The Visual Computer manuscript No.
(will be inserted by the editor)

Simultaneous Completion and Spatio-Temporal Grouping of Corrupted Motion Tracks

Antonio Agudo · Vincent Lepetit · Francesc Moreno-Noguer

Received: date / Accepted: date

Abstract Given an unordered list of 2D or 3D point trajectories corrupted by noise and partial observations, in this paper we introduce a framework to simultaneously recover the incomplete motion tracks and group the points into spatially and temporally coherent clusters. This advances existing work, which only addresses partial problems and without considering a unified and unsupervised solution. We cast this problem as a matrix completion one, in which point tracks are arranged into a matrix with the missing entries set as zeros. In order to perform the double clustering, the measurement matrix is assumed to be drawn from a dual union of spatio-temporal subspaces. The bases and the dimensionality for these subspaces, the affinity matrices used to encode the temporal and spatial clusters to which each point belongs, and the non-visible tracks, are then jointly estimated via Augmented Lagrange Multipliers in polynomial time. A thorough evaluation on incomplete motion tracks for multiple object typologies shows that the accuracy of the matrix we recover compares favorably to that obtained with existing low-rank matrix completion methods, specially under noisy measurements. In addition, besides recovering the incomplete tracks, the point trajectories are directly grouped into different object instances, and a number of semantically meaningful temporal primitive actions are automatically discovered.

Keywords Point Track Completion · Spatio-Temporal Clustering · Augmented Lagrangian Multiplier

Antonio Agudo and Francesc Moreno-Noguer
Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, 08028, Spain.
E-mail: {aagudo, fmoreno}@iri.upc.edu

Vincent Lepetit
Université de Bordeaux, cours de la Libération, Talance F-33405, France
E-mail: vincent.lepetit@u-bordeaux.fr

1 Introduction

Motion visual tracking is an important and essential component of perception that has been an active research area in computer vision for past two decades. The developments of 2D and 3D visual tracking algorithms have shown rapid progress thanks to the explosive growth of video data which in turn creates high demand for accuracy and speed of tracking methods. Current approaches are motivated to design faster and better methods in spite of the challenges that exist in this topic, especially robustness to large occlusions, drastic scale change, accurate localization, multi-object tracking, and recovery from failure (Hare et al. 2011; Jia et al. 2012). Despite the success in addressing numerous challenges under a wide range of scenarios, a number of core problems still remain unsolved. A major challenge in real scenarios is handling missing entries of the data, due to ad-hoc data collection, presence of outliers, sensor failure, or partial knowledge of relationships in a dataset. For instance, to recover object motions and deformations from video, the tracking algorithm may lose the track of features in some image frames due to lack of visibility or mismatches. In a similar manner, for 3D tracking, multi-camera systems (such as motion capture systems) (Van der Aa et al. 2011; Ionescu et al. 2014) are applied to obtain the time-varying evolution of a scenario. While these systems are now capable of recovering most of the observations, they can fail on real-world scenarios, such as those formed by multiple objects while are performing different activities, deforming, moving, and even interacting between them. In these cases, missing tracks continually appear, either as self-occlusions or occlusions between objects. It is worth mentioning that this is especially relevant in outdoors scenarios, where current algorithms to estimate motion tracks often produce partial solutions with a wide amount of missing entries.

In many fields, an underlying tenet is that the data may contain certain type of structure that enables intelligent processing and representation, and they can be characterized by using parametric models. Assuming that, the visual tracking completion problem can be addressed as a matrix completion one. To this end, one can use the well-known linear subspaces since they are easy to estimate, and often effective in many real-world applications. For instance, these models have been successfully used to characterize several types of visual data, such as motion (Yan and Pollefeys 2006; Rao et al. 2010a), shape (Liu and Yan 2011) and texture (Ma et al. 2007). Maybe, the most common choice it is to use the principal-component-analysis method, that is based on the hypothesis that the data are approximately drawn from a low-rank subspace. Unfortunately, real data from complex scenarios can rarely be well described by a single low-rank subspace. For these cases, a more reasonable model is to assume data are lying near several subspaces, i.e., the data are considered as samples approximately drawn from a union of several low-rank subspaces. The generality and importance of subspaces naturally leads to a challenging problem of subspace clustering, whose goal is to group data into clusters with every cluster corresponding to a different subspace. Solving clustering and finding low-dimensional representations of data are important unsupervised learning problems in machine learning with numerous applications, including image segmentation, system identification, data visualization and collaborative filtering to name just a few. The problem becomes even more complex if the data are partially observed, either due to sensor failure or visual occlusions (Agudo and Moreno-Noguer 2017a, 2015, 2018; Gotardo and Martinez 2011).

In this paper we propose, to the best of our knowledge, the first attempt to approximate high-dimensional data using a dual union of low-dimensional subspaces, accounting for two distinct criteria. Additionally, input data are assumed to be corrupted by partial observations and noise. We apply our approach to the specific case in which the input data encode 2D or 3D point trajectories of multiple dynamic objects with large percentages of missing entries, and we aim at hallucinating these missing tracks while simultaneously approximating the data using spatial and temporal subspaces, as well as filtering the noisy measurements.

We will formulate the problem as a matrix completion one. Input data will be arranged into a matrix with the missing entries set to zero. To encode data similarities, we introduce two affinity matrices to be learned. We will then devise an optimization scheme based on Augmented Lagrangian Multipliers (ALM) to simultaneously and efficiently estimate the missing entries, and the bases and dimensionality for of each low-rank subspace. The proposed algorithm is unsupervised, does not depend on the initialization nor relies on training data at all, and can be solved in polynomial time. An important corollary of our approach is that applying off-the-shelf state-of-the-art spectral clustering on the estimated affinity matrices, results in consistent temporal and spatial segmentations of the input data.

We evaluate the proposed algorithm on 2D and 3D incomplete motion-capture and real sequences of several objects performing complex actions and interacting with each other. We will show that the accuracy of the completed tracks we obtain improves that of state-of-the-art methods by a considerable margin, while we additionally provide a spatio-temporal clustering of the data, which in most cases has a direct physical interpretation (either the object identity or the type of motion it is performing).

2 Related Work

The most standard approach to perform matrix completion is to assume the underlying data lies in a single low-dimensional subspace. Early works (Knott and Bartholomew 1999; Tipping and Bishop 1999a) enforced this constraint based on expectation-maximization strategies to optimize non-convex functions of the model parameters and the missing entries. Other attempts constrain the solution space using trajectory (Gotardo and Martinez 2011), or spatio-temporal models (Akhter et al. 2012). Nevertheless, all these methods require a good initialization, and most importantly, they need to set the rank of the subspace a priori, performing poorly when the dimension of the subspace increases. Additionally, trajectory-based methods normally use a pre-defined basis, making them very problem specific. To address these limitations, another family of low-rank matrix-completion techniques has been recently proposed (Bhojanapalli and Jain 2013; Cai et al. 2010; Candès and Plan 2010; Chen et al. 2011; Cabral et al. 2013; Chiang et al. 2015). These methods estimate missing entries by optimizing the convex surrogate of the rank, i.e., by they enforce the nuclear norm of the complete matrix. These ideas were also applied in problems where the matrix directly includes visual tracking information, imposing smooth (Sui et al. 2015, 2016; Zhang et al. 2012) and sparse (Wang et al. 2013) representations. When the underlying subspace is not consistent with standard basis components and missing track locations are spread uniformly at random, these approaches are guaranteed to recover missing entries.

Unfortunately, matrix-completion techniques based on a single low-rank subspace cannot handle the challenging and more general scenario in which input data lie in a union of low-rank subspaces (e.g., when dealing with simultaneous and incomplete tracks of multiple objects performing complex motions). Data segmentation from full annotations was proposed by assuming a union of subspaces by means of a subspace clustering based on sparse representation (Elhamifar and Vidal 2013) or seeking the lowest rank one (Liu et al.

Approaches	Missing entries	Automatic rank	Temporal clustering	Spatial clustering	Unified framework
Category #1	✓	✓			
Category #2	✓		✓		
Category #3	✓	✓	✓		
Category #4	✓	✓	✓		✓
Ours	✓	✓	✓	✓	✓

Table 1 Qualitative comparison of our approach against competing techniques. We split the methods into four incremental categories, depending on a number of desirable properties, namely: robustness to missing entries, automatic estimation of rank, computation of temporal/spatial clusters, and unified formulation. Note that the method proposed in this paper is the only that can simultaneously offers all these properties. Some examples for every category are: #1 methods to solve completion using a single low-rank formulation Bhojanapalli and Jain 2013; Cai et al. 2010; Candès and Plan 2010; Chen et al. 2011; Cabral et al. 2013; Chiang et al. 2015; Sui et al. 2016, #2 methods to solve completion by means of multiple subspaces where the rank is known a priori Tipping and Bishop 1999b; Balzano et al. 2012, #3 methods able to automatically retrieve the matrix rank, such as Ma et al. 2008; Rao et al. 2010b; Yang et al. 2015, and finally #4, methods that can solve the problem in a unified manner Elhamifar 2016; Fan and Chow 2017. It is worth noting that (Agudo and Moreno-Noguer 2017b) used a spatio-temporal constraint for shape reconstruction, but not for shape completion as we do in this paper. Self-expressive models (Elhamifar and Vidal 2013; Liu et al. 2013) solved for one type of clustering but considering full data.

2013). Going back to the completion problem, the objective would extend to recovering the missing entries together with the clustering of the data according to the subspaces. Mixture of factor analyzers (Ghahramani and Hinton 1996), mixture of probabilistic principal component analysis (Tipping and Bishop 1999b) and incremental matrix completion algorithms with K-subspaces (Balzano et al. 2012), are some early methods used to address grouping and completion of multi-subspace data. Again, the performance of these methods highly depends on the initialization and degrades for large subspace ranks. A polynomial number of data points in the ambient space dimension is required in (Eriksson et al. 2012) which often cannot be met in high-dimensional datasets. Ma et al. (2008) proposed an algebraic approach to model data drawn from a union of subspaces based on generalized principal component analysis. Yet, due to the difficulty of estimating the polynomials from data, the method is sensitive to noise and is computationally very demanding. This strategy was extended in (Rao et al. 2010b), yielding more robust solutions but only for low dimensional input data and a reduced number of subspaces. A Lipschitz monotonic function was assumed to model the low-rank matrix in (Ganti et al. 2015), even though this cannot cover the case of multiple subspaces. Another family of solutions proposed solving completion and clustering as a two-stage problem (Yang et al. 2015), by first obtaining a similarity graph for clustering and then applying low-rank matrix completion to each cluster. While this is an interesting direction, the solution proposed in (Yang et al. 2015) is prone to fail when subspaces intersect or when the initial grouping is incorrect. To solve this limitation, Elhamifar (2016) has proposed self-expressive models for simultaneous clustering and completion of incomplete data. Along the same line, Fan and Chow (2017) have recently presented a sparse representation to solve the problem. However, these approaches can only cluster the data based on one single criterion. In parallel, some works have relied on neural networks to learn temporal clustering (Yang et al. 2016) and infer missing entries (Nguyen et al. 2019; Zheng et al. 2016), but solving just a single problem. In all cases, these approaches propose to exploit a loss function as we do in this paper, but they require a large amount of training data to learn the model and demand a specific hardware to complete the training step. Unfortunately, this cannot be assumed for generic scenarios, where an unknown number of unknown object typologies can deform, move, and even interact between them, doing the process of simultaneously obtaining training data for track completion, spatial groups and temporal ones very hard and expensive in practice. In contrast, our formulation can solve the problem in just few seconds in a commodity computer, without requiring sophisticated hardware, nor prior knowledge about the scenario to be solved. Moreover, none of them simultaneously solve multiple clustering and completion as we propose in this paper.

Our Contributions. We go beyond previous works by proposing an efficient and robust method that does not require initialization, and it can jointly perform two types of clustering (spatial and temporal), while recovering missing entries and filtering the rest. To the best of our knowledge, no previous approach has jointly addressed the three problems in a unified and unsupervised framework. To this end, we assume the input data to lie in a dual union of low-rank subspaces, where no a priori knowledge about the dimensionality of the subspaces or which data points belong to which subspace is required. It is worth noting that our approach does not require any training data at all. Additionally, the proposed solution can handle situations with complex motion patterns, affected by large degrees of overlapping and percentage of missing entries, in a completely unsupervised manner.

Table 1 summarizes a qualitative comparison of our approach and the aforementioned techniques to jointly solve completion and clustering.

3 Preliminaries and Problem Statement

Notation. Matrices are represented with boldface uppercase letters, e.g., $\mathbf{X}$. In particular, $\mathbf{I}_A$ is used to denote the identity matrix of size $A \times A$, and $\mathbf{1}_A$ a column-vector of ones of size $A \times 1$. The entries of matrices are denoted by means of subscripts $[\cdot]$. For instance, $\mathbf{X}_{[:j]}$ corresponds to the j-th column of the matrix $\mathbf{X}$, $\mathbf{X}_{[i:]}$ is the i-th row of the matrix

$\mathbf{X}$, and $\mathbf{X}_{[ij]}$ indicates its (i,j)-th entry. We also define two types of products: $\mathbf{X} \otimes \mathbf{Z}$ to denote the Kronecker product, and $\mathbf{X} \odot \mathbf{Z}$ to denote the Hadamard (or element-wise) one. The negative of a binary matrix $\mathbf{X}$ is denoted as $\bar{\mathbf{X}}$. We also define several norms on matrices: the l_∞-norm is defined as $\|\mathbf{X}\|_\infty = \max_{(i,j)} |\mathbf{X}_{[ij]}|$, and the $l_{2,1}$-norm as $\|\mathbf{X}\|_{2,1} = \Sigma_j \|\mathbf{X}_{[:j]}\|_2$, where the l_2-norm of a vector is denoted by $\|\mathbf{X}_{[:j]}\|_2$. The Frobenius and nuclear norms are represented as $\|\mathbf{X}\|_F$ and $\|\mathbf{X}\|_* = \Sigma_i \sigma_i(\mathbf{X})$, respectively, with $\sigma_i(\mathbf{X})$ being the i-th singular value of the matrix $\mathbf{X}$. Finally, the Euclidean inner product between two matrices is denoted as $\langle \mathbf{X}, \mathbf{Z} \rangle = \text{tr}(\mathbf{X}^\top \mathbf{Z})$, where tr$(\cdot)$ represents the trace of a matrix.

3.1 Problem Formulation

Let us consider F temporal subspaces $\{S_f\}_{f=1}^F$ of dimension $\{d_f > 0\}_{f=1}^F$ in a C-dimensional space, and G spatial subspaces $\{S_g\}_{g=1}^G$ of dimension $\{d_g > 0\}_{g=1}^G$ in a H-dimensional space. Let $\mathbf{Y} \in \mathbb{R}^{C \times T}$ be a matrix of T data points lying on the union of the temporal subspaces, and $\hat{\mathbf{Y}} \in \mathbb{R}^{H \times N}$ a matrix of N data points lying on the union of spatial subspaces. If we assume that both dimensions can be factorized by a factor D (i.e., $C = DN$ and $H = DT$), the two matrices $\mathbf{Y}$ and $\hat{\mathbf{Y}}$ can then contain exactly the same number of values but in a different arrangement. Additionally, we will assume that only some entries of these matrices are observed, i.e., some locations can include null values. To denote this, we include the matrix $\tilde{\mathbf{Y}}$, a sparse version of $\mathbf{Y}$, in which non-observed entries are set to zero.

Our problem consists in, given an incomplete and noisy matrix $\tilde{\mathbf{Y}}$ of data points from motion tracking, retrieving the full matrix, $\mathbf{Y}$ or $\hat{\mathbf{Y}}$, and clustering the data into the underlying temporal and spatial subspaces. To this end, we will encode the spatial and temporal subspaces using affinity matrices. It is worth noting that both the bases ($\{S_f\}_{f=1}^F$ and $\{S_g\}_{g=1}^G$) and dimensions of each subspace (d_f and d_g, respectively) are not known a priori, nor to which cluster each data point belongs to. The incomplete and noisy input matrix can be provided by any tracking algorithm, by considering, for instance, optimization (Hare et al. 2011; Jia et al. 2012) or deep-learning approaches (Joo et al. (2018)). We next describe our unsupervised and unified approach that can solve the problem without requiring any training data at all.

4 Spatio-Temporal Subspace Clustering

Drawing inspiration on the ideas of (Agudo and Moreno-Noguer 2017b) for reconstructing non-rigid shapes, we next generalize a spatio-temporal constraint for joint motion-track matrix completion and clustering. Note that this constraint was not used previously in the literature for completing missing entries as we present here. We first introduce the two types of interpretations of the tracking matrices we shall use. After that, and considering the previous interpretations, we will introduce the temporal and spatial constraints, extending our formulation to handle missing tracks.

4.1 Motion Tracking Matrix Interpretations

Let us consider a dynamic set of N D-dimensional points tracked along T time instances. For the particular case of $D = 3$, i.e., a tridimensional space, we shall denote by $\mathbf{x}_i^t = [x_i^t, y_i^t, z_i^t]^\top$ the spatial coordinates of the i-th point at time instant t. All acquired point coordinates can be collected into the matrix $\mathbf{Y} \in \mathbb{R}^{DN \times T}$ in an unordered manner in terms of any type of grouping, that stores the x, y, and z coordinates in a block matrix form as:

$$\mathbf{Y} = \begin{bmatrix} x_1^1 & \dots & x_N^1 & y_1^1 & \dots & y_N^1 & z_1^1 & \dots & z_N^1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_1^T & \dots & x_N^T & y_1^T & \dots & y_N^T & z_1^T & \dots & z_N^T \end{bmatrix}^\top .$$

We could assume the previous motion tracking matrix admits a low-rank decomposition of rank K ($K = 1$ for rigid objects), where K represents the number of bases in a single subspace. We know from the structure from motion theory this matrix is low-rank (Dai et al. (2012); Xiao et al. (2006)), but since no information about the motion is assumed, only a low-rank constraint can be considered. However, as discussed above, the single low-rank assumption may not have sufficient expressiveness power to model complex motion patterns of multi-object tracks. It is worth mentioning that if we know some kind of clustering or grouping of the T data points, we might handle this situation by enforcing the low-rank assumption to every particular cluster. In this work, however, the number and type of clusters is not known a priori, making the problem more challenging and generic. Consequently, we need to jointly solve for completion and clustering, without assuming any information about the dimensionality of the subspaces.

Since each column of the matrix $\mathbf{Y}$ encodes all points at a time instant, this matrix cannot be directly used to retrieve spatial similarities. To address this limitation, we consider a new $DT \times N$ matrix $\hat{\mathbf{Y}}$, for which each column stores the point tracks. Following the previous case of $D = 3$, this

matrix can be written as[1]:

$$\hat{\mathbf{Y}} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_N^1 \\ y_1^1 & y_2^1 & \dots & y_N^1 \\ z_1^1 & z_2^1 & \dots & z_N^1 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^T & x_2^T & \dots & x_N^T \\ y_1^T & y_2^T & \dots & y_N^T \\ z_1^T & z_2^T & \dots & z_N^T \end{bmatrix},$$

that is also low-rank as $\mathbf{Y}$ but differing in value.

Both matrices use two different matrix arrangements of the data points, but they include exactly the same information. We can map from $\mathbf{Y}$ matrix to $\hat{\mathbf{Y}}$ using the relation:

$$\hat{\mathbf{Y}} = (\mathbf{I}_D \otimes \mathbf{Y}^\top)\mathbf{A}, \tag{1}$$

where $\mathbf{A}$ is a $(D^D N) \times N$ binary matrix. The inverse mapping can be written as:

$$\mathbf{Y} = (\hat{\mathbf{Y}}^\top \otimes \mathbf{I}_D)\mathbf{B}, \tag{2}$$

where $\mathbf{B}$ is also a $(D^D T) \times T$ binary matrix. Both $\mathbf{A}$ and $\mathbf{B}$ matrices are known a priori, and they can be easily obtained by considering the data structure in data matrices $\hat{\mathbf{Y}}$ and $\mathbf{Y}$.

4.2 Dual Union of Spatio-Temporal Subspaces

The arrangement of the point tracks through the matrices $\mathbf{Y}$ or $\hat{\mathbf{Y}}$ gives two different interpretations, and each of it can be associated to a distinct subspace clustering process. For instance, when analyzing the temporal domain using $\mathbf{Y}$, we can define an affinity matrix to capture the temporal similarities between instances at different time steps. This relation can be written as:

$$\mathbf{Y} = \mathbf{Y}\mathbf{T} + \mathbf{E}_t, \tag{3}$$

where $\mathbf{T}$ encodes a temporal affinity $T \times T$ matrix, and $\mathbf{E}_t$ is a $DN \times T$ residual noise. In this context, the temporal affinity matrix $\mathbf{T}$ measures the similarities between D-dimensional poses along time. Using this relation, we enforce $\mathbf{Y}$ to lie in a union of S_f temporal subspaces, each of them with rank d_f. We could say that the matrix $\mathbf{T}$ is the lowest-rank representation of the data $\mathbf{Y}$ with respect to itself. It is worth noting that $\mathbf{T}$ will be block-diagonal when the data samples have been grouped together in $\mathbf{Y}$ according to the subspace memberships. This block pattern is lost for random entries, obtaining null entries when no affinities are provided. This type of self-expressive model was previously used by (Elhamifar and Vidal 2013; Liu et al. 2013) in the context of subspace clustering.

Similarly, we can analyze the spatial domain through the matrix $\hat{\mathbf{Y}}$, by introducing an affinity matrix associated with a union of spatial subspaces in the presence of noise. In this case, we can write:

$$\hat{\mathbf{Y}} = \hat{\mathbf{Y}}\mathbf{S} + \mathbf{E}_s, \tag{4}$$

where $\mathbf{S}$ encodes a spatial affinity $N \times N$ matrix, and $\mathbf{E}_s$ is a $DT \times N$ residual noise. In this case, we are enforcing $\hat{\mathbf{Y}}$ to lie in a union of S_g spatial subspaces of rank d_g, respectively, measuring the similarities between D-dimensional points in a same time instant. Basically, $\mathbf{S}$ and $\mathbf{T}$ are made of low-rank coefficients that define the union of subspaces in every domain, respectively. Once these affinity matrices are learned from data, off-the-shelf spectral clustering algorithms like (Chen et al. 2010) can be applied on each of them to discover the grouping in every domain. The temporal clustering splits the data into motion primitives, and the spatial one into different object instances.

Nevertheless, the previous formulation requires full measurements on the tracking matrices $\mathbf{Y}$ or $\hat{\mathbf{Y}}$, which is not often the case in real applications. Previous subspace clustering algorithms assume the observation matrices to be complete (Agudo and Moreno-Noguer 2017b; Elhamifar and Vidal 2013; Kanatani 2001; Liu et al. 2013; Zhu et al. 2014). As mentioned above, other approaches (Elhamifar 2016; Fan and Chow 2017) proposed an algorithm to jointly estimate missing entries and build a similarity graph for clustering, when considering a *single* union of temporal subspaces. The algorithm we present in the following section, goes beyond these approaches, and allows solving the matrix completion problem when considering the data to be spanned by *two different* union of subspaces. Our approach can handle high levels of missing entries and noisy measurements, and solve the problem by means of a one-stage optimization algorithm. This means our approach can produce more accurate solutions than competing techniques, while being more general.

5 Motion Tracking Completion and Spatio-Temporal Subspace Clustering

We next present our algorithm to simultaneously recover missing entries and estimate two similarity matrices for computing the spatial and temporal grouping. Note that no prior information nor training data is used at all. The input to our algorithm are incomplete motion tracks of N D-dimensional points observed along T time instances, that are arranged into the matrix $\tilde{\mathbf{Y}}$. In addition, we also introduce an observation matrix $\mathbf{O} \in \mathbb{R}^{N \times T}$ with binary entries that indicate whether the coordinates of a point at a specific time instant are observed or not.

[1] Note that other dimensions could be used, such as 2D motion tracking where $D = 2$. In this case we only need to eliminate the rows of the matrices $\mathbf{Y}$ and $\hat{\mathbf{Y}}$ corresponding to the z component. Theoretically, other dimensions can be also used, such as 1D image sensors.

5.1 Proposed Formulation

Let us denote by $\boldsymbol{\Theta} \equiv \{\mathbf{Y}, \hat{\mathbf{Y}}, \mathbf{T}, \mathbf{S}, \mathbf{E}_t, \mathbf{E}_s\}$ the set of model parameters we have to learn from the input data $\boldsymbol{\Gamma} \equiv \{\tilde{\mathbf{Y}}, \mathbf{O}\}$. We introduce an optimization framework ruled by a cost function that accounts for the spatio-temporal clustering constraints of Eqs. (3)-(4), and enforces the similarity matrices $\mathbf{T}$ and $\mathbf{S}$ to be spanned by low-rank subspaces. Consequently, the combination of both constraints, enforces the data in order to lie in a dual union of subspaces. Indeed, the single union of subspaces model can be seen as a degenerate case of our model (see Remark 1 below).

Since rank minimization is a non-convex NP-hard problem (Recht et al. 2010), the nuclear norm is approximated by its convex relaxation (Chen et al. 2011; Candès and Recht 2008). Additionally, in order to be able to deal with data corrupted by noise and outliers, we use $l_{2,1}$-norm regularization, as the convex relaxation of the $l_{2,0}$-norm (Liu et al. 2010). The objective function can therefore be written as:

$$\begin{aligned}\arg\min_{\boldsymbol{\Theta}} \; & \| \left(\mathbf{1}_D \otimes \mathbf{O}\right) \odot \left(\tilde{\mathbf{Y}} - \mathbf{Y}\right) \|_F^2 + \phi \left(\|\mathbf{T}\|_* + \|\mathbf{S}\|_*\right) \\ & + \gamma \|\mathbf{Y}\|_* + \lambda_t \|\mathbf{E}_t\|_{2,1} + \lambda_s \|\mathbf{E}_s\|_{2,1} \end{aligned} \tag{5}$$

$$\begin{aligned}\text{subject to } & \mathbf{Y} = \mathbf{Y}\mathbf{T} + \mathbf{E}_t \\ & \hat{\mathbf{Y}} = \hat{\mathbf{Y}}\mathbf{S} + \mathbf{E}_s \\ & (\mathbf{I}_D \otimes \mathbf{Y}^\top)\mathbf{A} = \hat{\mathbf{Y}}\end{aligned}$$

where $\{\phi, \gamma, \lambda_t, \lambda_s\}$ are predefined penalty term parameters.

Remark 1: When the data points are not connected in the spatial domain, it means that the affinity matrix $\mathbf{S}$ becomes the identity $\mathbf{I}_N$ (we assume the data points are clean in this domain, i.e., $\mathbf{E}_s = \mathbf{0}$), and hence our formulation degenerates to a union of temporal subspaces. On the other hand, when this occurs in the temporal domain ($\mathbf{T} = \mathbf{I}_T$ and $\mathbf{E}_t = \mathbf{0}$), our formulation degenerates to a union of spatial subspaces.

5.2 Efficient Augmented Lagrangian Multiplier Optimization

The optimization problem in Eq. (5) can be efficiently solved in a unified manner via an ALM method (Lin et al. 2010; Boyd et al. 2011). Without loss of generality, we set $\lambda \equiv \lambda_t \equiv \lambda_s$. In order to reduce the number of parameters and the complexity of the problem while improving convergence, we choose to bring the clustering constraints into the energy function using several Lagrange multipliers with a unique penalty weight $\beta > 0$. In addition, we introduce three support matrices $\mathbf{Y} \equiv \mathbf{M}$, $\mathbf{T} \equiv \mathbf{J}$, and $\mathbf{S} \equiv \mathbf{K}$, to obtain the corresponding augmented Lagrangian function, that can be written as:

$$\begin{aligned}\arg\min_{\boldsymbol{\Theta}_L} \; & \| \left(\mathbf{1}_D \otimes \mathbf{O}\right) \odot \left(\tilde{\mathbf{Y}} - \mathbf{Y}\right) \|_F^2 + \phi \left(\|\mathbf{J}\|_* + \|\mathbf{K}\|_*\right) \\ & + \gamma \|\mathbf{M}\|_* + \lambda \left(\|\mathbf{E}_t\|_{2,1} + \|\mathbf{E}_s\|_{2,1}\right) \\ & + \langle \mathbf{L}_1, \mathbf{Y} - \mathbf{Y}\mathbf{T} - \mathbf{E}_t \rangle + \frac{\beta}{2} \|\mathbf{Y} - \mathbf{Y}\mathbf{T} - \mathbf{E}_t\|_F^2 \\ & + \langle \mathbf{L}_2, \hat{\mathbf{Y}} - \hat{\mathbf{Y}}\mathbf{S} - \mathbf{E}_s \rangle + \frac{\beta}{2} \|\hat{\mathbf{Y}} - \hat{\mathbf{Y}}\mathbf{S} - \mathbf{E}_s\|_F^2 \\ & + \langle \mathbf{L}_3, (\mathbf{I}_D \otimes \mathbf{Y}^\top)\mathbf{A} - \hat{\mathbf{Y}} \rangle + \frac{\beta}{2} \|(\mathbf{I}_D \otimes \mathbf{Y}^\top)\mathbf{A} - \hat{\mathbf{Y}}\|_F^2 \\ & + \langle \mathbf{L}_4, \mathbf{Y} - \mathbf{M} \rangle + \frac{\beta}{2} \|\mathbf{Y} - \mathbf{M}\|_F^2 \\ & + \langle \mathbf{L}_5, \mathbf{T} - \mathbf{J} \rangle + \frac{\beta}{2} \|\mathbf{T} - \mathbf{J}\|_F^2 \\ & + \langle \mathbf{L}_6, \mathbf{S} - \mathbf{K} \rangle + \frac{\beta}{2} \|\mathbf{S} - \mathbf{K}\|_F^2 \end{aligned} \tag{6}$$

where $\boldsymbol{\Theta}_L \equiv \{\mathbf{M}, \mathbf{Y}, \mathbf{J}, \mathbf{T}, \mathbf{K}, \mathbf{S}, \hat{\mathbf{Y}}, \mathbf{E}_s, \mathbf{E}_t\}$ includes the tracking completion, spatio-temporal similarity parameters and residual noises. The Lagrange multipliers are defined as $\{\mathbf{L}_1, \mathbf{L}_4\} \in \mathbb{R}^{DN \times T}$, $\{\mathbf{L}_2, \mathbf{L}_3\} \in \mathbb{R}^{DT \times N}$, $\mathbf{L}_5 \in \mathbb{R}^{T \times T}$ and $\mathbf{L}_6 \in \mathbb{R}^{N \times N}$. Recall that we do not need to know the dimensions nor the bases of the temporal and spatial subspaces a priori, since Eq. (5) automatically selects the appropriate number of data points from every spatio-temporal subspace.

We propose to solve the problem in Eq. (6) by minimizing each variable individually and in closed form, while keeping fixed the rest of model parameters. Algorithm 1 explains the details. The expressions for estimating $\mathbf{Y}$, $\mathbf{T}$, $\mathbf{S}$ and $\hat{\mathbf{Y}}$ (steps 4, 6, 8 and 9) are obtained by computing the derivatives of Eq. (6) in $\mathbf{Y}$, $\mathbf{T}$, $\mathbf{S}$ and $\hat{\mathbf{Y}}$, respectively, and equating to zero. The subproblems to recover $\mathbf{M}$, $\mathbf{J}$, $\mathbf{K}$, $\mathbf{E}_t$ and $\mathbf{E}_s$ are convex and have closed-form solutions. Particularly, for steps 2, 5 and 7, we apply a singular value thresholding minimization (Cai et al. 2010) with a 'shrinkage operator' $S_{\frac{*}{\beta}}(x) = \max(0, x - \frac{*}{\beta})$ where $* = \{\phi, \gamma\}$. In order to optimize the noise terms $\mathbf{E}_t$ and $\mathbf{E}_s$ (steps 10 and 11, respectively), we apply the Lemma 4.1 in (Yang et al. 2009). After each iteration, the Lagrange multipliers are updated according to standard rules as shown in lines 12-13. Additionally, we also update the penalty weight β (step 14) to guarantee the convergence of our algorithm, following the upper bounded requirement of the alternating direction methods. Particularly, we apply a factor of 1.1 to increase β every iteration.

The theoretical convergence of our algorithms is not easy to proof, as the method is based on nine different blocks. However, we have empirically observed that for all experiments reported in the following section, the algorithm always converged in about $190 - 220$ iterations. Additionally, we observe the optimality gap obtained in every iteration to monotonically decrease. An example of this analysis is dis-

Algorithm 1: Algorithm for optimizing Eq. (6).

Input : Incomplete trajectories $\tilde{\mathbf{Y}}$ on a D-dimensional space. Parameters $\{\phi, \gamma, \lambda\}$, and β

Output: Matrix completion $\mathbf{Y}$ (or $\hat{\mathbf{Y}}$), temporal $\mathbf{T}$ and spatial $\mathbf{S}$ affinity matrices for clustering

1 **while** *not converged* **do**

/* Update Model Parameters */

2 $\mathbf{M} = \min \frac{\gamma}{\beta}\|\mathbf{M}\|_* + \frac{1}{2}\|\mathbf{M} - (\mathbf{Y} + \frac{\mathbf{L}_4}{\beta})\|_F^2$

3 $\mathbf{D} = \Big(\mathbf{M} - \frac{\mathbf{L}_4}{\beta} + \Big(((\hat{\mathbf{Y}} - \frac{\mathbf{L}_3}{\beta})^\top \otimes \mathbf{I}_D)\mathbf{B}\Big) + (\mathbf{E}_t - \frac{\mathbf{L}_1}{\beta})(\mathbf{I}_T - \mathbf{T}^\top)\Big)\big((\mathbf{I}_T - \mathbf{T})(\mathbf{I}_T - \mathbf{T}^\top) + 2\mathbf{I}_T\big)^{-1}$

4 $\mathbf{Y} = (\mathbf{1}_D \otimes \mathbf{O}) \odot \Big(\frac{1}{2+\beta}(2\tilde{\mathbf{Y}} + \beta\mathbf{D})\Big) + (\mathbf{1}_D \otimes \bar{\mathbf{O}}) \odot \mathbf{D}$

5 $\mathbf{J} = \min \frac{\phi}{\beta}\|\mathbf{J}\|_* + \frac{1}{2}\|\mathbf{J} - (\mathbf{T} + \frac{\mathbf{L}_5}{\beta})\|_F^2$

6 $\mathbf{T} = (\mathbf{Y}^\top\mathbf{Y} + \mathbf{I}_T)^{-1}(\mathbf{Y}^\top(\mathbf{Y} - \mathbf{E}_t) + \mathbf{J} + \frac{\mathbf{Y}^\top\mathbf{L}_1 - \mathbf{L}_5}{\beta})$

7 $\mathbf{K} = \min \frac{\phi}{\beta}\|\mathbf{K}\|_* + \frac{1}{2}\|\mathbf{K} - (\mathbf{S} + \frac{\mathbf{L}_6}{\beta})\|_F^2$

8 $\mathbf{S} = (\hat{\mathbf{Y}}^\top\hat{\mathbf{Y}} + \mathbf{I}_N)^{-1}(\hat{\mathbf{Y}}^\top(\hat{\mathbf{Y}} - \mathbf{E}_s) + \mathbf{K} + \frac{\hat{\mathbf{Y}}^\top\mathbf{L}_2 - \mathbf{L}_6}{\beta})$

9 $\hat{\mathbf{Y}} = \Big((\mathbf{E}_s - \frac{\mathbf{L}_2}{\beta})(\mathbf{I}_N - \mathbf{S}^\top) + \frac{\mathbf{L}_3}{\beta} + ((\mathbf{I}_D \otimes \mathbf{Y}^\top)\mathbf{A}\Big)\big((\mathbf{I}_N - \mathbf{S})(\mathbf{I}_N - \mathbf{S}^\top) + \mathbf{I}_N\big)^{-1}$

10 $\mathbf{E}_t = \min \frac{\lambda}{\beta}\|\mathbf{E}_t\|_{2,1} + \frac{1}{2}\|\mathbf{E}_t - (\mathbf{Y} - \mathbf{Y}\mathbf{T} + \frac{\mathbf{L}_1}{\beta})\|_F^2$

11 $\mathbf{E}_s = \min \frac{\lambda}{\beta}\|\mathbf{E}_s\|_{2,1} + \frac{1}{2}\|\mathbf{E}_s - (\hat{\mathbf{Y}} - \hat{\mathbf{Y}}\mathbf{S} + \frac{\mathbf{L}_2}{\beta})\|_F^2$

/* Update Lagrange Multipliers */

12 $\mathbf{L}_1 = \mathbf{L}_1 + \beta(\mathbf{Y} - \mathbf{Y}\mathbf{T} - \mathbf{E}_t)$;

13 $\mathbf{L}_2 = \mathbf{L}_2 + \beta(\hat{\mathbf{Y}} - \hat{\mathbf{Y}}\mathbf{S} - \mathbf{E}_s)$

14 $\mathbf{L}_3 = \mathbf{L}_3 + \beta\big((\mathbf{I}_D \otimes \mathbf{Y}^\top)\mathbf{A} - \hat{\mathbf{Y}}\big)$

15 $\mathbf{L}_4 = \mathbf{L}_4 + \beta(\mathbf{Y} - \mathbf{M})$

16 $\mathbf{L}_5 = \mathbf{L}_5 + \beta(\mathbf{T} - \mathbf{J})$

17 $\mathbf{L}_6 = \mathbf{L}_6 + \beta(\mathbf{S} - \mathbf{K})$

/* Update Penalty Weight */

18 $\beta = \min(1.1 \cdot \beta, \beta_{max})$

/* Check Convergence */

19 $\|\mathbf{Y} - \mathbf{Y}\mathbf{T} - \mathbf{E}_t\|_\infty < \epsilon$

20 $\|\hat{\mathbf{Y}} - \hat{\mathbf{Y}}\mathbf{S} - \mathbf{E}_s\|_\infty < \epsilon$

21 $\|(\mathbf{I}_D \otimes \mathbf{Y}^\top)\mathbf{A} - \hat{\mathbf{Y}}\|_\infty < \epsilon$

22 $\|\mathbf{Y} - \mathbf{M}\|_\infty < \epsilon$

23 $\|\mathbf{T} - \mathbf{J}\|_\infty < \epsilon$

24 $\|\mathbf{S} - \mathbf{K}\|_\infty < \epsilon$

25 Setting: In our experiments, we use $\epsilon = 10^{-8}$, and $\beta_{max} = 10^{12}$ since the values in $\mathbf{Y}$ are normalized within the range [-1,1].

played in Fig. 1, where both constraints and full errors in Eq. (6) are represented for a specific case. As it can be seen, after around 50 iterations all constraints are almost perfectly satisfied and the overall energy converges.

5.3 Complexity Analysis

The most computationally demanding parts of Algorithm 1 are the steps 2, 5 and 7, which require computing several SVD operations over matrices of size $DN \times T$, $T \times T$ and $N \times N$, respectively. Hence, our problem can be solved in polynomial time with a computational complexity of at most

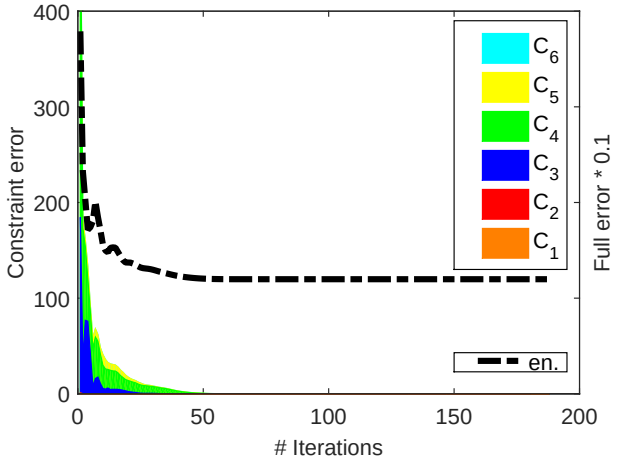

Fig. 1 Convergence analysis: energy reduction as a function of the number of iterations. Evolution of the error for the six constraints (denoted as C_c, with $c = \{1, \ldots, 6\}$) and the full energy in Eq. (6) as a function of the number of iterations until convergence (corresponding to the *Jump* scenario described in the results section). Note that two different scales are used to represent the errors of the constraints (left axis) and the full error (right axis). For visualization purposes, we plot the full energy scaled by a factor of 0.1.

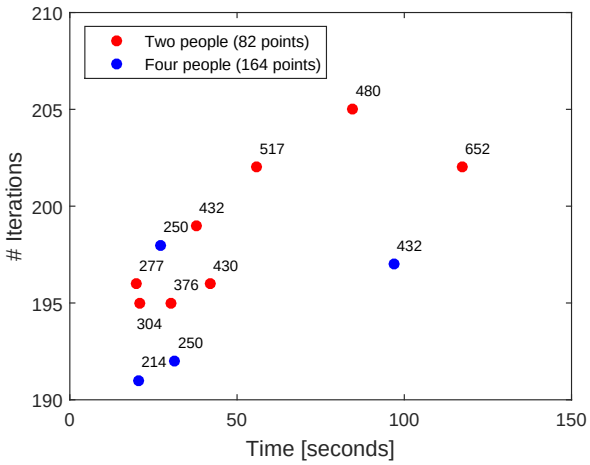

Fig. 2 Computation time as a function of the number of frames, points and iterations. Computation time vs. number of iterations until convergence on the mocap sequences described in the experimental-results section, for two (red dots) and four (blue dots) people. Next to each dot are indicated the number of images of the sequence. In all cases, the number of iterations until convergence always remains within reasonable bounds. The corresponding computation time depends on the number of frames and points.

of $\mathcal{O}(N^2T + T^3 + N^3)$ (Golub and Van Loan 1996). Note that this complexity could be easily reduced by orthogonalizing the columns of the matrices $\mathbf{Y}$ and $\hat{\mathbf{Y}}$. The computation times (in unoptimized Matlab code) on a commodity laptop with an Intel Core i7 processor at 2.4GHz for motion capture sequences for two and four people are displayed in Fig. 2. On average, the median computation time in experiments with sequences between $277 - 652$ frames, and two people ($N = 82$ points) was of 51 seconds. Processing between $214 - 432$ frames, and four people ($N = 164$ points)

required a median time of 44 seconds. In any case, to handle larger datasets we could use current results Yao et al. (2019) on the use of SVD operations on large datasets to address large-scale low-rank problems. This could really help to reduce the reported complexity. Moreover, our formulation could be extended to be employed in a sequential manner, being this a part of our future work.

6 Experimental Results

In this section we report the performance of our algorithm to solve motion tracking completion, as well as temporal and spatial clustering on several challenging datasets. For all cases, we denote by ρ the fraction of missing entries in the input data. In all experiments, we set $\phi = 1.0$, $\gamma = 2.0$ and $\lambda = 0.03$. It is worth pointing out that we do not need fine tuning these parameters, as the results were stable for wide range of values for $\phi \in [0.1, 10]$ and $\gamma \in [0.2, 20]$. Regarding the competing approaches, we will compare our algorithm, denoted as Spatio-Temporal Track Completion (ST2C), with the Low-Rank Matrix Completion (LRMC) (Cai et al. 2010), and the Bilinear Factorization Matrix Completion (BFMC) (Cabral et al. 2013), two approaches where the rank is automatically estimated. We do not include (Ghahramani and Hinton 1996; Balzano et al. 2012) as these methods require knowing the rank of every subspace a priori. Unfortunately, neither can we report the results of (Elhamifar 2016; Fan and Chow 2017) as its source code is not publicly available. Recall, however, that both approaches did only consider a single union of subspaces.

To establish a quantitative evaluation, we will compute three types of errors: the temporal e_{TC} and spatial e_{SC} clustering error as well as the motion tracking completion e_{MTC} (this error is equivalent to a matrix completion evaluation) that are defined as:

$$e_{TC} = \frac{\#\text{Misclassified frames}}{\#\text{All frames}}, \tag{7}$$

$$e_{SC} = \frac{\#\text{Misclassified points}}{\#\text{All points}}, \tag{8}$$

$$e_{MTC} = \frac{\|\mathbf{Y} - \mathbf{Y}_{GT}\|_F^2}{\|\mathbf{Y}_{GT}\|_F^2}. \tag{9}$$

where $\mathbf{Y}_{GT}$ and $\mathbf{Y}$ denote the ground true and the recovered matrices, respectively. For the temporal clustering error, we have obtained the ground truth segmentation over noise-free and complete measurement matrices by applying (Liu et al. 2013) to compute the similarity matrices and (Chen et al. 2010) to obtain the clusters. Spatial ground truth were annotated by hand. This means the evaluation we propose for temporal clustering is actually an implicit comparison with respect to the competing approach (Liu et al. 2013) by assuming clear measurements.

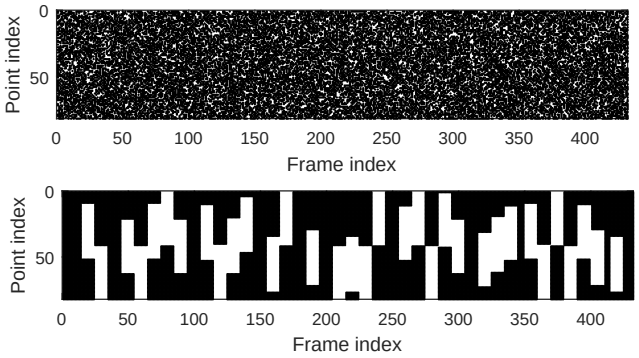

Fig. 3 Patterns of missing entries. $\mathbf{V}$ patterns used to simulate missing entries in the *Jump* sequence. White and black cells denote non-visible and visible points, respectively. **Top:** $\rho = 0.4$ of random missing entries. **Bottom:** $\rho = 0.4$ of structured missing entries.

6.1 Real Experiments on Motion Capture Data

We evaluate the proposed approach on the CMU MoCap dataset. We consider several scenarios with either two or four people interacting and performing complex motions in 3D. On average, the sequences we consider are 433 frames long, and the number of points per frame is either 82 (two people) or 164 (four people). Specifically, we select eight sequences with two people: *23_16 (Jump)*: subjects alternating synchronized jumping jacks; *19_05 (Pull)*: a subject pulls the other by the elbow; *22_20 (Violence)*: a subject picks up high stool and threatens to strike the other; *20_06 (Soldiers)*: subjects follow a soldiers march; *23_19 (Stares Down)*: a subject stares down the other and leans with hands on high stool; *22_12 (Stumbles)*: a person stumbles into the other; *20_09 (Nursery)*: people follow a nursery rhyme; and *22_10 (Shelters)*: a person shelters the other from harm. A total of four sequences with four people are considered, synthetically generated by combining pairs of sequences with two people.

All sequences are corrupted in three different ways: 1) randomly removing a fraction $\rho = \{0.1, \ldots, 0.8\}$ of entries of the measurement matrix $\mathbf{Y}$; 2) removing a structured fraction $\rho = \{0.1, \ldots, 0.4\}$ of entries of the measurement matrix $\mathbf{Y}$ where we emulate temporal self-occlusions or lack of visibility, by including patterns with 50% of structured missing entries per frame; and 3) adding noise to the observed points, according to a Gaussian distribution with standard deviation $\sigma_{noise} = \frac{\tau}{100}\psi$, where τ controls the amount of noise, and ψ represents the maximum distance of a point to the centroid of all the points. An example of these artifacts is shown in Fig. 3, for both random and structured missing entries.

Figures 4 and 5 summarize the results for two and four people, respectively. Each graph depicts the results of all 3 methods for one specific sequence, at increasing levels or missing data for the two types of cases we propose. Solid and dashed lines represent results for noise-free and noisy

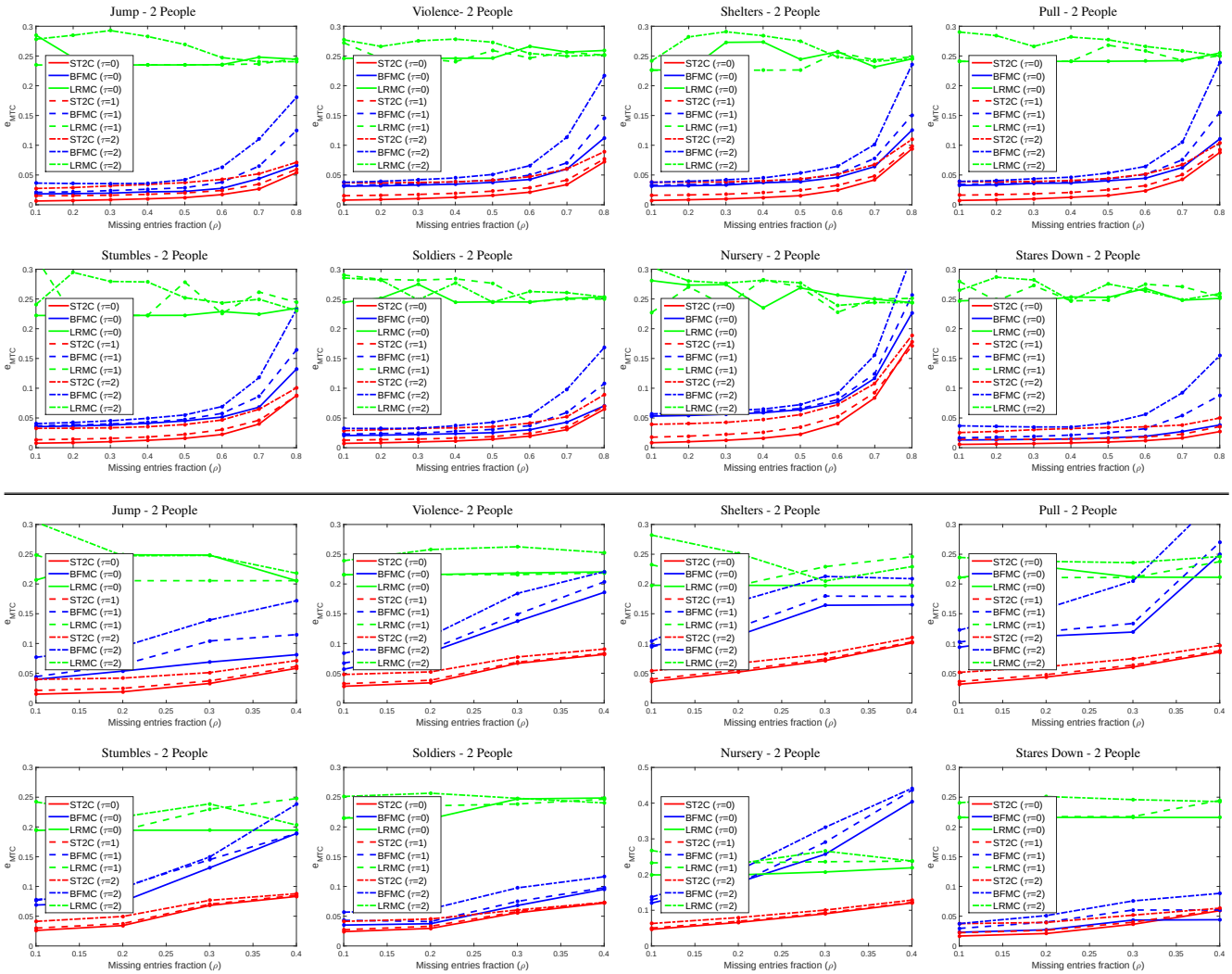

Fig. 4 Motion completion errors of different algorithms as a function of the missing entries rate ρ on motion capture sequences with *two* subjects. Each algorithm is evaluated under noise-free ($\tau = 0$) and noisy ($\tau = \{1, 2\}$) data. For visualization purposes, the error of LRMC has been divided by a factor 3.5 in all graphs. **Top:** Random missing entries. **Bottom:** Structured missing entries.

measurements, respectively. Our approach and BFMC (Cabral et al. 2013) show similar error patterns, even though ours being always consistently better. A breaking point is achieved earlier by BFMC (Cabral et al. 2013), showing our superiority in terms of robustness against this type of artifacts. As it can be seen, our solution by assuming noise can even provide better solutions than the competing approaches for clean annotations. The performance of LRMC (Cai et al. 2010) is far below the other two algorithms. We hypothesize this is due to the *pseudo-block structure* of the missing data, as each missing point does indeed represent three –recall that for this experiment, D=3– adjacent null elements in $\mathbf{Y}$. This is especially relevant when the missing entries are structured, as it can be seen in the bottom part of Figs. 4 and 5. Note also that BFMC (Cabral et al. 2013) and LRMC (Cai et al. 2010) are specifically designed for matrix completion. These algorithms do not provide any kind of affinity measure, that allows subsequent clustering. Some instances for several scenarios when the missing entries are random are displayed in Fig. 6. Moreover, our algorithm is faster than the competing approaches, producing an speed up of $2.7\times$ when BFMC (Cabral et al. 2013) is considered.

As we have commented above, our approach also estimates spatial and temporal clustering. Tables 2 and 3 summarize the mean error for each sequence and all levels of missing data for the random and structured cases, respectively, for noiseless ($\tau = 0$) and noisy ($\tau = \{1, 2\}$) measurements. As it can be seen, our approach produces very good results for most of the sequences, especially in terms of spatial clustering where we obtain an almost negligible clustering error. In fact, our algorithm produces better spatial clustering solutions with artifacts that the provided by LRR (Liu et al. 2013) even assuming full observations (remember that this method needs full data, i.e., $\rho = 0$), as it is

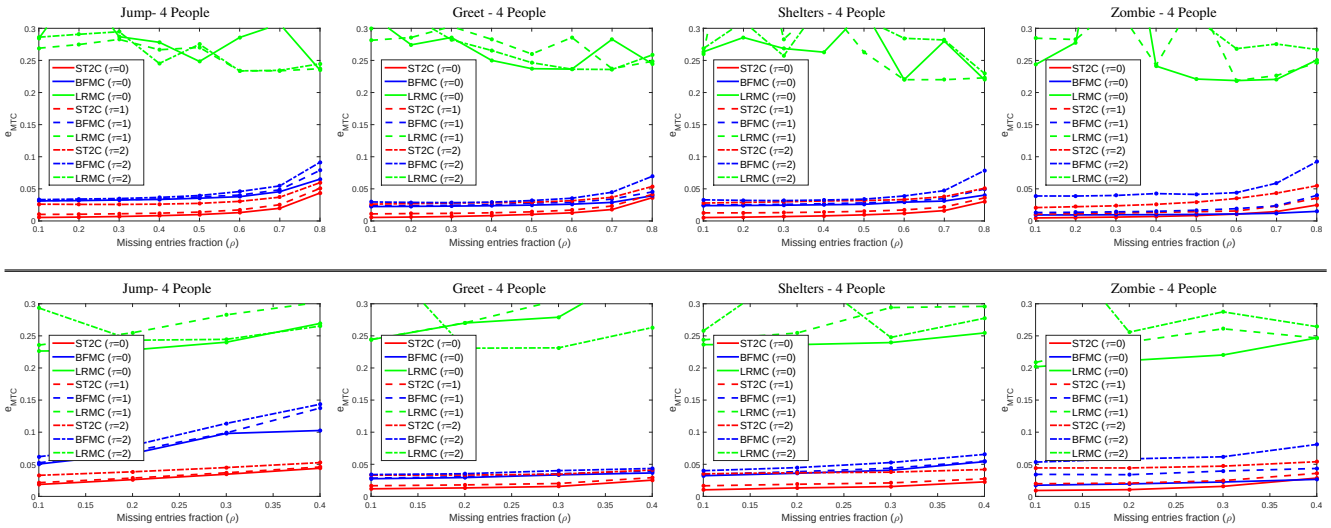

Fig. 5 Motion completion errors of different algorithms as a function of the missing entries rate ρ on motion capture sequences with *four* people. Again, every algorithm is evaluated under noise-free ($\tau = 0$) and noisy ($\tau = \{1, 2\}$) data. The error of LRMC has been divided by a factor 3.5 in all graphs. **Top:** Random missing entries. **Bottom:** Structured missing entries.

	(noise)	Jump	Violence	Shelters	Pull	Stumbles	Soldiers	Nursery	Stares	Jump4	Greet4	Shelters4	Zombie4
e_{MTC}	$\tau = 0$	4.23e-2	5.39e-2	5.98e-2	5.51e-2	5.52e-2	5.77e-2	9.66e-2	3.64e-2	2.70e-2	3.20e-2	3.05e-2	3.16e-2
	$\tau = 1$	4.91e-2	6.17e-2	6.60e-2	6.27e-2	6.06e-2	6.27e-2	10.40e-2	4.23e-2	3.14e-2	3.57e-2	3.62e-2	3.66e-2
	$\tau = 2$	6.62e-2	7.71e-2	8.31e-2	7.91e-2	7.78e-2	7.91e-2	12.70e-2	6.36e-2	4.35e-2	4.99e-2	4.97e-2	5.28e-2
e_{TC}	$\tau = 0$	1.39(2)	2.39(2)	2.51(2)	5.35(2)	1.31(3)	7.95(3)	26.23(2)	0.80(2)	3.94(2)	8.40(2)	4.40(2)	1.87(2)
[%]	$\tau = 1$	2.54(2)	3.45(2)	3.29(2)	5.58(2)	1.65(3)	9.03(3)	31.45(2)	0.93(2)	4.63(2)	8.40(2)	4.80(2)	2.80(2)
	$\tau = 2$	3.01(2)	3.99(2)	3.87(2)	5.81(2)	2.08(3)	10.11(3)	33.90(2)	1.25(2)	4.63(2)	9.60(2)	6.00(2)	2.80(2)
e_{SC}	$\tau = 0$	0.00(2)	0.00(2)	0.00(2)	0.00(2)	0.00(2)	0.00(2)	0.00(2)	0.00(2)	0.00(4)	0.00(4)	1.22(4)	0.00(4)
[%]	$\tau = 1$	0.00(2)	1.22(2)	1.22(2)	1.22(2)	0.00(2)	2.44(2)	1.22(2)	0.00(2)	0.00(4)	0.00(4)	1.22(4)	0.00(4)
	$\tau = 2$	0.00(2)	2.44(2)	3.66(2)	2.44(2)	0.00(2)	3.66(2)	1.22(2)	1.22(2)	0.00(4)	0.61(4)	1.22(4)	0.61(4)
e_{SC}	$\tau = 0$	0.00(2)	2.44(2)	4.88(2)	2.44(2)	50.00(2)	50.00(2)	0.00(2)	0.00(2)	0.00(4)	0.00(4)	16.46(4)	19.51(4)
[%]	$\tau = 1$	0.00(2)	2.44(2)	6.10(2)	2.44(2)	48.78(2)	50.00(2)	0.00(2)	0.00(2)	0.00(4)	17.67(4)	26.22(4)	11.58(4)
LRR ($\rho = 0$)	$\tau = 2$	0.00(2)	2.44(2)	7.31(2)	3.66(2)	48.78(2)	50.00(2)	0.00(2)	0.00(2)	8.53(4)	0.00(4)	18.29(4)	23.17(4)

Table 2 Completion and spatio-temporal clustering results on the CMU dataset as a function of noisy measurements for random missing entries. We represent the median values from $\rho = \{0.1, \ldots, 0.8\}$ for the motion track completion e_{MTC} as well as both temporal $e_{TC}[\%]$ and spatial $e_{SC}[\%]$ clustering errors considering noise-free ($\tau = 0$) and noisy ($\tau = \{1, 2\}$) measurements. In parenthesis it is represented the number of estimated temporal and spatial clusters our approach recovers, respectively. The last part in the table includes the $e_{SC}[\%]$ errors for the baseline LRR Liu et al. (2013) by considering full matrices, i.e., ($\rho = 0$).

	(noise)	Jump	Violence	Shelters	Pull	Stumbles	Soldiers	Nursery	Stares	Jump4	Greet4	Shelters4	Zombie4
e_{MTC}	$\tau = 0$	3.12e-2	5.25e-2	6.50e-2	5.52e-2	5.29e-2	4.54e-2	8.07e-2	3.33e-2	3.08e-2	1.64e-2	1.53e-2	1.59e-2
	$\tau = 1$	3.61e-2	5.56e-2	6.80e-2	5.87e-2	5.57e-2	4.79e-2	8.36e-2	3.74e-2	3.31e-2	2.10e-2	2.11e-2	2.52e-2
	$\tau = 2$	5.09e-2	6.70e-2	7.84e-2	7.09e-2	6.39e-2	5.55e-2	9.29e-2	4.81e-2	4.22e-2	3.55e-2	3.79e-2	4.74e-2
e_{TC}	$\tau = 0$	1.15(2)	2.19(2)	2.66(2)	5.29(2)	1.15(3)	7.22(3)	23.58(2)	0.52(2)	4.16(2)	5.60(2)	3.30(2)	1.52(2)
[%]	$\tau = 1$	1.62(2)	2.86(2)	3.43(2)	5.52(2)	1.32(3)	7.67(3)	25.11(2)	0.62(2)	4.86(2)	5.60(2)	3.80(2)	1.87(2)
	$\tau = 2$	2.08(2)	2.39(2)	3.87(2)	6.05(2)	1.97(3)	7.22(3)	27.61(2)	0.63(2)	4.40(2)	5.20(2)	4.40(2)	2.34(2)
e_{SC}	$\tau = 0$	0.00(2)	0.00(2)	0.00(2)	0.00(2)	0.00(2)	0.00(2)	0.00(2)	0.00(2)	0.00(4)	0.00(4)	0.61(4)	0.00(4)
[%]	$\tau = 1$	0.00(2)	2.44(2)	1.22(2)	1.22(2)	0.00(2)	2.44(2)	1.22(2)	0.00(2)	0.00(4)	0.00(4)	1.83(4)	0.61(4)
	$\tau = 2$	0.00(2)	2.44(2)	3.66(2)	3.66(2)	0.00(2)	3.66(2)	2.44(2)	1.23(2)	0.00(4)	0.61(4)	1.22(4)	0.61(4)

Table 3 Completion and spatio-temporal clustering results on the CMU dataset as a function of noisy measurements for structured missing entries. We represent the median values from $\rho = \{0.1, \ldots, 0.4\}$ (recall that patterns with 50% of missing entries were simulated) for the motion track completion e_{MTC} as well as both temporal $e_{TC}[\%]$ and spatial $e_{SC}[\%]$ clustering errors considering noise-free ($\tau = 0$) and noisy ($\tau = \{1, 2\}$) measurements. In parenthesis it is represented the number of estimated temporal and spatial clusters our approach recovers, respectively.

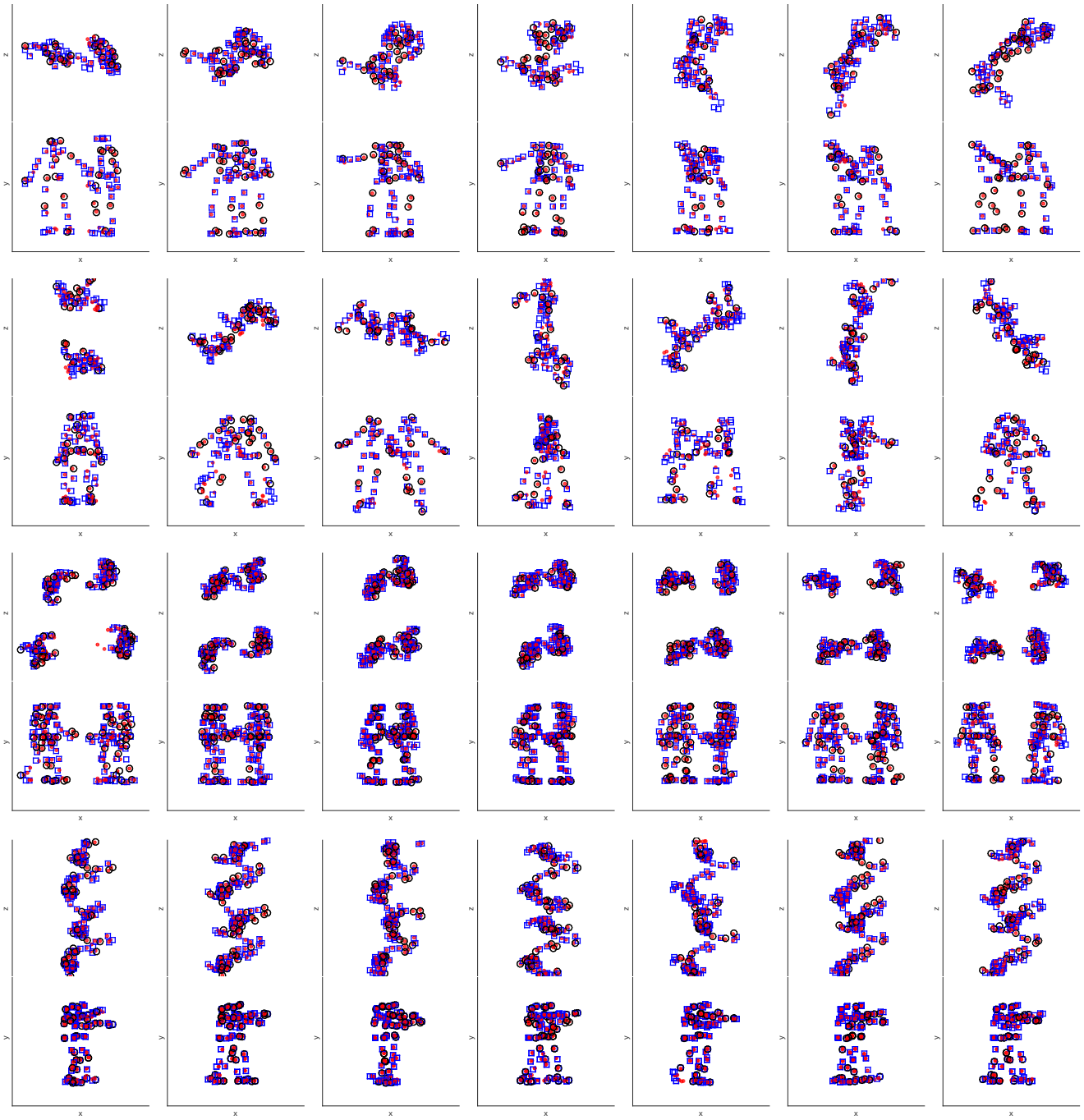

Fig. 6 3D motion track completion on multi-body scenarios, assuming missing entries ($\rho = 0.7$) and noisy measurements ($\tau = 1$). The sequences in order of appearance (from top to bottom) are: *Shelters*, *Nursery*, *Greet4* and *Zombie4*, respectively. For everyone, several instant frames are represented from two orthogonal viewpoints (z-x and y-z). 3D Ground truth is represented by circles and squares, where the color denoted if a point is visible (black circles) or not (blue squares). We represent our motion track completion by means of red dots. Observe that even for high levels of missing entries, our algorithm produces an accurate and clean completion. Although it is not represented in this figure, it is worth pointing out that our algorithm also recovers the spatio-temporal segmentation, even for large degrees of overlapping between the bodies, as it can be seen in the displayed scenarios. Best viewed in color.

observed in table 2. For temporal clustering, our solution is implicitly compared with respect to LRR Liu et al. (2013), showing consistent solutions as a function of the level of noisy. For both types of artifacts, the worst results are obtained for the sequences *Shelters* and *Nursery*, since the type of motion does not include many deformation cycles. In any case, even for these complex motions, our algorithm provides a good trade-off between accuracy and computational cost.

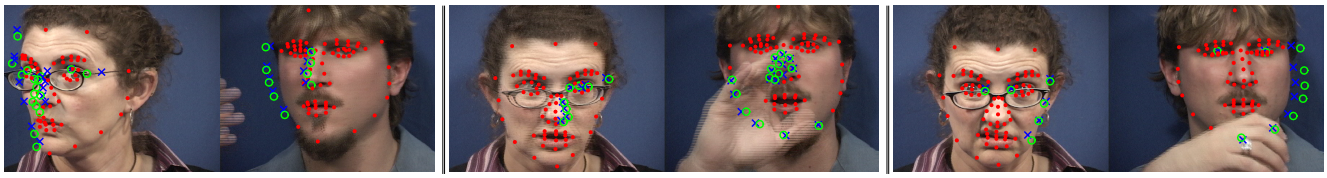

Fig. 7 2D Tracking completion for the ASL dataset. Results for three frames of the sequence. For every image, visible 2D tracking data are shown as red dots. To complete the non-visible tracks, we use our algorithm (blue crosses) and the low-trajectory-rank approach of (Gotardo and Martinez 2011) (green circles). Qualitative results show that our approach provides more accurate track completion for this challenging experiment. Best viewed in color.

6.2 Real Experiment on ASL Tracking Completion

We now consider the completion of time-series trajectories from a real monocular video. We use two American Sign Language (ASL) sequences (Gotardo and Martinez 2011) of 114 image frames, where 77 feature points per sequence are tracked. For the purpose of evaluating the spatial clustering ability of our algorithm, we merge the frames of the two sequences to render a unique video with two faces (with N=154 feature points). The face tracks are corrupted by missing entries (corresponding to $\rho = 0.1445$) due to partial occlusions produced by one or two hands (self-occlusion), or by the face self-rotation causing lack of visibility.

Results are shown in Fig. 7. We compare against (Gotardo and Martinez 2011), a completion algorithm that estimates missing tracks enforcing low-rank trajectory models. Note that this approach requires fine-tuning the rank of the subspace a priori, producing very different solutions when this is done. We use the rank value provided by the authors. For the non-visible points there is no ground truth, but from a qualitative inspection we observe our approach to be remarkably more accurate (see for instance the right-most frame of Fig. 7). We may nevertheless measure the accuracy of the estimated position for the visible points (red dots). For these, our method provides a solution 2.35 times more accurate than that obtained by (Gotardo and Martinez 2011) without assuming any rank knowledge.

6.3 Real Experiment on Multi-fish Data

Finally, we consider a very challenging multi-fish real sequence taken from the DAVIS dataset Perazzi et al. (2016). Particularly, this is a sequence of 51 frames where 33 points per image are tracked. The incomplete tracks are provided by hand, obtaining a level of missing entries of $\rho = 0.129$ (as a combination of random and structured missing tracks), due mainly to multiple partial occlusions produced by the dynamic motion of the animals. A qualitative evaluation of our algorithm is displayed in Fig. 8. As it can be seen, our algorithm can accurately recover the missing tracks without assuming any extra information about the type of observed scene, such as the number of objects, the type of deformations, or the rank of every subspace.

7 Conclusion

We have proposed an algorithm for simultaneous motion track completion and clustering based on two different criteria. For this purpose, we have devised a model that allows to jointly enforce the entries of the matrix to lie in a dual union of subspaces. This goes beyond state-of-the-art solutions, which were restricted to single union of subspaces. Using the machinery of the augmented Lagrange multipliers we have obtained an efficient solution to the problem, and applied it to the case of input data obtained from motion capture systems of multiple human motion, and to challenging real videos. Extensive evaluation demonstrates the ability of our approach to recover missing tracks, and segment input data into each of the objects being captured, and automatically discovering their motion primitives. Further theoretical analysis of the algorithm and convergence proofs will be investigated in the future. Moreover, we pretend to extend our formulation for sequential estimation as the data arrive.

Acknowledgments

This work has been partially supported by the Spanish State Research Agency through the María de Maeztu Seal of Excellence to IRI MDM-2016-0656, by the Spanish Ministry of Science and Innovation under project HuMoUR TIN2017-90086-R and the Salvador de Madariaga grant PRX19/00626, and by the ERA-net CHIST-ERA project IPALM PCI2019-103386.

References

Agudo A, Moreno-Noguer F (2015) Learning shape, motion and elastic models in force space. In: International Conference on Computer Vision, pp 756–764

Agudo A, Moreno-Noguer F (2017a) Combining local-physical and global-statistical models for sequential deformable shape from motion. International Journal of Computer Vision 122(2):371–387

Agudo A, Moreno-Noguer F (2017b) DUST: Dual union of spatio-temporal subspaces for monocular multiple ob-

Fig. 8 2D Tracking completion for the multi-fish sequence. Results for four frames of the sequence. For every image, visible 2D tracking data and hallucinated non-visible tracks by our algorithm are displayed as red dots and blue crosses, respectively. As it can be seen, our algorithm produces physically-aware estimations on this experiment. Best viewed in color.

ject 3D reconstruction. In: Computer Vision and Pattern Recognition, pp 1513–1521

Agudo A, Moreno-Noguer F (2018) Force-based representation for non-rigid shape and elastic model estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence 40(9):2137–2150

Akhter I, Simon T, Khan S, Matthews I, Sheikh Y (2012) Bilinear spatiotemporal basis models. TOG 31(2):17:1–17:12

Balzano L, Szlam A, Recht B, Nowak R (2012) K-subspaces with missing data. In: Statistical Signal Processing Workshop, pp 612–615

Bhojanapalli S, Jain P (2013) Universal matrix completion. In: International Conference on Machine Learning, pp 1881–1889

Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine Learning 3(1):1–122

Cabral R, de la Torre F, Costeira JP, Bernardino A (2013) Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition. In: International Conference on Computer Vision, pp 2488–2495

Cai J, Candes E, Shen Z (2010) A singular value thresholding algorithm for matrix completion. SIAM Journal on Optimization 20(4):1956–1982

Candès EJ, Plan Y (2010) Matrix completion with noise. IEEE Journals and Magazines 99(6):925–936

Candès EJ, Recht B (2008) Exact matrix completion via convex optimization. Foundations of Computational Mathematics 9(6):717

Chen WY, Song Y, Bai H, Lin C, Chang E (2010) Parallel spectral clustering in distributed systems. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(3):568–586

Chen Y, Xu H, Caramanis C, Sanghavi S (2011) Robust matrix completion with corrupted columns. In: International Conference on Machine Learning, pp 873–880

Chiang KY, Hsieh CJ, Dhillon IS (2015) Matrix completion with noisy side information. In: Neural Information Processing Systems, pp 3447–3455

Dai Y, Li H, He M (2012) A simple prior-free method for non-rigid structure from motion factorization. In: Computer Vision and Pattern Recognition

Elhamifar E (2016) High-rank matrix completion and clustering under self-expressive models. In: Neural Information Processing Systems, pp 73–81

Elhamifar E, Vidal R (2013) Sparse subspace clustering: Algorithm, theory, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(11):2765–2781

Eriksson B, Balzano L, Nowak R (2012) High-rank matrix completion and subspace clustering with missing. In: International Conference on Artificial Intelligence and Statistics

Fan J, Chow T (2017) Sparse subspace clustering for data with missing entries and high-rank matrix completion. Neural Networks 93(9):36–44

Ganti R, Balzano L, RWillett (2015) Matrix completion under monotonic single index models. In: Neural Information Processing Systems, pp 1873–1881

Ghahramani Z, Hinton GE (1996) The EM algorithm for mixtures of factor analyzers. Technical Report CRG-TR-96-1, Univ of Toronto

Golub G, Van Loan C (1996) Matrix computations. Johns Hopkins Univ Pr

Gotardo PFU, Martinez AM (2011) Computing smooth time-trajectories for camera and deformable shape in structure from motion with occlusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(10):2051–2065

Hare S, Saffari A, Torr P (2011) Struck: Structured output tracking with kernels. In: International Conference on Computer Vision, pp 263–270

Ionescu C, Papava D, Olaru V, Sminchisescu C (2014) Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. IEEE Transactions on Pattern Analysis and Machine Intelligence 36(7):1325–1339

Jia X, Lu H, Yang M (2012) Visual tracking via adaptive structural local sparse appearance model. In: Computer Vision and Pattern Recognition, pp 1822–1829

Joo H, Simon T, Sheikh Y (2018) Total capture: A 3D deformation model for tracking faces, hands, and bodies. In: CVPR

Kanatani K (2001) Motion segmentation by subspace separation and model selection. In: International Conference on Computer Vision, pp 586–591

Knott M, Bartholomew D (1999) Latent variables models and factor analysis. London: Edward Arnold

Lin Z, Chen M, Ma Y (2010) The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. Mathematical Programming

Liu G, Yan S (2011) Latent low-rank representation for subspace segmentation and feature extraction. In: International Conference on Computer Vision, pp 1615–1622

Liu G, Lin Z, Yu Y (2010) Robust subspace segmentation by low-rank representation. In: International Conference on Machine Learning, pp 663–670

Liu G, Lin Z, Yan S, Sun J, Yu Y, Ma Y (2013) Robust recovery of subspace structures by low-rank representation. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(1):171–184

Ma Y, Derksen H, Hong W, Wright J (2007) Segmentation of multivariable mixed data via lossy data coding and compression. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(9):1546–1562

Ma Y, Yang A, Derksen H, Fossum R (2008) Estimation of subspace arrangements with applications in modeling and segmenting mixed data. SIAM review 50(3):413–458

Nguyen DM, Calderbank R, Deligiannis N (2019) Geometric matrix completion with deep conditional random fields. IEEE Transactions on Neural Networks and Learning Systems

Perazzi F, Pont-Tuset J, McWilliams B, Van Gool L, Gross M, Sorkine-Hornung A (2016) A benchmark dataset and evaluation methodology for video object segmentation. In: CVPR

Rao S, Tron R, Vidal R, Ma Y (2010a) Motion segmentation in the presence of outlying, incomplete or corrupted trajectories. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(10):1832–1845

Rao S, Yang A, Sastry S, Ma Y (2010b) Robust algebraic segmentation of mixed rigid-body and planar motions in two views. International Journal of Computer Vision 88(3):425–446

Recht B, Fazel M, Parrilo PA (2010) Guaranteed minimum-rank solu- tions of linear matrix equations via nuclear norm minimization. SIAM review 52(3):471–501

Sui Y, Zhao X, Zhang S, Yu X, Zhao S, Zhang L (2015) Self-expressive tracking. Pattern Recognition 48(9):2872–2884

Sui Y, Wang G, Tang Y, Zhang L (2016) Tracking completion. In: European Conference on Computer Vision, pp 194–209

Tipping M, Bishop C (1999a) Probabilistic principal component analysis. Journal of the Royal Statistical 21(3):611–622

Tipping ME, Bishop CM (1999b) Mixtures of probabilistic principal component analysers. Neural Computing 11(2):443–482

Van der Aa NP, Luo X, Giezeman GJ, Tan RT, Veltkamp RC (2011) UMPM benchmark: A multi-person dataset with synchronized video and motion capture data for evaluation of articulated human motion and interaction. In: ICCVW, pp 1264–1269

Wang D, Lu H, Yang M (2013) Least soft-thresold squares tracking. In: Computer Vision and Pattern Recognition, pp 2371–2378

Xiao J, Chai J, Kanade T (2006) A closed-form solution to non-rigid shape and motion. International Journal of Computer Vision 67(2):233–246

Yan J, Pollefeys M (2006) A general framework for motion segmentation: independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In: European Conference on Computer Vision, pp 94–106

Yang C, Robinson D, Vidal R (2015) Sparse subspace clustering with missing entries. In: International Conference on Machine Learning, pp 2463–2472

Yang J, Yin W, Zhang Y, Wang Y (2009) A fast algorithm for edge-preserving variational multichannel image restoration. SIAM Journal on Imaging Sciences 2(2):569–592

Yang J, Parikh D, Batra D (2016) Joint unsupervised learning of deep representations and image clusters. In: CVPR

Yao Q, Kwok JT, Wang T, Liu T (2019) Large-scale low-rank matrix learning with nonconvex regularizers. IEEE Transactions on Pattern Analysis and Machine Intelligence 41(11):2628–2643

Zhang T, Ghanem B, Liu S, Ahuja N (2012) Low-rank sparse learning for robust visual tracking. In: European Conference on Computer Vision, pp 470–484

Zheng Y, Tang B, Ding W, Zhou H (2016) A neural autoregressive approach to collaborative filtering. In: ICML

Zhu Y, Huang D, de la Torre F, Lucey S (2014) Complex non-rigid motion 3D reconstruction by union of subspaces. In: Computer Vision and Pattern Recognition, pp 1542–1549