Chemometrics and Intelligent Laboratory Systems 250 (2024) 105153

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemometrics

Solving the missing value problem in PCA by Orthogonalized-Alternating Least Squares (O-ALS)

Adrián Gómez-Sánchez [a,b,*], Raffaele Vitale [b], Cyril Ruckebusch [b], Anna de Juan [a,**]

[a] Chemometrics Group, Universitat de Barcelona, Diagonal, 645, 08028, Barcelona, Spain
[b] U. Lille, CNRS, LASIRE, Laboratoire Avancé de Spectroscopie pour les Interactions, la Réactivité et l'Environnement, Cité Scientifique, F-59000, Lille, France

ARTICLE INFO

Keywords:
Principal Component Analysis (PCA)
Missing values
Nonlinear Estimation by Iterative Partial Least Squares (NIPALS)
Imputation
Singular Value Decomposition (SVD)
Orthogonalized-Alternating Least Squares (O-ALS)

ABSTRACT

Dealing with missing data poses a challenge in Principal Component Analysis (PCA) since the most common algorithms are not designed to handle them. Several approaches have been proposed to solve the missing value problem in PCA, such as Imputation based on SVD (I-SVD), where missing entries are filled by imputation and updated in every iteration until convergence of the PCA model, and the adaptation of the Nonlinear Iterative Partial Least Squares (NIPALS) algorithm, able to work skipping the missing entries during the least-squares estimation of scores and loadings. However, some limitations have been reported for both approaches. On the one hand, convergence of the I-SVD algorithm can be very slow for data sets with a high percentage of missing data. On the other hand, the orthogonality properties among scores and loadings might be lost when using NIPALS.

To solve these issues and perform PCA of data sets with missing values without the need of imputation steps, a novel algorithm called Orthogonalized-Alternating Least Squares (O-ALS) is proposed. The O-ALS algorithm is an alternating least-squares algorithm that estimates the scores and loadings subject to the Gram-Schmidt orthogonalization constraint. The way to estimate scores and loadings is adapted to work only with the available information.

In this study, the performance of O-ALS is tested and compared with NIPALS and I-SVD in simulated data sets and in a real case study. The results show that O-ALS is an accurate and fast algorithm to analyze data with any percentage and distribution pattern of missing entries, being able to provide correct scores and loadings in cases where I-SVD and NIPALS do not perform satisfactorily.

1. Introduction

Principal Component Analysis (PCA) [1–3] is one the most fundamental tools in chemometrics for data compression and visualization of complex data sets. It is widely employed as an exploratory tool for all kinds of data, such as in hyperspectral imaging [4,5], quality control [6], complex mixtures [7], as a basic tool for classification methods [8] or in process analytical technology [9,10], and for an endless number of different applications.

PCA is a bilinear factorization method that decomposes the data into so-called principal components, obtained using orthogonality and normalization constraints (Eq. (1)). The non-random variance of the initial data set can be described using a limited number of principal components, N, which allows for an easy visualization of the information. Each principal component is represented by a score and a loading vector, linked to the representation of samples (or row information) and variables (or column information) in the data matrix.

The PCA model is defined as in Eq. (1):

$$\mathbf{D} = \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{E} \qquad \textit{subject to}: \begin{cases} \mathbf{T}^{\mathrm{T}}\mathbf{T} = \mathbf{W}, \mathbf{W}_{ij} = \mathbf{0}\ \textit{for}\ i \neq j \\ \mathbf{P}^{\mathrm{T}}\mathbf{P} = \mathbf{I} \end{cases} \quad \text{Eq. 1}$$

where $\mathbf{D}$ is the data matrix sized $R \times C$, $\mathbf{T}$ is the scores matrix sized $R \times N$, $\mathbf{P}$ is the loadings matrix sized $C \times N$, and $\mathbf{E}$ is the residual matrix sized $R \times C$. R and C stand for the number of rows and columns of the initial data set, respectively. $\mathbf{W}$ is a diagonal matrix, sized $N \times N$, containing the eigenvalues of $\mathbf{D}\mathbf{D}^{\mathrm{T}}$ in its diagonal, while $\mathbf{I}$ is an identity matrix sized $N \times$

* Corresponding author. Chemometrics Group, Universitat de Barcelona, Diagonal, 645, 08028, Barcelona, Spain.
** Corresponding author.
E-mail addresses: gomez.sanchez.adr@gmail.com (A. Gómez-Sánchez), anna.dejuan@ub.edu (A. de Juan).

https://doi.org/10.1016/j.chemolab.2024.105153
Received 1 March 2024; Received in revised form 28 May 2024; Accepted 29 May 2024
Available online 31 May 2024

N. The PCA solution is unique up to sign and permutation ambiguity when principal components have pairwise non-equal eigenvalues [11].

Data sets often have missing values due to measurement errors, incomplete data collection or because of the nature of the measurements themselves [12–14]. The presence of missing values represents a significant challenge when applying PCA, since standard approaches, such as Singular Value Decomposition (SVD), cannot be straightforwardly applied.

The different patterns that missing values can adopt within a data set can affect differently and significantly the performance of the algorithms used (Fig. 1) [15]. Thus, randomly distributed missing values (Fig. 1A) do not tend to affect much the analysis because the estimation of missing entries is, in this case, quite easy. Instead, systematic patterns of missing values (Fig. 1B), e.g., those found in excitation-emission matrices where no signal is recorded below the excitation wavelength range or when scattering contributions are suppressed, provide a much more challenging scenario. Within the non-random patterns of missing entries, the missing block pattern is specifically associated with data fusion scenarios [16]. For instance, this pattern happens when data blocks related to several experiments monitored with different techniques are concatenated into a single structure (multiset) and the block coming from a particular technique may be missing for a specific experiment (Fig. 1C).

Depending on the percentage and pattern of missing values, different strategies may be adopted. For instance, if the pattern of missing data is randomly distributed (Fig. 1A) and the data presents a soft continuity across the variables, such as in spectroscopic data, an interpolation based on the nearest neighboring entries can be a simple and reliable solution to replace the missing observations [17]. However, when missing values show a systematic pattern, interpolation is not an option due to the absence of available neighboring entries, while extrapolation can be very risky and not recommended. This problem is even more dramatic in the case of missing data blocks in data fusion (Fig. 1C). In this scenario, there are some regions of the data set with full concatenated blocks of information, e.g., when measurements with different techniques were acquired in identical conditions, and some others where a block of information does not have an equivalence and is connected with a block of missing entries, providing the so-called incomplete multisets.

To perform PCA on an incomplete multiset, two main strategies can be adopted, either estimating the missing entries and working with traditional PCA algorithms, such as the Imputation based on SVD (I-SVD) [15,18], or adapting existing algorithms to work only with the available information, as could be done with Nonlinear Estimation by Iterative Partial Least Squares (NIPALS) [19].

I-SVD relies on applying the SVD algorithm after imputing the missing values with estimates. With I-SVD, such estimates are subsequently updated using the prediction of the SVD model, and a new SVD is conducted, until convergence. While I-SVD yields accurate orthogonal scores and loadings, the computational cost associated with the method is exceptionally high and convergence is not achieved in a reasonable time for challenging patterns of missing data.

NIPALS is a widely used method for the sequential extraction of principal components in multivariate data analysis. The technique is based on a one-by-one extraction of principal components through an alternating least squares (ALS) approach. Although the mathematical operations of NIPALS are easily adapted to handle missing values, it has been reported that the algorithm fails to provide orthogonal decompositions [18], and it only works properly when the (pseudo)rank of the data is 1, as might have pointed out by Anderson Christoffersson [20]. For higher ranks, the one-by-one extraction of subsequent components is not adequate as it results in non-orthogonal scores and loadings.

To overcome the limitations of the previous approaches, this work proposes a novel algorithm called Orthogonalized-Alternating Least Squares (O-ALS). O-ALS is adapted to work with missing entries, preserves orthogonality among components and achieves accurate results with a low computational cost. This algorithm operates iteratively performing adapted least-squares row-by-row and column-by-column calculations of scores and loadings for all components. In every iteration, the full matrices of scores and loadings are subjected to a Gram-Schmidt orthogonalization [21,22] to ensure orthogonality among the estimated profiles. The proposed O-ALS algorithm offers a promising solution for handling missing values in PCA and provides accurate results in few seconds.

In the remainder of this paper, the *modus operandi* of I-SVD, NIPALS and O-ALS for the analysis of data sets with missing information is described and their respective benefits and limitations assessed from a theoretical perspective. Afterwards, the performance of these algorithms is evaluated in simulated and real hyperspectral imaging data fusion examples, where missing block patterns are encountered.

2. Algorithm description

In this section, the calculation of PCA models in the presence of missing values by the I-SVD, NIPALS and O-ALS algorithms is described in detail. All of them were in-house encoded in MATLAB and are available on request.

2.1. Imputation based on SVD (I-SVD)

Let us consider $\mathbf{D}$ a full data set and $\mathbf{D}_m$ a derived data set with missing entries. To obtain the PCA model of the matrix $\mathbf{D}_m$, I-SVD works by estimating the missing entries of the data set [15,19]. The approach tries to impute values so that the original data space of the full data set $\mathbf{D}$ is preserved. The steps to apply SVD are as follows:

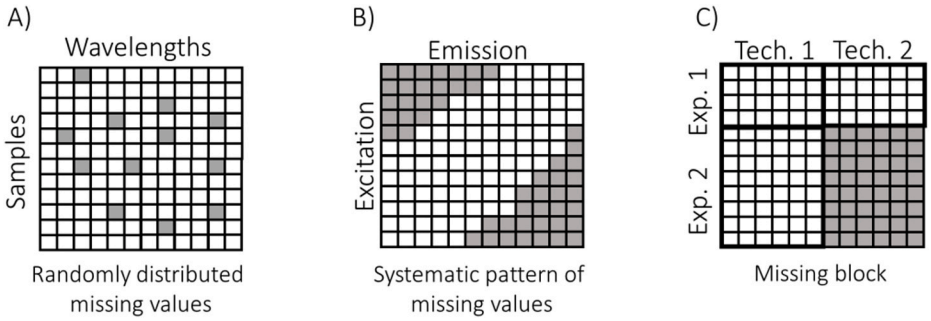

Fig. 1. Usual patterns of missing data found in data sets. In white, available entries. In gray, missing entries. A) Random distribution of missing values. B) Systematic pattern of missing values. C) Missing block pattern, where two experiments were conducted using Technique 1 and Technique 2, but the block of data corresponding to Experiment 2 measured with Technique 2 is missing.

Step 1) *Initial imputation of missing values in the data matrix* $\boldsymbol{D}_m$. This can be done by replacing missing entries with random values or by the row or column mean of the observed values.
Step 2) *SVD of the imputed data matrix.*
Step 3) *Update of the missing entries with the new predicted values from the SVD model calculated in step 2.* This is done reconstructing the full data $\widehat{\mathbf{D}}$ using a specified number of components equal to the rank of the data set. The missing entries on $\mathbf{D}_m$ are replaced by the predicted values in $\widehat{\mathbf{D}}$. Convergence is typically achieved when a small change in the SVD components is detected or after a predefined number of iterations.

This algorithm typically converges to the correct data space of **D** since the imputed entries must preserve the structure of the data space of the available data in $\mathbf{D}_m$, as long as the information of a component is preserved even in the presence of missing data. In other words, the final imputed values are estimated so that they represent the original space of **D**. The only assumption for I-SVD to work is that the rank chosen for the SVD reproduction of the data is correct.

2.2. Non-Iterative Partial Least Squares (NIPALS)

NIPALS is a well-known iterative algorithm that can be used to estimate sequentially the principal components of a given data set **D** [19]. The steps of the algorithm are as follows:

Step 1) *Find an initial guess for the loading vector* $\boldsymbol{p}$ *(*$C \times 1$*) of the first principal component.* A possible estimate can be a random vector, the mean column vector of the available data or the most intense row of the data set **D.** This initial estimate is used to start Step 2. Note that, alternatively, an initial estimate of the score vector $\mathbf{t}$ ($R \times 1$) can be also taken. If this is the case, the steps 2A and 2B will be swapped
Step 2) *Iterative ALS optimization of the score and the loading vector.*

2A) Given **D** and $\mathbf{p}^{\mathrm{T}}$, calculate the score vector by least squares.

$$\mathbf{t} = \mathbf{D}\left(\mathbf{p}^{\mathrm{T}}\right)^{+} \qquad \text{Eq. 2}$$

2B) Given **D** and **t**, calculate the loading vector.

$$\mathbf{p}^{\mathrm{T}} = \mathbf{t}^{+}\mathbf{D} \qquad \text{Eq. 3}$$

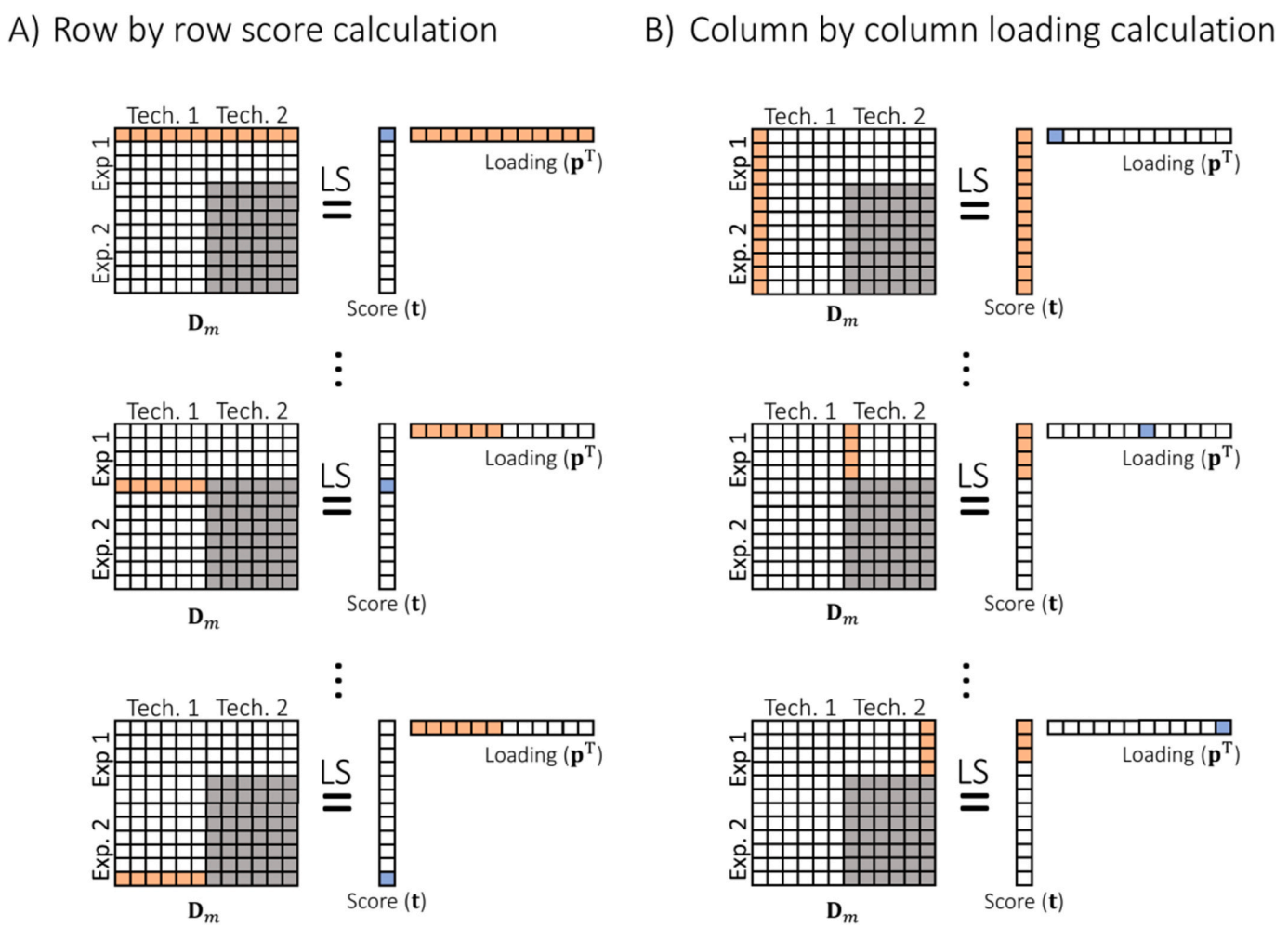

Fig. 2. A) Row-by-row calculation of the score profile by the NIPALS algorithm. Given a loading vector $\mathbf{p}^{\mathrm{T}}$ and a data row $\mathbf{d}_m(i,:)$ (both in orange), the corresponding score value $\mathbf{t}(i,1)$ is calculated (blue). If missing values are encountered, the loading vector is adapted to match the corresponding available entries in $\mathbf{d}_m(i,:)$. B) Column-by-column calculation of the loading profile by the NIPALS algorithm. Given a score vector $\mathbf{t}$ and a data column $\mathbf{d}_m(:,j)$ (both in orange), the corresponding score value $\mathbf{p}^{\mathrm{T}}(1,j)$ is calculated (blue). Similarly, if missing values are encountered, the score vector is adapted to match the corresponding available entries in $\mathbf{d}_m(:,j)$. The calculation is performed for all i and j which go from 1 to R and 1 to C, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

where "+" indicates the pseudoinversion operator. Steps 2A and 2B are repeated until convergence is reached, i.e., usually when a small change in the residual matrix **E** is detected or after a maximum number of iterations. When the convergence is achieved, **t** and $\mathbf{p}^T$ are retrieved.

Step 3) *Deflation of the initial data matrix **D**.* The data set is deflated by removing the variance explained by the first component (Eq. (4)).

$$\mathbf{D}_{new} = \mathbf{D} - \mathbf{t}\mathbf{p}^T \quad \text{Eq. 4}$$

The deflation process ensures that the variance of the next principal component will be orthogonal to the previous one, since the variation described by the first component has been removed from the data. If more components need to be calculated, once the data is deflated, steps 2 and 3 are repeated replacing **D** by $\mathbf{D}_{new}$ in Eqs. (2) and (3).

One of the claimed benefits of NIPALS is that it can handle missing values by skipping them during the ALS procedure, as represented in Fig. 2. To deal with the missing entries, the least squares calculations of steps 2A and 2B are done now row-by-row (Eq. (5)) and column-by-column (Eq. (6)), adapting the least-squares calculations to the available information in $\mathbf{D}_m$.

Step 2A)

$$\mathbf{t}(i,1) = \mathbf{d}_m(i,:)\left(\mathbf{p}^T\right)^+ \quad \text{Eq. 5}$$

Step 2B)

$$\mathbf{p}^T(1,j) = \mathbf{t}^+\mathbf{d}_m(:,j) \quad \text{Eq. 6}$$

Where $\mathbf{d}_m(i,:)$ is a row of matrix **D**, $\mathbf{d}_m(:,j)$ is a column of matrix **D**, i and j go from 1 to R and 1 to C, respectively. Then, when missing values are encountered along the vector $\mathbf{d}_m(i,:)$ in Eq. (5), the calculation of $\mathbf{t}(i,1)$ is done using only the available entries in $\mathbf{d}_m(i,:)$ (Fig. 2A) and the corresponding values of $\mathbf{p}^T$, which share the same position with the available entries in $\mathbf{d}_m(i,:)$. Such an operation is done to obtain all the elements of **t**, using every time the available entries in $\mathbf{d}_m(i,:)$ and the analogous information in $\mathbf{p}^T$. As can be inferred from the previous equations, every row of $\mathbf{D}_m$ can have a completely different number of available entries in completely different locations; therefore, the algorithm adapts to any pattern of missing information. Analogously, the same approach is exploited in Eq. (6) (Fig. 2B) to calculate every element of $\mathbf{p}^T$, using only the available entries in the column $\mathbf{d}_m(:,j)$ and the related information in **t**. Although this simple and elegant approach allows NIPALS to perform its least-squares calculations even in the presence of missing data, it will be shown that it can affect the orthogonality of the components estimated.

2.3. Orthogonalized-Alternating Least Squares (O-ALS)

The new algorithm Orthogonalized-Alternating Least Squares (O-ALS) is designed to work only with the available information to estimate the PCA model. In contrast to the NIPALS approach, this algorithm stands out for its ability to preserve the orthogonality of the score and loading profiles during the alternating least squares calculations. Summarizing, O-ALS is an iterative alternating least squares bilinear factorization that operates applying a Gram-Schmidt orthogonalization constraint. A key difference with the NIPALS algorithm is that all N components required to describe the variance of the data according to the desired rank are estimated simultaneously. The O-ALS steps are described below.

Step 1) *Generation of an initial estimate for the loadings matrix $\boldsymbol{P}^T$ ($N\times C$).* This can be done with random numbers. Note that, alternatively, an initial estimate of the score matrix **T** ($R \times N$) can also be taken. Then, the steps 2A and 2B will be swapped.

Step 2) *Iterative ALS optimization of scores and loadings under a Gram-Schmidt orthogonalization constraint.* This includes steps 2A and 2B, represented in Fig. 3 and described below.

Step 2A) Row-by-row least-squares estimation of the scores matrix **T** (Fig. 3A). For each data row, the corresponding row of the scores matrix is calculated using a least squares optimization process. The calculation of $\mathbf{t}(i,:)$ is performed by using only the available entries in $\mathbf{d}_m(i,:)$ and the corresponding values in $\mathbf{P}^T$ that share the same position with the available entries in $\mathbf{d}_m(i,:)$ (Eq. (7)).

$$\mathbf{t}(i,:) = \mathbf{d}_m(i,:)\left(\mathbf{P}^T\right)^+ \quad \text{Eq. 7}$$

where i goes from 1 to R. Once the calculation in Eq. (7) is finished, the Gram-Schmidt orthogonalization constraint is applied to the columns of **T** in order to preserve the orthogonality among the score profiles.

Step 2B) Column-by-column least-squares estimation of the loadings matrix, $\mathbf{P}^T$ (Fig. 3B). Every column of the loading matrix is estimated using only the available entries in $\mathbf{d}_m(:,j)$ and the analogous values in **T** that share the same position with the available entries in $\mathbf{d}_m(:,j)$ (Eq. (8)).

$$\mathbf{p}^T(:,j) = \mathbf{T}^+\mathbf{d}_m(:,j) \quad \text{Eq. 8}$$

where j goes from 1 to C. Similarly, the Gram-Schmidt orthogonalization constraint is applied to the rows of the matrix $\mathbf{P}^T$ in order to preserve the orthogonality among the loadings profiles, which in the end are individually normalized using the Euclidean norm.

Steps 2A and 2B are iteratively repeated until convergence is achieved. Convergence is typically by a small change in the principal component estimates is observed or after a predetermined number of iterations. Upon achieving convergence, the final orthogonal scores matrix **T** and orthogonal loadings matrix $\mathbf{P}^T$ are obtained.

3. Data sets

This section includes the details of the simulated and real examples of incomplete multisets with missing blocks dealt with in the present article. Since hyperspectral image fusion is a field where incomplete multisets are easily encountered, the simulations have been performed mimicking the fusion of a near-infrared (NIR) and a Raman hyperspectral image, considering various noise levels and missing data patterns. Additionally, a real example of NIR and Raman image fusion is investigated.

To set the scene, a hyperspectral image (HSI) consists of large number of spectra associated with a grid of points (pixels) spanning a scanned sample surface (Fig. 4A). A HSI can be represented as a data cube, with two spatial dimensions, sized x and y, that represent the pixel coordinates, and a third spectral dimension, sized λ, which represents the number of wavelength channels. To analyze a HSI, such an image cube is generally unfolded in the pixel direction by stacking each spectrum one under the other one to form a data matrix (Fig. 4B). When two or more HSIs need to be analyzed simultaneously (image fusion), a multiset is built by concatenating the spectra of the different HSIs related to the same pixels (blocks $\mathbf{D_2}$ and $\mathbf{D_3}$ in Fig. 4C) [16]. This multiset integrates spectral information from all the individual HSIs, facilitating joint analysis and exploration. Classical image fusion leading to a complete multiset, as in Fig. 4C, requires that the images to be combined cover the same scanned area and have the same pixel size. Thus, sometimes, it is needed to discard the non-common measured areas and lower the spatial resolution yielded by some of the employed imaging techniques in order to equal the pixel size among platforms. When pixels of one image without equivalent information in another platform are to be kept, an incomplete multiset is generated with a missing block of information (block $\mathbf{D_1}$ in Fig. 4D).

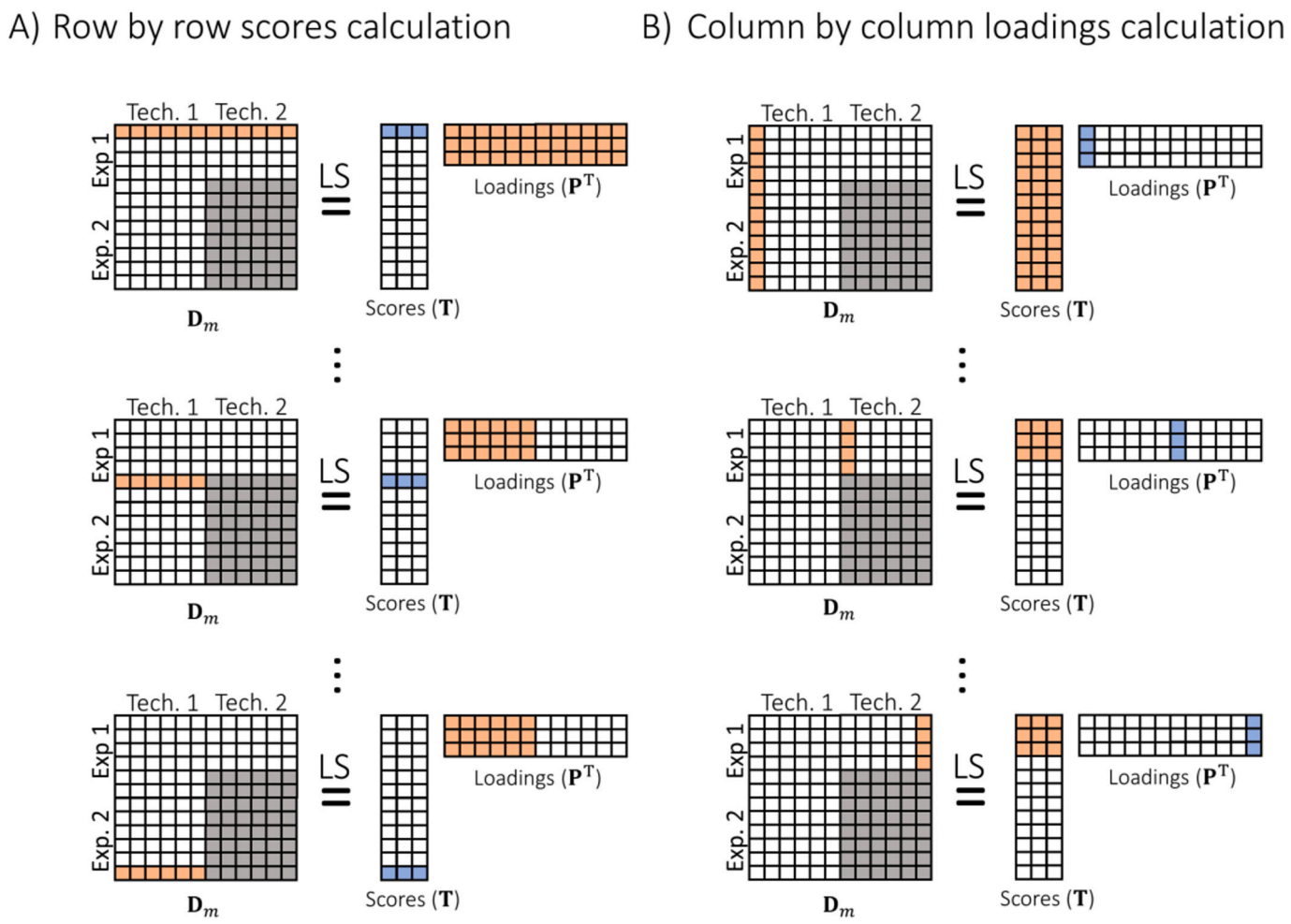

Fig. 3. A) Row-by-row calculation of scores by O-ALS for a three-component system. Given the loadings $\mathbf{P}^T$ and a data row $\mathbf{d}_m(i,:)$ (both in orange), the corresponding score values $\mathbf{t}(i,:)$ are calculated (blue). If missing values are encountered, the loadings are adapted to match the corresponding available entries in $\mathbf{d}_m(i,:)$. B) Column-by-column calculation of loadings by O-ALS for a three-component system. Given the scores $\mathbf{T}$ and a data column $\mathbf{d}_m(:,j)$ (both in orange), the corresponding score values $\mathbf{p}^T(:,j)$ are calculated (blue). Similarly, if missing values are encountered, the scores are adapted to match the corresponding available entries in $\mathbf{d}_m(:,j)$. The calculation is performed for all i and j which go from 1 to R and 1 to C, respectively. Figs. S1 and S2 in the Supplementary Information contain a similar graphical illustration for missing data patterns shown in Fig. 1A and B, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

3.1. Simulated data sets

The simulated incomplete multisets generated here mimic the fusion of a NIR and a Raman image following the scheme in Fig. 4. The images contain three different components and cover a total scanned area of 50 × 50 pixels. The Raman spectra span the spectral range from 700 to 1600 cm^{-1} and include 600 spectral channels. On the other hand, the NIR spectra cover the wavelength range from 935 to 1720 nm, with a total of 60 spectral channels. The pure components employed to generate the simulated data are shown in Fig. 5A. Two different levels of Poisson noise were accounted for (0 % and 7 % of the total simulated signal) to investigate the performance of the three algorithms described before in noise-free conditions and in scenarios encompassing the uncertainty commonly present in real photon-counting experiments. For each noise level, two distinct missing data patterns were considered: a) randomly distributed missing values with varying proportions of missing entries (5 %, 30 %, and 80 %) and b) missing block patterns, where an entire block of data is missing as if Raman measurements were covering only a part of the sample area. Similarly to the randomly distributed missing values case, missing blocks covered a proportion, 5 %, 30 %, and 80 % of the total entries of the data set.

In order to assess the effect of different patterns of missing data on the algorithm performance, two additional simulated case-studies were explored (see Fig. S3 in the Supplementary Information). These additional case-studies provide further evidence that O-ALS is capable of dealing with any percentage and pattern of missing data.

3.2. Real data set

A drawing done with three commercial blue pens, Uniball Signo (US), Bic Velocity Gel (BV) and Pilot V Ball Grip (PVG) on a conventional paper surface was scanned by a NIR and a Raman imaging device (Fig. 5B) [23]. As for the example illustrated in Fig. 4, all the surface of the drawing was scanned by NIR imaging, but only a small part of it by Raman imaging. This generates an incomplete multiset with 70 % of missing values. For this system, a 4-component PCA model is expected to properly capture the variation related to the three inks and the paper.

The NIR hyperspectral image was acquired by a pushbroom NIR camera (Specim FX17 by Spectral Imaging Ltd., Oulu, Finland) and is constituted by 224 spectral channels covering the 935–1720 nm spectral range and 246 × 225 pixels with a pixel size approximately of 106 × 106 μm^2. Savitzky-Golay [24] first-order derivation (second-order polynomial fitting, 15 spectral point-window) was applied to remove baseline signal contributions.

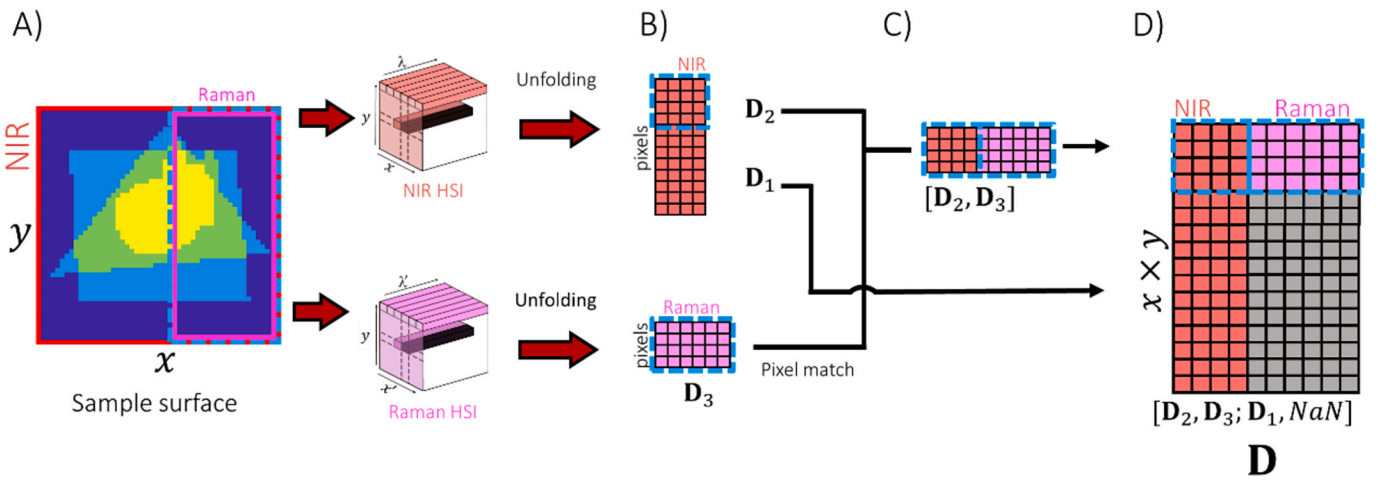

Fig. 4. Simulated case of image fusion A) Representation of the simulated scanned sample. The red square corresponds to the area scanned by NIR imaging. The pink square corresponds to the area scanned by Raman imaging. The dashed blue line denotes the common area scanned by both Raman and NIR imaging. Both HSIs have the same pixel size. B) The NIR and Raman spectra corresponding to the same scanned area, $\mathbf{D}_2$ and $\mathbf{D}_3$, are fused in a single complete multiset (C). Finally, the incomplete multiset is built by concatenating the multiset $[\mathbf{D}_2,\ \mathbf{D}_3]$ and $\mathbf{D}_1$ (representing the sample area scanned only by NIR imaging). The missing block (in gray) is filled with Not-a-Number (*NaN*) and corresponds to the area not scanned by Raman imaging. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

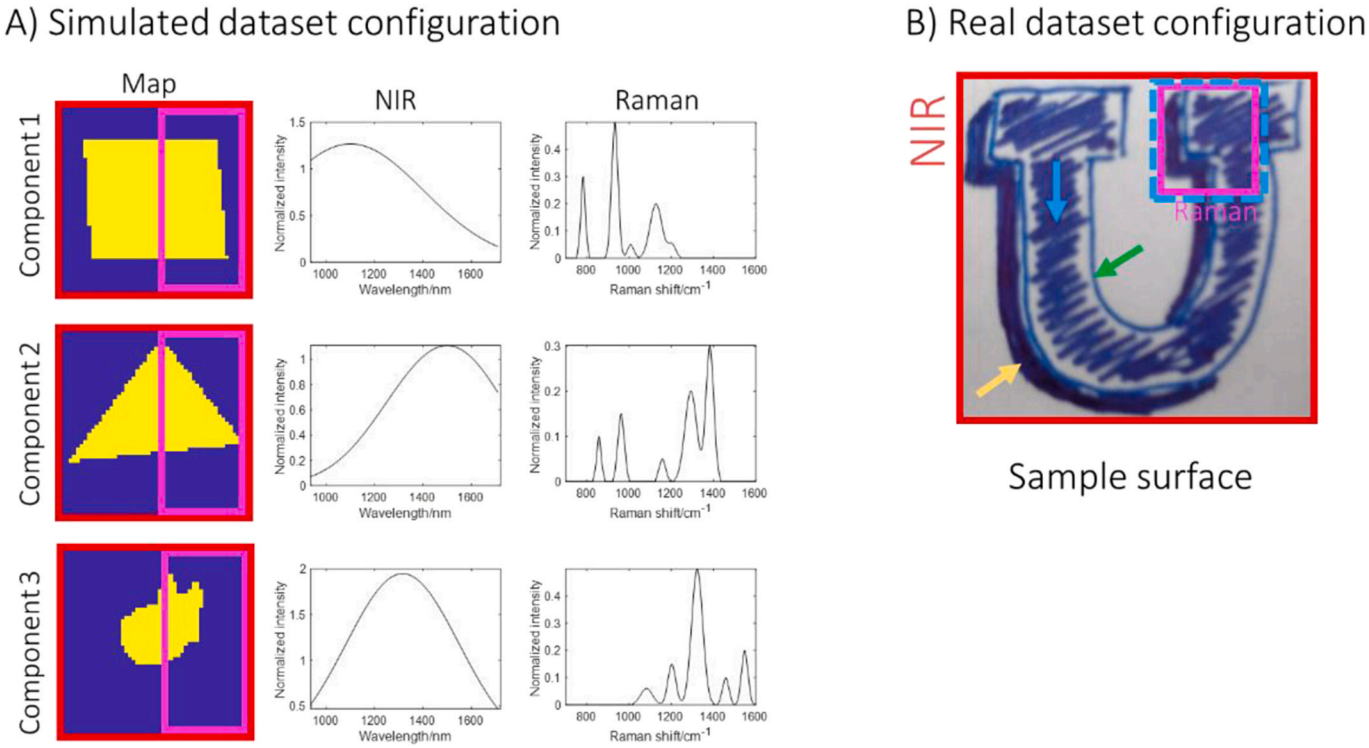

Fig. 5. A) Pure distribution maps and pure spectral profiles of the components used to simulate the image fusion case. The area enclosed by the pink rectangle corresponds to the common area scanned by both NIR and Raman imaging, while the area enclosed by the red rectangle corresponds to the area scanned only by NIR imaging. B) RGB image of the real sample. The yellow, blue and green arrows point at BV, PVG and US inks, respectively. The red square corresponds to the area scanned by NIR imaging. The pink square corresponds to the area scanned by Raman imaging. The dashed blue line highlights the common area scanned by both techniques. Image fusion was performed as in Fig. 4. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

The Raman hyperspectral image was collected using an INVIA RAMAN microscope (RENISHAW, Gloucestershire, UK). The investigated spectral range goes from 270 to 2015 cm^{-1}, with a spectral resolution of 1.55–1.95 cm^{-1} depending on the Raman shift sampled. Pixel size was approximately 26.5 × 26.5 μm^2. Also here, Savitzky-Golay first-order derivation (second-order polynomial fitting, 15 spectral point-window) was applied to remove baseline signal contributions. Raman pixels were binned to achieve the same pixel size as in the NIR image.

Following the scheme in Fig. 4, both images were fused in a single incomplete multiset.

4. Results

The simulated data sets were used to compare the NIPALS, I-SVD and O-ALS algorithms. The performance of the algorithms was assessed by comparing the scores and loadings profiles retrieved from the data set with missing values with those obtained from the corresponding full data set. Besides, the variance explained by the PCA model with a number of components equal to the rank of the simulated data is provided. The results of the analysis of the two additional simulated data sets are reported in Figs. S4 and S5 in the Supplementary Information.

4.1. Results of simulated data sets

The simulated data sets cover the relevant scenarios where the performance of the NIPALS, I-SVD and O-ALS algorithms can be properly tested, i.e., different noise levels (0 % and 7 % of the total generated signal), different patterns of missing data (random entries or missing blocks) and several percentages of missing data (5, 30 and 80 %). Since all algorithms perform an iterative optimization, the convergence criterion based on the difference in data reconstruction error between two consecutive iterations was set to 10^{-11} % (close to the machine precision) to avoid prematurely stopping the calculations. In most cases, random values were chosen as initial estimates. Results are summarized in Tables 1 and 2.

When missing entries are distributed randomly (Table 1), NIPALS, I-SVD and O-ALS perform all satisfactorily (all correlation coefficients between recovered and true scores and loadings profiles were found to be larger than 0.99). Here, some remarks must be considered. First, I-SVD and O-ALS return a perfect match between recovered and true scores and loadings in all scenarios, even in the presence of high percentages of missing data provided that all components are represented by the available information. We found that, if missing entries are randomly distributed, their percentage does not affect the accuracy of the solutions provided by the I-SVD and the O-ALS algorithms. The relevance of the percentage of missing data will be discussed in detail in the following subsections.

Regarding NIPALS, the retrieved scores and loadings profiles show a slight degradation as the order of the principal component extracted increases, as well as it can be observed in the explained variance. When the percentage of missing data and the noise level increase, this effect becomes even more noticeable.

In the presence of missing block patterns (Table 2), NIPALS results in strong degradations of the retrieved scores and loadings in all cases. The results of I-SVD are highly satisfactory in the absence of noise when 5 % and 30 % of missing data are concerned. On the other hand, I-SVD was unable to converge in a reasonable time (<6 h) when dealing with 80 % of missing data. Conversely, the O-ALS algorithm has excellent results for all cases, providing correct scores and loadings profiles in less than a minute.

Looking more carefully at the correlation coefficients in Tables 1 and 2, very interesting aspects could be investigated, e.g., why NIPALS fails, or why I-SVD and O-ALS achieve excellent results even for random patterns of 80 % of missing entries. The behavior of these results is addressed in the following specific subsections.

4.1.1. Presence of bias in the recovered scores and loadings

The bias that affects the components estimated by NIPALS in the presence of missing values was already reported by Bjørn Grung and Rolf Manne [18]. In this case, indeed, it is the one-by-one calculation scheme behind NIPALS that causes the incorrect retrieval of scores and loadings, as also illustrated in Fig. 6.

To understand the reason for this bias, it is important to remind that when $\mathbf{D}_m$ is analyzed, I-SVD and O-ALS estimate a PCA model with a number of components equal to the rank of the data, *N,* i.e., the models obtained are always describing the original space of the data. NIPALS, instead, computes PCA components sequentially and in every step it tries to describe $\mathbf{D}_m$, or the matrices resulting from its deflation, with a rank-1 approximation model. Fitting $\mathbf{D}_m$ skipping the missing entries generates a data space equivalent to impute the model $\mathbf{tp}^T$ on the missing entries of $\mathbf{D}_m$. Thus, when calculating the first component, the missing part of $\mathbf{D}_m$ comes from a rank-1 approximation $\mathbf{tp}^T$, while the rest of $\mathbf{D}_m$ is still rank *N*, being incoherent with the original data structure and generating incorrect components. The same effect occurs when the following components are calculated, but it is even aggravated because the deflation step performed using incorrect components adds to the bias due to the discrepancy between the rank of the available entries in the deflated $\mathbf{D}_m$ and the rank-1 approximation obtained with $\mathbf{tp}^T$ for the missing entries. It is expected then to observe an increase in the degradation of the scores and loadings as more components are extracted, as it can be observed in the results in Tables 1 and 2. Thus, the only instance

Table 1
Correlation coefficients between recovered and true scores and loadings (resulting from the complete data matrix) for the random missing value pattern case. Explained Variance explained by a model with 3 components (rank of the simulated data).

Missing pattern	Algorithm	Component		Noise-free			Noise 7 %		
				Percentage of missing data			Percentage of missing data		
				5 %	30 %	80 %	5 %	30 %	80 %
Random	**NIPALS**	**1**	**Score**	1.0000	0.9999	0.9998	1.0000	0.9998	0.9996
			Loading	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
		2	**Score**	1.0000	0.9998	0.9993	0.9999	0.9980	0.9966
			Loading	1.0000	1.0000	0.9999	1.0000	0.9999	0.9997
		3	**Score**	1.0000	1.0000	0.9996	0.9999	0.9937	0.9909
			Loading	1.0000	1.0000	0.9999	1.0000	0.9984	0.9982
		Explained variance (%)		100.00	99.99	99.97	99.53	99.53	99.51
	I-SVD	**1**	**Score**	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
			Loading	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
		2	**Score**	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
			Loading	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
		3	**Score**	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
			Loading	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
		Explained variance (%)		100.00	100.00	100.00	99.53	99.53	99.54
	O-ALS	**1**	**Score**	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
			Loading	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
		2	**Score**	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
			Loading	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
		3	**Score**	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
			Loading	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
		Explained variance (%)		100.00	100.00	100.00	99.53	99.53	99.54

Table 2
Correlation coefficients between recovered and true scores and loadings (resulting from the complete data matrix) for the missing block pattern case. Variance by a model with 3 components (rank of the simulated data).

Missing pattern	Algorithm	Component		Noise-free			Noise 7 %		
				Percentage of missing data			Percentage of missing data		
				5 %	30 %	80 %	5 %	30 %	80 %
Missing block	**NIPALS**	**1**	**Score**	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
			Loading	0.9997	0.9996	0.9993	0.9997	0.9996	0.9993
		2	**Score**	0.9967	0.9915	0.9829	0.9966	0.9914	0.9829
			Loading	0.9992	0.9976	0.9921	0.9992	0.9976	0.9921
		3	**Score**	0.9836	0.9835	0.9802	0.9826	0.9799	0.9738
			Loading	0.9987	0.9866	0.9600	0.9985	0.9861	0.9587
		Explained variance (%)		99.99	99.98	99.98	99.54	99.59	99.62
	I-SVD	**1**	**Score**	1.0000	1.0000	-[a]	1.0000	1.0000	1.0000[a]
			Loading	1.0000	1.0000	-[a]	1.0000	1.0000	1.0000[a]
		2	**Score**	1.0000	1.0000	-[a]	0.9999	0.9996	0.9994[a]
			Loading	1.0000	1.0000	-[a]	1.0000	1.0000	0.9998[a]
		3	**Score**	1.0000	1.0000	-[a]	0.9983	0.9956	0.9936[a]
			Loading	1.0000	1.0000	-[a]	1.0000	0.9998	0.9986[a]
		Explained variance (%)		100.00	100.00	-[a]	99.55	99.61	99.64
	O-ALS	**1**	**Score**	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
			Loading	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
		2	**Score**	1.0000	1.0000	1.0000	0.9999	0.9996	0.9994
			Loading	1.0000	1.0000	1.0000	1.0000	1.0000	0.9998
		3	**Score**	1.0000	1.0000	1.0000	0.9983	0.9956	0.9936
			Loading	1.0000	1.0000	1.0000	1.0000	0.9998	0.9986
		Explained variance (%)		100.00	100.00	100	99.55	99.61	99.64

[a] NIPALS solutions were here used as initial estimates due to the fact that I-SVD did not converge when initialized with random values.

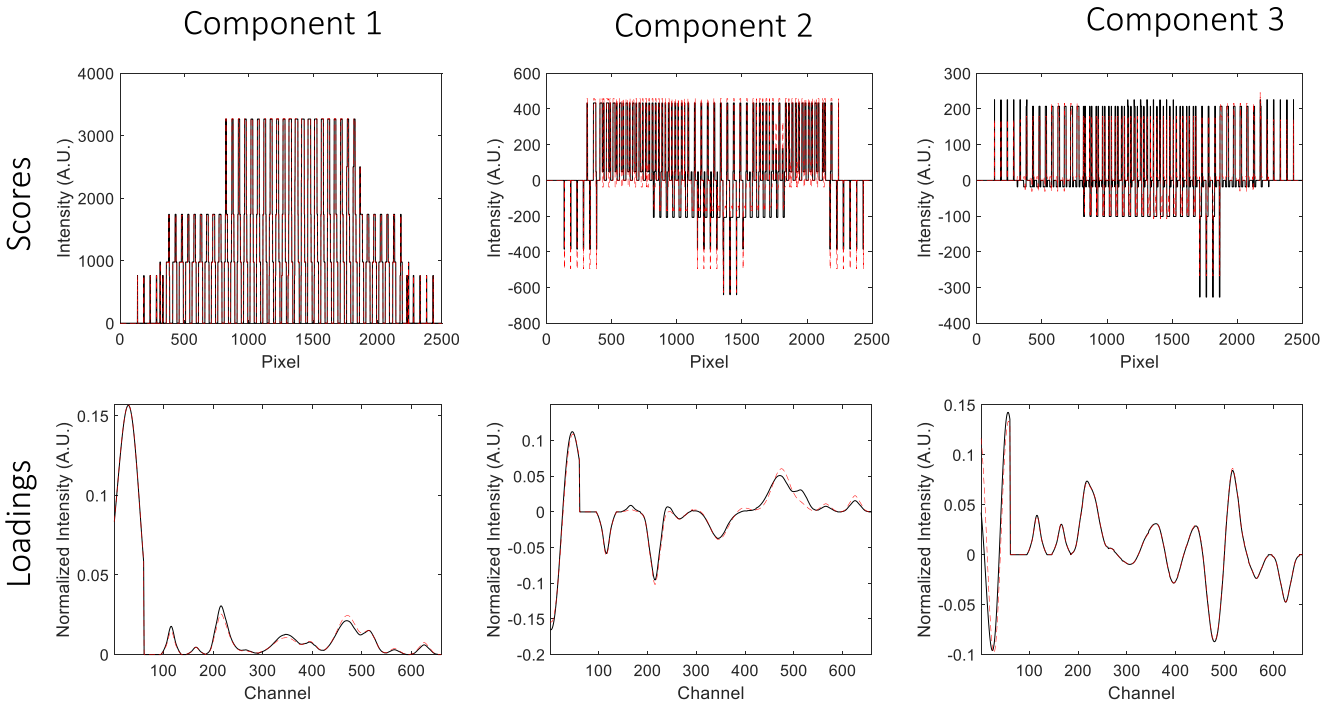

Fig. 6. Scores and loadings extracted by NIPALS (dashed red lines) and scores and loadings of the complete data matrix (solid black lines) for the noise-free missing block case encompassing 80 % of missing entries. Small deviations between NIPALS and true scores and loadings profiles can already be observed for the first principal component. These deviations significantly increase for the second and the third component. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

in which the use of the adapted NIPALS algorithm works in the presence of missing data is when the rank of $\mathbf{D}_m$ is equal to 1.

Additional evidence that supports the inaccurate extraction of scores and loadings by NIPALS are the non-orthogonality among the scores and loadings profiles obtained and the variance explained by the PCA model with the correct rank *N,* which is always lower than the variance expected and properly described by analogous I-SVD or O-ALS PCA models with the same number of components.

Although the bias mentioned above always occurs, the error induced by NIPALS can be more manageable when the number of missing entries and components are low, as it is shown in the results for the case of 5 % of missing entries. However, even in these conditions, the resulting bias is strongly data-dependent.

4.1.2. Effect of the pattern and the percentage of missing data on the obtained scores and loadings

The fundamental concept behind PCA is based on the assumption that the investigated data exhibit an underlying structure that can be captured by a low-rank (*N*-dimensional) representation of the systematic information they encode. When missing values are present, I-SVD works by imputing the missing values so that such a low-rank representation remains valid. On the other hand, O-ALS use only the available information to directly estimate the scores and loadings of this *N*-dimensional approximation. It is important to point out that both methods span the same low-dimensional data space (Fig. 7) which means that if the available information is sufficient to describe well this space, correct imputations will be achieved and correct scores and loadings will be retrieved. As mentioned in the previous subsection, the recovered scores and loadings with the NIPALS-based algorithm (in red in Fig. 7) show the bias already described.

Since the available information in the incomplete simulated data is enough to define properly their *N*-dimensional subspace, scores and loadings profiles in perfect agreement with the expected ones are obtained in the noise-free case (see Tables 1 and 2 and Fig. 7). Thus, we found that there is no dependence either of the missing pattern or of the percentage of missing values, but of the information contained in the available data.

Despite the correct retrieval of scores and loadings by I-SVD, the imputation step required by this algorithm may lead to an extremely slow convergence in practice. Indeed, due to the nature of the I-SVD algorithm, the imputed values play a relevant role in every iteration of the PCA model estimation. If the percentage of missing values is high and the missing block pattern does not allow for an easy imputation of the missing entries, the high leverage of these estimated values compared to the weight of the original available entries will result in an extremely long convergence time (hours or even days), making the use of this algorithm impractical. This is what occurs in the situation encompassing 80 % of missing data with a missing block pattern where a solution could not be found for the noise-free case. For the case in the presence of noise, the reconstructed model by the NIPALS solution had to be used as initial estimate for I-SVD, otherwise the algorithm did not converge in a reasonable time (hours at least).

The O-ALS method is not affected by the slow convergence problem since its *modus operandi* ignores the missing entries instead of pushing the predicted missing entries to the correct dataspace, thus saving time (see Fig. S6 in the Supplementary Information). In fact, O-ALS shows a very similar computation time for all percentages of missing entries, contrarily to algorithms based on imputation. To see clearly this effect, the data sets with a 7 % noise added following a random and a missing block pattern were analyzed 500 times by O-ALS and I-SVD using different random initial estimates in each run (Table 3).

The slow convergence of I-SVD for data sets with high percentage of missing entries is not observed when the missing value pattern is random (Table 3) and despite O-ALS was found to be always faster, no significant differences between the two approaches were found in practical terms. This can be easily explained in the light of the fact that good imputations can often be achieved when the missing entries follow random patterns since in such cases the underlying data structure of **D** is commonly very well preserved in $\mathbf{D}_m$.

On the other hand, when complete blocks of information are missing, the available information in $\mathbf{D}_m$ does not allow for a fast estimate of the missing entries. In this context, the results show that O-ALS spent 17 ± 6, 20 ± 21 and 19 ± 11s for the data sets containing 5, 30 and 80 % of

Table 3
Mean and standard deviation of the time spent by each algorithm in 500 runs for data sets with 7 % noise added. Random values were used as initial estimates.

Algorithm	Missing value random pattern			Missing block pattern		
	5 %	30 %	80 %	5 %	30 %	80 %
I-SVD	10±1s	27±3s	72±2s	114s ± 12s	17 ±1min	–
O-ALS	4 ± 0.8s	2.5 ± 0.6s	2.2 ± 0.5s	17±6s	20 ± 21s	19 ± 11s

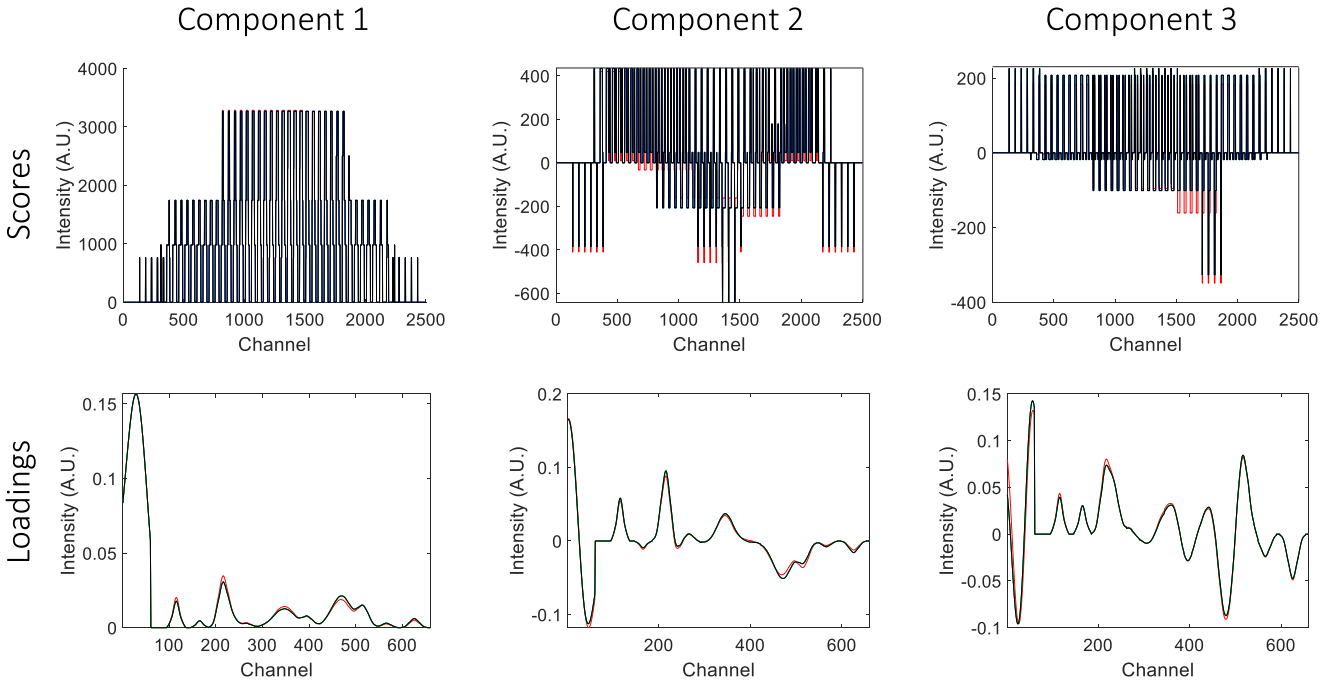

Fig. 7. Scores and loadings extracted by NIPALS (solid red lines), extracted by I-SVD (solid green lines), extracted by O-ALS (solid blue lines) and scores and loadings of the complete data matrix (solid black lines) for the noise-free missing block case encompassing 30 % of missing entries. Note that I-SVD, O-ALS and true profiles perfectly overlap. I-SVD and O-ALS algorithms yielded scores and loadings in perfect agreement with those obtained when the full data matrix was analyzed. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

missing values, respectively, whereas I-SVD took 114 ± 12s and 17 ± 1 min for the data sets containing 5 and 30 % of missing values, respectively. However, I-SVD was unable to converge in a reasonable time (>6 h) when dealing with the data including for the 80 % missing block pattern, becoming impractical for the analysis.

Whereas I-SVD can suffer from an excessively slow convergence, a possible limitation of the O-ALS algorithm is the possibility to fall into local minima. Possible local minima have been detected for the O-ALS algorithm when the missing block pattern was analyzed. Out of 500 runs, 257, 159 and 148 for the 5 %, 30 % and 80 % missing value-case, respectively, got stuck in a possible local minimum, detected because the variance explained by the model was slightly lower than the one expected. No local minima were detected for data random missing value patterns. The presence of local minima for bilinear factorizations when missing data are present was demonstrated by Ilin and Raiko in 2010 [25]. If this is the case, the high speed of the O-ALS algorithm easily solves the problem. Thus, the algorithm can be initialized several times during few iterations using different initial estimates each time. Then, the initial estimates that provided the best fit are selected and used to perform the full analysis until convergence is achieved. This enables the retrieval of accurate PCA models without increasing significantly the analysis time. Indeed, in the worst-case scenario handled here (80 % of missing data, missing block pattern) and using the best of 10 different initializations (as assessed after 35 iterations), the frequency of local minima was reduced from 32 % to <1 % while the analysis time increased from 19s to 120s.

4.2. Real data set

The O-ALS algorithm was applied to the real data set described in Section 3 which contains 70 % of missing values. The convergence criterion based on the difference in error among between two consecutive iterations was set as 10^{-11} % (close to the machine precision) to avoid stopping prematurely the algorithm. Random values were chosen as initial estimates and four principal components were extracted. Ten different random initializations were performed to avoid possible local minima. Results are shown in Fig. 8.

The variance explained by each retrieved component was 62.48 %, 25.87 %, 4.55 % and 0.47 %, respectively. The total variance explained by the model was 93.37 %, a reasonable percentage considering the typical noise content of real NIR and Raman images.

Fig. 8A displays the score plots of the solution provided by O-ALS. In the score plots, several clusters associated with the pixels of the pen inks and with the pixels of the paper can be observed in the PCA space. In a general view, the score plots show the presence of clusters. This may indicate the existence of pixels that contains similar spectra, which are attributed to the same ink. The first PC was not shown because of the lack of informative relevance (related to describe the mean of the data since the data set was not mean-centered). The PC 3 vs PC 2 score plot allows differentiating the pixels of paper (black circle) and of the US ink (green circle) from pixels of the rest of inks (see Fig. 8B for the corresponding representation of the position of the pixels of a cluster on the area scanned using the optical image). The signal from the US pen is very clear and selective in both NIR and Raman techniques; hence the differentiation in lower PCs.

However, the rest of the pixels related to the other pens remains mixed. This is due to the fact that the information distinguishing between US and the rest of the ink contains less variance compared to that distinguishing between inks with highly dissimilar spectral signatures. In other words, most of the channels already contain explained variance from the preceding PCs. As a result, the differentiation between the similar ink appears more prominently in the lower PCs. This can be observed in the PC 4 vs PC 3 score plot, which differentiates the pixels corresponding to US ink (green circle), BV ink (blue circle) and it can be slightly observed the cluster of the PVG ink (yellow circle). Finally, the PC 4 vs PC 2 score plot allows a clear separation of the pixels of the PVG ink (yellow circle) from others. The signals from PVG, BV and the paper are not very selective in Raman, and not selective at all in NIR. However,

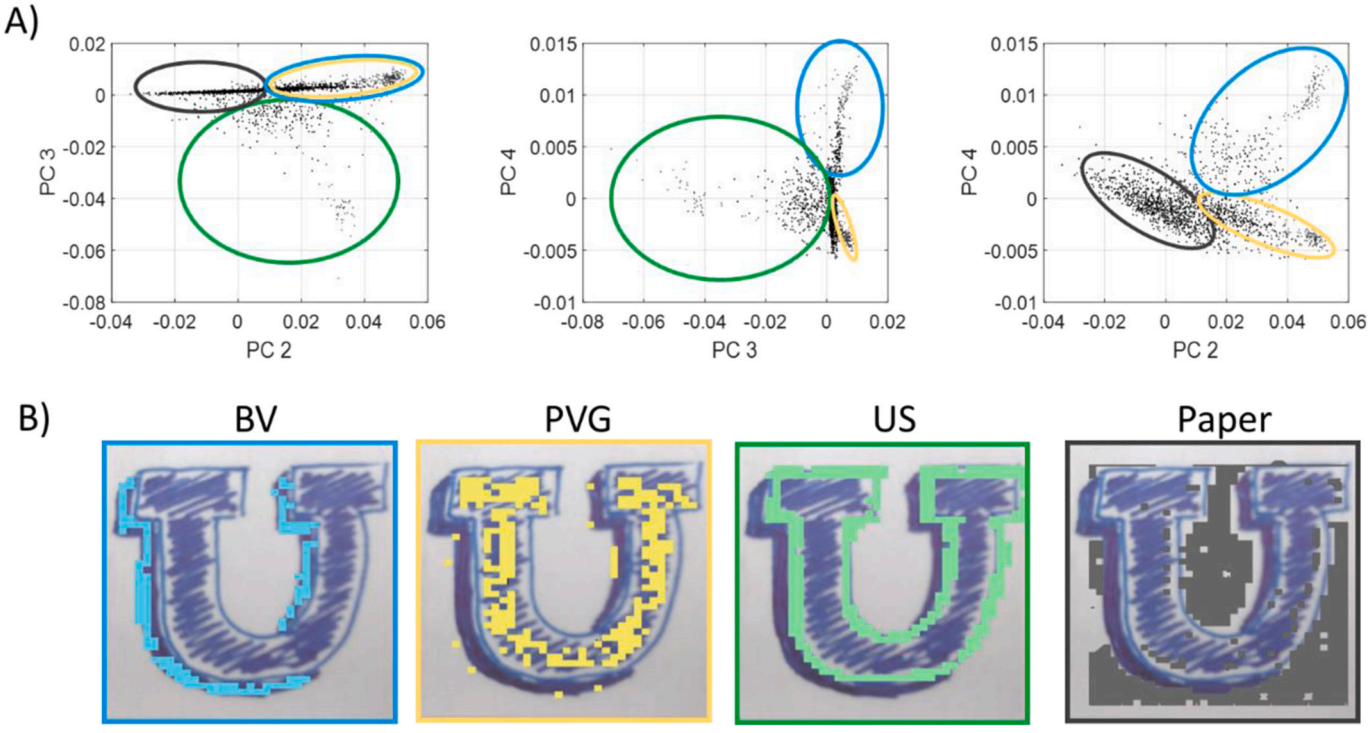

Fig. 8. A) Second, third and fourth principal component scores extracted by O-ALS. The presence of four different HSI pixel clusters can be observed: paper pixels (black circle), US ink pixels (green circle), BV ink pixels (blue circle) and PVG ink pixels (yellow circle). B) Spatial location of the pixels belonging to the four aforementioned clusters (BV ink – cyan pixels; PVG ink – yellow pixels; US ink – green pixels; paper – black pixels). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

the differences are enough to cluster correctly the pixels of all components.

Summarizing, the analysis suggests that O-ALS effectively detected the four expected clusters related to the signals from the three pen inks and the paper and captured the PCA space of the dataset despite the high percentage of missing entries and the challenging missing block pattern.

5. Conclusions

In this study, the new Orthogonalized- Alternating Least Squares algorithm (O-ALS) has been presented as a fast and accurate algorithm to provide PCA models for data sets with any kind of pattern and percentage of missing entries. O-ALS works only with the available entries in the data set and estimates simultaneously all necessary components in the PCA model using an alternating least-squares optimization of the scores and loadings under the Gram-Schmidt orthogonalization constraint. As for any soft-modeling method, the O-ALS algorithm provides correct scores and loadings as long as the available data contains information about all components sought.

Comparing the performance of O-ALS with the NIPALS and I-SVD algorithms in the same scenarios, some characteristics are key to understand the differences in the results provided by the different methodologies. A relevant fact is that O-ALS and I-SVD work always extracting simultaneously the N components required for the PCA model, whereas NIPALS proceeds with the one-at-a-time sequential extraction of components. Working in the correct rank-N space results in the consistent retrieval of accurate scores and loadings of O-ALS and I-SVD across various scenarios differing in noise level, pattern and percentage of missing entries. NIPALS instead suffers from a bias in the retrieved scores and loadings due to the rank-1 approximation used in the extraction of the components and a loss of orthogonality among the extracted profiles.

A fundamental difference between I-SVD and O-ALS is that the former works imputing the missing entries and the latter using only the available information. Whereas the *modus operandi* of the algorithms does not influence the accuracy of the results obtained, it has a significant effect in the computation time and convergence among them. O-ALS remains a fast algorithm whatever the pattern and percentage of missing entries in the data. Instead, the convergence of the I-SVD algorithm is compromised for data sets with high percentage of missing entries and complex missing block patterns. Finally, the only limitation identified for the O-ALS algorithm is the possibility to fall in local minima in extreme cases of missing block patterns with high percentage of missing entries. In this instance, starting with a small number of initializations for a few iterations and selecting the model with the best fit to obtain the definitive PCA model clearly solves the problem.

Although the O-ALS algorithm is suitable for any kind of data set with missing entries, a very promising application field is image fusion, where missing block patterns always exist, the proportion of missing values can reach up to 80 % and the size of the data sets can be massive. In this scenario, the fast and accurate O-ALS algorithm can be an excellent tool for exploratory purposes and to, eventually reconstruct the missing blocks of information if required.

CRediT authorship contribution statement

Adrián Gómez-Sánchez: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Raffaele Vitale:** Writing – review & editing, Investigation, Conceptualization. **Cyril Ruckebusch:** Writing – review & editing, Software, Resources, Investigation, Conceptualization. **Anna de Juan:** Writing – review & editing, Writing – original draft, Supervision, Resources, Project administration, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

A. G.-S. warmly thanks Martina Beese, PhD student at the Leibniz-Institut für Katalyse and at the Institut für Mathematik of the University of Rostock for fruitful discussions. A. G.-S. and A. d. J. acknowledge financial support from the Catalan government (2021 SGR 00449). A. G.-S. acknowledges scholarships from the MOBLILEX ULille program, from Santander bank and from Fundació Montcelimar.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.chemolab.2024.105153.

References

[1] I.T. Jolliffe, J. Cadima, Principal component analysis: a review and recent developments, Phil. Trans. Math. Phys. Eng. Sci. 374.2065 (2016) 20150202.
[2] S. Wold, K. Esbensen, P. Geladi, Principal component analysis, Chemometr. Intell. Lab. Syst. 2 (1–3) (1987) 37–52.
[3] J. Camacho, J. Picó, A. Ferrer, Data understanding with PCA: structural and variance information plots, Chemometr. Intell. Lab. Syst. 100 (1) (2010) 48–56.
[4] H. Grahn, P. Geladi (Eds.), Techniques and Applications of Hyperspectral Image Analysis, John Wiley & Sons, 2007.
[5] J.M. Amigo, I. Martí, A. Gowen, Hyperspectral imaging and chemometrics: a perfect combination for the analysis of food structure, composition and quality, Data Handling Sci. Technol. 28 (2013) 343–370. Elsevier.
[6] B. Torres-Cobos, et al., Varietal authentication of virgin olive oil: proving the efficiency of sesquiterpene fingerprinting for Mediterranean Arbequina oils, Food Control 128 (2021) 108200.
[7] R. Tauler, E. Casassas, Principal component analysis applied to the study of successive complex formation data in Cu (II)–ethanolamine systems, J. Chemometr. 3 (S1) (1989) 151–161.
[8] S. Wold, M. Sjöström, SIMCA: a method for analyzing chemical data in terms of similarity and analogy. Vol. 52, ACS (Am. Chem. Soc.) Symp. Ser. (1977) 243–282.
[9] C.R. Avila, et al., Process monitoring of moisture content and mass transfer rate in a fluidised bed with a low-cost inline MEMS NIR sensor, Pharmaceut. Res. 37 (2020) 1–19.
[10] T. Kourti, Quality by design in the pharmaceutical industry: process modelling, monitoring and control using latent variable methods, IFAC Proceedings 42 (11) (2009) 36–41.
[11] L.N. Trefethen, D. Bau, Numerical Linear Algebra, SIAM, Philadelphia, PA, 1997.
[12] M. Alier, R. Tauler, Multivariate curve resolution of incomplete data multisets, Chemometr. Intell. Lab. Syst. 127 (2013) 17–28.
[13] M. De Luca, G. Ragno, G. Ioele, R. Tauler, Multivariate curve resolution of incomplete fused multiset data from chromatographic and spectrophotometric analyses for drug photostability studies, Anal. Chim. Acta 837 (2014) 31–37.
[14] S. Piqueras, et al., Handling different spatial resolutions in image fusion by multivariate curve resolution-alternating least squares for incomplete image multisets, Anal. Chem. 90 (2018) 6757–6765.
[15] B. Walczak, D.L. Massart, Dealing with missing data: Part I, Chemometr. Intell. Lab. Syst. 58 (1) (2001) 15–27.
[16] A. de Juan, R.R. de Oliveira, A. Gómez-Sánchez, Multiset analysis by multivariate curve resolution: the unmixing methodology to handle hyperspectral image fusion scenarios, Data Handling Sci. Technol. 33 (2024) 111–132. Elsevier.
[17] M. Bahram, et al., Handling of Rayleigh and Raman scatter for PARAFAC modeling of fluorescence data using interpolation, J. Chemometr.: A J. Chem. Soc. 20 (3-4) (2006) 99–105.
[18] B. Grung, R. Manne, Missing values in principal component analysis, Chemometr. Intell. Lab. Syst. 42 (1–2) (1998) 125–139.
[19] H. Wold, Soft modeling by latent variables: the nonlinear iterative partial least squares (NIPALS) approach, J. Appl. Probab. 12 (S1) (1975) 117–142.
[20] A. Christoffersson, The One Component Model with Incomplete Data, University of Uppsala, Sweden, 1970. Doctoral Thesis.
[21] E. Schmidt, Zur Theorie der linearen und nichtlinearen Integralgleichungen I. Teil: Entwicklung willkürlicher Funktionen nach Systemen vorgeschriebener, Math. Ann. 63 (1907) 433–476.

[22] Gram, J. P. Ueber die Entwickelung reeller Functionen in Reihen mittels der Methode der kleinsten Quadrate. J. für die Reine Angewandte Math. (Crelle's J.) 94 (1883): 41–73.
[23] F.D.S.L. Borba, R.S. Honorato, A. de Juan, Use of Raman spectroscopy and chemometrics to distinguish blue ballpoint pen inks, Forensic Sci. Int. 249 (2015) 73–82.
[24] A. Savitzky, M.J. Golay, Smoothing and differentiation of data by simplified least squares procedures, Anal. Chem. 36 (8) (1964) 1627–1639.
[25] A. Ilin, T. Raiko, Practical approaches to principal component analysis in the presence of missing values, J. Mach. Learn. Res. 11 (2010) 1957–2000.