Decomposition and dictionary learning for 3D trajectories

Quentin Barthélemy, Anthony Larue, Jerome I. Mars

Decomposition and Dictionary Learning for 3D Trajectories

Q. Barthélemy[a,b,*], A. Larue[a], J.I. Mars[b]

[a] *CEA, LIST, Data Analysis Tools Laboratory, Gif-sur-Yvette Cedex, 91191, France*
[b] *GIPSA-lab, DIS, UMR 5216 CNRS, Grenoble INP, Grenoble, 38402, France*

Abstract

A new model for describing a three-dimensional (3D) trajectory is proposed in this article. The studied trajectory is viewed as a linear combination of rotatable 3D patterns. The resulting model is thus 3D rotation invariant (3DRI). Moreover, the temporal patterns are considered as shift-invariant. This article is divided into two parts based on this model. On the one hand, the 3DRI decomposition estimates the active patterns, their coefficients, their rotations and their shift parameters. Based on sparse approximation, this is carried out by two non-convex optimizations: 3DRI matching pursuit (3DRI-MP) and 3DRI orthogonal matching pursuit (3DRI-OMP). On the other hand, a 3DRI learning method learns the characteristic patterns of a database through a 3DRI dictionary learning algorithm (3DRI-DLA). The proposed algorithms are first applied to simulation data to evaluate their performances and to compare them to other algorithms. Then, they are applied to real motion data of cued speech, to learn the 3D trajectory patterns characteristic of this gestural language.

Keywords: 3D motion trajectory, rotation invariant, shift-invariant, Procrustes registration, orthogonal matching pursuit, dictionary learning.

1. Introduction

Different communities (computer vision, signal processing, statistics, robotics and machine learning) deal with 3D rotations. Although with different terminologies, these domains are interested in the same challenges. In 3D space, a time-varying 3D trajectory of N temporal samples is decomposed on elementary patterns, and thus described as the sum of K basis vectors. Different models described thereafter can be considered to study it.

1.1. The models

For computer vision, Bregler et al. [1] described a non-rigid 3D object of P points as N successive instantaneous 3D shapes (or point sets). These shapes are decomposed on a *shape* basis, and this is a common way to analyze 3D data [2, 3, 4, 5]. Recently, the duality between *shape* basis and *trajectory* basis has been shown by Akhter et al. [6]. They described the 3D object of P points as P temporal trajectories of N samples [6]. In our work, we focus on the 3D trajectory of a single point. Thus, the 3D trajectory $y \in \mathbb{R}^{3\times N}$ is defined as:

$$y = \sum_{k=1}^{K} a_k \, f_k \,, \tag{1}$$

where $a_k \in \mathbb{R}^{3\times 1}$ are the coefficients, and $f_k \in \mathbb{R}^{1\times N}$ are the trajectory basis vectors. Then, the trajectory y is the sum of K trajectory basis vectors $\{f_k\}_{k=1}^{K}$. With this model, the discrete cosine transform (DCT) appears to be a well-adapted generic basis to study motion signals [6]. This new *trajectory* model to study 3D data opens many prospects.

In signal processing, a multicomponent temporal signal is described in [7] as the sum of the multicomponent patterns. Considering here the particular case of tricomponent data, a 3D trajectory of N samples is viewed as the sum of K 3D trajectories. The trajectory $y \in \mathbb{R}^{3\times N}$ is defined as:

$$y = \sum_{k=1}^{K} x_k \, \phi_k \,, \tag{2}$$

where $x_k \in \mathbb{R}$ are the coefficients, and $\phi_k \in \mathbb{R}^{3\times N}$ are the 3D patterns. This model is different from the Akhter model. Indeed, in model (1), each unicomponent trajectory θ_k (1D pattern) is multiplied by three coefficients, one by dimension. In model (2), each tricomponent trajectory ϕ_k (3D pattern) is multiplied by a scale factor. Thereby, the trajectory y is viewed as a weighted sum of 3D patterns. The advantage of model (2) is to deal with 3D trajectory patterns $\phi_k \in \mathbb{R}^{3\times N}$ where the three components can be different, contrary to model (1), which has the same pattern on the three components. The differences between model (1), known as the *multichannel* framework, and model (2), known as the *multivariate* framework, are detailed in [7].

In order to be clear, our study deals with tricomponent signals, and not tridimensional ones. Classically, a temporal sig-

*Corresponding author. CEA, LIST, Data Analysis Tools Laboratory, Bat DIGITEO 565 PC 192, Gif-sur-Yvette Cedex, 91191, France. Tel: + 33 1 69 08 84 39.

Email address: q.barthelemy@gmail.com, anthony.larue@cea.fr, jerome.mars@gipsa-lab.grenoble-inp.fr (J.I. Mars)

nal $y \in \mathbb{R}^{3\times N}$ composed of N temporal 3D coordinates is incorrectly called tridimensional. In effect, a such signal is not tridimensional, but tricomponent (either trivariate or trichannel, depending on the model). A tridimensional signal would be $y \in \mathbb{R}^{N_1\times N_2\times N_3}$, such as video frames or other cubic data.

The purpose of this article is to provide a 3D rotation invariant (3DRI) model for 3D trajectories. Thus, a rotation matrix $R_k \in \mathbb{R}^{3\times 3}$ is added to each 3D pattern ϕ_k and model (2) becomes:

$$y = \sum_{k=1}^{K} x_k \, R_k \, \phi_k \, . \tag{3}$$

Each rotation matrix R_k has to be orthogonal, so they have to verify the condition: $R_k R_k^T = Id$ (C_o). This trajectory y is represented as a weighted sum of rotatable 3D patterns. The differences between these three models are illustrated in [8]. As explained in the following paragraph, two problems can be handled on this model: the first one estimates coefficients $\boldsymbol{x} = \{x_k\}_{k=1}^K$ and matrices $\boldsymbol{R} = \{R_k\}_{k=1}^K$ when $\boldsymbol{\Phi}$ is fixed, and the second one estimates the best $\boldsymbol{\Phi} = \{\phi_k\}_{k=1}^K$ from data.

1.2. The organization

This article is divided into two parts: one for 3DRI decomposition extending the previous work [8], and one for 3DRI learning, with both based on the new model (3).

In the first part, we want to solve the estimation of coefficients $\boldsymbol{x}$ and rotation matrices $\boldsymbol{R}$. The problem is expressed as:

$$\min_{\boldsymbol{x},\boldsymbol{R}} \left\| y - \sum_{k=1}^{K} x_k \, R_k \, \phi_k \right\|^2 \quad \text{s.t. } \forall k \in \mathbb{N}_K, R_k R_k^T = Id \, , \tag{4}$$

where $\|.\|$ is the Frobenius norm, $\langle A, B\rangle = \mathrm{Tr}(AB^T)$ is the associated matrix inner product, and $(.)^T$ is the transpose operator. This problem has not been addressed, and we ignore if an analytic solution exists to solve it. It can be viewed as a generalization of the orthogonal Procrustes problem [9, 10, 11, 12], which usually deals with the registration of a single pattern. In our study, the shift-invariant case will be considered hereafter. Using a sparsity constraint, we propose in a first part two non-convex optimizations to solve this more complex problem (shift-invariant case). They are based on the matching pursuit (MP) principle: 3DRI-MP and 3DRI-orthogonal MP (OMP).

In the second part, we are interested in the 3D patterns $\boldsymbol{\Phi}$. The goal is to learn the best basis adapted to the data studied. Algorithms based on expectation-maximization (EM) allow the learning of the basis of the structure-from-motion domain [2, 13]. In signal processing and machine learning communities, a redundant basis, called a dictionary, provides a more efficient representation than a basis [14, 15, 16]: it is more robust to noise, it has more flexibility for matching features in the data, and it allows a more compact representation. Dictionary learning algorithms (DLAs) learn a dictionary that is adapted to the data [17, 14, 18]. Based on model (3), we present the 3DRI-DLA that learns a dictionary composed of trivariate patterns invariant to 3D rotations.

In this article, existing methods to make 3D registration are first presented in Section 2. Then, the context is narrowed in Section 3, with the introduction of the shift-invariant case. The 3D rotation invariant MP and OMP are introduced in Sections 4 and 5. The second part begins with a presentation of the existing methods for learning 3D patterns, in Section 6. The 3D rotation invariant dictionary learning algorithm is explained in Section 7. As validation, experiments on simulation data are shown in Section 8, and on real data of French cued speech in Section 9.

2. 3D decomposition: state of the art

In this section, 3D decomposition problems related to problem (4) are mentioned.

2.1. Rigid 3D registration or orthogonal Procrustes problem

A rigid transformation composed of a 3D rotation R and a spatial translation T is considered here, between the trivariate pattern ϕ and the original signal y. The orthogonal Procrustes problem consists of finding parameters R and T such that:

$$\min_{R,T} \| \, y - R \, \phi - T \, \|^2 \quad \text{s.t. } RR^T = Id \, . \tag{5}$$

Eggert et al. [10] reviewed the main methods that give an analytical solution to this rigid 3D registration problem: singular value decomposition (SVD) [9, 19], unit quaternions [20], orthonormal matrix [21], and dual quaternions [22].

In [11], Gower and Dijksterhuis reviewed multiple different Procrustes problems and many generalizations. However, they did not address our problem in Eq. (4). It is the same for the multiview challenge which reconstructs a 3D object from several overlaping observations taken for different angles [23].

2.2. 3D matching

Compared to problem (5), the spatial translation is not considered here any more, and $\psi(t)$ is a short shiftable pattern (zero-padded to have N samples). The 3D curve matching consists of solving:

$$\min_{R,\tau} \| \, y(t) - R \, \psi(t-\tau) \, \|^2 \quad \text{s.t. } RR^T = Id \, , \tag{6}$$

where τ is the sample shift. This problem is solved by calculating the optimal registration matrix R using a method based on the orthonormal matrix for each sample τ [24]. If there is no shift, the simple rigid 3D registration problem (5) is recovered.

Remark that methods based on rotation invariant shape signatures/features (as curvature or torsion for example) [25, 26, 27] or rotation invariant metrics [28] are not considered here since they do not make the estimation of the rotation matrix. Moreover, they match a 3D curve with an other, and not a 3D curve with a linear combination of 3D patterns.

Finally, note that the iterative closest point (ICP) algorithm [29] allows the matching of two 3D sets, where one is a subset of the other, and thus does not solve Eq. (4).

2.3. Tricomponent decompositions

Different models dealing with tricomponent signals are reviewed here. As indicated in the Introduction, a 3D object can be described as a linear combination of shape basis vectors [1, 4]. Using the dual model (1), the object is now a linear combination of trajectory basis vectors [6]. Note that these two contributions come from the structure-from-motion of the computer vision community, so an orthogonal projection is used to estimate the 3D structure from 2D motions.

Studying signals of length N, Akhter et al. [6] sought compactness for their representations, using only K vectors, with $K \ll N$. However, using PCA or DCT for their decompositions, they ignored how to choose the constant K, and how to select the K vectors among the N possible. So, a manual trade-off (between K and the residual error) is used on an exhaustive search on all possibilities to determinate the optimal vectors [6]. To solve this problem, multichannel sparse approximations [30],[7] could be used.

In the signal processing community, a redundant basis composed of $M > N$ elements is called a dictionary. In this case, elements of the dictionary are not called vectors any more, but atoms. Model (2) was introduced in this domain, and the choice of atoms and coefficient estimations are achieved by multivariate sparse approximation (see Section 3.2). The introduced model (3) allows atoms to rotate but needs an appropriate approximation method to estimate the associated rotation matrices as well.

3. Shift and 3D Rotation Invariant Model

If some decompositions of 3D patterns reviewed previously involve both shapes and trajectories, we insist on the fact that our work concern 3D temporal trajectories. In this section, the context is narrowed: the shift invariance, the sparse approximation, and the shift and 3D rotation invariance are detailed.

3.1. The shift-invariant case

In the shift-invariant case, we want to sparsely code the temporal signal y as a sum of a few short structures, known as kernels, that are characterized independent of their positions. This model is usually applied to time-series data, and it provides a compact kernel dictionary [31].

The L shiftable kernels of the compact dictionary $\mathbf{\Psi}$ are replicated at all of the positions, to provide the M atoms of the dictionary $\mathbf{\Phi}$. The N samples of the signal y, the residual error ϵ, and the atoms ϕ_m are indexed[1] by t. The kernels $\{\psi_l\}_{l=1}^{L}$ can have different lengths. The kernel $\psi_l(t)$ is shifted in the τ samples to generate the atom $\psi_l(t-\tau)$: zero padding is carried out to have N samples. The subset σ_l collects the translations τ of the kernel $\psi_l(t)$. For the few kernels that generate all of the atoms, we have:

$$y(t) = \sum_{m=1}^{M} x_m \, \phi_m(t) + \epsilon(t) \tag{7}$$

$$= \sum_{l=1}^{L} \sum_{\tau \in \sigma_l} x_{l,\tau} \, \psi_l(t-\tau) + \epsilon(t) \, . \tag{8}$$

As a result, the signal y is approximated as a weighted sum of a few shiftable kernels ψ_l.

3.2. Sparse approximation

Due to shift invariance, the dictionary $\mathbf{\Phi}$ is the concatenation of L Toeplitz matrices [32] and is L times overcomplete. Since $M > N$, the dictionary is redundant and the linear system is thus under-determined and has multiple solutions. The introduction of constraints such as *sparsity* allows the solution to be regularized. The sparse approximation selects only K active atoms among the M that are possible, and computes the associated coefficients vector $\boldsymbol{x}$ to have a better approximation of the signal y. One way to formalize the sparse approximation is:

$$\min_{\boldsymbol{x}} \left\| y(t) - \sum_{l=1}^{L} \sum_{\tau \in \sigma_l} x_{l,\tau} \, \psi_l(t-\tau) \right\|^2 \quad \text{s.t.} \quad \|\boldsymbol{x}\|_0 \leq K \, , \tag{9}$$

where $K \ll M$ is a constant, and $\|\boldsymbol{x}\|_0$ is the number of nonzero elements of vector $\boldsymbol{x}$. But this problem is NP-hard [33], and non-convex pursuits tackle it sequentially, such as MP [34]. The OMP [35] assures that coefficients $\boldsymbol{x}$ are the orthogonal projection of the signal over the selected atoms. Using only K active atoms among the M that are possible, sparsity provides the compactness that was so searched for by [6]. From the beginning of Section 3, explanations were given for univariate signals. However, they are extended to trivariate signals by multivariate OMP (M-OMP) [7].

3.3. The shift & 3D rotation invariant case

Now, by combining shift and 3D rotation invariances problems, we obtain the following equation to solve:

$$\min_{\boldsymbol{x},\boldsymbol{R}} \left\| y(t) - \sum_{l=1}^{L} \sum_{\tau \in \sigma_l} x_{l,\tau} \, R_{l,\tau} \, \psi_l(t-\tau) \right\|^2$$

$$\text{s.t.} \quad \|\boldsymbol{x}\|_0 \leq K \;\; \text{and} \;\; \forall l \in \mathbb{N}_L, \forall \tau \in \sigma_l, \; R_{l,\tau} R_{l,\tau}^T = Id \, . \tag{10}$$

More than Eq. (4), Eq. (10) is the real issue that is addressed in this article. Eq. (10) combines Eq. (4), which we ignore if an analytic solution exists, and Eq. (9), which is NP-hard. As briefly introduced in [8], we propose two non-convex optimizations to solve this particularly hard problem.

Eq. (10) has two particular cases already solved: when $K = 1$, the 3D curve matching of Eq. (6) is retrieved; and when each $R_k = Id$, the sparse approximation of Eq. (9) is retrieved, with trivariate signals, and this case is solved by M-OMP. Note that 2DRI-OMP [7] simply tackles Eq. (10) in the 2D case using complexes. The presented article can be viewed as a non-trivial 3D extension based on a generalized Procrustes problem, that explains the names of the methods presented.

[1] Note that $a(t)$ and $a(t - t_0)$ do not represent samples, but the signal a and its translation of t_0 samples.

4. 3D Rotation Invariant Matching Pursuit

In the two following sections, our proposed sparse 3DRI decomposition algorithms are going to be introduced. In this section, we first detail the chosen method for the 3D registration, which will be the core of the introduced algorithm. Then, a nonconvex optimization based on the MP principle is introduced to solve Eq. (10), which is called 3DRI-MP.

4.1. 3D registration by SVD

Registration problem (5) is considered here with a normalized trivariate pattern $\phi \in \mathbb{R}^{3\times N}$, but without spatial translation. Sought parameters are the rotation R and the scale factor x:

$$\min_{x,R} \| y - x\, R\, \phi \|^2 \quad \text{s.t. } RR^T = Id\,. \tag{11}$$

For solving this 3D registration equation, the SVD method is chosen among the other possible methods because it is the cheapest and it simply deals with the particular cases of noise and planar patterns [10]. Introduced by [9] to solve the orthogonal Procrustes problem, the SVD method fails in the particular cases mentioned above. It was finally improved by [19], where it was ensured that R is a rotation ($\det R = 1$) and not a reflection ($\det R = -1$).

The method chosen described in Algorithm 1 is resumed in five steps. After having computed the correlation matrix $M_c = y\phi^T \in \mathbb{R}^{3\times 3}$ (step 1), its SVD is carried out: $(U, \Sigma_1, V) = \text{SVD}(M_c)$ (step 2). Defining matrix Σ_2 such that (step 3):

$$\Sigma_2 = \begin{bmatrix} 1 & & \\ & 1 & \\ & & \det(UV^T) \end{bmatrix}, \tag{12}$$

the optimal rotation is: $R = U\Sigma_2 V^T$ (step 4). The correlation value which provides the scale factor is computed such that: $x = \text{Tr}(R\phi y^T) = \text{Tr}(\Sigma_2\Sigma_1) \geq 0$ (step 5).

Algorithm 1 : $(x, R) = \text{Reg\_SVD}\ (\phi, y)$

1: Correlation Matrix: $M_c \leftarrow y\phi^T$
2: SVD: $(U, \Sigma_1, V) \leftarrow \text{SVD}(M_c)$
3: Matrix Σ_2: $\Sigma_2 \leftarrow \text{diag}(1, 1, \det(UV^T))$
4: Optimal Rotation Matrix: $R \leftarrow U\Sigma_2 V^T$
5: Correlation Value: $x \leftarrow \text{Tr}(\Sigma_2\Sigma_1)$

This registration method is the core of the following algorithms.

4.2. Description of 3DRI-MP

In this section, the 3DRI-MP is going to be explained step by step. A trivariate signal $y \in \mathbb{R}^{3\times N}$ and a dictionary $\mathbf{\Psi}$ of shiftable trivariate kernels are considered. It is important to note that the dictionary is normalized, which means that each kernel is normalized. Given this redundant trivariate dictionary, 3DRI-MP produces a sparse approximation of the signal y described in Algorithm 2.

The initialization (step 1) allocates the studied signal y to the residue ϵ^0. At the current iteration k, the algorithm selects the atom that produces the absolute strongest decrease in the mean square error (MSE) $\left\|\epsilon^{k-1}\right\|^2$. Denoting $\epsilon^{k-1}(t) = x_{l,\tau} R_{l,\tau} \psi_l(t - \tau) + \epsilon^k(t)$, and using the rule of the derivative of a matrix trace [36], we have:

$$\begin{aligned} \frac{\partial \left\|\epsilon^{k-1}(t)\right\|^2}{\partial x_{l,\tau}} &= \frac{\partial \text{Tr}(\epsilon^{k-1}(t)\epsilon^{k-1}(t)^T)}{\partial x_{l,\tau}} \\ &= 2\ \text{Tr}(R_{l,\tau}\psi_l(t-\tau)\ \epsilon^{k-1}(t)^T) = 2\ \left\langle R_{l,\tau}\psi_l(t-\tau), \epsilon^{k-1}(t)\right\rangle\,. \end{aligned} \tag{13}$$

This is thus equivalent to finding the registered atom that is the most correlated to the residue ϵ^{k-1}. The correlation value $x^k_{l,\tau} = \text{Tr}(R^k_{l,\tau}\ \psi_l(t-\tau)\ \epsilon^{k-1}(t)^T)$ is computed for each shift τ, with $R^k_{l,\tau}$ the optimal rotation matrix to register $\psi_l(t-\tau)$ on $\epsilon^{k-1}(t)$. To carry out this step, algorithm Reg_SVD is applied for each τ and each $l = 1..L$ (step 5), and then, the maximum of the values $x^k_{l,\tau} (\geq 0)$ is searched for to select the optimal atom (step 7), which is characterized by its kernel index l^k and its position τ^k. Selected atoms form an active dictionary. The vector $\boldsymbol{x}$ accumulates the active (*i.e.* nonzero) coefficients that are the maximum correlation values (step 8). Associated rotation matrices are grouped in $\boldsymbol{R}$ (step 9) and the current residue is computed (step 10).

Different stopping criteria (step 12) can be used: a threshold on k for the number of iterations, a threshold on the relative root MSE (rRMSE) $\left\|\epsilon^k\right\| / \|y\|$, or a threshold on the decrease in the rRMSE. In the end, the 3DRI-MP provides a K-sparse approximation of y using the K selected active elements:

$$\hat{y}^K = \sum_{k=1}^{K} x^k_{l^k,\tau^k}\, R^k_{l^k,\tau^k}\, \psi_{l^k}(t - \tau^k)\,. \tag{14}$$

Algorithm 2 : $(\boldsymbol{x}, \boldsymbol{R}) = \text{3DRI\_MP}\ (y, \mathbf{\Psi})$

1: **initialization**: $k = 1$, residue $\epsilon^0 = y$, $\boldsymbol{x} = \varnothing$, $\boldsymbol{R} = \varnothing$
2: **repeat**
3: **for** $l \leftarrow 1, L$ **do**
4: 3D Registration for each τ:
5: $(x^k_{l,\tau}, R^k_{l,\tau}) \leftarrow \text{Reg\_SVD}\ (\ \psi_l(t-\tau),\ \epsilon^{k-1}(t)\ )$
6: **end for**
7: Selection: $(l^k, \tau^k) \leftarrow \arg\max_{l,\tau} x^k_{l,\tau}$
8: Active Coefficients: $\boldsymbol{x} \leftarrow \boldsymbol{x} \cup x^k_{l^k,\tau^k}$
9: Active Matrices: $\boldsymbol{R} \leftarrow \boldsymbol{R} \cup R^k_{l^k,\tau^k}$
10: Residue: $\epsilon^k \leftarrow \epsilon^{k-1} - x^k_{l^k,\tau^k} R^k_{l^k,\tau^k} \psi_{l^k}(t - \tau^k)$
11: $k \leftarrow k + 1$
12: **until** stopping criterion

4.3. Comments on the 3DRI-MP

Without considering the nonconvexity of the algorithm, if there is no overlap between the selected atoms, 3DRI-MP gives the orthogonal projection of the signal on the active dictionary in Eq. (10). Otherwise, it is suboptimal, since atom overlaps generate cross terms that are not treated by 3DRI-MP, as observed in [8]. The difference with 3DRI-OMP will be explained below.

Note that the description of Algorithm 2 is detailed in order to be clear, although its complexity is $O(N^2)$. A more rapid implementation is possible. Indeed, each of the nine elements of the N correlation matrices $M_c(l,\tau) = \epsilon^{k-1}(t)\ \psi_l(t-\tau)^T$ can be first computed by fast Fourier transform in $O(N\log N)$ for every τ, and then the N registrations are computed in $O(N)$. The resulting complexity is $O(N\log N)$.

Note also that the method presented can be easily extended to a higher dimension $D > 3$, considering $y \in \mathbb{R}^{D\times N}$. In this case, the physical signification of the orthogonal matrix $R \in \mathbb{R}^{D\times D}$ is obviously lost. The extension, which can be called nDRI-MP [37], modifies only the registration (steps 4-5), extending the definition of the inner variable $\Sigma_2 = \mathrm{diag}(1,\ldots,1,\det(UV^T)) \in \mathbb{R}^{D\times D}$.

5. 3D Rotation Invariant Orthogonal Matching Pursuit

After having introduced 3DRI-MP, we present 3DRI-OMP, its orthogonal version. The similarities and differences are first explained, and the algorithm is then detailed.

5.1. Description of 3DRI-OMP

In 3DRI-MP described in Algorithm 2, the coding coefficient $x^k_{l^k,\tau^k}$ of the selected atom $\psi_{l^k}(t-\tau^k)$ is the optimal correlation value (step 8), and equally for the rotation matrix $R^k_{l^k,\tau^k}$ (step 9). It provides an approximated solution for Eq. (10), although this is not optimal in the least-squares meaning, as it does not take into account the correlation terms due to overlaps between the selected atoms. In 3DRI-OMP, coding coefficients and rotation matrices are computed via the orthogonal projection of the signal y on the selected atoms of the active dictionary. Previous coefficients and matrices values are corrected to take into account the new atom and to give the least-squares solution of Eq. (10). These are modified only if there are correlations between the new atom and the old ones.

3DRI-OMP solves the least-squares Eq. (10) sequentially, by increasing K iteratively (Algorithm 3). 3DRI-OMP described in Algorithm 3 is similar to 3DRI-MP for steps 1 to 10, but computes the least-squares solutions $\boldsymbol{x}$ and $\boldsymbol{R}$ at each iteration k in step 11. This allows better selection at the following iteration $k+1$. Thereafter, the superscript k on variables x_{l^k,τ^k} and R_{l^k,τ^k} is omitted, to lighten notations, and index $\kappa = 1..k$ nominates the different elements that were already selected at iteration k. In 3DRI-OMP, at the current iteration k,

- active coefficients $\boldsymbol{x} = \{x_{l^\kappa,\tau^\kappa}\}_{\kappa=1}^k$
- and active matrices $\boldsymbol{R} = \{R_{l^\kappa,\tau^\kappa}\}_{\kappa=1}^k$

are the solutions of Eq. (15) defined as:

$$\min_{\boldsymbol{x},\boldsymbol{R}} \left\| y(t) - \sum_{\kappa=1}^{k} x_{l^\kappa,\tau^\kappa}\, R_{l^\kappa,\tau^\kappa}\, \psi_{l^\kappa}(t-\tau^\kappa) \right\|^2$$
$$\text{s.t. } \forall \kappa \in \mathbb{N}_k,\ R_{l^\kappa,\tau^\kappa} R_{l^\kappa,\tau^\kappa}^T = Id\,. \qquad (15)$$

The optimization procedure (step 11) that tackles this problem is detailed in the following paragraph.

The stopping criteria (step 14) are the same as for 3DRI-MP. Finally, 3DRI-OMP gives K-sparse approximations using the least-squares coefficients and matrices.

Algorithm 3 : $(\boldsymbol{x},\boldsymbol{R}) = \text{3DRI\_OMP}\ (y, \boldsymbol{\Psi})$

1: **initialization**: $k = 1$, residue $\epsilon^0 = y$, $\boldsymbol{x} = \varnothing$, $\boldsymbol{R} = \varnothing$
2: **repeat**
3: **for** $l \leftarrow 1, L$ **do**
4: 3D Registration for each τ:
5: $(x_{l,\tau}, R_{l,\tau}) \leftarrow \text{Reg\_SVD}\ (\ \psi_l(t-\tau),\ \epsilon^{k-1}(t)\ )$
6: **end for**
7: Selection: $(l^k,\tau^k) \leftarrow \arg\max_{l,\tau} x_{l,\tau}$
8: Active Coefficients: $\boldsymbol{x} \leftarrow \boldsymbol{x} \cup x_{l^k,\tau^k}$
9: Active Matrices: $\boldsymbol{R} \leftarrow \boldsymbol{R} \cup R_{l^k,\tau^k}$
10: Residue: $\epsilon^k \leftarrow \epsilon^{k-1} - x_{l^k,\tau^k} R_{l^k,\tau^k} \psi_{l^k}(t-\tau^k)$
11: Optimization Procedure: $(\boldsymbol{x},\boldsymbol{R}) \leftarrow \arg\min_{\boldsymbol{x},\boldsymbol{R}}$ (15)
12: Residue: $\epsilon^k \leftarrow y - \hat{y}^k$
13: $k \leftarrow k+1$
14: **until** stopping criterion

5.2. Optimization procedure for coefficients and matrices

In this paragraph, the optimization procedure to solve Eq. (15) at step 11 of Algorithm 3 is detailed. Problem (15) is solved by alternating updates on coefficients $\boldsymbol{x}$ and rotation matrices $\boldsymbol{R}$, updating one when the others are fixed. Moreover, each update is based on gradient descent.

The 3DRI-MP solution is a good initialization for this optimization, as it is not so far from the optimal solution of Eq. (15). Thus, the procedure uses the 3DRI-MP solution given by steps 8, 9 and 10 as the initial $\boldsymbol{x}$, $\boldsymbol{R}$ and ϵ^k of the optimization. Both updates are now detailed. Thereafter, a superscript i is added to variables to denote the current optimization procedure iteration.

Update on coefficients $\boldsymbol{x}$

Based on the least mean squares (LMS) method [38], Eq. (15) is derived with respect to x_{l^κ,τ^κ}, and each coefficient is updated such that:

$$x^i_{l^\kappa,\tau^\kappa} = x^{i-1}_{l^\kappa,\tau^\kappa} + \lambda^i_1 \cdot \mathrm{Tr}\big(\ R^{i-1}_{l^\kappa,\tau^\kappa}\, \psi_{l^\kappa}(t-\tau^\kappa)\, \epsilon^k(t)^T\ \big)\,, \qquad (16)$$

where $\lambda^i_1 = 1/i^{0.6}$ is the adaptive descent step. Solving the update on coefficients $\boldsymbol{x}$ with the gradient method is better than giving an exact solution by pseudo-inverse. Indeed, in solving the coefficients update so well in comparison with the matrices, there is the risk of getting the global optimization stuck in a local minimum.

Update on rotation matrices $\boldsymbol{R}$

This update has to maintain the orthogonality of rotation matrices $\boldsymbol{R}$. In the same manner as previously, LMS can be used for matrices, giving an additive update. To guarantee the orthogonality of the updated matrices, a second stage with an orthogonal penalization has to be added [39]. The drawbacks are that this update is empirically less robust to noise and is made in two stages rather than one. So, we choose a multiplicative update

which intrinsically keeps the rotation matrix orthogonal in the orthogonal Stiefel manifold [40, 39]. Each rotation matrix R_{l^κ,τ^κ} is updated following the geodesic of the manifold:

$$R^i_{l^\kappa,\tau^\kappa} = \text{expm}(-\mu G^{i-1}_\kappa)\, R^{i-1}_{l^\kappa,\tau^\kappa}\,, \tag{17}$$

where G^{i-1}_κ is defined in Eq. (18). The constant μ is set up to 0.1 in the following and expm is the matrix exponential function.

$$\begin{aligned} G^{i-1}_\kappa \;=\; x^i_{l^\kappa,\tau^\kappa} \Big(& R^{i-1}_{l^\kappa,\tau^\kappa}\, \psi_{l^\kappa}(t-\tau^\kappa)\, \epsilon^k(t)^T \\ & - \epsilon^k(t)\, \psi_{l^\kappa}(t-\tau^\kappa)^T\, {R^{i-1}_{l^\kappa,\tau^\kappa}}^T \Big). \end{aligned} \tag{18}$$

Alternating optimization

The optimization procedure which solves Eq. (15) at step 11 is resumed:

1: **initialization** of $\boldsymbol{x}$, $\boldsymbol{R}$ and ϵ^k on the 3DRI-MP solution
2: **for** $i \leftarrow 1, I$ **do**
3: **for** $\kappa \leftarrow 1, k$ **do**
4: Update of the coefficient $x^i_{l^\kappa,\tau^\kappa}$ with Eq. (16)
5: **end for**
6: Update of residue ϵ^k
7: **for** $\kappa \leftarrow 1, k$ **do**
8: Update of the matrix $R^i_{l^\kappa,\tau^\kappa}$ with Eq. (17) and update of residue ϵ^k
9: **end for**
10: **end for** .

The number of iterations I can be chosen as constant, or as a function of k (in this way, the last coefficients have more iterations to approach the least-squares solution than the first ones). Note that it is no use to carry out this optimization procedure at the first iteration $k = 1$, as there is no overlap in the active dictionary reduced to a single atom.

Due to non-convexity of the alternating updates, even with enough iterations, this optimization procedure is not guaranteed to converge to the optimal least-squares solution of Eq. (15).

5.3. Comments on the 3DRI-OMP

The first things to note is that 3DRI-OMP does not have a unique implementation. Here, we have presented one possible implementation of the optimization procedure to solve Eq. (15). However, other choices can be made to improve this least-squares optimization.

Note that 3DRI-MP can be viewed as 3DRI-OMP without optimization of the problem (15), *i.e.* setting $I = 0$. This explains why 3DRI-MP is more rapid, but sub-optimal.

To continue the nDRI extension evoked in Section 4.3, the optimization procedure is unchanged to give the nDRI-OMP.

6. 3D Learning: state of the art

After having presented sparse 3DRI decomposition algorithms, in this section, learning methods for 3D objects are reviewed. These aim at learning 3D patterns from a signal set.

6.1. Basis learning

Basis learning methods are mostly based on the expectation-maximization (EM) algorithm [41]. They alternate between an expectation step that estimates the coefficients, and a maximization step that optimizes the vectors. As explained in the Introduction, Bregler et al. described a 3D object as linear combinations of shape basis vectors [1]. As shape basis is dedicated to a studied object, it has to be re-estimated for each new object. Torresani et al. [2] use a generalized EM algorithm to learn 3D shapes that were adapted to the data they studied, and this approach was improved by [4]. Akhter et al. represents a 3D object with trajectory basis vectors in the dual model (1), independent of the studied object; so the generic basis of DCT can be used [6]. However, Su et al. [13] used an EM-like algorithm to learn 1D trajectories adapted to the data studied, and obtained better results than [2] and [6].

With the advantages of dictionary over basis mentioned in Section 1.2, dictionary learning will now be detailed.

6.2. Dictionary learning

Dictionary learning algorithms (DLAs) empirically learn a dictionary that is dedicated to a signal set [17, 14, 42, 31, 18, 43], [7]. These algorithms alternate between two steps: extraction of the main patterns (the sparse decomposition step) that are then learned (the dictionary update step). At the end of the learning, each signal of the set can be approximated sparsely with this dictionary. DLAs are different from EM as they use a sparsity constraint in the decomposition/ expectation step: sparsity makes informative patterns emerge from data. As seen in [7], the EM approach gives less sparsity than DLAs. This learning approach provides adapted dictionaries that can outperform generic ones, such as gammatones, wavelets, DCT, and others [31], [7]. In addition, the advantages of DLA over PCA and ICA are detailed in [14].

Considering a set of trivariate signals that represents instantaneous 3D objects, Zhang et al. learn a shape dictionary [5], but without time-varying aspect. Studying trivariate temporal trajectories, the multivariate DLA (M-DLA) [7] based on model (2) computes a trivariate temporal dictionary. However, there is no rotation between the three components. So, based on model (3), a 3D rotation invariant DLA (3DRI-DLA), which learns a rotatable trivariate temporal dictionary, is now proposed.

7. 3D Rotation Invariant Dictionary Learning Algorithm

In this section, we explain how to compute a 3DRI dictionary from a trivariate training set.

7.1. Description of the 3D rotation invariant dictionary learning algorithm

A training set of trivariate signals $\boldsymbol{Y} = \left\{y_q\right\}_{q=1}^{Q}$ is considered, and the index q is added to the variables. In our learning algorithm, named as the 3DRI-DLA described in Algorithm 4, each training signal y_q is treated one at a time. This is an *online* alternation between two steps: a sparse 3DRI decomposition and a

dictionary update. The sparse 3DRI decomposition (steps 4-5) is carried out by 3DRI-OMP:

$$\boldsymbol{x}_q, \boldsymbol{R}_q = \arg\min_{\boldsymbol{x},\boldsymbol{R}} \left\| y_q(t) - \sum_{l=1}^{L} \sum_{\tau\in\sigma_l} x_{l,\tau}\, R_{l,\tau}\, \psi_l(t-\tau) \right\|^2$$

$$\text{s.t.} \quad \|\boldsymbol{x}\|_0 \leq K \quad \text{and} \quad \forall l \in \mathbb{N}_L, \forall \tau \in \sigma_l,\ R_{l,\tau} R_{l,\tau}^T = Id\ . \tag{19}$$

The dictionary update (steps 6-7) is based on maximum likelihood criterion [17], on the assumption of Gaussian noise:

$$\boldsymbol{\Psi} = \arg\min_{\boldsymbol{\Psi}} \left\| y_q(t) - \sum_{l=1}^{L} \sum_{\tau\in\sigma_l} x_{l,\tau;q}\, R_{l,\tau;q}\, \psi_l(t-\tau) \right\|^2$$

$$\text{s.t.} \quad \forall\, l \in \mathbb{N}_L,\ \|\psi_l\| = 1\ . \tag{20}$$

This criterion is usually optimized by gradient descent [17, 31, 32]. To achieve this optimization, we prefer a stochastic gradient descent (SGD) based on the LMS. It has the particularity to deal with 3D rotation matrices, so we name it 3DRI-LMS and it is derived using the derivative rules of trace matrix [36]:

$$-\frac{\partial \left\| \epsilon_q(t) \right\|^2}{\partial \psi_l} = 2 \sum_{\tau\in\sigma_l} x_{l,\tau;q}\, R_{l,\tau;q}^T\, \underline{\epsilon}_\tau(t)\ , \tag{21}$$

with $\underline{\epsilon}_\tau$ denoting the residue localized at the shift τ and limited to the temporal support of ψ_l, *i.e.* $\underline{\epsilon}_\tau = \epsilon|_{t=\tau..\tau+T_l}$, with T_l the length of ψ_l. The current iteration is denoted as j. For $l = 1..L$, each trivariate kernel ψ_l is updated such that:

$$\psi_l^j(\underline{t}) = \psi_l^{j-1}(\underline{t}) + \lambda_2^j \cdot \sum_{\tau\in\sigma_l} x_{l,\tau;q}^j\, {R_{l,\tau;q}^j}^T\, \epsilon_q^{j-1}(\underline{t}+\tau)\ , \tag{22}$$

where $\underline{t}$ are the indices limited to the ψ_l temporal support, and λ_2 is the adaptive descent step. This is chosen such that $\lambda_2^j = 1/j$. The three components of the trivariate kernel ψ_l are updated simultaneously. Kernels are normalized at the end of each iteration, and their lengths can be modified. They are lengthened if there is some energy in their edges, and they are shortened otherwise.

At the beginning of the algorithm (step 1), the kernels are initialized as white uniform noise (between 0 and 1) and they are normalized. At the end, different stopping criteria (step 10) can be used: a threshold on the rRMSE computed for the whole of the training set, or a threshold on j, the number of iterations.

At the end of the algorithm, elementary patterns that are characteristic of the training set $\boldsymbol{Y}$ have been learned empirically in the optimal dictionary, which jointly gives sparse approximations for all of the signals of this set.

7.2. Comments on the 3D rotation invariant dictionary learning algorithm

During the learning, we observe sometimes that some kernels do not converge, and they are not used in decompositions as they are similar to white noise. Consequently, they are suppressed from the dictionary at the end of the learning.

Algorithm 4 : $\boldsymbol{\Psi}$ = 3DRI_DLA $\left(\left\{y_q\right\}_{q=1}^{Q}\right)$

1: **initialization**: $j = 1, \boldsymbol{\Psi}^0 = \{L$ kernels of white noise$\}$
2: **repeat**
3: **for** $q \leftarrow 1, Q$ **do**
4: Sparse 3DRI Decomposition:
5: $(\boldsymbol{x}_q^j, \boldsymbol{R}_q^j) \leftarrow$ 3DRI-OMP $(y_q, \boldsymbol{\Psi}^{j-1})$
6: Dictionary Update:
7: $\boldsymbol{\Psi}^j \leftarrow$ 3DRI-LMS $(y_q, \boldsymbol{x}_q^j, \boldsymbol{R}_q^j, \boldsymbol{\Psi}^{j-1})$
8: $j \leftarrow j + 1$
9: **end for**
10: **until** stopping criterion

In 3DRI-DLA, the 3DRI-OMP is stopped by a threshold on the number of iterations. We cannot use rRMSE here, because at the beginning of the learning, the kernels of white noise cannot span a given part of the space studied. Moreover, at the first iteration of the 3DRI-DLA ($j = 1$), optimization of Eq. (15) of the 3DRI-OMP (step 11) is not carried out. So, the constant is set up as: $I = j - 1$.

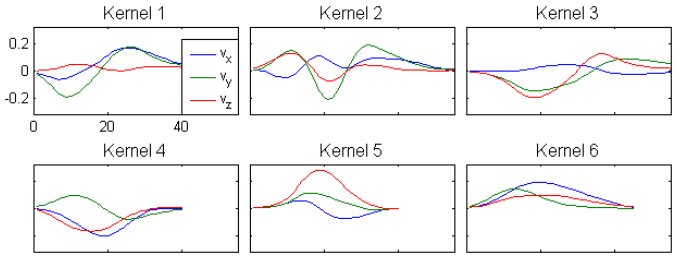

Figure 1: Dictionary of $L = 6$ trivariate kernels at the end of the learning. The solid, dashed and dotted lines represent the three components v_x, v_y and v_z of the velocity kernels. Note that the axes are all the same for each subfigure.

To illustrate this on real data, we apply 3DRI-DLA to trivariate velocity signals to learned an adapted dictionary. In Fig. 1, the dictionary of $L = 6$ trivariate kernels is shown at the end of the learning. The solid, dashed and dotted lines represent, respectively, the three components v_x, v_y and v_z. This convention for the line style will be used henceforth.

To finish the nDRI extension begun in Sections 4.3 and 5.3, Eq. (22) is unchanged for the dictionary update of the nDRI-DLA, using a nDRI-OMP for the sparse decomposition.

8. Experiments on Simulation Data

In this section, the introduced methods are applied to simulation data, and compared to evaluate their performance.

8.1. Experiment 1: 3D rotation invariant decompositions

This first experiment compares the 3DRI-MP and 3DRI-OMP performances. A dictionary $\boldsymbol{\Psi}$ of $L = 50$ trivariate kernels is randomly created: the normalized kernels are drawn from white Gaussian noise. The kernel length is $T = 65$ samples. One hundred signals of $N = 250$ samples are composed of the

sum of $K = 15$ atoms, for which the coefficients (strictly positive), rotation matrices, kernel indices, and shift parameters are randomly drawn on uniform distributions. As a consequence, the different atoms overlap. Each signal is approximated by 3DRI-MP and 3DRI-OMP with $K = 15$ iterations, to recover the 15 simulated atoms. The M-OMP used in a trivariate case is also tested on these signals. A normalized univariate dictionary is computed avering the trivariate one, and the multichannel OMP (Mch-OMP) [30] used in a trichannel case is also compared. The rRMSE $\left\|\epsilon^k\right\|/\|y\|$ is averaged (mean and standard deviation) over the 100 signals and is plotted in Fig. 2 as a function of the inner iterations $k = 1..2K$ of the four algorithms.

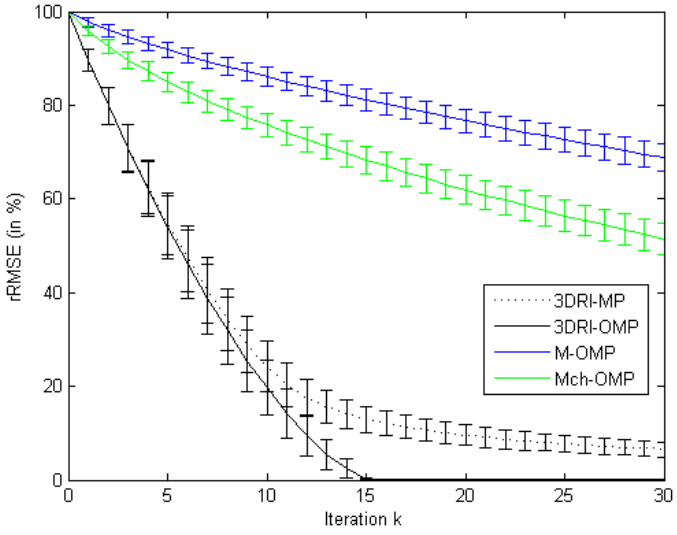

Figure 2: Comparison between the performances of 3DRI-MP, 3DRI-OMP, M-OMP and Mch-OMP. The rRMSE averaged over 100 signals is plotted as a function of the inner iteration k.

We observe that the 3DRI-OMP gives better approximation performances than the 3DRI-MP. At the beginning, both algorithms have similar behaviors, as there are not many active atoms in the decompositions, and so the optimization procedure (see Section 5.2) of the 3DRI-OMP does not have a significant impact. At the end of the K iterations, the 3DRI-OMP has a rRMSE of 0.2%, whereas 3DRI-MP has a rRMSE of 13.4%. This shows the optimality of the 3DRI-OMP, which provides the least-square solution, contrary to the 3DRI-MP. Sometimes, when the overlaps between atoms are sizeable, the 3DRI-OMP does not recover the good parameters $\left\{l^k\right\}_{k=1}^K$ and $\left\{\tau^k\right\}_{k=1}^K$. This explains why the averaged rRMSE in Fig. 2 is not exactly equal to 0 at the end of the 3DRI-OMP ($k = 15$). Concerning M-OMP and Mch-OMP, their rRMSE are huge since they do not recover the good atoms. So, these algorithms are not appropriate for rotated data. The rRMSE for iterations $k = K..2K$ are shown to verify that there is no breakpoint in the curves.

This experiment highlights the relevance of the 3DRI algorithms for the decomposition of revolved data, and the optimality of 3DRI-OMP over 3DRI-MP.

8.2. Experiment 2: 3D rotation invariant dictionary learning

This experiment was designed to test the recovery ability of the 3DRI-DLA. The experimental protocol described hereafter was inspired by [43, 7, 44] which tested shift-invariant DLAs. A dictionary $\mathbf{\Psi}$ of $L = 45$ normalized trivariate kernels is created from white uniform noise, and the kernels length is $T = 18$ samples. The training set $\boldsymbol{Y}_1$ is composed of $Q = 2000$ signals of length $N = 20$, and it is synthetically generated from this dictionary. For the kernels, circular shifts are not allowed, and so only three shifts are possible. Each training signal is composed of the sum of three atoms, for which the coefficients (strictly positive), rotation matrices, kernel indices, and shift parameters are randomly drawn. White Gaussian noise is also added at several levels: a signal-to-noise ratio of 10, 20 and 30 dB, and without noise. The dictionary initialization is made on the training set, and the learned dictionary $\hat{\mathbf{\Psi}}$ is returned after 80 iterations over the training set (*i.e.* $j \leq 80 \times Q$). For the first 40 iterations, the adaptive step is chosen as: $\lambda_2^j = 1/(j - q + 1)^{1.5}$, as constant for each loop of the training set, and it is then kept constant for the last iterations: $\lambda_2^j = 1/40^{1.5}$.
The experimental protocol was slightly changed to give the training set $\boldsymbol{Y}_2$. In this case, there is no more rotation when composing the training signals with the trivariate kernels (*i.e.* $R_{l,\tau} = Id$).

In the experiment, a learned kernel $\hat{\psi}_l$ is considered as detected, *i.e.* recovered, if its correlation value μ_l, after 3D registration, with its corresponding original kernel ψ_l respects the following rule:

$$\mu_l \geq 0.99 \;\text{ with }\; (\mu_l, .) = \text{Reg\_SVD}(\hat{\psi}_l, \psi_l)\,. \tag{23}$$

As the M-DLA used in a trivariate case, is also tested for comparison, its detection condition is simply:

$$\mu_l = \left|\left\langle \psi_l, \hat{\psi}_l \right\rangle\right| \geq 0.99\,. \tag{24}$$

Table 1 summarizes the detection rates averaged over 10 tests, given in percentages as a function of the noise levels (columns) and the DLA type (rows). 3DRI-DLA (a) gives the results for the training set $\boldsymbol{Y}_1$ and the detection condition (23); 3DRI-DLA (b) gives the results for the training set $\boldsymbol{Y}_2$ and the detection condition (23); and M-DLA gives the results for the training set $\boldsymbol{Y}_2$ and the detection condition (24).

Table 1: Detection rate results (in %) as a function of the noise level and the DLA type.

Noise level (in dB)	10	20	30	No noise
3DRI-DLA (a)	96.8	95.8	95.1	96.8
3DRI-DLA (b)	58.0	72.2	76.3	77.8
M-DLA	56.4	57.3	61.1	61.1

Note that a similar experiment was done by [7], to compare different shift-invariant DLAs, but in the univariate case. First, we note that the results of the M-DLA in the trivariate case are

similar to the univariate one [7]. Thus, the trivariate nature of the signals has no influence on the results considered. However, the differences in the results between the three rows of Table 1 can seem surprising.

On the first hand, trivariate kernels are white noised but are nevertheless correlated since they are short ($T = 18$). For 100 000 tries, the averaged correlations between two trivariate normalized white Gaussian noise are 0.2, and 0.4 after 3D registration. Thus, in this case, the registration doubles the correlation value. The 3DRI approach intrinsically improves the ability to recover the kernels, which explains the better results of 3DRI-DLA over M-DLA used in the trivariate case.

On the other hand, random rotations in the training set $\boldsymbol{Y}_1$ provide a faster and a more stable convergence of the non-convex 3DRI-DLA, compared to $\boldsymbol{Y}_2$. Kernels, which are the rotational invariants of the data, are multiplied by rotation matrices with different angles in $\boldsymbol{Y}_1$. As different rotations are observed, algorithm separates faster the kernel from the rotations. In a nutshell, when everything rotates, it is easy to identify what does not rotate. That explains why results 3DRI-DLA(a) are better than 3DRI-DLA(b) (which give similar results with more iterations).

This phenomenon is in agreement with stochastic optimization theory [45]. In non-convex optimization, several local minima exist. When optimization gets stuck in a local minimum, the cost to go out is function to the data diversity/entropy in the optimization space. Consequently, the convergence of an iterative non-convex optimization for high diversity data is faster than data with low diversity.

To conclude this section, our proposed algorithms have been successfully evaluated and compared on simulation data.

9. Experiments on Real Data

In this section, our methods are now applied to real data. First, the data are presented, and then methods are applied to the original data and to the revolved data.

9.1. Application data

Our methods are applied to motion signals of French cued speech, which is a gestural language, to complement speech reading [46, 47]. This language associates speech articulation to cues formed by the hand. In the following, only the motions of the hand are studied. The hand can be put in place in different shapes (finger configurations corresponding to consonants) as shown in Fig. 3, and in different placements (locations on the face corresponding to vowels).

To make the acquisition, retroreflective markers are put on the hands of a skilled cuer who usually practices French cued speech [46, 47]. Data are acquired by 12 cameras which record the 3D coordinates of these markers using a Vicon® Motion Capture System, as shown in Fig. 4. At the end of the acquisition, tricomponent coordinates are obtained for each marker at 120 Hz. These raw data are parsed in $Q = 57$ positions signals, and are then derived to give velocity signals v_x, v_y and v_z. These signals $v = [\ v_x\ ;\ v_y\ ;\ v_z\ ]^T$ are going to be the input to our algorithms. As a single point/ marker is studied, among all of the

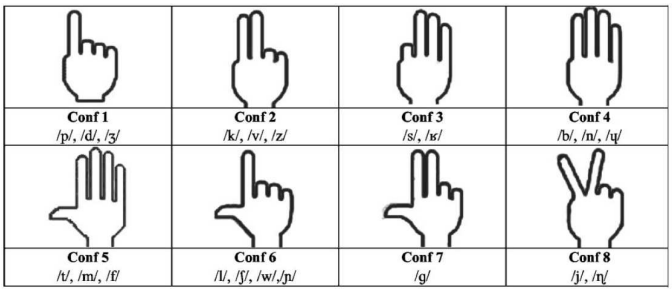

Figure 3: Hand shapes of French cued speech (reprinted from [46]).

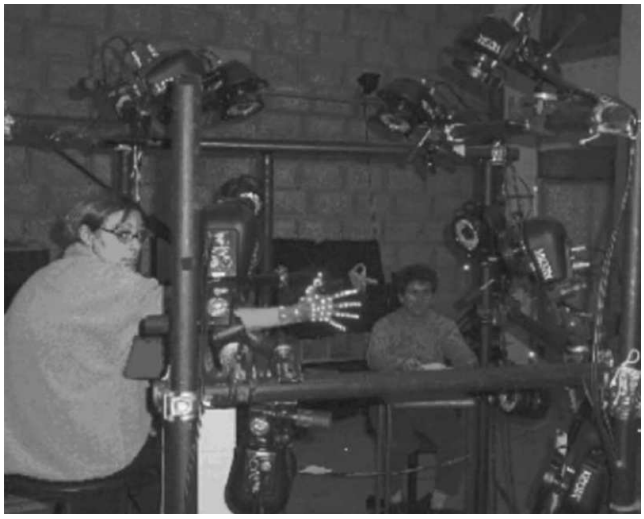

Figure 4: Acquiring motion data: the 3D positions of the retroreflective markers are given by a Vicon® Motion Capture System (reprinted from [46]).

available markers shown in Fig. 4, we arbitrarily chose the one located on the top on the thumb. The data units are centimetres (cm) for positions and centimetres per second (cm/s) for velocities. To lighten figures, units will be omitted hereafter.

Note that the introduced algorithms are independent of the motion capture system: inertial, magnetic and optical systems (active or passive markers). All of these data acquisitions finally give 3D spatial signals, which are then processed by our algorithms.

9.2. Experiment 3: dictionaries learning

In this experiment, we applied 3DRI-DLA to the trivariate dataset described in the previous paragraph. We also made a comparison with the M-DLA used in the trivariate case. A dictionary processed by M-DLA is called an *oriented* learned dictionary (OLD), as kernels are learned in a fixed orientation, without possible rotation. A dictionary processed by 3DRI-DLA is called a *non-oriented* learned dictionary (NOLD), as kernels are invariant to rotation and can be used in all possible orientations. A DCT used with Mch-OMP (Mch-DCT) is also considered to compare results with model (1).

Hyper-parameters have been chosen empirically. Parameter L corresponds to the number of underlying motion primitives,

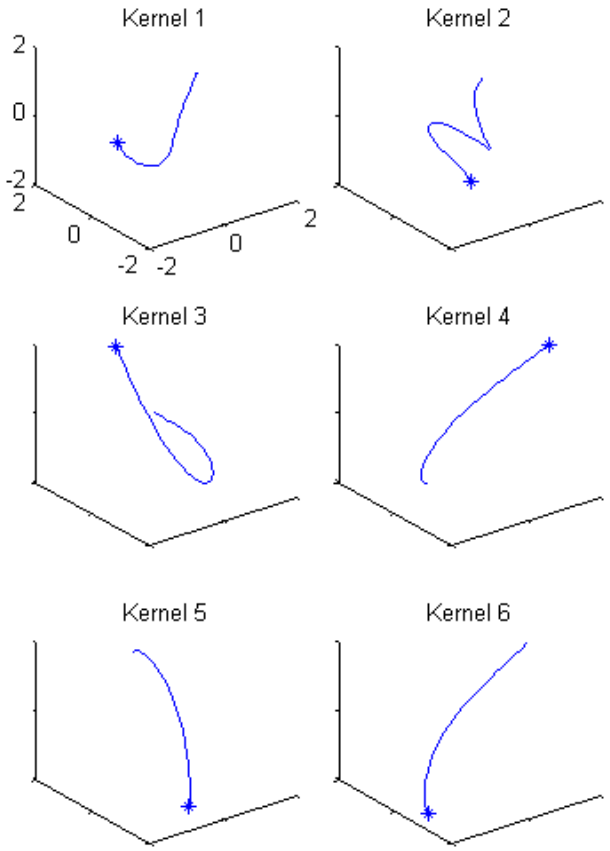

Figure 5: Rotatable 3D trajectories associated with the nonoriented learned dictionary (NOLD) processed by 3DRI-DLA. Axes are all the same.

ie kernels, which the user expects to be present in database signals. Since primitives are invariant to rotation, their number L can be reduced [7]. Parameter K corresponds to the number of actives primitives, ie atoms, which compose each signal. The user has to determinate $K = K_1 + K_2$ in order to provide K_1 primary primitives coding main energy, and K_2 secondary primitives coding small variabilities between signals.

A NOLD is processed with $L = 6$ kernels, as the dictionary shown in Fig. 1. The rRMSE averaged over the dataset is 18.2%, with $K = 20$ atoms in each decomposition. The velocity signals are integrated only to provide a more visual representation of the dictionary, even though it is not easy to see a 3D trajectory on a 2D figure. Fig. 5 shows the rotatable 3D trajectories associated with the NOLD, and the stars represent the beginning of the trajectories. Due to integration, different velocity kernels can provide very similar trajectories. These trajectories, which were extracted by 3DRI-DLA, correspond to the elementary patterns of these data. They are the motion primitives of the cued speech gestural language.

In the same manner, different OLDs are learned with $L = 6$, 10 and 14 kernels. With $K = 20$ atoms, the averaged rRMSE are respectively $27.7\%, 25.7\%$ and 24.2%. Even with kernels over twice those of the NOLD, the oriented learning cannot span the space as much as the non-oriented learning. For the Mch-DCT, the averaged rRMSE is 48.1%. These results already show the relevance of the non-oriented approach, which provides an efficient kernel dictionary that is more compact than the oriented ones, which are themselves better than the Mch-DCT.

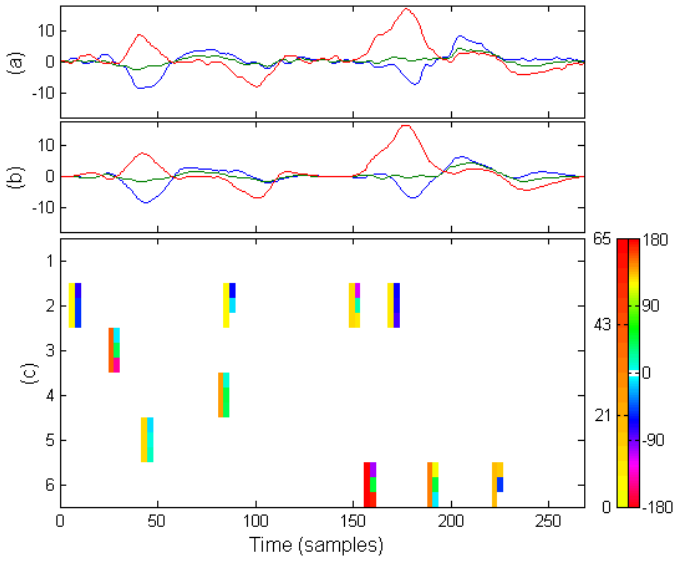

Figure 6: Original y_{52} (a) and approximated $\hat{y}_{52}$ (b) velocity signals, and the associated spikegram (c) which is a time-kernel representation.

9.3. *Experiment 4: test on data*

We then applied 3DRI-OMP with the NOLD, to give an adapted and non-oriented decomposition. We now explain how to visualize the coefficients obtained from a shift and 3DRI decomposition. Usually, real coding coefficients $x_{l,\tau}$ are displayed by a time-kernel representation called a spikegram [48]. This condenses three indications:

- the temporal position τ (abscissa),
- the kernel index l (ordinate),
- the coefficient amplitude $x_{l,\tau}$ (gray level of the spike).

This presentation allows an intuitive readability of the decomposition. With complex coefficients, the coefficient modulus is used for the amplitude and its argument gives the angle value, which is written next to the spike [7]. This coefficient presentation provides clear visualization.

In the present case, each spike has a coding coefficient $x_{l,\tau}$ and a rotation matrix $R_{l,\tau}$, which means that there are several parameters to display for each coefficient. To maintain good visualization, each rotation matrix is converted in a univocal way into 3 Euler angles [49]. Two gray shading levels are set up for this spikegram: one for coefficient amplitude and one for Euler angles. The angles scale, defined from -180° to +180°, is visually circular: a negative angle just above -180° thus appears visually close to a positive one just below +180°. Finally, the decomposition parameters are thus displayed with six indications:

- the temporal position τ (abscissa),
- the kernel index l (ordinate),
- the coefficient amplitude $x_{l,\tau} \geq 0$ (colorbar),
- the 3 Euler angles $\theta^1_{l,\tau}$, $\theta^2_{l,\tau}$, $\theta^3_{l,\tau}$ displayed vertically (circular colorbar).

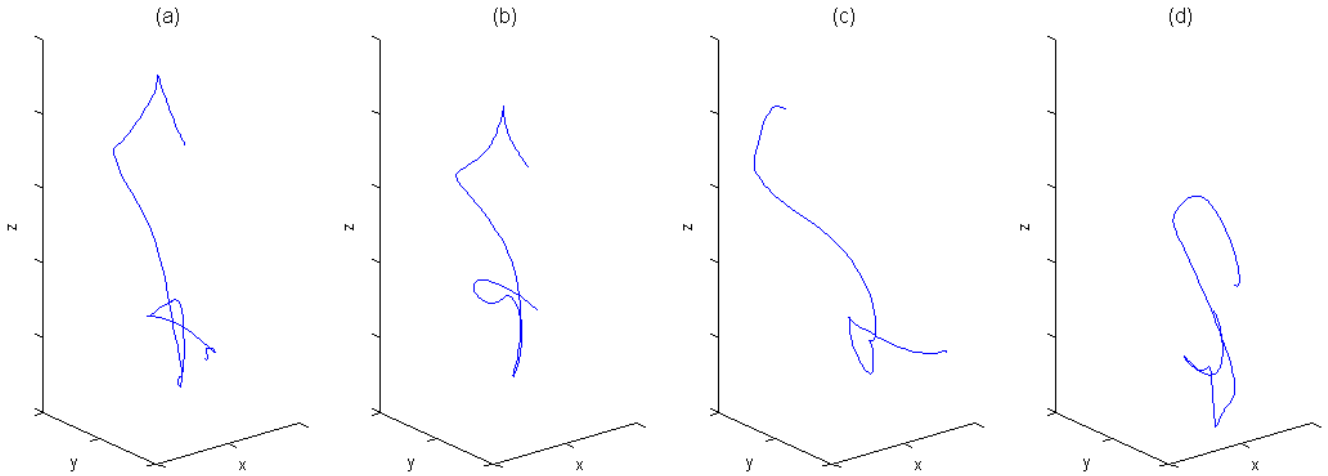

Figure 7: Signal y_{52}. Original trajectory (a) and approximations with $K = 5$ atoms for nonoriented (b), oriented (c) and Mch-DCT (d) approaches.

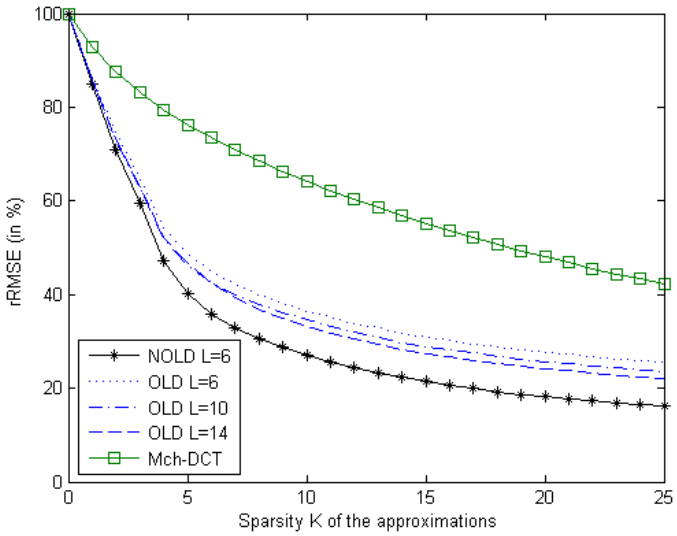

Figure 8: rRMSE on cued speech data as a function of the sparsity K of the approximation for the different dictionaries.

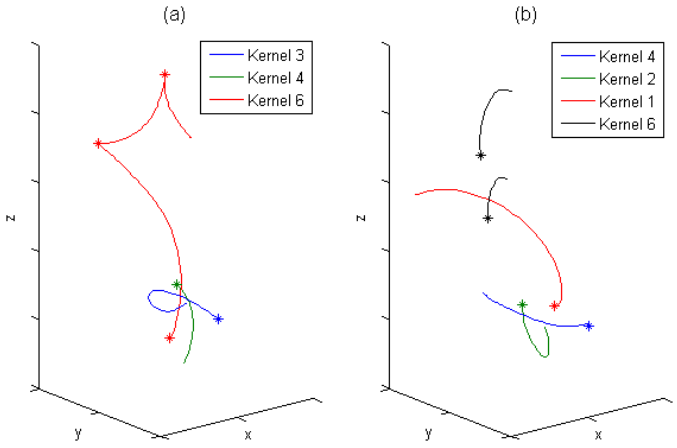

Figure 9: Signal y_{52}. Nonoriented reconstructed (a), and oriented reconstructed (b) trajectories using primary atoms.

To illustrate this, the signal y_{52} is processed by 3DRI-OMP with the NOLD and is presented in Fig. 6. The signal (a) is the original signal y_{52} composed of three components (solid, dashed and dotted lines), and the signal (b) is the approximated signal $\hat{y}_{52}$ with $K = 10$ atoms. The associated spikegram is plotted in (c) and can be viewed as the result of the signal deconvolution through the learned dictionary. The low number of atoms used in the signal approximation shows the sparsity of the decomposition, which is called *sparse code*. Primary atoms are the largest amplitude ones, like kernels 3, 4 and 6, and they concentrate the signal energy. The secondary atoms code the variabilities between the different realizations of the same gesture.

Approximated trajectories of the signal y_{52} with $K = 5$ atoms are plotted in Fig. 7. The original 3D trajectory is plotted in Fig. 7(a), the non-oriented approximated one using 3DRI-OMP with NOLD in Fig. 7(b), the oriented approximated one using M-OMP with OLD ($L = 6$) in Fig. 7(c), and the Mch-DCT approximated one in Fig. 7(d). On this visualization, quality degradation of approximed trajectories is obviously observed To quantitatively confirm this observation, rRMSE for different values of the sparsity K are plotted in Fig. 8. These results confirm the efficiency of the non-oriented dictionary with respect to the the oriented ones, which are themselves better than the Mch-DCT.

Now, we are interested in the contributions of the primary atoms. The trajectory of the signal y_{52} is reconstructed using its five primary atoms in Fig. 9. For instance, for the reconstruction of Fig. 9(a), the y_{52} is rebuilt as the sum of the NOLD kernels 3, 4 and 6 (used three times), which are specified by the amplitudes and rotations of the spikegram (Fig. 6). The non-oriented reconstruction and the oriented one are now compared. Reconstruction is not possible for DCT since its atoms are not time-localized. Considering the original 3D trajectory plotted in Fig. 7(a), the non-oriented reconstructed trajectory is plotted in Fig. 9(a) and the oriented reconstructed one in Fig. 9(b). In Fig. 9(a), we observe that the NOLD kernel 6 is

used three times with different orientations (see the Euler angles of Fig. 6), whereas in Fig. 9(b), the OLD kernel 6 is used two times, but without the possibility to change the orientation to provide better matching of the original trajectory. Notice that, in the 3D oriented case treated by M-OMP, a negative coefficient $x_{l,\tau}$ gives a reflection of the associated trajectory.

To conclude this experiment, we have observed that the 3DRI approach provides an efficient and compact rotatable kernel dictionary and favors better fitting of the kernels, allowing their rotations.

9.4. Experiment 5: test on revolved data

In this experiment, the signals of the dataset are revolved by different angles, which were randomly chosen and are now denoted by $(.)'$. The NOLD and the OLDs learned in the previous experiment are kept to carry out the decompositions of the revolved signals. With $K = 20$, the rRMSE averaged over the dataset is 18.2% for the non-oriented case, 48.5% for the oriented one with $L = 6$, and 48.1% for the Mch-DCT. rRMSE for different values of the sparsity K are plotted in Fig. 10. Comparing Fig. 8 and 10, we observe that rRMSE curves of the NOLD are identical, that proves its rotation invariance, contrary to OLDs with decreasing performances. We also observe that Mch-DCT is not sensible to data rotation.

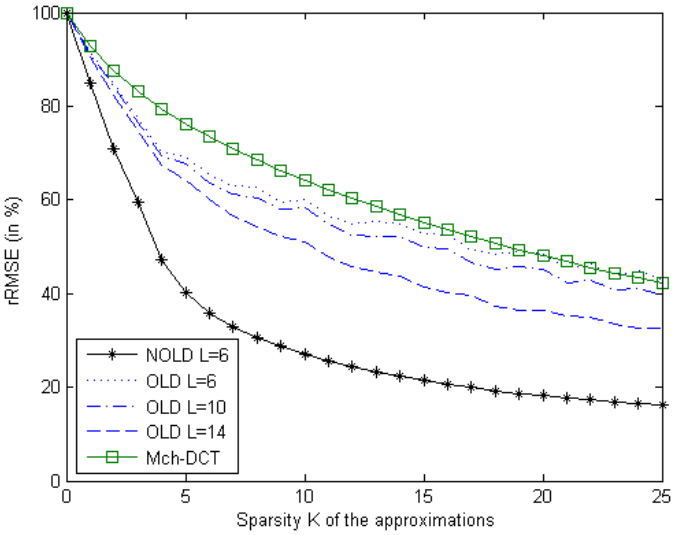

Figure 10: rRMSE on revolved cued speech data as a function of the sparsity K of the approximation for the different dictionaries.

The revolved signal y'_{52} is processed by 3DRI-OMP with the NOLD and is shown in Fig. 11. Signal (a) is the revolved signal y'_{52}, the signal (b) is the approximated signal $\hat{y}'_{52}$ (b) with $K = 10$ atoms. The associated spikegram is displayed in (c). The two spikegrams of Fig. 6(c) and Fig. 11(c), come from non-oriented decompositions of signals y_{52} and y'_{52}, and they are now compared. The kernel indices, shift parameters and coefficient amplitudes are the same. However, as the rotations are visually displayed, it is not possible to see if the angle differences correspond to the applied random rotation. The random rotation matrix is denoted by R^r, the estimated rotation matrices

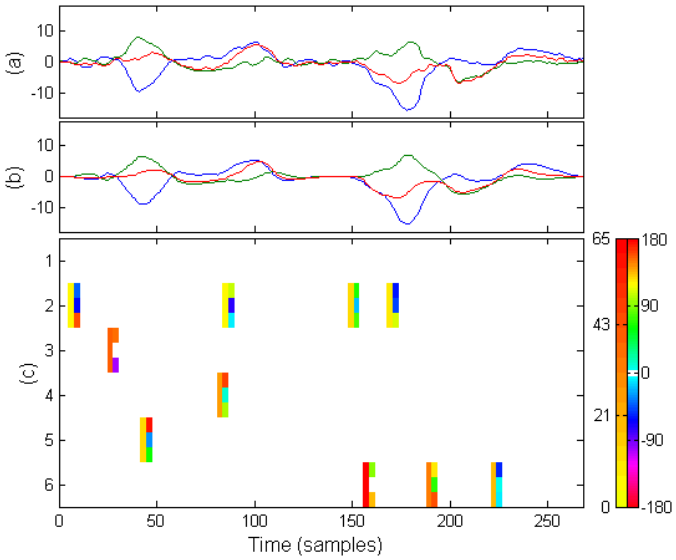

Figure 11: Revolved y'_{52} (a) and approximated $\hat{y}'_{52}$ (b) velocity signals, and the associated spikegram (c).

are denoted by R^{e4} for Experiment 4 and by R^{e5} for Experiment 5. For each signal y_q, the error e_q between rotation matrices is computed such that:

$$e_q = \frac{1}{K} \sum_{k=1}^{K} \left\| R^{e5}_{l^k,\tau^k;q} - R^r_q \, R^{e4}_{l^k,\tau^k;q} \right\| \Big/ \left\| R^r_q \, R^{e4}_{l^k,\tau^k;q} \right\| . \qquad (25)$$

This error is averaged over the dataset and it is null. Thus, the differences between the rotation parameters of Experiment 4 and Experiment 5 correspond exactly to the random rotations applied to the signals. This proves the 3D rotation invariance of our nonoriented method.

As in the previous experiment, the trajectory of the revolved signal y'_{52} is reconstructed on the primary atoms in Fig. 12. The 3D trajectory of y'_{52} is plotted in Fig. 12(a), the non-oriented reconstructed one in Fig. 12(b), and the oriented reconstructed one in Fig. 12(c). The reconstructions of Fig. 9(a) and Fig. 12(b) are similar, taking into account the rotation. The reconstructions of Fig. 9(b) and Fig. 12(c) are totally different, notably concerning the atoms used. This shows the limitation of an oriented dictionary fixed in a particular orientation, and thus not appropriate for multiple orientation data.

Better results of the nonoriented approach over the oriented approach have already been noted in the previous experiment, even without rotation of the data, but they are obviously highlighted in this experiment that deals with revolved data. The nonoriented approach is robust to rotation: the rRMSE is equal and the selected atoms are identical whatever the rotation. To conclude this section, our 3DRI methods have been applied to trajectories analysis, and the comparative experiments have shown the necessity and the relevance of these methods.

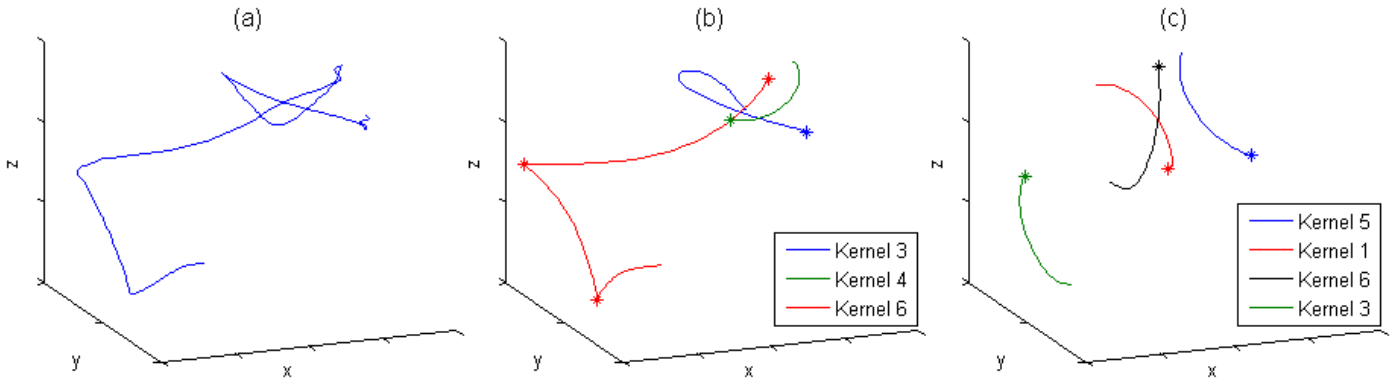

Figure 12: Revolved signal y'_{52}. Original (a), nonoriented reconstructed (b), and oriented reconstructed (c) trajectories using primary atoms.

10. Discussions and Prospects

Different applications and reflections are discussed in this section.

10.1. Gesture recognition

Mallat recently noted that the key for the classification is not the representation sparsity, but its invariances [50, 51]. 3D rotation is a difficult problem, which is often circled around rather than evaluated. For making recognition, most of the methods do not evaluate rotation parameters, but use rotation invariant descriptors, which are often called shape signatures/ features [25, 26]. As rotation parameters are not computed, there is a loss of information. Our decomposition methods compute the parameters of the 3D transformation of a trivariate signal modeled as a linear combination of rotatable 3D patterns. In the 3DRI case, the decompositions are invariant to temporal shift (parameter τ), to rotation (parameter $R_{l,\tau}$), to scale (parameter $x_{l,\tau}$) and to spatial translation (use of velocity signals instead of position signals).

Considering these reflections, we are also working on the classification of sparse codes, to carry out gesture recognition. Some classification methods have been set up in [52, 53] to deal with model (2) with good results, but have to be adapted to model (3).

10.2. Motion primitives learning for robotics

Considering a dataset that has acquired a phenomenon, dictionary learning can extract the generating causes of this phenomenon [17, 31]. The learned patterns correspond to the underlying structures of the observed phenomenon and provide efficient coding. In the same way, studying 2D handwritten characters, Barthélemy et al. [7] learnt 2D gestural primitives of the handwriting, and in this article, 3D gestural primitives of the French cued speech have been learned. It is thus related to trajectory data mining [54] and motion primitives learning [55, 56].

In robotics, differents studies have learnt motion primitives for specific locomotion modes of biped robots [57, 58, 59]. However, a heavy parametric formalism was used to model the robot kinematics. It is possible to solve this problem with 3DRI-DLA, which is a nonparametric approach. Using a dataset containing several signals of specific motion realized by a subject, 3DRI learning is carried out for each robot joint (hip, knee and ankle). This gives motion primitives dedicated to specific locomotion modes.

11. Conclusion

This article proposes a new model for describing a time-varying 3D object as the sum of the rotatable 3D patterns. The model considered combines the 3D rotation invariance and the shift-invariance of the patterns. Based on sparse approximation, 3DRI-MP and 3DRI-OMP carry out 3DRI decompositions, and 3DRI-DLA carries out the learning of 3DRI patterns. Such an approach provides a compact rotatable kernel dictionary, is robust to 3D rotations, and is efficient even when the studied trivariate data are not revolved. As validation, these algorithms were here applied to motion signals of French cued speech.

There are multiple applications in various domains: non-rigid structure-from-motion, 3D curve matching, 3D tracking, gesture representation and analysis, motion primitives learning, trajectory data mining, 3D pattern discovery and all other processings based on 3DRI decomposition.

The considered prospects are to extend the presented methods to the multisensors case, when physically linked P points/ markers/ sensors are studied (cf. Introduction), and to add a classification step adapted to our model to provide gesture recognition.

Acknowledgements

The authors thank C. Richard and anonymous reviewers for their comments, A. Souloumiac from CEA, LIST for his plentiful advice about the Procrustes problem, J. Atif from LRI, TAO for his help about optimization theory, F. Elisei from GIPSA-Lab, DPC for his explanations about cued speech data (acquired by Attitude Studio during the RNRT 01/37 ARTUS project), and C. Berrie for his help about English usage.

References

[1] C. Bregler, A. Hertzmann, H. Biermann, Recovering non-rigid 3D shape from image streams, in: Proc. IEEE Conf. Computer Vision and Pattern Recognition CVPR, pp. 690–696.

[2] L. Torresani, A. Hertzmann, C. Bregler, Learning non-rigid 3D shape from 2D motion, in: Advances in Neural Information Processing Systems NIPS '04, pp. 1555–1562.

[3] A. Srivastava, S. Joshi, W. Mio, X. Liu, Statistical shape analysis: clustering, learning, and testing, IEEE Trans. Pattern. Anal. Mach. Intell. 27 (2005) 590–602.

[4] L. Torresani, A. Hertzmann, C. Bregler, Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors, IEEE Trans. Pattern. Anal. Mach. Intell. 30 (2008) 878–892.

[5] S. Zhang, Y. Zhan, Y. Zhou, M. Uzunbas, D. Metaxas, Shape prior modeling using sparse representation and online dictionary learning, in: Medical Image Computing and Computer-Assisted Intervention - MICCAI 2012, volume 7512 of *Lecture Notes in Computer Science*, pp. 435–442.

[6] I. Akhter, Y. Sheikh, S. Khan, T. Kanade, Trajectory space: A dual representation for nonrigid structure from motion, IEEE Trans. Pattern. Anal. Mach. Intell. 33 (2011) 1442–1456.

[7] Q. Barthélemy, A. Larue, A. Mayoue, D. Mercier, J. Mars, Shift & 2D rotation invariant sparse coding for multivariate signals, IEEE Trans. Signal Process. 60 (2012) 1597–1611.

[8] Q. Barthélemy, A. Larue, J. Mars, 3D rotation invariant decomposition of motion signals, in: Computer Vision – ECCV 2012, volume 7585 of *Lecture Notes in Computer Science*, pp. 172–182.

[9] P. Schonemann, A generalized solution of the orthogonal Procrustes problem, Psychometrika 31 (1966) 1–10.

[10] D. Eggert, A. Lorusso, R. Fisher, Estimating 3-D rigid body transformations: a comparison of four major algorithms, Mach. Vis. Appl. 9 (1997) 272–290.

[11] J. Gower, G. Dijksterhuis, Procrustes Problems, Oxford Statistical Science Series, 30, 2004.

[12] J. Gower, Procrustes methods, Wiley Interdiscip. Rev. Comput. Stat. 2 (2010) 503–508.

[13] Y. Su, L. Liu, Y. Yang, Optimal trajectory space finding for nonrigid structure from motion, in: Advanced Concepts for Intelligent Vision Systems ACIVS '10, volume 6474 of *Lecture Notes in Computer Science*, pp. 357–366.

[14] M. Lewicki, T. Sejnowski, Learning overcomplete representations, Neural Comput. 12 (1998) 337–365.

[15] S. Mallat, A Wavelet Tour of signal processing, 3rd edition, New-York : Academic, 2009.

[16] I. Tošić, P. Frossard, Dictionary learning, IEEE Signal Process. Mag. 28 (2011) 27–38.

[17] B. Olshausen, D. Field, Sparse coding with an overcomplete basis set: a strategy employed by V1?, Vision Res. 37 (1997) 3311–3325.

[18] K. Skretting, J. Husøy, S. Aase, General design algorithm for sparse frame expansions, Signal Process. 86 (2006) 117–126.

[19] K. Kanatani, Analysis of 3-D rotation fitting, IEEE Trans. Pattern. Anal. Mach. Intell. 16 (1994) 543–549.

[20] B. Horn, Closed-form solution of absolute orientation using unit quaternions, J. Opt. Soc. Am. A 4 (1987) 629–642.

[21] B. Horn, Closed-form solution of absolute orientation using orthonormal matrices, J. Opt. Soc. Am. A 4 (1988) 1127–1135.

[22] M. Walker, L. Shao, R. Volz, Estimating 3-D location parameters using dual number quaternions, CVGIP: Image Understanding 53 (1991) 358–367.

[23] R. Bergevin, M. Soucy, H. Gagnon, D. Laurendeau, Towards a general multi-view registration technique, IEEE Trans. Pattern. Anal. Mach. Intell. 18 (1996) 540–547.

[24] C. Bastuscheck, E. Schonberg, J. Schwartz, M. Sharir, Object recognition by 3-Dimensional curve matching, Int. J. Intell. Syst. 1 (1986) 105–132.

[25] A. Croitoru, P. Agouris, A. Stefanidis, 3D trajectory matching by pose normalization, in: Proc. ACM Int. Workshop on Geographic Information Systems GIS '05, pp. 153–162.

[26] J. Bandera, R. Marfil, A. Bandera, J. Rodriguez, L. Molina-Tanco, F. Sandoval, Fast gesture recognition based on a two-level representation, Pattern Recognit. Lett. 30 (2009) 1181–1189.

[27] H.-T. Pham, J. Kim, Y. Won, A cost effective method for matching the 3d motion trajectories, in: IT Convergence and Security 2012, volume 215 of *Lecture Notes in Electrical Engineering*, pp. 889–895.

[28] M. Bakircioğlu, U. Grenander, N. Khaneja, M. Miller, Curve matching on brain surfaces using induced frenet distance metrics, Hum. Brain Mapp. 6 (1998) 329–331.

[29] P. Besl, H. McKay, A method for registration of 3-D shapes, IEEE Trans. Pattern. Anal. Mach. Intell. 14 (1992) 239–256.

[30] R. Gribonval, H. Rauhut, K. Schnass, P. Vandergheynst, Atoms of all channels, unite! Average case analysis of multi-channel sparse recovery using greedy algorithms, Technical Report PI-1848, IRISA, 2007.

[31] E. Smith, M. Lewicki, Efficient auditory coding, Nature 439 (2006) 978–982.

[32] T. Blumensath, M. Davies, Sparse and shift-invariant representations of music, IEEE Trans. Audio Speech Lang. Processing 14 (2006) 50–57.

[33] G. Davis, Adaptive Nonlinear Approximations, Ph.D. thesis, New York University, 1994.

[34] S. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, IEEE Trans. Signal Process. 41 (1993) 3397–3415.

[35] Y. Pati, R. Rezaiifar, P. Krishnaprasad, Orthogonal Matching Pursuit: recursive function approximation with applications to wavelet decomposition, in: Conf. Record of the Asilomar Conf. on Signals, Systems and Comput., pp. 40–44.

[36] K. Petersen, M. Pedersen, The matrix cookbook, Technical Report, Technical University of Denmark, 2008.

[37] Q. Barthélemy, Représentations parcimonieuses pour les signaux multivariés, Ph.D. thesis, Université de Grenoble, 2013.

[38] B. Widrow, Adaptive Filters, Aspects of Network and System Theory, pp. 563–586.

[39] M. Plumbley, Geometrical methods for non-negative ICA: Manifolds, Lie groups and toral subalgebras, Neurocomputing 67 (2005) 161–197.

[40] Y. Nishimori, S. Akaho, Learning algorithms utilizing quasi-geodesic flows on the Stiefel manifold, Neurocomputing 67 (2005) 106–135.

[41] T. Moon, The Expectation-Maximization algorithm, IEEE Signal Process. Mag. 13 (1996) 47–60.

[42] K. Engan, S. Aase, J. Husøy, Multi-frame compression: theory and design, Signal Process. 80 (2000) 2121–2140.

[43] M. Aharon, Overcomplete Dictionaries for Sparse Representation of Signals, Ph.D. thesis, Technion - Israel Institute of Technology, 2006.

[44] C. Rusu, B. Dumitrescu, S. Tsaftaris, Explicit shift-invariant dictionary learning, IEEE Signal Process. Lett. 21 (2014) 6–9.

[45] J. Spall, Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control, John Wiley and Sons, Inc., 2003.

[46] G. Gibert, G. Bailly, D. Beautemps, F. Elisei, R. Brun, Analysis and synthesis of the three-dimensional movements of the head, face, and hand of a speaker using cued speech, J. Acoust. Soc. Am. 118 (2005) 1144–1153.

[47] G. Gibert, Conception et évaluation d'un système de synthèse 3D de Langue française Parle Complétée (LPC) à partir du texte, Ph.D. thesis, Institut National Polytechnique de Grenoble, 2006.

[48] E. Smith, M. Lewicki, Efficient coding of time-relative structure using spikes, Neural Comput. 17 (2005) 19–45.

[49] A. Hanson, Visualizing quaternions, Morgan-Kaufmann/Elsevier, 2006.

[50] S. Mallat, Group invariant scattering, Commun. Pure Appl. Math. 65 (2012) 1331–1398.

[51] J. Bruna, S. Mallat, Invariant scattering convolution networks, IEEE Trans. Pattern. Anal. Mach. Intell. 35 (2013) 1872–1886.

[52] A. Mouraud, Q. Barthélemy, A. Mayoue, C. Gouy-Pailler, A. Larue, H. Paugam-Moisy, From neuronal cost-based metrics to sparse coded signals classification, in: Proc. Eur. Symp. Artificial Neural Networks, Computational Intelligence and Machine Learning ESANN, pp. 311–316.

[53] A. Mayoue, Q. Barthélemy, S. Onis, A. Larue, Preprocessing for classification of sparse data: Application to trajectory recognition, in: Proc. IEEE Workshop on Statistical Signal Processing SSP '12, pp. 37–40.

[54] S. Satpathy, L. Sharma, A. Akasapu, N. Sharma, Towards mining approaches for trajectory data, Int. J. Advances in Science and Technology 2 (2011) 38–43.

[55] T. Kim, G. Shakhnarovich, R. Urtasun, Sparse coding for learning interpretable spatio-temporal primitives, in: Advances in Neural Information Processing Systems NIPS, pp. 1117–1125.

[56] C. Vollmer, J. Eggert, H.-M. Gross, Modeling human motion trajectories by sparse activation of motion primitives learned from unpartitioned data,

in: KI 2012: Advances in Artificial Intelligence, volume 7526 of *Lecture Notes in Computer Science*, pp. 168–179.

[57] Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, K. Tanie, Planning walking patterns for a biped robot, IEEE Trans. Rob. Autom. 17 (2001) 280–289.

[58] D. Tlalolini, C. Chevallereau, Y. Aoustin, Human-like walking: Optimal motion of a bipedal robot with toe-rotation motion, IEEE/ASME Trans. Mechatron. 16 (2011) 310–320.

[59] M. Powell, H. Zhao, A. Ames, Motion primitives for human-inspired bipedal robotic locomotion walking and stair climbing, in: Proc. IEEE Int. Conf. Robotics and Automation ICRA, pp. 543–549.