

SPLINE HUMAN MOTION RECOVERY

Antonio Agudo

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

ABSTRACT

Simultaneous camera pose, 4D reconstruction of an object and deformation clustering from incomplete 2D point tracks in a video is a challenging problem. To solve it, in this work we introduce a union of piecewise subspaces to encode the 4D shape, where two modalities based on B-splines and Catmull-Rom curves are considered. We demonstrate that formulating the problem in terms of B-spline or Catmull-Rom functions, allows for a better physical interpretation of the resulting priors while C^1 and C^2 continuities are automatically imposed without needing any additional constraint. An optimization framework is proposed to sort out the problem in a unified, accurate, unsupervised and efficient manner. We extensively validate our claims on a wide range of human motions, including articulated and continuous deformations as well as those cases with noisy and missing measurements where our approach provides competing joint solutions.

Index Terms— 4D Reconstruction, Catmull-Rom, B-splines, Clustering, Optimization.

1. INTRODUCTION

The problem of simultaneously recovering the 3D reconstruction of a dynamic object and the camera motion from incomplete 2D point tracks in a video is coined in the literature as non-rigid structure from motion [1, 2, 3]. Despite being a very challenging problem with many real-world applications in several domains, solving it without 3D supervision is an ill-posed problem that requires exploring the art of priors to be tractable. Probably, the most standard prior in this context is that based on a low-rank subspace in terms of shapes [4, 5], trajectories [6, 7], shape-trajectory models [8, 9, 10, 11] or forces to induce the deformations [12]. These models for shapes [13] and trajectories [14] were also considered in a deep-learning context, where large amounts of training data are needed to learn the model. According to literature, most shape- and force-based approaches need to estimate the basis from data. However, a trajectory basis for a wide variety of motions could be pre-defined a priori, by using some signal representations as the discrete cosine transform [6, 7, 14]

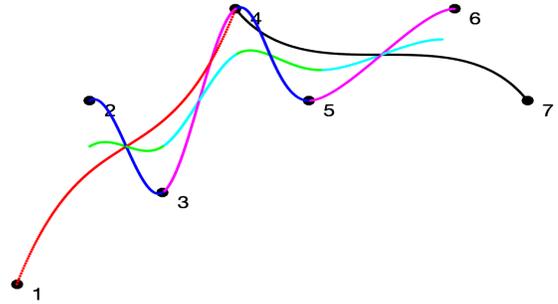


Fig. 1. Human motion modeling by means of piecewise curves. Considering a parametric scenario with $K = 7$ control points, several performances can be obtained depending on the trajectory basis: 1) Catmull-Rom functions that interpolate the points (blue and magenta pieces), 2) B-spline functions that approximate the points (green and cyan pieces), and 3) Bézier functions that include some points in the solution (red and black pieces). As it can be seen, the Bézier curve is not smooth between pieces. Best viewed in color.

where the problem is handled in a global manner. More recently, these methods were extended to piece-varying motion, where the trajectory basis is local-aware, more controllable but it needs to impose restrictive continuity constraints to smoothly connect the pieces [15]. The low-rank constraint has also been imposed by directly minimizing the rank of a matrix representing the 4D shape, considering the data lie in a single [16, 17], in a union of temporal [18, 19], in a dual union of spatio-temporal [20, 21], or in multiple [22] subspaces. In these cases, some temporal and/or spatial segmentations were also included in the solution.

We here depart from previous work in that our approach exploits a union of piecewise subspaces to encode 3D point trajectories that are observed in 2D data. The key factor of our approach is that we use pre-defined Catmull-Rom and B-spline functions to define the trajectory basis that directly guarantees C^1 and C^2 continuities, respectively. Our piecewise model is parametrized by a set of control points that are interpolated/approximated in the case of Catmull-Rom/B-spline functions, respectively (a comparison of both representations is displayed in Fig. 1). We present a novel unified, accurate, unsupervised, and efficient approach that can exploit both modalities naturally, and it produces competing results with respect to state-of-the-art solutions. To the best of our knowledge, this is the first time a union of B-spline/Catmull-Rom subspaces is introduced.

This work has been partially supported by the Spanish Ministry of Science and Innovation under project MoHuCo PID2020-120049RB.

2. SMOOTH-AWARE SPLINE CURVES

The use of parametric curves defined by a set of control points have been widely employed in mechanical engineering for computer-aided design [23]; in the context of computer graphics for object modeling, camera paths or vector fonts as well as in many computer-vision and image-processing applications such as image segmentation, compression [24] and recognition [25]. Without loss of generality, a curve $\mathbf{c}(s)$ can be defined by means of a linear combination of K basis of degree $K - 1$ restricted to the continuous interval $s = [0, \dots, 1]$ as $\mathbf{c}(s) = \sum_{k=1}^K b_k(s)\mathbf{p}_k$, where $b_k(s)$ and \mathbf{p}_k are the k -th basis and control point, respectively. The control points, for any d -dimensional space, can be applied independently for every dimension, being they directly defined in a 3D space by its coordinates as $\mathbf{p}_k = [p_{xk}, p_{yk}, p_{zk}]^\top$.

While a curve could be approximated by high-order Bernstein polynomials, such as in a Bézier curve, the alternative will normally produce unwanted wiggles or over-smoothing solutions that do not consider rich and local information. Moreover, the obtained global approximation would be very hard to control. To solve that, splines can be considered to obtain a piecewise smooth curve that is coded by low-order –up to a degree of 3– polynomials. In this context, maybe the most standard way is the use of low-order Bézier curves [15]. For Bézier curves, the solution is contained in the convex hull of the control polygon and, the use of quadratic or cubic functions is a direct consequence of the number of control points to encode the global curve. 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 (see Fig. 1). Unfortunately, and as it can be seen in the figure, this construction does not guarantee C^1 and C^2 continuities that instead it should be enforced via constraints.

Looking for a more compact representation, we could consider the use of Catmull-Rom splines that provides local controllability –like in piecewise Bézier curves– while both C^0 and C^1 continuities will be automatically guaranteed. In this interpolation case, all control points excepting the first and the last are included in the global curve (see Fig. 1). To achieve that, a different set basis needs to be used but, in contrast to piecewise Bézier curves, this time with the same basis for every piece (see Fig. 2-left with cubic Catmull-Rom functions). In addition to that, we could also consider uniform B-spline basis functions to solve the problem, achieving C^2 continuity and local controllability in a unified formulation. To this end, B-spline basis functions (see Fig. 2-middle with cubic B-spline functions) are used to produce global curves that do not interpolate any of control points (an example it can be observed in Fig. 1), i.e., the set of control points is approximated by piecewise and continuous sections.

Table 1 shows the number of pieces as a function of the number of control points for both B-spline and Catmull-Rom modalities. It is worth noting that a unique set of cubic func-

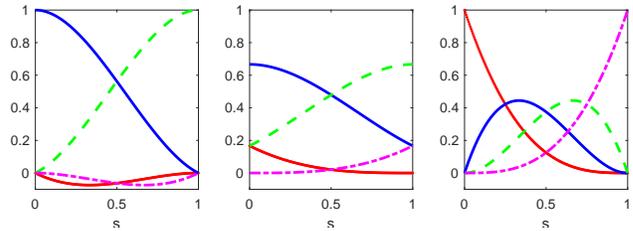


Fig. 2. Cubic spline basis functions. **Left:** Catmull-Rom. **Middle:** B-spline. **Right:** Bézier. Every color represents a k -th basis $b_k(s)$. Best viewed in color.

K	P	Modality	N	K	P	Modality	N
4	1	B-spline/Catmull	0/2	11	8	B-spline/Catmull	0/9
5	2	B-spline/Catmull	0/3	12	9	B-spline/Catmull	0/10
6	3	B-spline/Catmull	0/4	13	10	B-spline/Catmull	0/11
7	4	B-spline/Catmull	0/5	14	11	B-spline/Catmull	0/12
8	5	B-spline/Catmull	0/6	15	12	B-spline/Catmull	0/13
9	6	B-spline/Catmull	0/7	16	13	B-spline/Catmull	0/14
10	7	B-spline/Catmull	0/8	17	14	B-spline/Catmull	0/15

Table 1. Curve modeling using B-spline and Catmull-Rom functions. The table shows how a generic curve can be approximated/interpolated by several B-spline/Catmull-Rom curves, respectively, as a function of the number of control points (K), the number of pieces (P) to be used, and the number of control points to lie in the final curve (N). For every modality, the same cubic functions are used per piece.

tions is employed for every modality instead of a combination of quadratic and cubic pieces that are used for piecewise Bézier curves [15]. In general terms, the number of pieces is bigger and, therefore, the global curve is more local-aware controllable, while C^1 and C^2 continuities between consecutive pieces can be automatically enforced without including any additional constraint. In the next section, we introduce both B-spline and Catmull-Rom curves to encode 3D motion, fitting it from 2D point tracks in a sequence of color images.

3. MODELING 3D SPLINES FROM 2D MOTION

Let $\mathbf{s}_n^i = [x_n^i, y_n^i, z_n^i]^\top$ be the n -th 3D point at image i of an object. Assuming an orthographic camera, its 2D projection can be denoted as $\mathbf{w}_n^i = [u_n^i, v_n^i]^\top$. After detecting the N points in the I images, and subtracting a 2D translation to obtain zero-mean measurements, the 3D-to-2D projection is:

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

where \mathbf{W} is a $2I \times N$ measurement matrix to collect the 2D point tracks, \mathbf{G} is a $2I \times 3I$ block diagonal matrix, made of the I truncated 2×3 camera rotations \mathbf{R}^i , and \mathbf{S} is a $3I \times N$ shape matrix with the 3D locations of the corresponding dynamic points. The inverse problem to infer both camera pose and the 3D non-rigid reconstruction from 2D data consists in factoring \mathbf{W} into motion \mathbf{G} and shape \mathbf{S} factors.

One possibility to handle the previous problem is the use of trajectory-based models [6, 14, 15, 26, 27], where the location of each point coordinate over time is encoded by a linear combination of K low-frequency basis vectors. These methods were first applied in a global manner without being useful to recover piece-varying motions and later, in a local fashion by means of the use of piecewise Bézier curves [15]. Despite providing striking results and local controllability, the global curve could not be smooth enough as some continuities need to be enforced by additional constraints. To sort out this limitation, we propose the use of B-splines or Catmull-Rom functions, achieving so a more compact representation where the continuities are automatically enforced. To this end, and following [15], we consider N $3K$ -dimensional vectors as $\boldsymbol{\kappa}^n = [p_{x1}^n, \dots, p_{xK}^n, p_{y1}^n, \dots, p_{yK}^n, p_{z1}^n, \dots, p_{zK}^n]^\top$, to collect the K control-point locations associated with the n -th object point. Additionally, we define I K -dimensional vectors as $\mathbf{b}^i(s) = [b_1(s), \dots, b_K(s)]^\top$, where the entries $b_k(s)$ represent the k -th B-spline or Catmull-Rom basis component at instant s . As in both cases we use cubic functions, all entries excepting four in $\mathbf{b}^i(s)$ will be null. To find those particular entries in the i -th image, we consider the total number of control points to obtain the number of pieces (according to Table 1) and assign the corresponding number of images to each piece defined by just four control points. Then, for every piece, it is established a direct correspondence between the number of images and the interval $[0, \dots, 1]$, obtaining the s value. Control-point locations and basis functions are combined to encode a 3D image-varying shape as:

$$\mathbf{S} = \underbrace{\begin{bmatrix} \mathbf{I}_3 \otimes (\mathbf{b}^1(s))^\top \\ \vdots \\ \mathbf{I}_3 \otimes (\mathbf{b}^I(s))^\top \end{bmatrix}}_{\mathbf{B} \in \mathbb{R}^{3I \times 3K}} \underbrace{\begin{bmatrix} \boldsymbol{\kappa}^1 & \dots & \boldsymbol{\kappa}^N \end{bmatrix}}_{\mathbf{C} \in \mathbb{R}^{3K \times N}}, \quad (1)$$

where \mathbf{I} is a 3×3 identity matrix, and \otimes represents the Kronecker product. Finally, we assume that the image-varying shape encoded by B-spline or Catmull-Rom curves lie in a union of temporal subspaces. To this end, we introduce the matrix $\hat{\mathbf{S}}$ that rearranges the entries of \mathbf{S} into a new $3N \times I$ matrix [18, 16] and then define $\hat{\mathbf{S}} = \hat{\mathbf{S}}\mathbf{T} + \mathbf{E}$, where \mathbf{T} is a $I \times I$ similarity matrix to encode the deformation clustering, and \mathbf{E} is a $3N \times I$ residual noise. To solve the problem, a low-rank constraint over $\hat{\mathbf{S}}$ is enforced as the deformation can be coded by a reduced combination of modes. \mathbf{T} is also low rank and that value represents the number of clusters.

4. OPTIMIZATION PROBLEM

Our goal is to jointly retrieve the camera pose, the 3D non-rigid shape, as well as the deformation clustering of an object, all of them, from partial 2D point tracks. To this end, we exploit a union of B-spline or Catmull-Rom subspaces – two modalities are possible –, as it was introduced in sec. 3.

As it was said above, we must impose some low-rank constraints that is a non-convex NP-hard problem, being handled by means of a nuclear norm as its convex relaxation [28]. Finally, we enforce orthonormality constraints on camera rotations and we also use a fourth-order temporal filter in our formulation by means of the expression $\hat{\mathbf{S}}\mathbf{F} = \mathbf{0}$, as it was introduced in [18]. As a result, our optimization strategy is unsupervised, unified, accurate and efficient.

With these considerations, let $\Psi \equiv \{\mathbf{W}, \mathbf{G}, \mathbf{S}, \hat{\mathbf{S}}, \mathbf{T}, \mathbf{C}, \mathbf{E}\}$ be the set of all model parameters to be estimated. As input data, we consider partial 2D point tracks in $\bar{\mathbf{W}}$, and the corresponding visibility matrix $\mathbf{V} \in \mathbb{R}^{I \times N}$, with $\{0, 1\}$ values indicating whether a point in a specific image is missing or not. On overall, the problem can be written as:

$$\arg \min_{\Psi} \quad \|(\mathbf{V} \otimes \mathbf{1}_2) \odot (\mathbf{W} - \bar{\mathbf{W}})\|_F^2 + \beta \|\mathbf{W}\|_* + \zeta g(\mathbf{R}^i) \\ + \alpha \|\mathbf{S} - \mathbf{BC}\|_F + \gamma (\|\hat{\mathbf{S}}\|_* + \|\mathbf{T}\|_*) + \lambda \|\mathbf{E}\|_{2,1} \quad (2)$$

$$\text{subject to} \quad \mathbf{W} = \mathbf{GS} \\ \mathbf{S} = q(\hat{\mathbf{S}}) \\ \hat{\mathbf{S}} = \hat{\mathbf{S}}\mathbf{T} + \mathbf{E} \\ \hat{\mathbf{S}}\mathbf{F} = \mathbf{0} \\ \mathbf{R}^i \mathbf{R}^{i\top} = \mathbf{I}_2, \quad 1 \leq i \leq I$$

where $\mathbf{1}$ is a vector of ones, and \odot represents a Hadamard product. $\|\cdot\|_F$, $\|\cdot\|_*$ and $\|\cdot\|_{2,1}$ indicate the Frobenius, nuclear and $l_{2,1}$ -norms, respectively. $\{\beta, \zeta, \alpha, \gamma, \lambda\}$ are weight coefficients. Finally, we denote by $g(\cdot)$ the function to enforce smooth solutions on the camera rotation, and by $q(\cdot)$ the function to map $\hat{\mathbf{S}}$ into \mathbf{S} . Problem in Eq. (2) can be solved by means of a two-step approach in which: 1) complete missing tracks \mathbf{W} , 2) estimate rotation \mathbf{G} , 4D shape $(\mathbf{S}, \hat{\mathbf{S}}, \mathbf{C}, \mathbf{E})$ and clustering \mathbf{T} parameters. Both problems are solved by augmented Lagrange multipliers [20].

5. EXPERIMENTAL EVALUATION

We provide experimental results on several human motion sequences, considering different body configurations with articulated and continuous deformation as well as with partial and dense entries. For quantitative evaluation, we use the articulated human motion dataset introduced in [6] which includes five types of actions; nine competing methods are considered for comparison: MP [4], PTA [6], CSF [9], KSTA [8], BMM [16], PPTA [27], URS [18], TRUS [3], and PBS [15]; and under two situations: noise-free and noisy 2D point tracks as it was done in [27]. As in [9, 16, 27], we provide the mean rotation error e_R and the normalized mean 3D error e_S . For further details, we refer the reader to these papers. We also report the deformation clustering error e_C [20], after applying spectral clustering [29] over the similarity matrix \mathbf{T} .

As in piecewise Bézier curves [15], our approach –with two modalities– needs to fix the number of control points K .

Data	Met.	MP [4]		PTA [6]		CSF [9]		KSTA [8]		BMM [16]		PPTA [27]		URS [18]		TRUS [3]		PBS [15]		BS (Ours)		CR (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	$e_C[\%]$	e_R	e_S	$e_C[\%]$	e_R	e_S	$e_C[\%]$	e_R	e_S	$e_C[\%]$			
<i>Noise-free observations</i>																												
Drink		.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)	.005	.008	0.6(2)	.005	.009	0.6(2)
Stretch		.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)	.048	.062	4.3(3)	.047	.061	4.0(3)
Yoga		.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)	.076	.110	0.1(2)	.075	.109	0.1(2)
Pick-up		.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)	.104	.137	1.1(3)	.104	.136	0.8(3)
Dance		-	.271(5)	-	.296(5)	-	.271(2)	-	.249(4)	-	.188(10)	-	.229(4)	-	.165	-	-	.150	-	-	.143	-	-	.140	-	-	.141	-
Average error:		.549	-	.166	-	.148	-	.143	-	.121	-	.143	-	.119	-	-	.112	-	-	.093	-	-	.091	-	-	.091	-	
Relative error:		6.01	-	1.82	-	1.63	-	1.57	-	1.33	-	1.57	-	1.31	-	-	1.23	-	-	1.02	-	-	1.00	-	-	1.00	-	
<i>Noisy observations</i>																												
Drink		.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)	.037	.035	1.3(2)	.037	.036	1.3(2)
Stretch		.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)	.091	.119	4.3(3)	.091	.119	4.3(3)
Yoga		.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)	.114	.164	0.2(2)	.114	.163	0.2(2)
Pick-up		.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)	.103	.136	1.2(3)	.103	.135	0.8(3)
Dance		-	.282(5)	-	.299(5)	-	.266(2)	-	.248(4)	-	.236(10)	-	.222(4)	-	.164	-	.157	-	.146	-	.144	-	-	.144	-	-	.144	-
Average error:		.594	-	.178	-	.165	-	.170	-	.176	-	.157	-	.140	-	-	.135	-	-	.120	-	-	.119	-	-	.119	-	
Relative error:		4.98	-	1.49	-	1.38	-	1.42	-	1.48	-	1.32	-	1.17	-	-	1.13	-	-	1.01	-	-	1.00	-	-	1.00	-	

Table 2. Quantitative comparison on human mocap sequences. Rotation e_R and 3D reconstruction e_S errors for competing techniques: MP [4], PTA [6], CSF [9], KSTA [8], BMM [16], PPTA [27], URS [18], TRUS [3], and PBS [15]; and for our methods BS and CR, 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 w.r.t. the best reconstruction. When possible, $e_C[\%]$ and the number of motion clusters in parentheses are provided. “-” means that ground truth is not available.

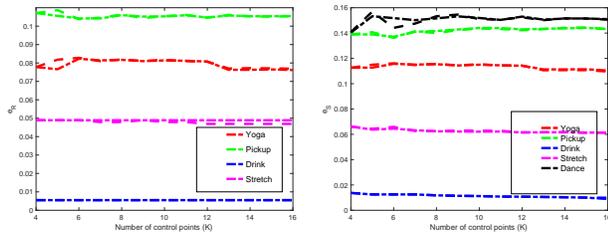


Fig. 3. e_R and e_S errors as a function of K for BS (-) and CR (-) functions. Results on the five human motion capture sequences.

Fortunately, our approach is not very sensitive to this selection, providing an error reduction as K increases, while the solution is within reasonable bounds (see Fig. 3). Thanks to this behavior, our approach does not need fine tuning any parameter and K could be set according to other factors such as external physical priors or computational resources. Table 2 summarizes both e_R and e_S for all methods, actions, and situations we consider. It is worth pointing out that our approaches provide competitive solutions in terms of joint estimations, improving e_S with respect to state-of-the-art approaches by large margins between the 2% and 601% for noise-free, and between the 1% and 498% for noisy observations, respectively. As a consequence of the improvement on 3D error, our approaches also achieve good clustering results e_C when available. The median computation time for both modalities in unoptimized Matlab code for these experiments was of 81 seconds, on a commodity laptop with an Intel Core i7 processor at 2.4GHz, i.e., giving an speed up of 27% in comparison with [15] due to no additional constraint to enforce spatial continuity is needed.

We provide results against a 11.5% of missing entries by processing an American-sign-language sequence where a human face is deforming and moving; and using dense (back and heart) data by running two sequences taken from [17]. Figure 4-top shows some images and our joint estimations for all points in every case by using Catmull-Rom functions. Our algorithm obtains qualitatively accurate and physically possible solutions in comparison to [3, 15]. For instance, three clusters in the face sequence (open and closed mouth with open eyes,

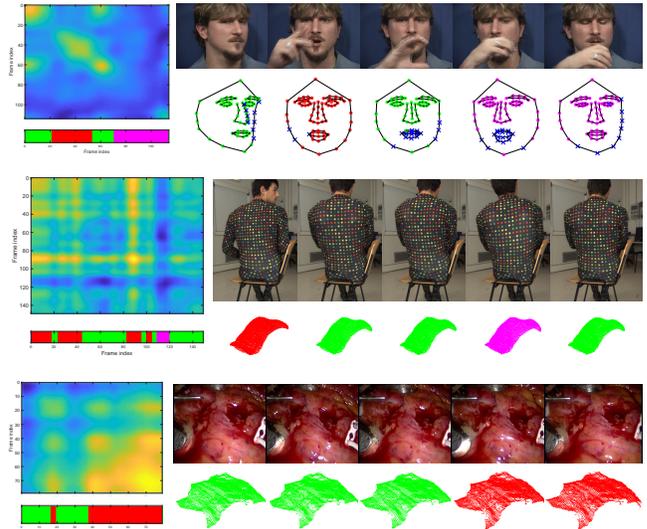


Fig. 4. Qualitative comparison on real Face, Back and Heart videos. The same information is provided in all cases. **Left:** Deformation similarity matrix T and the corresponding clustering bar we recover. **Right:** Images and 3D reconstruction for a novel point of view. Every color corresponds to a deformation cluster. Blue crosses represent missing tracks.

and closed eyes) are detected for our approach.

6. CONCLUSION

We have introduced a union of piecewise subspaces with two modalities based on B-splines and Catmull-Rom trajectory functions to encode a 4D shape that is observed with a color camera. This model is combined with some additional priors but without enforcing explicitly a spatial smooth continuity. All model parameters are recovered in a unified, accurate, unsupervised, and efficient manner by means of an optimization algorithm. Experimental results on human motion sequences show a good trade-off between accuracy and computational cost in comparison to state-of-the-art approaches. Our future work is oriented to extend our formulation to spline surfaces.

7. REFERENCES

- [1] 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.
- [2] M. Lee, C. H. Choi, and S. Oh, “A procrustean Markov process for non-rigid structure recovery,” in *CVPR*, 2014.
- [3] A. Agudo, “Segmentation and 3D reconstruction of non-rigid shape from RGB video,” in *ICIP*, 2020.
- [4] 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.
- [5] M. Lee, J. Cho, C. H. Choi, and S. Oh, “Procrustean normal distribution for non-rigid structure from motion,” in *CVPR*, 2013.
- [6] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, “Non-rigid structure from motion in trajectory space,” in *NIPS*, 2008.
- [7] 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.
- [8] P. F. U. Gotardo and A. M. Martinez, “Kernel non-rigid structure from motion,” in *ICCV*, 2011.
- [9] 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.
- [10] T. Simon, J. Valmadre, I. Matthews, and Y. Sheikh, “Separable spatiotemporal priors for convex reconstruction of time-varying 3D point clouds,” in *ECCV*, 2014.
- [11] X. Xu and E. Dunn, “Discrete Laplace operator estimation for dynamic 3D reconstruction,” in *ICCV*, 2019.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] A. Agudo, “Piecewise bézier space: Recovering 3D dynamic motion from video,” in *ICIP*, 2021.
- [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. Agudo, “Unsupervised 3D reconstruction and grouping of rigid and non-rigid categories,” *TPAMI*, vol. 44, no. 1, pp. 519–532, 2022.
- [23] G. E. Farin and J. Hoschek; M.-S. Kim, *Handbook of Computer Aided Geometric Design*, Elsevier, 2002.
- [24] S. Biswas and B. C. Lovell, *Bézier and Splines in Image Processing and Machine Vision*, Springer-Verlag, London, 2008.
- [25] E. Ghorbel, R. Boutteau, J. Boonaert, X. Savaier, and S. Lecoecuche, “3D real-time human actin recognition using a spline interpolation approach,” in *IPTA*, 2015.
- [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] 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.