordinates that are defined as:

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_3 \otimes (\mathbf{b}^1(s))^\top \\ \vdots \\ \mathbf{I}_3 \otimes (\mathbf{b}^F(s))^\top \end{bmatrix}, \;\; \mathbf{C} = \begin{bmatrix} \boldsymbol{\kappa}^1 & \dots & \boldsymbol{\kappa}^N \end{bmatrix}, \qquad (2)$$

where $\mathbf{I}$ is a 3×3 identity matrix, and $\otimes$ represents the Kronecker product. A direct duality can be found between classical trajectory-based formulations [5, 26, 27] and our approach, since the global basis is now represented in pieces, and the trajectory weights correspond to control-point coordinates, *i.e.*, they have a direct physical meaning.

The expression $\mathbf{S} = \mathbf{BC}$ guarantees the C^0 continuity of the piecewise curve for every point coordinate but not the continuities C^1 and C^2. To achieve that, we define the matrices $\{\mathbf{M}, \mathbf{N}\} \in \mathbb{R}^{vT \times K}$, with all but two elements in each row equal to 0 (see constraints in section 2). The parameter $v = \{1, 2\}$ is to impose just a C^1 continuity or both, respectively. The constraints can be applied on the control-point locations as $(\mathbf{I}_3 \otimes \mathbf{M})\mathbf{C} = (\mathbf{I}_3 \otimes \mathbf{N})\mathbf{C}$.

Finally, we incorporate the deformation-grouping quality to our formulation by assuming that the 3D shape coded by piecewise Bézier curves lies in a union of temporal subspaces. To this end, we define the matrix $\hat{\mathbf{S}}$ that rearranges the entries of $\mathbf{S}$ into a new $3N \times F$ matrix [18]. Both matrices can be related by the expressions $\mathbf{S} = (\mathbf{I}_3 \otimes \hat{\mathbf{S}}^\top)\mathbf{A}$ and $\hat{\mathbf{S}} = (\mathbf{S}^\top \otimes \mathbf{I}_3)\mathbf{B}$, where $\mathbf{A}$ and $\mathbf{B}$ are binary matrices of size $9N \times N$ and $9F \times F$, respectively. Thanks to this interpretation, we can finally define $\hat{\mathbf{S}} = \hat{\mathbf{S}}\mathbf{T} + \mathbf{E}$, where $\mathbf{T}$ is a $F \times F$ affinity matrix to encode the deformation grouping, and $\mathbf{E}$ is a $3N \times F$ residual noise. The most standard approach to solve the problem is to enforce a low-rank constraint over $\hat{\mathbf{S}}$ since the 3D shape configurations can be modeled as a reduced combination of vectors. In the same line, $\mathbf{T}$ is also low rank because it could be factorized as $\mathbf{LL}^\top$, where the rank of $\mathbf{L}$ corresponds to the number of groups in the sequence.

4. 3D DYNAMIC SHAPE, GROUPING AND MOTION FROM 2D POINT TRAJECTORIES

Our aim is to jointly recover the 3D dynamic shape and the deformation grouping of an object, as well as the camera motion, all of them, from partial 2D point trajectories in a RGB video. We propose an unsupervised, accurate, unified and efficient optimization strategy that does not need any training data at all. To this end, we formulate the full problem by introducing a union of piecewise Bézier subspaces, as it was defined in sections 2 and 3. As it was introduced in section 3, some matrices in our formulation are low-rank and enforce that is a non-convex NP-hard problem. Fortunately, we can use a nuclear norm instead, which is its convex relaxation [28]. Moreover, to improve robustness, we use a fourth-order temporal filter in our formulation by means of the expression $\hat{\mathbf{S}}\mathbf{F} = \mathbf{0}$, as it was done recently in [18].

With these ingredients, we denote the set of all model parameters to be recovered by $\boldsymbol{\Phi} \equiv \{\mathbf{W}, \mathbf{G}, \mathbf{S}, \hat{\mathbf{S}}, \mathbf{T}, \mathbf{C}, \mathbf{E}\}$. Our input data consists of partial 2D point trajectories in a RGB video $\bar{\mathbf{W}}$, and the corresponding visibility matrix $\mathbf{V} \in \mathbb{R}^{F \times N}$, with $\{0, 1\}$ entries indicating whether a point in a specific frame is missing or not. Taking into account all previous definitions along with the orthonormality constraints on camera rotations, our problem can be written as:

$$\begin{aligned} \arg\min_{\boldsymbol{\Phi}} \;\; & \| (\mathbf{V} \otimes \mathbf{1}_2) \odot \left(\mathbf{W} - \bar{\mathbf{W}}\right) \|_F^2 + \beta \|\mathbf{W}\|_* + \zeta g(\mathbf{R}^f) \\ & + \alpha(\|\mathbf{S} - \mathbf{BC}\|_F + \|(\mathbf{I}_3 \otimes (\mathbf{M} - \mathbf{N}))\mathbf{C}\|_F) \\ & + \gamma(\|\hat{\mathbf{S}}\|_* + \|\mathbf{T}\|_*) + \lambda \|\mathbf{E}\|_{2,1} \qquad (3) \\ \text{subject to} \;\; & \mathbf{W} = \mathbf{GS} \\ & \mathbf{S} = (\mathbf{I}_3 \otimes \hat{\mathbf{S}}^\top)\mathbf{A} \\ & \hat{\mathbf{S}} = \hat{\mathbf{S}}\mathbf{T} + \mathbf{E} \\ & \hat{\mathbf{S}}\mathbf{F} = \mathbf{0} \\ & \mathbf{R}^f {\mathbf{R}^f}^\top = \mathbf{I}_2 \,, \quad 1 \leq f \leq F \end{aligned}$$

where $\mathbf{1}$ denotes a vector of ones, and $\odot$ represents a Hadamard product. $\|\cdot\|_*$, $\|\cdot\|_F$ and $\|\cdot\|_{2,1}$ indicate the nuclear, Frobenius and $l_{2,1}$-norms, respectively. $\{\beta, \zeta, \alpha, \gamma, \lambda\}$ are penalty weight coefficients. Finally, we denote by $g(\cdot)$ the function to impose smooth solutions on the camera rotation.

Energy in Eq. (3) can be minimized by means of a two-step factorization approach in which: 1) complete missing entries $\mathbf{W}$, 2) estimate jointly motion $\mathbf{G}$, shape $(\mathbf{S}, \hat{\mathbf{S}}, \mathbf{C}, \mathbf{E})$ and grouping $\mathbf{T}$ parameters. In order to solve both problems, we apply augmented Lagrange multipliers [29].

5. EXPERIMENTAL EVALUATION

We now present our experimental evaluation on several human motion videos, including articulated and continuous deformation, several body configurations and scenarios with missing or dense entries. For quantitative comparison, we apply our approach on the articulated human motion dataset introduced in [5], which includes five types of activities, and nine competing methods are considered: EM-PPCA [4], MP [2], PTA [5], CSF [8], KSTA [7], BMM [16], PPTA [27], URS [18], and TRUS [23]; under two scenarios: noise-free and noisy 2D point trajectories as it was done in [27]. As in the literature [8, 16, 27], we provide the normalized mean 3D error e_S, and the mean rotation error e_R. For further details, we refer the reader to these papers. We also report the deformation grouping error e_G as defined in [20], after applying spectral clustering [30] over the estimated affinity matrix $\mathbf{T}$.

As it occurs for some methods with the subspace rank, we need to tune the number of control points K by hand. Fortunately, as it can be seen in Fig. 2, the error reduction is consistent as K increases, doing the error always remains within reasonable bounds. In contrast to other approaches, this is

Met. / Data	EM-PPCA [4]		MP [2]		PTA [5]		CSF [8]		KSTA [7]		BMM [16]		PPTA [27]		URS [18]			TRUS [23]			(Ours)		
	e_R	$e_S(K)$	e_R	$e_S(K)$	e_R	$e_S(K)$	e_R	$e_S(K)$	e_R	$e_S(K)$	e_R	$e_S(K)$	e_R	$e_S(K)$	e_R	e_S	$e_G[\%]$	e_R	e_S	$e_G[\%]$	e_R	e_S	$e_G[\%]$
Noise-free observations																									
Drink	.186	.261(7)	.330	.357(12)	.006	.025(13)	.006	.022(6)	.006	.020(12)	.007	.027(12)	.006	.011(30)	.006	**.009**	0.8(2)	.006	**.009**	0.6(2)	.005	**.009**	0.6(2)
Stretch	.749	.458(7)	.832	.900(8)	.055	.109(12)	.049	.071(8)	.049	.064(11)	.068	.103(11)	.058	.084(11)	.058	.061	4.1(3)	.058	**.060**	4.1(3)	.048	.062	4.3(3)
Yoga	.688	.445(8)	.854	.786(2)	.106	.163(11)	.102	.147(7)	.102	.148(7)	.088	.115(10)	.106	.158(11)	.106	.143	0.3(2)	.091	.133	0.2(2)	.076	**.111**	0.1(2)
Pick-up	.417	.423(14)	.249	.429(5)	.155	.237(12)	.155	.230(6)	.155	.233(6)	.121	.173(12)	.154	.235(12)	.154	.221	3.7(3)	.147	.209	3.0(3)	.104	**.138**	1.4(3)
Dance	–	.339(4)	–	.271(5)	–	.296(5)	–	.271(2)	–	.249(4)	–	.188(10)	–	.229(4)	–	.165	–	–	.150	–	–	**.143**	–
Average error:		.385		.549		.166		.148		.143		.121		.143		.119			.112			**.092**	
Relative error:		4.16		5.93		1.79		1.60		1.54		1.31		1.54		1.28			1.21			**1.00**	
Noisy observations																									
Drink	.231	.250(7)	.329	.517(12)	.043	.045(13)	.043	.044(6)	.043	.042(12)	.044	.056(12)	.042	.038(30)	.042	.044	3.6(2)	.036	**.034**	1.4(2)	.037	.036	1.3(2)
Stretch	.819	.886(7)	.872	.975(8)	.091	.144(12)	.091	.121(8)	.091	.166(11)	.098	.183(11)	.091	.123(11)	.091	**.119**	8.4(3)	.091	**.119**	5.1(3)	.091	.120	4.9(3)
Yoga	.700	.507(8)	.858	.791(2)	.124	.174(11)	.125	.168(7)	.125	.172(7)	.136	.195(10)	.124	.174(11)	.125	.167	0.0(2)	.112	**.162**	0.2(2)	.115	.164	0.2(2)
Pick-up	.499	.807(14)	.250	.407(5)	.148	.228(12)	.148	.224(6)	.148	.222(6)	.141	.212(12)	.148	.228(12)	.148	.207	3.1(3)	.147	.205	2.5(3)	.103	**.136**	1.2(3)
Dance	–	.336(4)	–	.282(5)	–	.299(5)	–	.266(2)	–	.248(4)	–	.236(10)	–	.222(4)	–	.164	–	–	.157	–	–	**.146**	–
Average error:		.557		.594		.178		.165		.170		.176		.157		.140			.135			**.120**	
Relative error:		4.64		4.95		1.48		1.37		1.42		1.47		1.31		1.17			1.12			**1.00**	

Table 2. Quantitative evaluation on human motion capture videos. Rotation e_R and 3D reconstruction e_S errors for competing techniques: EM-PPCA [4], MP [2], PTA [5], CSF [8], KSTA [7], BMM [16], PPTA [27], URS [18], and TRUS [23]; and for our approach, considering both noise-free and noisy observations. The table also indicates in parentheses the rank K of the linear subspace that produced the lowest e_S. Relative error is always computed with respect to our reconstruction. When possible, $e_G[\%]$ and the number of motion groups in parentheses are provided. "–" means that ground truth is not available.

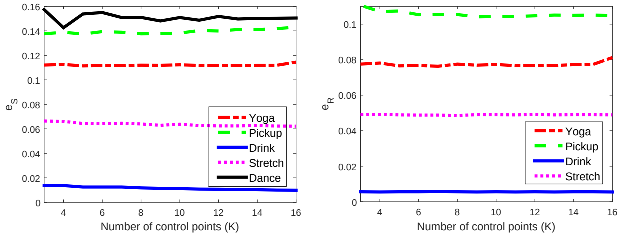

Fig. 2. 3D Reconstruction and rotation errors as a function of K. Results on the five human motion capture videos.

a key factor for our method since it does not require tuning accurately any parameter and could be done a priori according to other factors as the computational resources. Table 2 summarizes both e_S and e_R for all methods, sequences, and scenarios. It is worth mentioning that our approach outperforms consistently the state of the art in terms of joint estimations, reducing the 3D error of other methods by large margins between the 21% and 593% for noise-free, and between the 12% and 495% for noisy observations, respectively. We also include e_G when available. As our approach produces better 3D reconstructions on average, the partition on deformation groups is very competitive with respect to competing approaches. The median computation time in unoptimized Matlab code for these experiments was 103 seconds, on a commodity laptop with an Intel Core i7 processor at 2.4GHz.

We next show the availability of our approach against missing entries due to self-occlusions or lack of visibility, by processing an American-sign-language video, where a human face is moving and gesturing [11]. Figure 3-top displays some images and our 3D reconstruction for all points. Moreover, our algorithm detects three groups in this video: open and closed mouth with open eyes, and closed eyes. Finally, we also validate our approach on dense data by running two videos with 20,561 and 68,295 2D point trajectories taken from [17], where a back and a heart are deforming and moving, respectively. In the same figure is represented our joint solution in these videos, obtaining qualitatively accurate and physically possible solutions in comparison to [23].

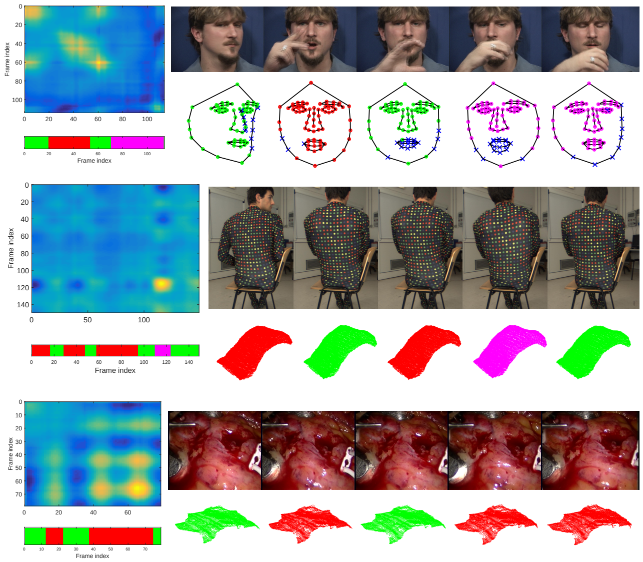

Fig. 3. Qualitative evaluation on real *Face*, *Back* and *Heart* videos. In all cases, we display the same information. **Left:** Deformation affinity matrix **T** we recover, and the corresponding grouping. **Right:** Images and 3D reconstruction using another point of view. Every color corresponds to a deformation group. Blue crosses represent missing points.

6. CONCLUSION

We have introduced a union of piecewise Bézier subspaces in combination with temporal smoothness priors and spatial continuities to model 3D trajectories of a dynamic object observed with a RGB camera. To this end, we have presented an energy-based formulation that can recover all model parameters without assuming any training data at all in an unsupervised, unified, accurate and efficient manner. Experimental results on human motion videos show our solution provides competitive joint estimations with respect to state-of-the-art approaches. Our future work is oriented to extend our formulation for sequential processing as the data arrive.

7. REFERENCES

[1] A. Agudo, J. M. M. Montiel, L. Agapito, and B. Calvo, "Modal space: A physics-based model for sequential estimation of time-varying shape from monocular video," *JMIV*, vol. 57, no. 1, pp. 75–98, 2017.

[2] M. Paladini, A. Del Bue, M. Stosic, M. Dodig, J. Xavier, and L. Agapito, "Factorization for non-rigid and articulated structure using metric projections," in *CVPR*, 2009.

[3] M. Lee, J. Cho, C. H. Choi, and S. Oh, "Procrustean normal distribution for non-rigid structure from motion," in *CVPR*, 2013.

[4] L. Torresani, A. Hertzmann, and C. Bregler, "Non-rigid structure-from-motion: estimating shape and motion with hierarchical priors," *TPAMI*, vol. 30, no. 5, pp. 878–892, 2008.

[5] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, "Non-rigid structure from motion in trajectory space," in *NIPS*, 2008.

[6] H. S. Park, T. Shiratori, I. Matthews, and Y. Sheikh, "3D reconstruction of a moving point from a series of 2D projections," in *ECCV*, 2010.

[7] P. F. U. Gotardo and A. M. Martinez, "Kernel non-rigid structure from motion," in *ICCV*, 2011.

[8] P. F. U. Gotardo and A. M. Martinez, "Computing smooth time-trajectories for camera and deformable shape in structure from motion with occlusion," *TPAMI*, vol. 33, no. 10, pp. 2051–2065, 2011.

[9] T. Simon, J. Valmadre, I. Matthews, and Y. Sheikh, "Separable spatiotemporal priors for convex reconstruction of time-varying 3D point clouds," in *ECCV*, 2014.

[10] X. Xu and E. Dunn, "Discrete Laplace operator estimation for dynamic 3D reconstruction," in *ICCV*, 2019.

[11] A. Agudo and F. Moreno-Noguer, "Force-based representation for non-rigid shape and elastic model estimation," *TPAMI*, vol. 40, no. 9, pp. 2137–2150, 2018.

[12] D. Novotny, N. Ravi, B. Graham, N. Neverova, and A. Vedaldi, "C3DPO: Canonical 3D pose networks for non-rigid structure from motion," in *ICCV*, 2019.

[13] V. Sidhu, E. Tretschk, V. Golyanik, A. Agudo, and C. Theobalt, "Neural dense non-rigid structure from motion with latent space constraints," in *ECCV*, 2020.

[14] A. Agudo, B. Calvo, and J. M. M. Montiel, "FEM models to code non-rigid EKF monocular SLAM," in *ICCVW*, 2011.

[15] J. Fayad, L. Agapito, and A. Del Bue, "Piecewise quadratic reconstruction of non-rigid surfaces from monocular sequences," in *ECCV*, 2010.

[16] Y. Dai, H. Li, and M. He, "A simple prior-free method for non-rigid structure from motion factorization," in *CVPR*, 2012.

[17] R. Garg, A. Roussos, and L. Agapito, "Dense variational reconstruction of non-rigid surfaces from monocular video," in *CVPR*, 2013.

[18] A. Agudo and F. Moreno-Noguer, "Deformable motion 3D reconstruction by union of regularized subspaces," in *ICIP*, 2018.

[19] Y. Zhu, D. Huang, F. de la Torre, and S. Lucey, "Complex non-rigid motion 3D reconstruction by union of subspaces," in *CVPR*, 2014.

[20] A. Agudo and F. Moreno-Noguer, "DUST: Dual union of spatio-temporal subspaces for monocular multiple object 3D reconstruction," in *CVPR*, 2017.

[21] S. Kumar, Y. Dai, and H. Li, "Spatio-temporal union of subspaces for multi-body non-rigid structure-from-motion," *PR*, vol. 77, no. 11, pp. 428–443, 2017.

[22] A. Bartoli, V. Gay-Bellile, U. Castellani, J. Peyras, S. Olsen, and P. Sayd, "Coarse-to-fine low-rank structure-from-motion," in *CVPR*, 2008.

[23] A. Agudo, "Segmentation and 3D reconstruction of non-rigid shape from RGB video," in *ICIP*, 2020.

[24] S. Biswas and B. C. Lovell, *Bezier and Splines in Image Processing and Machine Vision*, Springer-Verlag, London, 2008.

[25] G. E. Farin and J. Hoschek; M.-S. Kim, *Handbook of Computer Aided Geometric Design*, Elsevier, 2002.

[26] J. Valmadre and S. Lucey, "General trajectory prior for non-rigid reconstruction," in *CVPR*, 2012.

[27] A. Agudo and F. Moreno-Noguer, "A scalable, efficient, and accurate solution to non-rigid structure from motion," *CVIU*, vol. 167, no. 2, pp. 121–133, 2018.

[28] Y. Chen, H. Xu, C. Caramanis, and S. Sanghavi, "Robust matrix completion with corrupted columns," in *ICML*, 2011.

[29] A. Agudo, "Unsupervised 3D reconstruction and grouping of rigid and non-rigid categories," *TPAMI, to appear*, 2021.

[30] W. Y. Chen, Y. Song, H. Bai, C.J. Lin, and E. Chang, "Parallel spectral clustering in distributed systems," *TPAMI*, vol. 33, no. 3, pp. 568–586, 2010.