

Multivariate functional outlier detection using the fast massive unsupervised outlier detection indices

Oluwasegun Taiwo Ojo^{1,2}  | Antonio Fernández Anta¹  | Marc G. Genton³  | Rosa E. Lillo^{4,5} 

¹IMDEA Networks Institute, Leganés, Madrid 28918, Spain

²Universidad Carlos III de Madrid, Getafe, Madrid 28903, Spain

³Statistics Program, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia

⁴uc3m-Santander Big Data Institute, Getafe, Madrid 28903, Spain

⁵Department of Statistics, Universidad Carlos III de Madrid, Getafe, Madrid 28903, Spain

Correspondence

Oluwasegun Taiwo Ojo, IMDEA Networks Institute, Leganés, Madrid 28918, Spain.
 Email: oluwasegun.ojo@imdea.org

Funding information

Comunidad de Madrid; King Abdullah University of Science and Technology; Ministry of Science and Innovation - State Research Agency, Spain, Grant/Award Number: PID2019-104901RB-I00

We present definitions and properties of the fast massive unsupervised outlier detection (FastMUOD) indices, used for outlier detection (OD) in functional data. FastMUOD detects outliers by computing, for each curve, an amplitude, magnitude, and shape index meant to target the corresponding types of outliers. Some methods adapting FastMUOD to outlier detection in multivariate functional data are then proposed. These include applying FastMUOD on the components of the multivariate data and using random projections. Moreover, these techniques are tested on various simulated and real multivariate functional datasets. Compared with the state of the art in multivariate functional OD, the use of random projections showed the most effective results with similar, and in some cases improved, OD performance. Based on the proportion of random projections that flag each multivariate function as an outlier, we propose a new graphical tool, the magnitude-shape-amplitude (MSA) plot, useful for visualizing the magnitude, shape and amplitude outlyingness of multivariate functional data.

KEYWORDS

FastMUOD, functional data, functional outlier detection, multivariate functional data, outlier classification, video data

1 | INTRODUCTION

We consider the problem of detecting outliers in a collection of multivariate functional observations. In particular, we consider observations of the form: $\{Y_i(t), t \in \mathcal{I}\}_{i=1}^n$, wherein a vector $Y_i(t) \in \mathbb{R}^d, d \in \mathbb{N}$, is observed at a domain point t in the interval \mathcal{I} . Such vector-valued functional observations are increasingly observed in real-life studies and various physical and environmental applications. Thus, exploratory methods for multivariate functional data have been recently garnering considerable interest.

Outlier detection (OD), a part of the exploratory data analysis process, involves identifying observations that differ from the bulk of the data, either because they come from a different distribution compared with the bulk or because they lie at the extremes of the distribution of the data. However, identifying outliers is more complicated when observations are functions observed on a domain, that is, functional data. Functional observations demonstrate different outlying behaviours, for example, a vertical shift, compared with the bulk of the data (magnitude outliers) or a horizontal shift, in which case the outlying function is not well aligned with the bulk of the data. Functional outliers can also have different shapes or follow different paths compared with the bulk of the data. Hubert et al. (2015) proposed a taxonomy for different types of functional outliers based on the different outlying behaviours they exhibit, and whether such behaviours can be observed in a small part of the domain or throughout the domain (see also Dai et al., 2020).

To identify outliers among multivariate (nonfunctional) observations (i.e., vector observations $X \in \mathbb{R}^d$), it is typical to order the observations, from the centre outward using a notion of *statistical depth*. Then, the observations having the lowest depth values can be closely examined for

outlying behaviours. This procedure is convenient because most depth notions are nonparametric and they do not require any assumption concerning the underlying data distribution.

The approach mentioned above has also caught on in the analysis of functional observations, where several OD methods are based on notions of functional depths. For example, the *functional boxplot* (Sun & Genton, 2011) uses the modified band depth (López-Pintado & Romo, 2009) to order functional observations and define a 50% central region. Then, the outliers are functions that lie outside of the central region inflated by 1.5, similar to the classical boxplot. Other proposals around this theme include Sguera et al. (2016) and Febrero et al. (2008), where functional depth measures were used for OD.

On the other hand, several functional OD methods are based on “custom-built” outlyingness indices, metrics, or pseudo-depths directly targeted towards OD, instead of ordering (as with functional depth notions). Examples along this line include the *magnitude-shape plot* (MS-plot) (Dai & Genton, 2018), based on the directional outlyingness proposed by Dai and Genton (2019); the *functional outlier map* (FOM), based on another (functional) directional outlyingness proposed by Rousseeuw et al. (2018); the *modified shape similarity index* (MSS) proposed in Huang and Sun (2019); the (robustified) *functional tangential angle* (rFUNTA) proposed in Kuhnt and Rehage (2016); and an earlier proposal of Hubert et al. (2015) in which the *bag distance* and *skewness adjusted projection depth* were proposed for functional OD.

Finally, certain functional OD procedures are based on either a combination of depth notions and outlyingness indices, or the use of more primitive methods (such as dimension reduction or transformation). Some of these include the *outliergram*, based on the modified epigraph index (López-Pintado & Romo, 2011) and the modified band depth; the *functional bagplots*, and the *functional highest density regions* (Hyndman & Shang, 2010), both using the first two robust principal components of the functional data to construct plots used for detecting functional outliers. Likewise, Dai et al. (2020) proposed detecting functional outliers using a sequence of (functional) data transformations, each followed by a functional boxplot to detect different types of outliers. Recently, Herrmann and Scheipl (2021) proposed using multidimensional scaling (Cox & Cox, 2008) to reduce functional data to lower dimensional embeddings. Then, an OD method such as the local outlier factors (Breunig et al., 2000) was applied on the embeddings to detect outlying curves.

Fast massive unsupervised outlier detection (FastMUOD), introduced by Ojo et al. (2021), belongs to the second group of functional OD methods (outlined above) because it uses three indices, each targeting different outlying behaviours that functional outliers may exhibit. The FastMUOD indices are the *magnitude index*, which targets magnitude outliers; the *shape index*, which targets shape outliers; and the *amplitude index*, which targets amplitude outliers. Because these indices target different outlier types, the outliers identified are also classified as per their types, unsupervised, without the need for inspection or visualisation of the data. The method is fast and simple, making it scalable to (and suitable for) “big” functional data analysis.

Nevertheless, despite its advantages, FastMUOD has its limitations. First, its indices are designed for univariate functional data. Second, it is not exactly clear from Ojo et al. (2021) why the FastMUOD indices are suitable for OD from a theoretical perspective, despite the good and scalable performance observed on simulated and real datasets. In this study, we aim to address these two issues by (1) exploring the properties of the FastMUOD indices rigorously and by (2) extending the FastMUOD indices to OD in multivariate functional data.

The rest of the article is organised as follows. In Section 2, we present definitions and properties of the FastMUOD indices. Next, in Section 3, we describe several extensions of the FastMUOD indices to outlier detection in multivariate functional data. In Section 4, we evaluate the extensions presented in Section 3 in a simulation study. We demonstrate the extensions on two real data applications in Section 5. We end the article with some concluding remarks in Section 6.

2 | DEFINITIONS AND PROPERTIES OF THE FASTMUOD INDICES

We assume that functions are defined on the unit interval $[0,1]$ and denote by $L^2([0,1])$ the space of all square-integrable functions defined over $[0,1]$. We denote by $\langle f, g \rangle$ the inner product of $f, g \in L^2([0,1])$. The norm induced by $\langle \cdot, \cdot \rangle$ is denoted by $\|\cdot\|$.

2.1 | Definitions of the univariate FastMUOD indices

Definition 1 **Definitions of univariate FastMUOD indices.** Let X be a stochastic process in $L^2([0,1])$ with distribution F_X and $\mu(t) = \mathbb{E}[X(t)]$ be its population mean function. We define the **shape index** of a function $y \in L^2([0,1])$ (which may be a realisation of X) with respect to (w.r.t.) F_X as

$$I_S(y, F_X) := 1 - \frac{\int \tilde{y}(t) \tilde{\mu}(t) dt}{\left[\int \tilde{y}(t)^2 dt \right]^{1/2} \left[\int \tilde{\mu}(t)^2 dt \right]^{1/2}} = 1 - \frac{\langle \tilde{y}, \tilde{\mu} \rangle}{\|\tilde{y}\| \cdot \|\tilde{\mu}\|},$$

where $\tilde{y}(t)$ and $\tilde{\mu}(t)$ denote the centred curves given by: $\tilde{y}(t) := y(t) - \int y(r)dr$, and $\tilde{\mu}(t) := \mu(t) - \int \mu(r)dr$, respectively. We define the **amplitude index** of y w.r.t. F_X as

$$I_A(y, F_X) := \frac{\int \tilde{y}(t)\tilde{\mu}(t)dt}{\int \tilde{\mu}(t)^2 dt} - 1 = \frac{\langle \tilde{y}, \tilde{\mu} \rangle}{\|\tilde{\mu}\|^2} - 1.$$

Finally, we define the **magnitude index** of a function y w.r.t. F_X as

$$I_M(y, F_X) := \int y(t)dt - \beta(y) \int \mu(t)dt,$$

where $\beta(y) = I_A(y, F_X) + 1$.

In practice, functions are usually observed on a finite number of points in the domain. In this case, a discrete approximation to the FastMUOD indices can be obtained by replacing the integral with a summation. Moreover, the sample versions of the FastMUOD indices can be defined by replacing the mean function, μ , of X with an appropriate empirical estimate, for example, the pointwise mean function, given by $\bar{X}(t) = n^{-1} \sum_{i=1}^n X_i(t)$. In fact, any central function with a consistent sample estimate (e.g., the median function defined by a depth) can be used in the definitions of the FastMUOD indices. We provide a sample version and discrete approximation of the FastMUOD indices in the Supporting Information (Section S1).

2.2 | Properties of the univariate FastMUOD indices

We present some properties of the FastMUOD indices—under certain simple transformations—which makes them ideal for detecting outliers.

Proposition 1 **Properties of univariate FastMUOD indices.** *Let X be a stochastic process in $L^2([0,1])$ with distribution F_X and mean function $\mu(t)$. Let y and z be other functions in $L^2([0,1])$ (which may be realisations of X) and let $a, b \in \mathbb{R}$. Then, the following statements hold:*

1. For a new function $y'(t) = ay(t) + b$ we have $I_M(y', F_X) = aI_M(y, F_X) + b$; $I_A(y', F_X) = aI_A(y, F_X) + a - 1$; and if $a \neq 0$, then $I_S(y, F_X) = I_S(y', F_X)$.
2. For a new function $y'(t) = y(t) + z(t)$ we have $I_M(y', F_X) = I_M(y, F_X) + I_M(z, F_X)$; $I_A(y', F_X) = I_A(y, F_X) \Leftrightarrow \langle \tilde{z}, \tilde{\mu} \rangle = 0$; and $I_S(y, F_X) = I_S(y', F_X) \Leftrightarrow \frac{\langle \tilde{y}, \tilde{\mu} \rangle}{\|\tilde{y}\|} = \frac{\langle \tilde{y}, \tilde{\mu} \rangle + \langle \tilde{z}, \tilde{\mu} \rangle}{\|\tilde{y} + \tilde{z}\|}$.
3. For a new function $y'(t) = z(t)y(t)$, we have $I_A(y, F_X) = I_A(y', F_X) \Leftrightarrow \langle \tilde{y}, \tilde{\mu} \rangle = \langle \tilde{z}\tilde{y}, \tilde{\mu} \rangle$; and $I_S(y, F_X) = I_S(y', F_X) \Leftrightarrow \frac{\langle \tilde{y}, \tilde{\mu} \rangle}{\|\tilde{y}\|} = \frac{\langle \tilde{z}\tilde{y}, \tilde{\mu} \rangle}{\|\tilde{z}\tilde{y}\|}$.

Proof. See Section S2 of the Supporting Information. □

Proposition 1 provides insights into how the different FastMUOD indices behave under transformations and hence why they are useful for targeting the corresponding types of outliers. The first property (Proposition 1.1) demonstrates that I_M is sensitive to the translation and scaling of a function (by real numbers), which is a desirable property, because I_M is intended to be a measure of magnitude outlyingness; it should consequently capture any magnitude shift to be such a worthy measure (of magnitude outlyingness). This property is shown in the first row of Figure 1 where a realisation $y(t)$, out of 100, of a process $X_i(t)$ is transformed by scaling ($a = 2$) and shifting it ($b = 3$). The second column of Figure 1 shows that the index of the transformed function ($I_M(y', F_X)$) is equal to that of $y(t)$ scaled and shifted with same values ($aI_M(y, F_X) + b$).

Proposition 1.2, also illustrated in Figure 1, shows that I_M preserves the functional addition operation, which is desirable because functional addition causes a shift in magnitude, and this shift is captured by the I_M . Thus, for $y'(t) = y(t) + z(t)$, $I_M(y') = I_M(y) \Leftrightarrow I_M(z) = 0$.

Unlike I_M , I_A is not sensitive to shifting by a scalar (Proposition 1.1). Because shifting a (periodic) function does not inherently change its amplitude, a good measure of amplitude outlyingness should ignore such transformation. However, the index I_A is sensitive to scaling as this transformation changes the amplitude of a function. In fact, Proposition 1.1 indicates that for $y'(t) = ay(t)$, $a \in \mathbb{R}$, $I_A(y') = I_A(y) \Leftrightarrow a = 1$. This property is illustrated in Figure 2. Finally, I_S is neither sensitive to scaling nor shifting (Figure 3). The remaining properties in Proposition 1 establish conditions

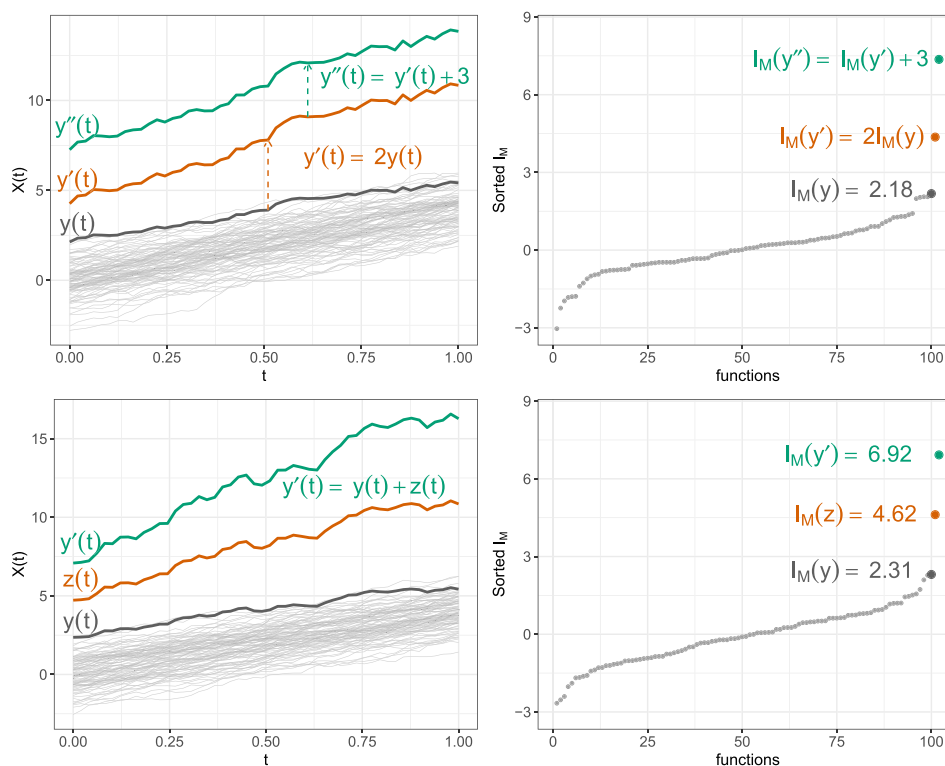


FIGURE 1 Illustration of the magnitude indices under scaling and translation. Functions and their sorted magnitude indices are shown in the first and second columns, respectively. Functions in grey are the bulk of the data. The function in black is $y(t)$. Functions in orange and green are transformed functions. The same colour code applies to points representing the indices.

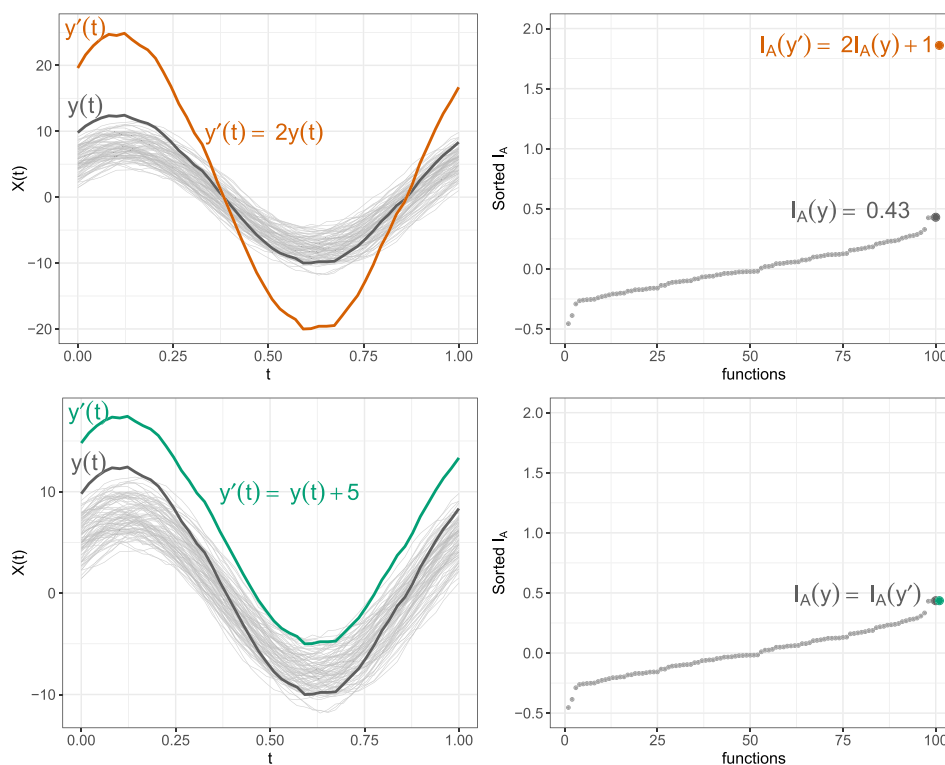


FIGURE 2 Illustration of the amplitude indices under simple transformation. Functions and their sorted amplitude indices are shown in the first and second columns, respectively. The functions in grey are the bulk of the data. The function in black is $y(t)$. The functions in orange and green are transformed functions. The same colour code applies to the points representing the indices.

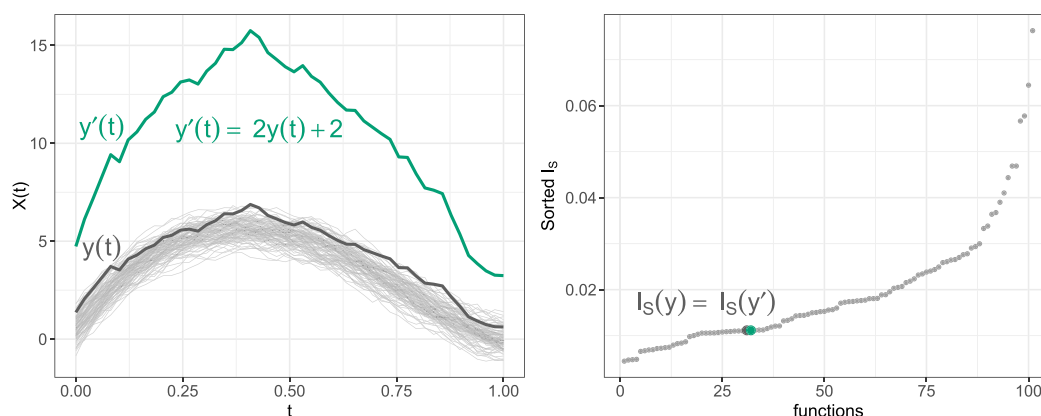


FIGURE 3 Illustration of the shape indices under simple transformation. Functions and their sorted shape indices are shown in the first and second columns, respectively. The functions in grey are the bulk of the data. The function in black is $y(t)$. The function in green is the transformed function ($y'(t)$). The same colour code applies to the points representing the indices.

under which the I_A and I_S indices of a transformed function remain the same. The FastMUOD indices defined above slightly differ from those used in Ojo et al. (2021). The original amplitude and magnitude indices, I_{A_v} and I_{M_v} , in Ojo et al. (2021) had absolute values that guaranteed these indices were positive. We provide definitions and properties of I_{A_v} and I_{M_v} in Section S3 of the Supporting Information.

2.3 | Implementation and cutoffs for FastMUOD Indices

The sample versions of the indices, denoted by I_{S_n} , I_{A_n} , and I_{M_n} , were implemented in R (R Core Team, 2022). The point-wise median is used in the implementation as it is more robust to outliers (compared with $\bar{X}(t)$). We consider both the upper and lower whiskers of a boxplot applied on I_{A_n} (and I_{M_n}) as cutoffs to find amplitude (and magnitude) outliers, because these outliers have indices that occur on both tails of the distribution. To find shape outliers, we consider only the upper whisker of a boxplot applied on I_{S_n} as a cutoff because this index is always positive and right skewed, with outliers having indices on the right tail of the distribution.

3 | EXTENSIONS TO MULTIVARIATE FUNCTIONAL DATA

FastMUOD was proposed for univariate functional data; however, many real functional data are multivariate in nature. Consequently, we present some techniques for detecting outliers in multivariate functional data using FastMUOD indices. The proposed techniques all involve applying the univariate FastMUOD indices on univariate functional datasets obtained from the multivariate functional data of interest; hence, for our multivariate applications, the definitions and properties presented in Section 2 are relevant.

3.1 | Marginal outlier detection with FastMUOD indices

Suppose $\{Y(t), t \in [0, 1]\}$ is a stochastic process taking values in \mathbb{R}^d . Let the distribution of $Y(t)$ be $F_{Y(t)}$ and let $F_{Y^j(t)}$ be the distribution of the j^{th} marginal component of $Y(t)$, with $j = 1, \dots, d$. Consider a set of n realisations of Y : $\{Y_i(t)\}_{i=1}^n$. To identify outliers in $\{Y_i(t)\}_{i=1}^n$, a first option is to apply FastMUOD to the d marginals of the observed curves, (i.e., $Y_i^j(t)$) and identify Y_i as an outlier if it is an outlier (of any type: shape, amplitude, or magnitude) in any of the d margins. This technique has a limitation of not detecting “joint-outliers,” that is, observations that are not outliers in any of the components but are outlying compared with the joint distribution of the data. It is also prone to false positives (FPs) because the final FPs is the union of the FPs of the three indices for each margin of the multivariate functional data to which FastMUOD is applied.

3.2 | Stringing marginal functions into univariate functional data

For a multivariate functional observation, $Y_i(t)$, we can concatenate or “string” its d univariate dimensions (i.e., $Y_i^1(t), \dots, Y_i^d(t)$) together into a single univariate function. Thus, we can obtain univariate curves $Z_i(t')$ defined on $[0, d]$ from the original multivariate curves $\{Y_i(t)\}_{i=1}^n$ given by

$Z_i(t') := Y_i^j(t' - j + 1)$, whenever $t' \in (j - 1, j]$, for $j = 1, \dots, d$ (set $Z_i(0) := Y_i^1(0)$). FastMUOD can be applied on $\{Z_i(t')\}_{i=1}^n$ by estimating the indices $I_{S_n}(Z_i, F_{Z_n})$, $I_{A_n}(Z_i, F_{Z_n})$ and $I_{M_n}(Z_i, F_{Z_n})$ and applying the cutoffs described in Section 2.3. However, the d univariate functions $(Y_i^1(t), Y_i^2(t), \dots, Y_i^d(t))$ may have different ranges so it might be convenient to scale the dimensions of $Y_i(t)$ into the same range (e.g., using a min-max scaling). Also, changing the order of “stringing” might have an effect on which observations are detected as outliers, for example, for $\{R(t)\}_{i=1}^n$, the “strung” functions $\{A_i(t') = R_i^j(t' - j + 1)\}_{i=1}^n$ and $\{B_i(t') = R_i^{d+1-j}(t' - j + 1)\}_{i=1}^n$ might produce different outliers.

3.3 | Random projections

Due to the limitations of the two techniques in Sections 3.1 and 3.2, we introduce a technique based on random projections. For $\{Y_i(t)\}_{i=1}^n$, we generate L random unit vectors $\{\hat{a}_l \in \mathbb{R}^d\}_{l=1}^L$ and project $Y_i(t)$ in the direction of \hat{a}_l : $Y_{il}(t) := \hat{a}_l^\top Y_i(t) \in \mathbb{R}$. Then, FastMUOD can be applied on the univariate functional data $\{Y_{il}(t)\}_{i=1}^n$ by estimating the indices $I_{S_n}(Y_{il}, F_{Y_{nl}})$, $I_{A_n}(Y_{il}, F_{Y_{nl}})$, and $I_{M_n}(Y_{il}, F_{Y_{nl}})$, where $F_{Y_{nl}}$ is the empirical distribution of $\{Y_i(t)\}_{i=1}^n$ projected on \hat{a}_l . Applying the cutoff described in Section 2.3 on the sets $\{I_{S_n}(Y_{il}, F_{Y_{nl}})\}_{i=1}^n$, $\{I_{A_n}(Y_{il}, F_{Y_{nl}})\}_{i=1}^n$, and $\{I_{M_n}(Y_{il}, F_{Y_{nl}})\}_{i=1}^n$ reveals whether $Y_i(t)$ is an outlier (of a specific type) when projected in the direction of \hat{a}_l .

3.3.1 | Threshold for the random projections

To combine all information from the L projections, we adopt a “voting system” in which a multivariate function is flagged as an outlier of a specific type if it is an outlier of that type in more than a fixed proportion of the projection directions. To this end, we define the following indicator functions:

$$\begin{aligned} O_{S,l}(Y_i) &:= \mathbb{1}\{\text{if } Y_{il}(t) \text{ is a shape outlier}\}, \\ O_{A,l}(Y_i) &:= \mathbb{1}\{\text{if } Y_{il}(t) \text{ is an amplitude outlier}\}, \\ O_{M,l}(Y_i) &:= \mathbb{1}\{\text{if } Y_{il}(t) \text{ is a magnitude outlier}\}. \end{aligned} \quad (1)$$

Then, we fix the threshold triple $Q = (\tau_M, \tau_S, \tau_A)$, where $\tau_M, \tau_S, \tau_A \in [0, 1]$, and declare $Y_i(t)$ a “shape” outlier if $\mathbb{E}_l[O_{S,l}(Y_i)] \geq \tau_S$ (and as an amplitude/magnitude outlier whenever $\mathbb{E}_l[O_{A,l}(Y_i)] \geq \tau_A$ or $\mathbb{E}_l[O_{M,l}(Y_i)] \geq \tau_M$). The classification of an outlier indicates that the function is an outlier of a given type in at least $\tau \in Q$ proportion of the projections. This classification is also not necessarily disjoint (e.g., see Section 4.3).

The threshold triple Q helps to control the false positive rate (FPR). The lower the value of $\tau \in Q$, the more aggressive the procedure is in flagging an observation as an outlier (because flagging an outlier requires less number of “votes” from the projections). Higher values of τ , on the other hand, make the procedure more conservative. For amplitude and magnitude outliers, we find (in our simulation tests) that limiting the value of both τ_A and τ_M to $[0.3, 0.7]$ works well for most applications (see Section S8 in the Supporting Information). In the case when there are magnitude (or amplitude) outliers in the projected data, τ_M (or τ_A) should be close to the lower bound of 0.3, which is sufficiently low to allow for flagging the outliers without introducing many FPs. When there are no magnitude (amplitude) outliers, τ_M (or τ_A) should be close to 0.7, which is a sufficiently high proportion to prevent FPs. When there are magnitude (or amplitude) outliers, some random projections of the data will not detect the true outliers, and therefore it is imperative not to set τ_M (or τ_A) to a high proportion in this case. However, setting τ_M (or τ_A) to a very low proportion, even when there are magnitude (amplitude) outliers, will yield many FPs because some non-outliers will be erroneously flagged as outliers in some of the projections. For shape outliers, we suggest limiting τ_S to the interval $[0.4, 0.7]$ because we know from previous studies that the shape index is more prone to FPs than the magnitude and amplitude indices (partly because of its skewed distribution; see Ojo et al., 2021).

3.3.2 | Selecting the thresholds Q

It is possible to fix the values in Q (within the intervals $[0.3, 0.7]$ for τ_A/τ_M and $[0.4, 0.7]$ for τ_S) in a data-driven way if the distribution of the functional data is known; for example, consider the following model for $T \in \{M, S, A\}$ (where M, S , and A denote magnitude, shape, and amplitude, respectively):

$$\tau_T := \begin{cases} \gamma_T - \eta_T \frac{\Delta_{PT}}{\Delta_C} & \text{if } \frac{\Delta_{PT}}{\Delta_C} \in [0, 1], \\ \gamma_T - \eta_T & \text{if } \frac{\Delta_{PT}}{\Delta_C} > 1, \\ \gamma_T & \text{otherwise,} \end{cases} \quad (2)$$

where $\gamma_T, \eta_T \in [0,1]$. The term Δ_{PT} is an estimate of the proportion of outliers of type T present in the data computed by subtracting the expected FPR of type T under the null model (a model where there are no outliers) from the average proportions of outliers of type T in all projections:

$$\Delta_{PT} = \frac{\sum_{l=1}^L \sum_{i=1}^n \hat{O}_{T,l}(Y_i)}{n \times L} - \hat{B}_T, \quad (3)$$

where \hat{B}_T is an estimate of the “baseline” expected FPR of type T under the null model and $\hat{O}_{T,l}$ is an estimate of the indicator functions in Equation (1). Likewise, Δ_C is an estimate of the proportion of all unique outliers (regardless of their type) present in the data computed by subtracting the expected proportion of total FPs under the null model from the average proportions of total unique outliers found over all L projections:

$$\Delta_C = \frac{\sum_{l=1}^L \sum_{i=1}^n \hat{O}_l(Y_i)}{n \times L} - \hat{B}_C, \quad (4)$$

where \hat{O}_l is an estimate of the indicator function $O_l(Y_i) := 1\{\text{if any } O_{T,l}(Y_i) = 1 \text{ for } T \in \{M, S, A\}\}$ and \hat{B}_C is an estimate of the “baseline” expected proportion of total FPs (of any type) under the null model. To ensure that τ_T is within an interval $[a, b] \subset [0,1]$ of interest in Equation (2), it suffices to set $\gamma_T = b$ and $\eta_T = b - a$. For example, to ensure that $\tau_S \in [0.4, 0.7]$, we can set $\gamma_S = 0.7$ and $\eta_S = 0.3$ in Equation (2). The intuition is that if there are only shape outliers in the data, the proportion $\frac{\Delta_{PS}}{\Delta_C}$ will be close to 1, resulting in $\tau_S \approx 0.4$, which is the lower bound of the suggested interval $[0.4, 0.7]$ for shape outliers. On the other hand, if there are no shape outliers, $\frac{\Delta_{PS}}{\Delta_C}$ will be close to 0 so that $\tau_S \approx 0.7$, which is the upper bound of the suggested interval $[0.4, 0.7]$, thereby controlling for FPs. However, to estimate $\frac{\Delta_{PS}}{\Delta_C}$, it is necessary to have an estimate of \hat{B}_S and \hat{B}_C in Equations (3) and (4), respectively. If the model or distribution of the data is known, it is possible to estimate \hat{B}_S and \hat{B}_C by simulating the null model (observations without outliers) and estimating the proportion of FP of type S (\hat{B}_S) and the proportion of all FPs (\hat{B}_C).

However, for real applications, the distribution or model from which the data come is usually unknown, and therefore it is impossible to estimate the baselines B_T and B_C ; hence, the model in Equation (2) cannot be used to fix the threshold values in Q . An obvious option is to consider as an outlier of type T any observation that is flagged as an outlier of type T in at least one projection, i.e., flag $Y_i(t)$ as an outlier of type T if $\mathbb{E}_l[O_{T,l}(Y_i)] > 0$. This has the downside of being prone to FPs because it does not control for any FP due to the projection directions and the FastMUOD indices. Another option, which we recommend, is to use the threshold triple $Q = (\tau_M, \tau_S, \tau_A) = (0.3, 0.4, 0.3)$ that we have found to have a well-balanced performance across various scenarios in our simulation studies (see Section S8 of the Supporting Information).

3.3.3 | The magnitude-shape-amplitude plot

The proportion of projections in which a multivariate function Y_i is flagged as an outlier of type T for $T \in \{M, S, A\}$ is given by $v_T(Y_i) := \mathbb{E}_l[O_{T,l}(Y_i)] \in [0,1]$. Thus, we propose the Magnitude-Shape-Amplitude plot (MSA-plot), which is a scatterplot of points (v_S, v_M) , colored by v_A . The MSA-plot is an extension to MS-plot (Dai & Genton, 2018) for multivariate functional data as it visualises the proportions of projections in which an observation is flagged as a shape, magnitude and amplitude outlier. Pure magnitude outliers will be clustered in the top-left corner of this plot, whereas pure shape outliers will be clustered around the lower-right corner. Magnitude and shape outliers will be clustered in the top right corner, whereas outlier of all types will have a deep purple hue in addition to being clustered in the top right corner. Figure 4 shows MSA-plots for two simulated data, containing pure magnitude and amplitude-shape outliers, generated from Simulation Models 1 and 3, respectively (presented in Section 4.1 and shown in Figure S2 of the Supporting Information).

4 | SIMULATION STUDY

4.1 | Simulation models

We simulated trivariate ($d=3$) functional datasets from models based on the truncated Karhunen–Loève expansion for multivariate functional data (Happ & Greven, 2018):

$$Y_i(t) = \mu(t) + \sum_{m=1}^M \rho_{i,m} \psi_m(t) + \epsilon(t), \quad i = 1, \dots, n, M \in \mathbb{N},$$

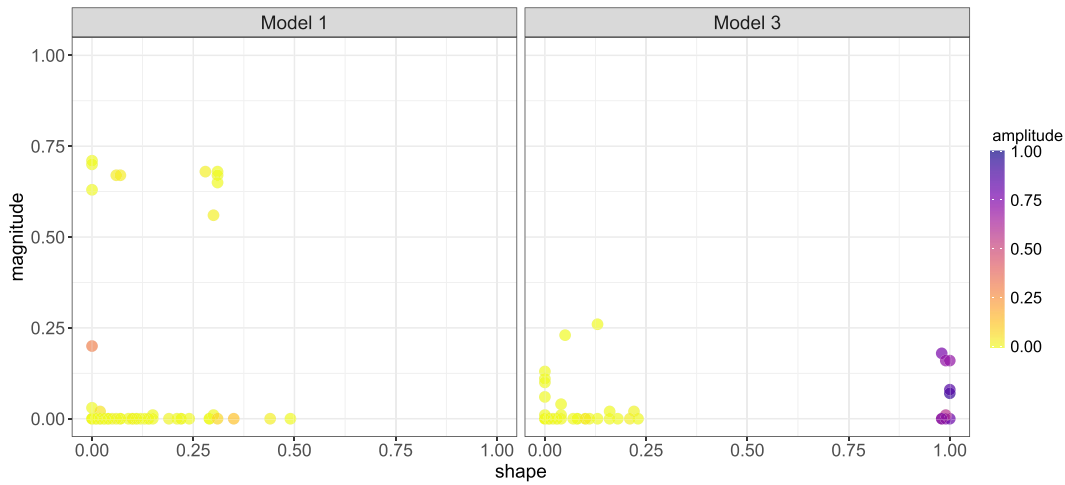


FIGURE 4 MSA-plots for two simulated datasets from Simulation Models 1 and 3.

where $\mathbf{Y}_i(t) \in \mathbb{R}^3$, $\boldsymbol{\mu}(t) \in \mathbb{R}^3$ is the multivariate mean function, and $\boldsymbol{\psi}_m(t) \in \mathbb{R}^3$, for $m=1, \dots, M$ are multivariate eigenfunctions. The scores $\rho_{i,m} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \nu_m)$ for eigenvalues ν_m that are linearly decreasing ($\nu_m = \frac{M+1-m}{M}$). The errors $\epsilon(t) \in \mathbb{R}^3$, and $\epsilon(t) \stackrel{\text{iid}}{\sim} \mathcal{N}_3(\mathbf{0}, \Sigma)$, with $\Sigma = \text{diag}(\sigma_1, \sigma_2, \sigma_3)$, where $\sigma_i \stackrel{\text{iid}}{\sim} U[0.1, 0.3]$, for $i=1, 2, 3$. The eigenfunctions $\boldsymbol{\psi}_m(t)$ were constructed by splitting orthonormal Fourier functions into $d=3$ pieces and shifting them to the required domain (Happ-Kurz, 2020). We set $M=9$ basis functions. The sample size for each dataset is $n=100$ and we considered a contamination rate of 10%. The simulated functions are evaluated at 50 equidistant points in $[0, 1]$. For each simulation model considered, the non-outliers were generated from a main model while the outliers were generated from a contaminated model, both listed below. Some sample data from these models are shown in Figures S2 and S3 of the Supporting Information.

1. Simulation Model 0 (No outliers):

Main Model: $\mathbf{Y}_i(t) = \boldsymbol{\mu}(t) + \sum_{m=1}^M \rho_{i,m} \boldsymbol{\psi}_m(t) + \epsilon(t)$; where $\boldsymbol{\mu}(t) = (4t, 30t(1-t)^{\frac{3}{2}}, 5(t-1)^2)^\top$.

2. Simulation Model 1 (Persistent magnitude outliers):

Main Model: The same as Model 0.

Contamination Model: $\mathbf{Y}_i(t) = \boldsymbol{\mu}(t) + \mathbf{u}(t) + \sum_{m=1}^M \rho_{i,m} \boldsymbol{\psi}_m(t) + \epsilon(t)$; where $\boldsymbol{\mu}(t)$ is the same as in Model 0 and $\mathbf{u}(t)$ is given by: $u_j^i(t) = 8W_j$ for $j=1, 2, 3$. W_j is sampled from $\{-1, 1\}$ with equal probability.

3. Simulation Model 2 (Non-persistent magnitude outliers):

Main Model: The same as Model 0 but with $\boldsymbol{\mu}(t) = (5\sin(2\pi t), 5\cos(2\pi t), 5(t-1)^2)^\top$.

Contamination Model: $\mathbf{Y}_i(t) = \boldsymbol{\mu}(t) + \mathbf{u}(t) + \sum_{m=1}^M \rho_{i,m} \boldsymbol{\psi}_m(t) + \epsilon(t)$ where $\boldsymbol{\mu}(t)$ is the same as the Main Model above, and $\mathbf{u}(t)$ is given by: $u_j^i(t) = 8W_j \cdot \mathbb{1}\{\text{if } t \in [T_q, T_q + 0.1]\}$. W_j is the same as in Model 1 and $T_q \sim U[0, 0.9]$.

4. Simulation Model 3 (Shape outlier I):

Main Model: The same as Model 0 but with $\boldsymbol{\mu}(t) = (5\sin(2\pi t), 5\cos(2\pi t), 5(t-1)^2)^\top$.

Contamination Model: The same as Model 0 but with $\boldsymbol{\mu}(t)$ changed to: $\boldsymbol{\mu}(t) = (5\sin(2\pi(t-0.3)), 5\cos(2\pi(t-0.2)), 5(0.1-t)^2)^\top$.

5. Simulation Model 4 (Shape outlier II):

Main Model: $\mathbf{Y}_i(t) = \boldsymbol{\mu}(t) + \mathbf{u}(t) + \sum_{m=1}^M \rho_{i,m} \boldsymbol{\psi}_m(t) + \epsilon(t)$; where $\boldsymbol{\mu}(t) = (5\sin(2\pi t), 5\cos(2\pi t), 5(t-1)^2)^\top$, and $\mathbf{u}(t)$ is given by $u_j^i(t) = q_j$, with $q_j \stackrel{\text{iid}}{\sim} U[-2.1, 2.1]$.

Contamination Model: Same as the Main Model above but with $\mathbf{u}(t)$ changed to: $\mathbf{u}(t) = (2\sin(4\pi t), 2\cos(4\pi t), 2\cos(8\pi t))^\top$.

6. Simulation Model 5 (Amplitude outliers):

Main Model: The same as Model 0 but with $\boldsymbol{\mu}(t) = (5\sin(2\pi t), 5\cos(2\pi t), 5(t-1)^2)^\top$.

Contamination Model: $\mathbf{Y}_i(t) = \boldsymbol{\mu}(t) + \mathbf{u}(t) + \sum_{m=1}^M \rho_{i,m} \boldsymbol{\psi}_m(t) + \epsilon(t)$; where $\boldsymbol{\mu}(t)$ is the same as the Main Model above and $\mathbf{u}(t)$ is given by: $\mathbf{u}_i(t) = ((2+R_i^1)\mu^1(t), (2+R_i^2)\mu^2(t), (2+R_i^3)\mu^3(t) - 6)^\top$; $\mu^j(t)$ are the components of $\boldsymbol{\mu}(t)$ in the Main Model above and $R_i^j \stackrel{\text{iid}}{\sim} \text{Exp}(2)$, for $j=1, 2, 3$.

7. Simulation Model 6 (Shape outlier III):

Main Model: $Y_i(t) = \mu(t) + u(t) + \sum_{m=1}^M \rho_{i,m} \psi_m(t) + \epsilon(t)$; where $\mu(t) = (5\sin(2\pi t), 5\cos(2\pi t), 5(t-1)^2)^\top$, and $u(t) = (8t\sin(\pi t), t\cos(\pi t), 6\sin(2\pi t) - 3)^\top$.

Contamination Model: Same as the Main Model above but with $u(t)$ changed to: $u(t) = (10t\sin(\pi t), 11t\cos(\pi t), 10\sin(2\pi t) - 6)^\top$.

4.2 | Outlier detection methods

Data were simulated from the seven models presented in Section 4.1. For each model, we compared the OD performance of the proposed extensions in Section 3. We also compared our proposals with other multivariate OD methods such as MS-plot (Dai & Genton, 2018), FOM, and functional adjusted outlyingness (FAO) (Rousseeuw et al., 2018). Because our FastMUOD-based proposals use indices that target different types of outliers, we can consider the OD performance of each index or consider a union of outliers flagged by the three indices. Thus, we considered the following methods in our comparison:

- FST-MAR: This is the union of all outlier types detected by applying FastMUOD to the marginal distributions of the multivariate functional data (described in Section 3.1). Consequently, an observation is an outlier if it is flagged as an outlier, of any type, in any of the three dimensions of the simulated multivariate functional data.
- FST-STR: This is the union of all outlier types detected by applying FastMUOD to the univariate functional data obtained by stringing marginal functions together (described in Section 3.2).
- FST-PRJ: This is the union of all outlier types detected by applying FastMUOD using random projections (described in Section 3.3). For our simulation tests, 60 random directions were used for projection. To generate each random direction, the components of the vector follow $\mathcal{N}(0,1)$ and the resulting vector was then normalised to have a unit norm. Generating the random directions in this manner is straightforward and fast with limited computational burden. We used Equation (2) to determine the threshold triple $Q = (\tau_M, \tau_S, \tau_A)$. Because we knew the base model (Model 0) from which the simulated data were generated, we could estimate the values of B_T and B_C used in Equations (3) and (4), respectively. For this purpose, we simulated data from Model 0 (null model without outliers) and computed the total FPR (of all outlier types) and the FPR of each type of outliers (shape, magnitude, and amplitude). Then, we used the computed FPR of all outliers as an estimate of B_C and the computed FPRs of outliers of each type as an estimate of B_T , for $T \in \{M, S, A\}$. Our simulation results yielded the following baseline values: $B_A = B_M = 0.009$, $B_S = 0.075$, and $B_C = 0.09$, which we then used in the estimation of $\frac{\Delta_{PT}}{\Delta_C}$ in Equation (2). Finally, we set the parameters γ_T and η_T in Equation (2) to $\gamma_T = 0.7$ for $T \in \{M, S, A\}$, $\eta_S = 0.3$, and $\eta_A, \eta_M = 0.4$ so that $\tau_S \in [0.4, 0.7]$ and $\tau_A, \tau_M \in [0.3, 0.7]$, as reported in Section 3.3.
- FST-PRJ1-MG, FST-PRJ1-AM, FST-PRJ1-SH: These are the “magnitude,” “amplitude,” and “shape” outliers, detected using FST-PRJ above.
- FST-PRJ1: This is similar to FST-PRJ, but uses the threshold triple $Q = (\tau_M, \tau_S, \tau_A) = (0.3, 0.4, 0.3)$, which we recommend in real application when it is impossible to use Equation (2) to determine the values of Q .
- FST-PRJ1-MG, FST-PRJ1-AM, FST-PRJ1-SH: These are the “magnitude,” “amplitude,” and “shape” outliers, detected using FST-PRJ1.
- FST-PRJ2: This is similar to FST-PRJ, but rather than using Equation (2) to select the threshold Q , an observation is an outlier of type T if it is flagged as an outlier of that type in ANY projection, that is, an observation is an outlier of type T if $\mathbb{E}_i[O_{T,j}(Y_i)] > 0$ for $T \in \{M, S, A\}$.
- FST-PRJ2-MG, FST-PRJ2-AM, FST-PRJ2-SH: These are the “magnitude,” “amplitude,” and “shape” outliers, detected using FST-PRJ2.
- MSPLOT: This is a multivariate functional outlier detection method based on the directional outlyingness (Dai & Genton, 2018).
- FOM: This is functional outlier map proposed in Rousseeuw et al. (2018).
- FAO: The functional adjusted outlyingness (FAO) is based on the functional adjusted outlyingness (fAO) (Brys et al., 2005; Hubert et al., 2015).

4.3 | Simulation results

We tested the proposed methods on the simulation models in Section 4.1. Contamination rate is set to 10% and 200 repetitions were performed for each model. The results are presented in Table 1. Because Model 0 contains no outliers, we show the FPRs of the techniques. For Models 1-6, we show the true positive rate (TPR) and the FPR together with their standard deviations in parentheses.

The results of Model 0 show that FST-MAR and FST-PRJ2 both have very high FPRs. For FST-MAR, FastMUOD was independently applied on each of the three dimensions of the trivariate functional data. For each dimension, the three FastMUOD indices contributed some FPs and the union of all these FPs yielded the overall FPR of about 26% for Model 0. Moreover, the extremely high FPR of FST-PRJ2 justifies the need to impose the threshold $Q = (\tau_M, \tau_S, \tau_A)$ used for determining if an observation is an outlier. Simply flagging an observation as an outlier if it is detected as an outlier in any random projection does not work for the FastMUOD indices. First, because a nonoutlier might sometimes appear to be an outlier in the projected direction, and second because the FastMUOD indices (especially the shape index) and the boxplot cutoff procedure

TABLE 1 Mean and standard deviation (in parentheses) of the true positive rate (TPR) and the false positive rate (FPR) (in percentage) over 200 repetitions for each model. Sample size $n = 100$, evaluation points $t_j = 50$, and contamination rate is 10%. Comparatively high TPRs ($\geq 95\%$) and low FPRs ($\leq 1\%$) are marked in bold. The proposed techniques are in italics.

		Model 1		Model 2		Model 3	
Method	Model 0 FPR	TPR	FPR	TPR	FPR	TPR	FPR
FST-MAR	26.2(4.2)	100.0(0.0)	25.2(4.8)	99.9(1.0)	13.9(3.2)	100.0(0.0)	13.4(3.3)
FST-STR	4.6(2.5)	100.0(0.0)	2.3(1.6)	90.4(10.9)	2.3(1.8)	100.0(0.0)	1.9(1.8)
FST-PRJ	1.7(2.6)	100.0(0.0)	0.2(0.5)	98.7(4.3)	0.7(0.9)	100.0(0.0)	0.8(1.0)
FST-PRJ-SH	1.7(2.6)	0.2(1.4)	0.2(0.5)	98.7(4.3)	0.7(0.9)	100.0(0.0)	0.8(1.0)
FST-PRJ-AM	0.0(0.3)	0.0(0.0)	0.0(0.0)	0.0(0.7)	0.0(0.0)	100.0(0.0)	0.0(0.2)
FST-PRJ-MG	0.0(0.2)	100.0(0.0)	0.0(0.1)	0.0(0.0)	0.0(0.0)	0.0(0.0)	0.0(0.0)
FST-PRJ1	3.7(1.9)	100.0(0.0)	3.4(1.9)	99.2(2.8)	1.1(1.2)	100.0(0.0)	1.0(1.1)
FST-PRJ1-SH	3.5(1.8)	4.2(6.5)	3.3(1.9)	98.9(3.6)	0.8(1.0)	100.0(0.0)	0.8(0.9)
FST-PRJ1-AM	0.2(0.4)	0.1(1.0)	0.2(0.5)	4.0(6.6)	0.1(0.4)	100.0(0.0)	0.0(0.2)
FST-PRJ1-MG	0.1(0.4)	100.0(0.0)	0.0(0.2)	2.9(5.3)	0.2(0.6)	2.1(4.7)	0.2(0.5)
FST-PRJ2	52.2(3.6)	100.0(0.0)	51.0(4.0)	100.0(0.0)	35.7(3.8)	100.0(0.0)	34.8(4.1)
FST-PRJ2-SH	47.0(3.3)	47.8(14.7)	47.1(4.0)	100.0(0.0)	28.2(3.0)	100.0(0.0)	27.4(3.3)
FST-PRJ2-AM	12.4(3.6)	12.6(10.5)	12.4(3.8)	64.3(14.6)	10.1(3.5)	100.0(0.0)	6.6(3.2)
FST-PRJ2-MG	14.0(4.1)	100.0(0.0)	7.2(3.1)	37.7(16.6)	8.9(3.6)	44.1(18.7)	9.0(3.6)
MSPLOT	1.6(1.7)	100.0(0.0)	0.5(0.8)	100.0(0.0)	0.9(1.2)	100.0(0.0)	1.1(1.4)
FOM	0.3(0.6)	100.0(0.0)	0.1(0.3)	96.9(7.5)	0.0(0.3)	70.4(29.0)	0.1(0.3)
FAO	0.3(0.6)	100.0(0.0)	0.0(0.2)	84.9(17.1)	0.1(0.3)	43.5(35.3)	0.1(0.3)
	Model 4		Model 5		Model 6		
Method	TPR	FPR	TPR	FPR	TPR	FPR	
FST-MAR	76.3(17.0)	15.7(3.6)	100.0(0.0)	25.4(4.0)	100.0(0.0)	14.0(3.5)	
FST-STR	49.7(27.1)	2.4(1.5)	100.0(0.0)	4.7(2.4)	99.9(1.0)	2.3(1.6)	
FST-PRJ	17.0(23.7)	0.3(0.8)	100.0(0.0)	0.2(0.4)	99.6(2.0)	0.6(0.8)	
FST-PRJ-SH	15.4(22.8)	0.2(0.7)	0.0(0.0)	0.1(0.4)	93.1(13.2)	0.5(0.8)	
FST-PRJ-AM	0.0(0.0)	0.0(0.2)	100.0(0.0)	0.0(0.2)	1.3(6.3)	0.0(0.1)	
FST-PRJ-MG	2.1(6.7)	0.1(0.3)	45.5(30.3)	0.0(0.1)	90.1(15.8)	0.1(0.3)	
FST-PRJ1	45.2(19.2)	1.2(1.2)	100.0(0.0)	3.9(1.9)	99.7(2.2)	0.9(0.9)	
FST-PRJ1-SH	41.8(18.7)	1.0(1.1)	0.0(0.0)	3.8(1.9)	98.2(5.6)	0.8(0.9)	
FST-PRJ1-AM	0.1(1.2)	0.2(0.5)	100.0(0.0)	0.0(0.1)	18.4(16.4)	0.1(0.3)	
FST-PRJ1-MG	5.6(8.5)	0.1(0.4)	86.6(12.3)	0.1(0.4)	90.6(11.9)	0.0(0.2)	
FST-PRJ2	96.9(6.7)	37.6(3.9)	100.0(0.0)	51.8(3.9)	100.0(0.0)	39.1(3.9)	
FST-PRJ2-SH	94.1(8.2)	29.9(3.6)	0.7(2.5)	49.3(3.7)	100.0(0.0)	34.1(3.9)	
FST-PRJ2-AM	16.8(12.4)	13.2(3.7)	100.0(0.0)	5.9(2.9)	95.4(7.6)	9.5(3.3)	
FST-PRJ2-MG	41.6(22.3)	8.3(3.2)	99.4(2.5)	6.0(2.8)	99.8(1.9)	7.5(3.1)	
MSPLOT	34.8(21.9)	1.1(1.4)	100.0(0.0)	0.9(1.1)	95.6(7.2)	1.0(1.2)	
FOM	1.9(4.6)	0.1(0.3)	100.0(0.0)	0.1(0.3)	51.4(33.9)	0.1(0.3)	
FAO	1.1(3.6)	0.1(0.3)	99.9(1.0)	0.1(0.3)	25.6(29.9)	0.0(0.1)	

described in Section 2.3 also produce some FPs. These high FPRs can be observed for both methods (FST-MAR and FST-PRJ2) across all tested models.

For Model 1, all methods performed well, except for FST-MAR and FST-PRJ2 because of their high FPRs. The high TPRs of FST-PRJ1-MG and FST-PRJ-MG demonstrate that the magnitude indices detect the magnitude outliers while the other indices (FST-PRJ-AM, FST-PRJ1-AM, FST-PRJ-SH, and FST-PRJ1-SH) do not contribute significantly to the FPs, thus yielding overall good results for FST-PRJ and FST-PRJ1. This also

shows that the multivariate magnitude outliers in Model 1 remained magnitude outliers after the projection procedure because only the magnitude indices were activated. FST-STR, which used the stringing procedure described in Section 3.2, also showed a very high TPR with low FPR on this magnitude model. In Model 2, FST-PRJ and FST-PRJ1 show very good OD performance but this time powered by their shape indices (FST-PRJ-SH and FST-PRJ1-SH). FAO however struggled with this model with less than 90% TPR and high standard deviation. Both FOM and FAO did not perform well in Model 3, whereas FST-PRJ and FST-PRJ1 showed excellent OD performance on this model, helped by their amplitude and shape indices. This reiterates that outlier classification is not necessarily disjoint because on average, all the outliers in Model 3 were flagged as both amplitude and shape outliers. The “non-disjoint” classification of outliers can also be observed in the results of FST-PRJ and FST-PRJ1 on Models 5 and 6. All the methods showed poor TPRs in Model 4, because it contains pure shape outliers that are hidden within the bulk of the data. Apart from Model 4, MSPLOT maintains an excellent performance across all other models, except for Model 6 where it did not perform quite as well with a TPR of 95% compared with 100% for FST-PRJ1 and FST-PRJ. Most of the simulation models used in this study have outliers outlying in all three dimensions (except for Model 6).

In the Supporting Information (Section S7), we show more simulation results for different contamination rates. Moreover, most of the simulation models used in this study have outliers outlying in all three dimensions (except for Model 6). In Section S6 of the Supporting Information, we show the performance of the proposed methods with similar simulation models but with the outliers only outlying in one or two of the three dimensions of the functional data.

To summarise, FST-PRJ and FST-PRJ1 showed the best performance across all tested simulation models. We recommend FST-PRJ1 in most usual applications because the underlying data distribution will be unknown, and it will consequently be impossible to compute the threshold Q for FST-PRJ.

5 | DATA EXAMPLES

5.1 | Character dataset

The character dataset comprises bivariate functional data of trajectories of a pen tip along the x and y axes while a subject repeatedly writes various letters of the English alphabet. The original data were provided as part of the Character Trajectories dataset on the UCI machine learning repository (Williams et al., 2006). The versions of the dataset used in this study are for the letters “i” (without the dot) and “a” provided in the *mrfDepth* R package (Segaert et al., 2017). For the letter “i,” the dataset consists of $n_i = 174$ bivariate functions, observed at 100 times points, whereas for letter the “a,” there are $n_a = 171$ bivariate functions observed at the same number of time points. The bivariate functions in both datasets are the vertical and horizontal coordinates of the pen tip while the subject wrote n_i or n_a copies of each corresponding letter. The first row of Figures 5 and 7 show the bivariate functions for letters “i” and “a,” respectively.

Plotting the vertical coordinates against the horizontal coordinates in both datasets reveals the handwritten characters (Figures 5 and 7). The aim is to use the FastMUOD via projections (FST-PRJ1) to detect outliers in both datasets. For each dataset, we generated 60 random unit vectors in \mathbb{R}^2 and projected the data. FastMUOD was then applied on the projections and we used a threshold triple of $Q = (0.3, 0.4, 0.3)$ to determine which observations are outliers of the various types.

5.1.1 | Letter “i”

For the letter “i,” curves 41 and 46 were flagged as magnitude outliers; curves 39, 46, and 67 were flagged as amplitude outliers and curves 3, 5, 6, 9, 35, 39, 40, 41, 46, 64, 90, and 102 were flagged as shape outliers. Thus, 13 unique outliers were flagged in total. All magnitude outliers are also shape outliers, and curve 46 is an outlier of all types. The “magnitude” and “amplitude” outliers are shown in Figure 5. Curve 41 deviates from the overall trend of the data while curve 46 does not have enough “follow through” compared with other curves. Some of the shape outliers are shown on the bottom right plot of Figure 5. Like curve 41, curves 40 and 64 deviate from the overall trend of the data, whereas the curve 102 looks rather similar to a “v” instead of an “i.” Although curve 9 seems to follow the overall trend of the data and does not appear to be an outlier, a closer look at its horizontal and vertical coordinate curves (see Figure S4 of the Supporting Information) reveals that the minimum points of both curves are horizontally shifted (to the right) compared with other curves; hence, it was flagged as a shape outlier. Curves 3, 5, 6, and 90 (shown in Figure S4 of the Supporting Information) were also flagged as outliers for this same reason. Figure 6 shows the MSA-plot for the character “i” data.

For comparison, we applied MSPLOT on the character dataset for letter “i.” MSPLOT discovered a total of 18 unique outliers. The curves flagged by MSPLOT were: 3, 5, 6, 9, 11, 12, 14, 39, 40, 41, 67, 73, 90, 102, 109, 110, 111, and 141. Among the 13 unique outliers flagged by FastMUOD, 10 were flagged by MSPLOT. The functions flagged by only FastMUOD are curves: 35, 46, and 64; while those flagged by only MSPLOT are curves: 11, 12, 14, 73, 109, 110, 111, and 141. These curves are shown in Figure S5 of the Supporting Information.

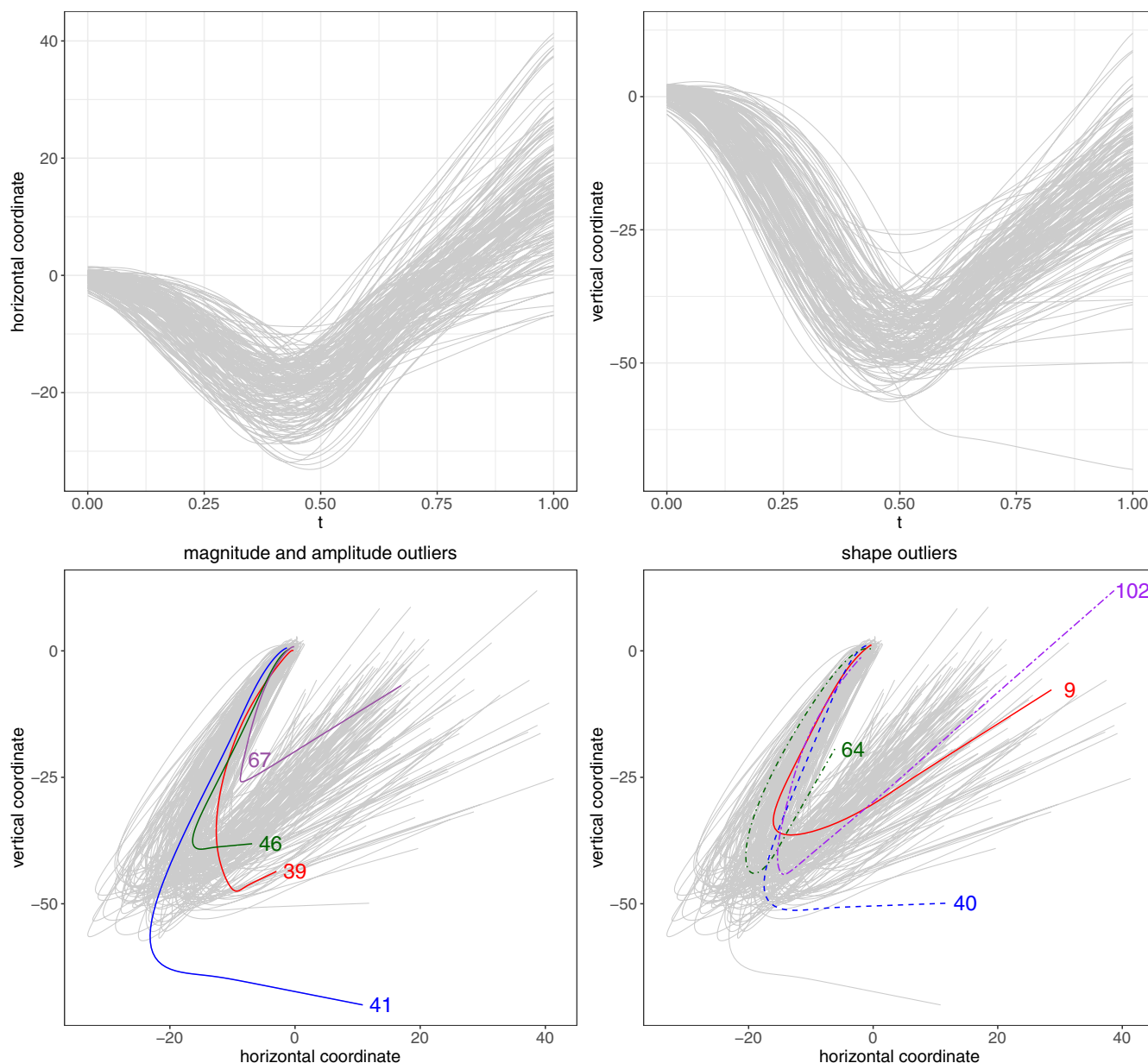


FIGURE 5 First row: Horizontal and vertical trajectories for letter “i” data. Second row: All magnitude outliers and some shape outliers detected in letter “i” data.

Considering the curves detected by only FastMUOD and MSPLOT reveals certain interesting features in the data. Curves 35, 46, and 64, detected by only FastMUOD, clearly show some deviation from the trend of the data, especially in their follow-through. On the other hand, curve 111, flagged by only MSPLOT, seems to resemble a slanted “v” rather than an “i.” To summarise, MSPLOT seems to be more aggressive in declaring curves as outliers compared with FastMUOD. Finally, we compared the results obtained to those of FOM, which only flagged curve 41 (shown in the bottom left plot of Figure 5) as an outlier, probably because FOM is more suited to detecting magnitude outliers rather than shape outliers. Curve 41 demonstrates a clear magnitude deviation in its vertical axis (Figure S6 of the Supporting Information).

5.1.2 | Letter “a”

For letter “a,” FST-PRJ1 flagged 17 curves as outliers. Curves: 49, 56, 88, 94, and 131 were flagged as magnitude outliers. Curves 1 and 166 were flagged as amplitude outliers, whereas curves 21, 56, 58, 75, 100, 101, 102, 114, 117, 125, and 136 were flagged as shape outliers. The curves flagged as magnitude and amplitude outliers (shown in the middle row of Figure 7) show a shift, either in the vertical, horizontal, or both axes.

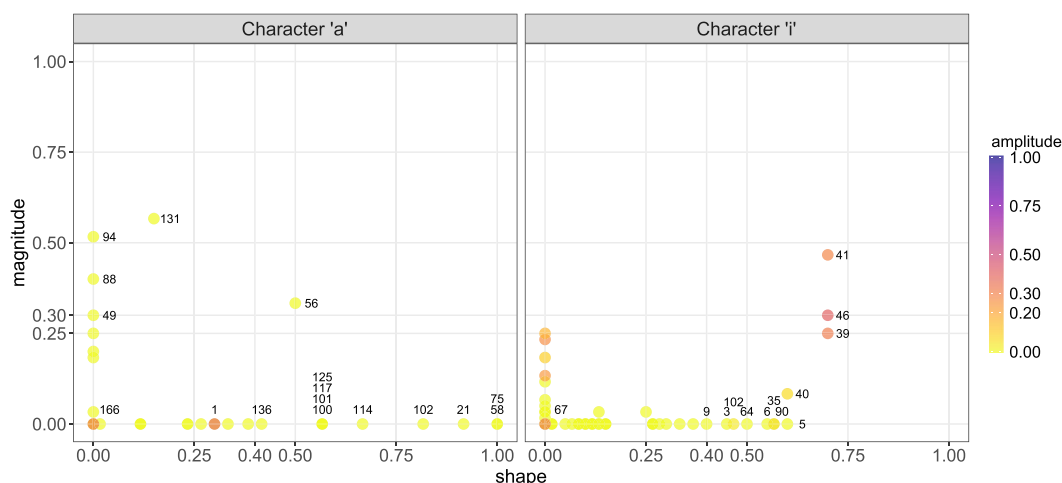


FIGURE 6 MSA-plots for character “a” and “i” data. Outlying curves are labelled.

Some of the flagged shape outliers have their peaks shifted to the right, particularly in the vertical axis, compared with the bulk of the data, resulting in letter “a”s with very small follow-through (when both axes are plotted) compared with the bulk of the data. Some of these functions are shown in the bottom row of Figure 7. On the other hand, some of the shape outliers have their peaks shifted to the left, which results in letter “a”s with a long follow through compared with the bulk of the data, thereby making the corresponding letters look like a “q” rather than an “a.” These functions are also shown in Figure 7 (see Figures S7 and S8 of the Supporting Information for the plots of the horizontal and vertical coordinates of these curves). Also, the MSA-plot for the character “a” data is shown in the left plot of Figure 6 with the outliers annotated.

We applied MSPLOT on the data for letter “a” and MSPLOT flagged 12 unique outliers compared with the 17 outliers flagged by FastMUOD (FST-PRJ1). The outliers flagged by MSPLOT are the curves: 1, 21, 23, 49, 56, 58, 75, 102, 114, 125, 131, and 158. Among these 12 unique outliers, 10 were also flagged by FST-PRJ1, indicating a good overlap between the outliers flagged by MSPLOT and FastMUOD. Curves 23 and 158 were flagged as outliers by MSPLOT but not by FastMUOD, whereas the curves 88, 94, 100, 101, 117, 136, and 166 were flagged by FastMUOD but not by MSPLOT. Some of these curves are shown in Figure S9 of the Supporting Information. FOM detected only 3 unique outliers. These are curves 23, 56, and 58, which can be seen in Figure 7 (and Figures S7 and S9 of the Supporting Information).

5.2 | Video data

In the second application, we applied FST-PRJ1 on a surveillance video data named “WalkByShop1front” (made available by the the EC Funded CAVIAR project/IST 2001 37540 at: homepages.inf.ed.ac.uk/rbf/CAVIAR/). The video consists of a 94-s-long recording of a surveillance camera in front of a clothing shop in a shopping mall in Lisbon. At various time stamps in the course of the video clip, people passed by the front of the shop; sometimes they entered the shop to explore the products too. The aim is to identify time frames during which people passed by or entered the shop. This video dataset was analysed in Ojo et al. (2021); they converted the video to greyscale and used the original FastMUOD to analyse the resulting univariate functional data (with each frame represented as a function and each pixel being an evaluation point on the curve). Because the original video is coloured, some information is lost in the conversion to greyscale. We represent the coloured video as trivariate functional data with each dimension being the RGB values of each pixel. We aim to apply FST-PRJ1 to the trivariate functional data and compare the performance to the univariate analysis of the greyscale values done in Ojo et al. (2021).

The video clip is provided at 25 frames per second, and there are a total of 2359 frames. The resolution of the video is 384×288 , and therefore each frame contains $384 \times 288 = 110,592$ pixels. For each frame, we arranged the RGB pixel values into an array of size $110,592 \times 3$. Thus, the trivariate functional dataset is of dimension $2,359 \times 110,592 \times 3$ representing 2359 functions (the frames) evaluated at 110,592 points (the pixels) where the value of each point is a vector in \mathbb{R}^3 (the RGB pixels intensity). Then, we projected the constructed trivariate functional data on 30 random unit vectors in \mathbb{R}^3 and applied FastMUOD (FST-PRJ1) on the 30 univariate functional data of size $2359 \times 110,592$. We then set the threshold triple to $Q = (0.3, 0.4, 0.3)$.

In total, 356 unique frames were flagged as outliers with 213, 270, and 226 frames flagged as shape, amplitude, and magnitude outliers, respectively (Figure 8). A total of 143 frames were flagged as outliers of all types. The 356 unique outliers flagged are an improvement over the 294 unique frames detected as outliers in the previous analysis of the greyscale pixel values performed by Ojo et al. (2021). Similar to the analysis in Ojo et al. (2021), all the frames flagged as outliers correspond to frames during which people pass by or enter the shop. This improvement

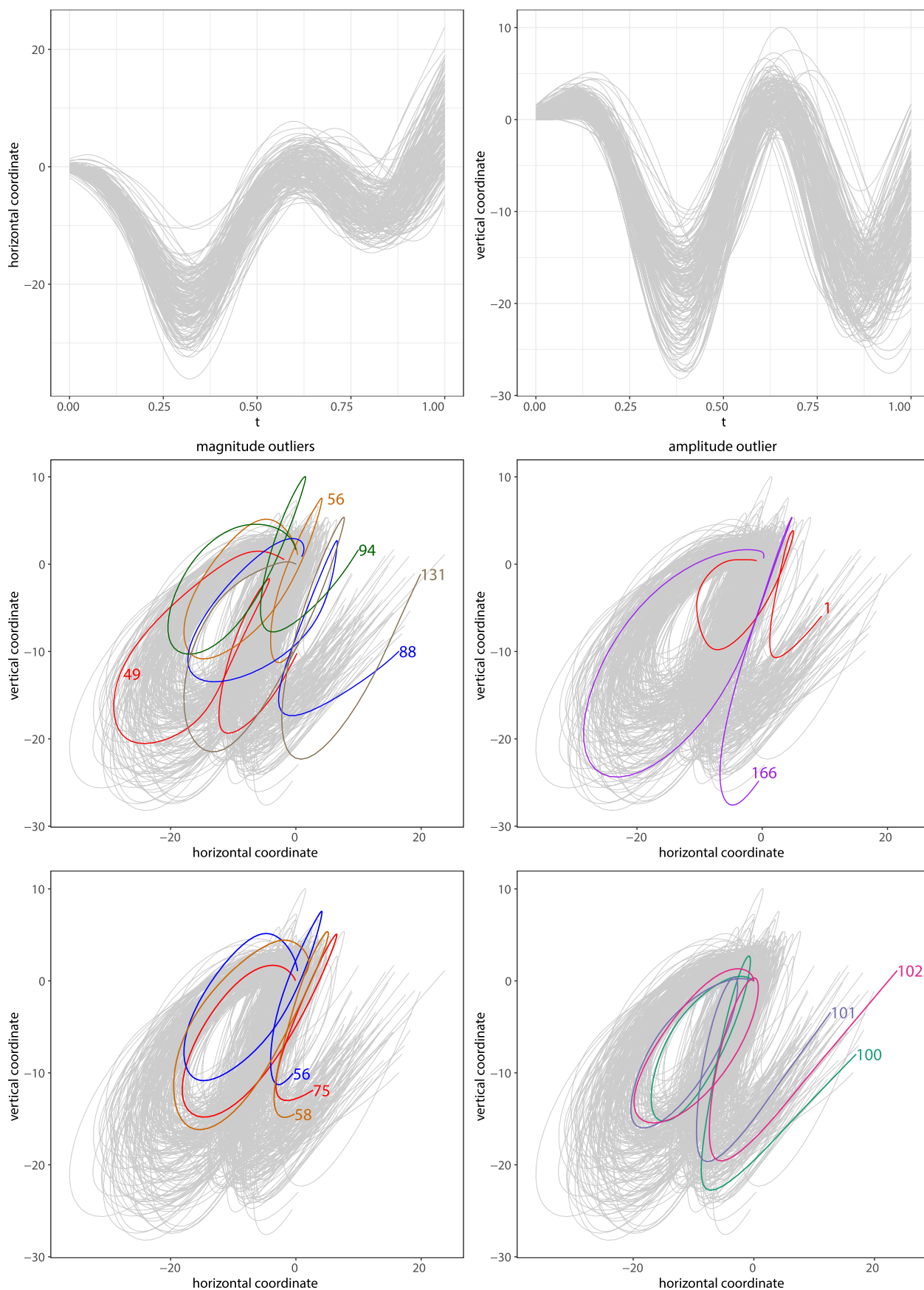


FIGURE 7 First row: Horizontal and vertical trajectories for letter “a” data. Second row: Magnitude and amplitude outliers detected by FastMUOD (FST-PRJ1). Third row: Shape outliers with short (left) and long (right) “follow-throughs” respectively.

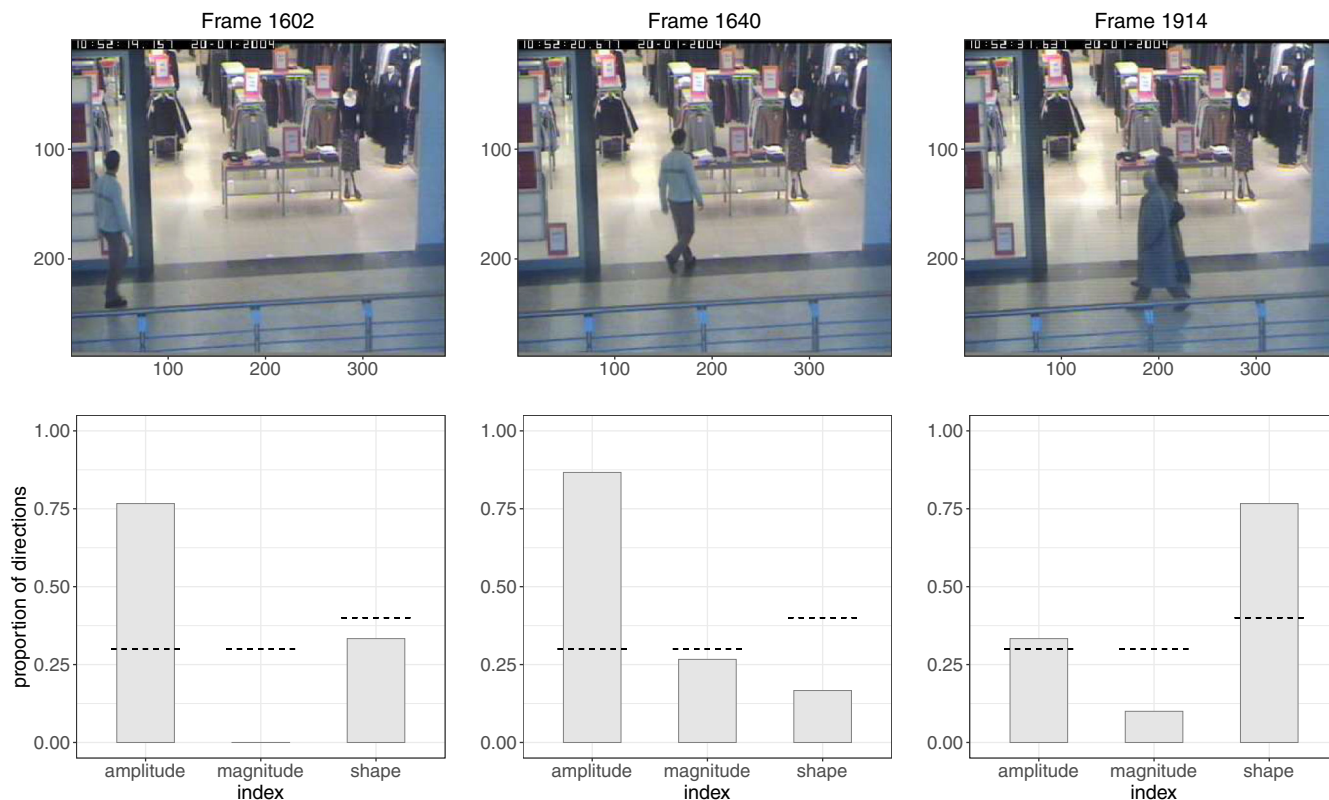


FIGURE 8 Some selected frames detected as outliers by FST-PRJ1. The bar charts below each frame show the proportion of projections in which that corresponding frame was flagged as an outlier of a particular type. The dotted lines indicate threshold values in Q .

underlines the advantage of using the multivariate data of the video data compared with performing a univariate analysis of the greyscale values as done in Ojo et al. (2021).

To evaluate the performance of FastMUOD (with projections) in detecting the video frames of interest, it is necessary to understand the distribution of the outlying video frames. The video itself contains three major segments during which various people passed by or entered the front of the shop. The first segment contains frames 804–908, during which a woman passed by the front of the shop. The second segment contains frames 1588–2000, when a man entered the shop (to check the products on sale) and two other women passed by the shop. The third segment contains frames 2073–2359, which show another man entering the shop. All frames detected as outliers are within the frames of the three main segments, and so there are no FPs. However, similar to the results obtained by Ojo et al. (2021), there are pockets of timestamps in these segments not flagged as outliers. For instance, the first frames detected as outliers are in the frames of the first segment (frames 803–908) with FastMUOD flagging frames 815, 830–851, 855–857, 864, and 881–903 as outliers while missing some frames at the beginning (frames 804–814 and 816–830), middle (frames 858–863 and 865–880), and end (frames 904–908) of this segment. We observed the same behaviour for the second and third segments, with certain pockets of a few frames in the beginning, middle and end of the segments not flagged as outliers. Usually, the pocket of frames not flagged as outliers in the middle of the segments correspond to timestamps when someone enters the shop and stands beside the products on display, yielding insufficient contrast in the pixel values of the person and the products on display in the shop. This is shown in Figure 9, which shows some frames in the second segment of outlying frames that were not flagged as outliers. The first frame, frame 1597, shows when a man just entered the camera view. The second frame, frame 1700, shows the same man in the store checking out the products. Figure 8, on the other hand, shows some selected frames in the second segment that were flagged as outliers. In addition to frames shown in Figures 8 and 9, the proportion of directions in which the frames are detected as outliers of each type are also shown.

Because the frames detected as outliers depend on the threshold triple Q , it is useful to visualise the frames of the video together with an animation of the proportion of directions in which the frames are flagged as outliers of different types. Such an animation can be seen by clicking on this [URL](#) and it shows the variation in the proportion of directions in which frames are outlying as people pass by or enter the shop.

For comparison, we applied MSPLIT on the same video data. MSPLIT flagged 1001 frames (out of the 2359 frames in the data) as outliers; although most frames in the three segments of interest were flagged as outliers, about 200 additional frames that are clearly out of the outlying segments were detected as outliers. However, FOM performed very well on the data, flagging 774 outliers with all the flagged frames coming from the three outlying segments in the video data. FOM excels in the analysis of image and video data because it computes a directional

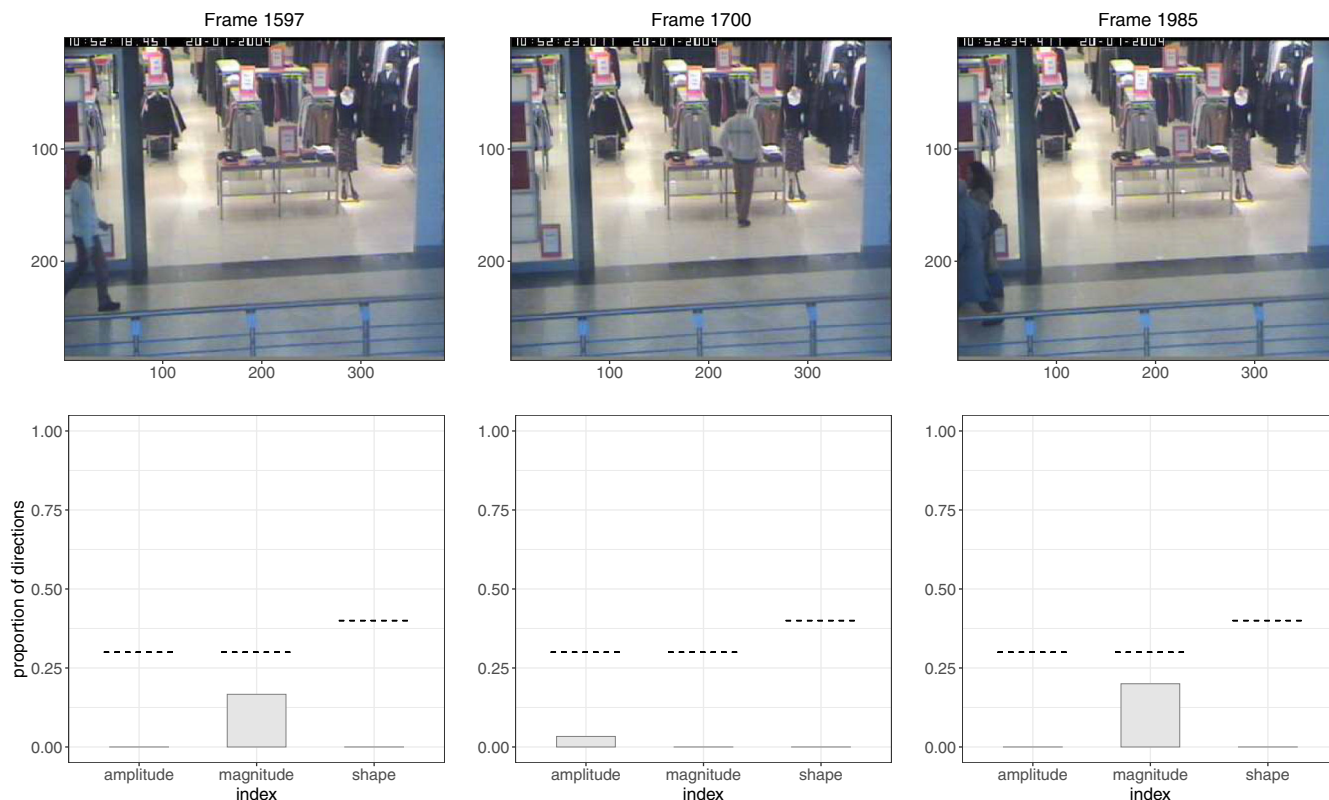


FIGURE 9 Some selected frames not detected as outliers by FST-PRJ1. The bar charts below each frame show the proportion of projections in which that corresponding frame was flagged as an outlier of a particular type. The dotted lines indicate threshold values in Q .

TABLE 2 Computational time in minutes for video data.

Method	Time (minutes)
FOM	47.4
FastMUOD	51.1
MSPLIT	679.7

outlyingness at each grid point of the functional data, and image and video data usually consist of thousands of gridpoints (or pixels) per observation (image or frame). Only a few frames from the beginning and end of the outlying segments were not flagged by FOM.

To briefly examine the computational burden of the three methods, Table 2 shows the computational time for each method to analyse the video data. FOM is the fastest requiring about 47 min to complete the analysis. FastMUOD with 30 projections used about 51 min, whereas MSPLIT required over 679 min (>11 h) to complete the analysis. Although FOM had the fastest running time, it also required the largest amount of random access memory (RAM) for the analysis. MSPLIT and FastMUOD ran in a computer with 64 GB memory, whereas FOM required ≥ 64 GB memory to complete the analysis. The running time experiment was performed on an Ubuntu Server containing an AMD Opteron CPU containing 64 cores (each running at 2.3 GHz) with 512 GB of RAM. The codes used in this experiment are those provided with the published paper on the methods or those provided on the website of the authors without any prior optimisation.

6 | CONCLUSION

The FastMUOD indices, introduced by Ojo et al. (2021), are useful and scalable tools for OD for functional data. However, their use presented in Ojo et al. (2021) was limited to univariate functional data. We sought to address that in this work by presenting some techniques for using these indices for outlier detection in multivariate functional data settings.

To do that, we presented definitions, sample estimates, and the corresponding finite dimensional versions of the FastMUOD indices. Next, we presented and illustrated certain properties of these indices that make them useful for OD in functional data. We then proposed various

techniques for applying the FastMUOD indices to multivariate functional data. Among the various proposed techniques, using random projections showed the most effective results. This involves projecting the multivariate functional data of interest on different unit vectors and then applying FastMUOD indices on the resulting projected univariate functional data. Then, an observation is flagged as an outlier if it is detected as an outlier in at least a fixed proportion of the projection directions. We also proposed the MSA-plot, which visualises the outlyingness of a multivariate observation in terms of the proportion of projection directions in which it is flagged as an outlier (of a specific type).

We demonstrated the proposed methods on various simulated and real datasets and compared their performance to other multivariate functional OD methods. Our simulation results show the need for adequate selection of a threshold triple $Q = (\tau_M, \tau_S, \tau_A)$ of the proportion of projection directions used in determining whether an observation is an outlier (of a particular type). Instead of declaring an observation as an outlier if it is detected as an outlier in any direction (possibly resulting in a high FPR), carefully selecting the threshold helps to control the FPR. A possible direction for improvement is to develop a method to select these threshold values even when the distribution and base model of the data are unknown. With the proposed techniques, the FastMUOD indices add to the available options of OD tools for multivariate functional data.

ACKNOWLEDGEMENT

This work has been partially supported by the Comunidad de Madrid grant EdgeData-CM (P2018/TCS4499, cofunded by FSE & FEDER) and grants TED2021-131264B-I00 (SocialProbing) and PID2019-104901RB-I00, funded by Ministry of Science and Innovation - State Research Agency, Spain, MCIN/AEI/10.13039/ 501100011033 and the European Union "NextGenerationEU"/PRTR. The work is also part of the agreements of the Community of Madrid (Ministry of Education, Universities, Science and Spokesperson) with the Carlos III University of Madrid and the IMDEA Networks Institute for the funding of research projects on SARS-CoV-2 and COVID-19 disease, project names "PredCov-CM: Multi-source and multi-method prediction to support COVID-19 policy decision making" and "COMODIN-CM: COVID-19 Monitoring via Data-Intensive Analysis," which are supported with REACT-EU funds from the European regional development fund "a way of making Europe." Marc G. Genton and Oluwasegun Ojo's research was supported by the King Abdullah University of Science and Technology (KAUST).

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interests.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available in CAVIAR Test Cases at <https://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/>. These data were derived from the following resources available in the public domain: - WalkByShop1front, <http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA2/WalkByShop1front/WalkByShop1front.tar.gz>

ORCID

Oluwasegun Taiwo Ojo  <https://orcid.org/0000-0001-9629-6990>

Antonio Fernández Anta  <https://orcid.org/0000-0001-6501-2377>

Marc G. Genton  <https://orcid.org/0000-0001-6467-2998>

Rosa E. Lillo  <https://orcid.org/0000-0003-0802-4691>

REFERENCES

- Breunig, M. M., Kriegel, H.-P., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers, *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00. New York, NY, USA: Association for Computing Machinery, pp. 93–104.
- Brys, G., Hubert, M., & Rousseeuw, P. J. (2005). A robustification of independent component analysis. *Journal of Chemometrics*, 19(5-7), 364–375.
- Cox, M. A. A., & Cox, T. F. (2008). Multidimensional scaling, *Handbook of data visualization*: Springer Berlin Heidelberg, pp. 315–347.
- Dai, W., & Genton, M. G. (2018). Multivariate functional data visualization and outlier detection. *Journal of Computational and Graphical Statistics*, 27(4), 923–934.
- Dai, W., & Genton, M. G. (2019). Directional outlyingness for multivariate functional data. *Computational Statistics & Data Analysis*, 131, 50–65.
- Dai, W., Mrkvička, T., Sun, Y., & Genton, M. G. (2020). Functional outlier detection and taxonomy by sequential transformations. *Computational Statistics & Data Analysis*, 149, 106960.
- Febrero, M., Galeano, P., & González-Manteiga, W. (2008). Outlier detection in functional data by depth measures, with application to identify abnormal NO_x levels. *Environmetrics*, 19(4), 331–345. <https://onlinelibrary.wiley.com/doi/pdf/10.1002/env.878>
- Happ, C., & Greven, S. (2018). Multivariate functional principal component analysis for data observed on different (dimensional) domains. *Journal of the American Statistical Association*, 113(522), 649–659.
- Happ-Kurz, C. (2020). Object-oriented software for functional data. *Journal of Statistical Software*, 93(5), 1–38.
- Herrmann, M., & Scheipl, F. (2021). A geometric perspective on functional outlier detection. *Stats*, 4(4), 971–1011.
- Huang, H., & Sun, Y. (2019). A decomposition of total variation depth for understanding functional outliers. *Technometrics*, 61(4), 445–458. <https://doi.org/10.1080/00401706.2019.1574241>
- Hubert, M., Rousseeuw, P. J., & Segal, P. (2015). Multivariate functional outlier detection. *Statistical Methods & Applications*, 24(2), 177–202.

- Hyndman, R. J., & Shang, H. L. (2010). Rainbow plots, bagplots, and boxplots for functional data. *Journal of Computational and Graphical Statistics*, 19(1), 29–45. <https://doi.org/10.1198/jcgs.2009.08158>
- Kuhnt, S., & Rehage, A. (2016). An angle-based multivariate functional pseudo-depth for shape outlier detection. *Journal of Multivariate Analysis*, 146, 325–340. Special Issue on Statistical Models and Methods for High or Infinite Dimensional Spaces.
- López-Pintado, S., & Romo, J. (2009). On the concept of depth for functional data. *Journal of the American Statistical Association*, 104(486), 718–734.
- López-Pintado, S., & Romo, J. (2011). A half-region depth for functional data. *Computational Statistics & Data Analysis*, 55(4), 1679–1695.
- Ojo, O. T., Fernández Anta, A., Lillo, R. E., & Sguera, C. (2021). Detecting and classifying outliers in big functional data. *Advances in Data Analysis and Classification*, 2021, 1–36.
- R Core Team (2022). R: A language and environment for statistical computing [Computer software manual]. Vienna, Austria.
- Rousseeuw, P. J., Raymaekers, J., & Hubert, M. (2018). A measure of directional outlyingness with applications to image data and video. *Journal of Computational and Graphical Statistics*, 27(2), 345–359.
- Segaert, P., Hubert, M., & Rousseeuw, P. (2017). mrfDepth: Depth measures in multivariate, regression and functional settings. <https://cran.rproject.org/package=mrfDepth>
- Sguera, C., Galeano, P., & Lillo, R. E. (2016). Functional outlier detection by a local depth with application to NO_x levels. *Stochastic Environmental Research and Risk Assessment*, 30(4), 1115–1130.
- Sun, Y., & Genton, M. G. (2011). Functional boxplots. *Journal of Computational and Graphical Statistics*, 20(2), 316–334. <https://doi.org/10.1198/jcgs.2011.09224>
- Williams, B. H., Toussaint, M., & Storkey, A. J. (2006). Extracting motion primitives from natural handwriting data. In *International Conference on Artificial Neural Networks*, Springer, pp. 634–643.

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Ojo, O. T., Fernández Anta, A., Genton, M. G., & Lillo, R. E. (2023). Multivariate functional outlier detection using the fast massive unsupervised outlier detection indices. *Stat*, 12(1), e567. <https://doi.org/10.1002/sta4.567>