Abstract

This report presents a new strategy to correct the Earth data corrupted by spurious samples that are randomly included in the multiplexed data stream provided by the MADRAS instrument. The proposed strategy relies on the construction of a trellis associated with each scan of the multi-channel image, modeling the possible occurrences of these erroneous data. A specific weight that promotes the smooth behavior of the signals recorded in each channel is assigned to each transition between trellis states. The joint detection and correction of the erroneous data are conducted using a dynamic programming algorithm for minimizing the overall cost function throughout the trellis. Simulation results obtained on synthetic and real MADRAS data demonstrate the effectiveness of the proposed solution.

Index Terms

MADRAS, multi-band imaging, multiplexing, dynamic programming.

1This report provides complementary results to the paper [1].
I. INTRODUCTION

Born from a close collaboration between the Indian and French space agencies (namely, ISRO and CNES, respectively), the MEGHA-Tropiques mission aims at developing a monitoring system dedicated to the study of the tropical atmosphere [2]. The measurements collected over the inter-tropical belt by multiple sensors embedded on the spacecraft platform allow various ocean and atmospheric parameters of interest (e.g., rain rate, profile of water vapor content, sea surface wind) to be determined with high spatial and temporal sampling [3]. These climate and atmospheric parameters are disseminated over the scientific community through academic institutions and national agencies, whose objectives are, e.g., climate research, weather forecasting, and prediction of major events (i.e., monsoons) [4].

The satellite payload is composed of four instruments: GPS-ROSA, a GPS occultation sensor designed to provide atmospheric temperature and humidity profiles; SCARAB, an optical radiometer retrieving the radiation parameters; SAPHIR, a microwave sensor for vertical humidity profiling; MADRAS, a microwave imager used to provide rain and cloud properties [5]. This later sensor, jointly developed by ISRO (for the scan mechanisms) and CNES (for the radio-frequency subsystems), is a passive conical microwave imager measuring the radiation at nine frequency bands, at vertical and horizontal polarizations. Exploitation of the scientific data collected by MADRAS has already motivated several studies to retrieve rainfall parameters, demonstrating the interest in these parameters by the science data users [6]–[10]. However, after a few weeks in orbit, an anomaly in the communication chain between two electronic devices was detected [11]. This anomaly leads to a mixing of the channels that compose the images provided by MADRAS. More precisely, additional data can be randomly inserted into the main data streams associated with each column of the MADRAS images. Visually, these corruptions result in the occurrence of vertical stripe noise, i.e., vertically and contiguously distributed erroneous pixels in the columns that compose the MADRAS images. Stripe noise, which generally comes from undesirable gain and/or offset variations of the sensors, is a common and well-known degradation that affects, for instance, images acquired by push-broom scanners. Thus, destriping has motivated numerous research works for several decades, not only for Earth remote sensing images [12]–[14], but also for biomedical [15], [16] images and astronomical data [17], [18]. Most of these destriping methods consist of locating the affected pixels in the image domain.
or using an appropriate representation (e.g., subspace, wavelet or histogram), and then replacing
them by spatially interpolated or more probable values. However, in the case of the MADRAS
applicative context, such interpolation-like techniques remain prohibited to maintain the highest
integrity of the scientific data and also to guarantee the confidence the scientists may have in their
results. This constraint makes inapplicable all the destriping methods proposed in the literature.
Fortunately, after thorough analysis of the corruption process that affects the MADRAS images,
it appears that the corrupted data streams still contains most of the measurements of interest, but
in a wrong order. By removing the spurious extra data, one may expect that the correct order of
the measurements can be reestablished to obtain exploitable scientific data. This finding opens
the door for a correction method that fulfills the initial requirement of avoiding any creation of
new pixel values, e.g., by interpolation, which is precisely the objective of this paper.

In this paper, we focus on the correction of the anomalies in the scientific data, called
Earth data, collected by the MADRAS instrument. The problem is formulated as the detection
and the removal of spurious data in a data stream resulting from a cyclic multiplexing of
several individual signals. This problem may be encountered in various applicative contexts
where physical data are measured in several channels, such as multi-band (e.g., multispectral
or hyperspectral) imaging. The proposed solution relies on the construction of an oriented
trellis, modeling the possible occurrences of abnormal samples in the data stream. Application-driven weights are proposed and associated with transitions between trellis states. A
similar approach has been adopted in [19] to detect and correct errors encountered in automatic
identification systems benefitting from a cyclic redundancy check. Finally, a Viterbi-like dynamic
programming algorithm [20]–[25] is designed to recover the optimal path of minimal cumulative
weight through the trellis.

This paper is organized as follows. The MADRAS multi-channel images and the problem
to be solved are described in Section II. The strategy proposed to detect and correct possible
anomalies in the MADRAS data is introduced in Section III. Section IV reports experimental
results. Finally, some conclusions are drawn in Section V.
II. PROBLEM STATEMENT

A. The MADRAS data

The scientific data acquired by MADRAS takes the form of a multi-channel image, as depicted in Fig. 5 (1st row) using an arbitrary composition color. This image, composed of \( M = 11 \) individual channels, consists of a set of \( P \) contiguous scans, where one given scan corresponds to a unique column of this image. After sampling correction, each scan is composed of \( T \) multi-valued pixels, called frames. Each frame is thus a vector of \( M \) individual samples and corresponds to a given pixel observed in the \( M \) channels. The number \( T \) of frames depends on the type of acquired data: Earth data, which are considered in this work, are composed of \( T = 526 \) frames. A typical example of a scan is depicted in Fig. 1, where the signals recorded in the \( M \) channels are depicted in distinct colors.

![Fig. 1. Example of a scan without corruption. The Earth data of interest are those not masked by pink areas.](image)

To summarize, a scan can be given in form of a matrix of size \( M \) (channels) \( \times \) \( T \) (frames), where the first (last, resp.) row corresponds to samples assigned to channel \( #M \) (#1, resp.), and the first (last, resp.) column contains the frame \( #1 \) (resp, \( #T \)). However, a given scan of \( T \) frames of \( M \) channels actually results from the reordering of a unique data stream. This data flow contains a cyclic sequence of \( N = MT \) samples that are sequentially and periodically acquired in the \( M \) channels. These notations are gathered in Table I. The relation between a scan and the corresponding data stream is schematically illustrated in Fig. 2.
### TABLE I
NOTATIONS AND NOMENCLATURE

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P$</td>
<td>number of scans (i.e., image columns)</td>
</tr>
<tr>
<td>$T$</td>
<td>number of frames (i.e., pixels per image column)</td>
</tr>
<tr>
<td>$M$</td>
<td>number of channels (i.e., pixel dimension)</td>
</tr>
<tr>
<td>$N$</td>
<td>total number of samples per scan, i.e., in the data stream ($= TM$)</td>
</tr>
<tr>
<td>$x(j)$</td>
<td>sample $#j$ in the data stream</td>
</tr>
<tr>
<td>$x(N)$</td>
<td>full data stream composed of $N$ samples ($= [x(1), \ldots, x(N)]$)</td>
</tr>
<tr>
<td>$S$</td>
<td>number of trellis states ($= $ max. number of glitches per scan)</td>
</tr>
<tr>
<td>$c_{k,j}$</td>
<td>trellis node associated with sample $x(j)$</td>
</tr>
<tr>
<td>$v_{0,k,j}$</td>
<td>branch connecting $c_{k,j-1}$ and $c_{k,j}$ (“$x(j)$ is not a glitch”)</td>
</tr>
<tr>
<td>$v_{1,k,j}$</td>
<td>branch connecting $c_{k-1,j-1}$ and $c_{k,j}$ (“$x(j)$ is a glitch”)</td>
</tr>
<tr>
<td>$d_{0}(k,j)$</td>
<td>weight associated with branch $v_{0,k,j}$</td>
</tr>
<tr>
<td>$d_{1}(k,j)$</td>
<td>weight associated with branch $v_{1,k,j}$</td>
</tr>
<tr>
<td>$D(k,j)$</td>
<td>cumulative weights along the unique path reaching the node $c_{k,j}$</td>
</tr>
<tr>
<td>$N_{k,j}$</td>
<td>number of accepted samples along the path reaching $c_{k,j}$</td>
</tr>
<tr>
<td>$\hat{x}_{k,j}$</td>
<td>sequence of the $N_{k,j}$ samples along the path reaching $c_{k,j}$</td>
</tr>
</tbody>
</table>

### B. Anomalies

The anomalies considered in this work consist of additions of extra samples in the data stream associated with each scan. In the sequel of this paper, these extra samples will be called *glitches*, while the valid samples will be called *measurements*. The term *sample* will thus now stand for undifferentiated data that could be either a glitch, either a measurement. These multiple and random valued insertions in a given scan result in

- the presence of erroneous samples in the multiplexed data stream (the glitches),
- cyclic permutations of the channels after recombining the data stream by demultiplexing (due to the presence of glitches).

More precisely, assume that the successive samples are periodically assigned to channels $M, M - 1, \ldots, 1$. If a glitch appears when acquiring a measurement for channel 3, then this channel will receive an outlier (this glitch), channel 2 will receive the measurement initially
intended for channel 3, channel 1 will receive the measurement initially intended for channel 2, and the shift is propagated into the next frame and until the end of the scan. Note that, due to the $M$-periodic cyclic reordering of the data stream samples into the $M$ channels, the insertion of $M$ glitches between the sample $#n_1$ (belonging to the frame $#f_1$) and the sample $#n_2$ (belonging to the frame $#f_2 > #f_1$) leads to an assignment of the samples after the sample $#n_2$ to the correct channels, subjected to a simple delay of 1 frame. In other words, in this case, the ideal and corrupted scans only differ by a single shift of all the frames after the frame $#f_1$.

A schematic view of the corruption affecting a scan is depicted in Fig. 2, where for conciseness, only $M = 4$ channels have been considered and the data flow is split into $T = 3$ successive frames.

![Diagram of data stream corruption](image)

Fig. 2. Schematic view of anomaly in the data stream. The symbol "G" is used to represent the glitch that has appeared in channel #3.

A typical example of a scan corrupted by several glitches is depicted in Fig. 3, and typical examples of resulting MADRAS images are plotted in Fig. 5 and 6 (2nd rows, left).

### III. Proposed Algorithm

This section describes a new algorithm that is proposed for the detection and removal of glitches in the MADRAS data. The algorithm operates on each scan individually, and the technical developments that follow are related to the analysis of the corresponding data stream
Fig. 3. Example of Earth data corrupted by several anomalies.

composed of $N$ samples denoted as $x(N)$. Moreover, for clarity, the set of the first $j$ observed samples of the data stream is denoted by $x(j) \triangleq [x(1), \ldots, x(j)]$.

A. Trellis design

A trellis is an oriented graph whose nodes are organized into vertical stacks that identify all the possible states of a given system at the same discrete time step. Each node is connected to at least one node from the previous time step and at least one node from the next time step. In this work, a given discrete time step corresponds to a given sample of the data stream, and the states are defined by the number of glitches that have already been detected in the data stream preceding a given sample.

More precisely, the trellis is defined by the following characteristics.

- The trellis contains $S$ nodes per received sample $x(j)$, which are denoted by $c_{k,j}$ ($k = 0, \ldots, S - 1$), where $S$ is the maximum number of glitches. The node $c_{k,j}$ corresponds to the presence of $k$ glitches (modulo $S - 1$) in the set of received samples $x(j)$.
- Each node $c_{k,j}$ is connected to the nodes $c_{k-1,j-1}$ and $c_{k,j-1}$ associated with the previous sample, and $c_{k,j+1}$ and $c_{k+1,j+1}$ associated with the next sample. The trellis is circular in the sense that node $c_{0,j}$ is connected to node $c_{S-1,j-1}$, and node $c_{S-1,j}$ is connected to node $c_{0,j+1}$.
- The vertices connecting the nodes are referred to as branches, or transitions. The branch connecting nodes $c_{k,j-1}$ and $c_{k,j}$ is denoted by $v_{k,j}^0$, and the branch connecting nodes $c_{k-1,j-1}$ and $c_{k,j}$ is denoted by $v_{k,j}^1$. 
• The branch $v_{k,j}^1$ corresponds to the proposition “$x(j)$ is a glitch” for state $k$ whereas the branch $v_{k,j}^0$ corresponds to the proposition “$x(j)$ is not a glitch” for state $k$ (i.e., $x(j)$ is a valid measurement).

• The branches $v_{k,j}^0$ and $v_{k,j}^1$ are assigned weights $d_0(k,j)$ and $d_1(k,j)$, respectively. These weights can be interpreted as the (inverse) probabilities of reaching node $c_{k,j}$ from nodes $c_{k,j-1}$ or $c_{k-1,j-1}$, respectively, given the new observed sample $x(j)$. Consequently, these weights should penalize the transitions $v_{k,j}^0$ and $v_{k,j}^1$ according to their respective likelihoods.

The choice of the weights $d_0(k,j)$ and $d_1(k,j)$ is discussed in Section III-C.

Based on this trellis, the most likely configuration of glitch occurrences in the set of samples $x(1), \ldots, x(j)$ can be identified by the path connecting the series of $j$ successive nodes of minimum cumulative costs. This optimal path can be recovered by a Viterbi-like dynamic programming algorithm described in the next subsection.

B. Viterbi algorithm

The Viterbi algorithm removes at each time instant $j$ all branches but one reaching the states $c_{k,j}$, such that each state $c_{k,j}$ can be reached by only one unique path through the trellis. More specifically, the following rules are applied to sequentially prune the trellis.

• At time $j-1$, each node $c_{k,j-1}$ has been assigned the cumulative weight $D(k,j-1)$ defined as the sum of the weights of the branches of the unique path reaching it.

• At node $c_{k,j}$, if the sum of $D(k,j-1)$ and $d_0(k,j)$ is smaller than the sum of $D(k-1,j-1)$ and $d_1(k,j)$:
  
  – The branch $v_{k,j}^1$ is removed from the trellis, and the branch $v_{k,j}^0$ is kept.
  
  – The sample $x(j)$ is accepted as a valid measurement at state $k$.
  
  – The sequence of $N_{k,j}$ samples that have been accepted along the unique path reaching the node $c_{k,j}$ is denoted by $\hat{x}_{k,j}$, with final value $\hat{x}_{k,j}(N_{k,j}) = x(j)$. Note that $N_{k,j} = j - k$ since $k$ glitches have been detected at state $k$ among the $j$ already analyzed samples.

• Otherwise, if the sum of $D(k,j-1)$ and $d_0(k,j)$ is larger than the sum of $D(k-1,j-1)$ and $d_1(k,j)$ at node $c_{k,j}$:
  
  – The branch $v_{k,j}^0$ is removed from the trellis, and the branch $v_{k,j}^1$ is kept.
– The sample \( x(j) \) is identified as a glitch and is not accepted as a valid measurement by state \( k \).

– The resulting sequence of \( N_{k,j} \) valid measurements \( \hat{x}_{k,j} \) is hence given by \( \hat{x}_{k,j} = \hat{x}_{k-1,j-1} \), with final value \( \hat{x}_{k,j}(N_{k,j}) = \hat{x}_{k-1,j-1}(N_{k-1,j-1}) \). Note also that in this case \( N_{k,j} = N_{k-1,j-1} \).

After receiving the last sample \( x(N) \), the path through the trellis with the smallest cumulative weight \( D(k,N) \), terminating at the optimal node, denoted \( c_{k,N} \), is chosen. The corrected set of \( N_{k,N} \) samples is given by the vector \( \hat{x}_{k,N} \).

The design of the trellis and of the Viterbi algorithm are illustrated in Fig. 4 on a toy example with a data stream composed of 12 samples, including 1 glitch and 11 valid measurements, distributed into \( T = 3 \) frames of \( M = 4 \) channels (the toy example corresponds to that shown in Fig. 3), using a trellis of \( S = 5 \) possible states with constant and artificial branch weights. The trellis is depicted at the instant of reception of the 10th sample \( x(10) \).

C. Branch weights

As previously stated, the choice of weights must promote the most likely transition from a node at time instant \( j - 1 \) to a connected node at time instant \( j \), given the new sample \( x(j) \). For the application considered in this paper, the weights assigned to the trellis branches are based on local derivatives of the samples \( x(j) \) with future and past received samples. Indeed, since the MADRAS instrument records physical parameters, the evolution between two successive valid measurements in a given channel is expected to be rather smooth while the difference in value between distinct channels is large, as illustrated in Fig. 1. More precisely, the weights are defined as follows.

**Weights** \( d_0(k,j) \). The weight assigned to the branch \( v_{k,j}^0 \) connecting nodes \( c_{k,j-1} \) and \( c_{k,j} \) is given by the square root of the absolute difference between \( x(j) \) and the last valid measurement received by state \( k \) that is supposed to belong to the same channel as \( x(j) \), namely \( \hat{x}_{k,j}(N_{k,j} - M + 1) \):

\[
d_0(k,j) = \left( |x(j) - \hat{x}_{k,j}(N_{k,j} - M + 1)| \right)^p. \tag{1}
\]

As expected, this weight will be small if the new sample \( x(j) \) is not a glitch and should be assigned to the same channel as \( \hat{x}_{k,j}(N_{k,j} - M + 1) \).
Based on this trellis, the most likely configuration of glitch occurrences in the set of samples \( k, j \) is discussed in Section III-C.

**Weights** \( d_1(k, j) \). The weights \( d_1(k, j) \) assigned to the branches \( v_{k,j}^1 \) connecting nodes \( c_{k-1,j-1} \) and \( c_{k,j} \) are chosen equal for all the transitions \( v_{k,j}^1, k = 0, \ldots, S - 1 \), hence \( d_1(j) = d_1(k, j) \).

The weight \( d_1(j) \) is derived from a “robustified” mean of the absolute differences between the last valid measurement \( \hat{x}_{k,j}(N_{k,j} - M + 1) \) that the state \( k \) has received and the \( N_f \) future samples \( x(j + 1), \ldots, x(j + N_f) \). More precisely,

\[
d_1(j) = \frac{\alpha}{S} \sum_{k=0}^{S-1} \bar{\gamma}_{k,j}, \quad \text{where} \quad \bar{\gamma}_{k,j} = \frac{2}{N_f} \sum_{i=1}^{N_f/2} \gamma_{k,j}(i) \quad \tag{2}
\]

\[
\gamma_{k,j} = \text{sort}_{i=1,\ldots,N_f}^+ \left\{ \left( x(j + i) - \hat{x}_{k,j}(N_{k,j} - M + 1) \right)^p \right\}. \quad \tag{3}
\]

Only the \( N_f/2 \) smallest differences associated with each state are considered in the average to discard any distance that could correspond with the presence of another but not yet detected

![Trellis diagram](image-url)
glitch in the future samples. The parameters $N_f$, $p$ and $\alpha$ have been chosen as $N_f = 10$, $p = \frac{1}{2}$ and $\alpha = 1.77$ after testing different possible values and keeping those providing the best results, i.e., according to a cross-validation technique, for the application to MADRAS data considered here.

IV. Experiments

A. Simulated data

To assess the performance of the proposed algorithm, an anomaly-free MADRAS image has been artificially corrupted by simulated glitches. The simulated glitch corruptions are designed to closely resemble those observed on corrupted MADRAS images and consist of random values (drawn in the image dynamics range) that are inserted at (groups of) random positions in the data stream. Four simulated datasets with several degrees of anomaly severities have been considered: from scenario 1, which corresponds to relatively clean data, to scenario 4, which corresponds to highly corrupted data. The numbers of glitches and corrupted samples for the four scenarios are reported in Tables II and III. The proposed algorithm has been also compared with two standard destriping methods from the literature. The first one, denoted as wFFT, consists of a combined wavelet-FFT filtering [15]. The second method, denoted as TV, formulates the destriping task as a TV-regularized optimization problem [26], solved here using an alternating direction method of multipliers (ADMM). Three kinds of performance analysis have been conducted. In a first analysis, reported in paragraph IV-A1, the corrected image is visually compared with the original uncorrupted image. Then, in paragraph IV-A2, a quantitative analysis is conducted on the synthetic datasets to evaluate the ability of the proposed algorithm to detect glitches. More precisely, the $\ell_0$-norm of the correction error is computed to measure (i.e., to count), the number of badly corrected samples and frames (i.e., pixels) between the original (uncorrupted) image and the corrected one. Let $X = [x_1, \ldots, x_P]$ denote the matrix of the $P$ data streams $x_p = [x_p(1), \ldots, x_p(N)]^T$ associated with the $P$ contiguous scans that compose the original image. Denote as $\hat{x}_p(n)$ ($p = 1, \ldots, P$, $n = 1, \ldots, N$) the corrected samples. The proposed error measure is

$$e_0 = \|X\|_0 = \sum_{p=1}^P \|x_p - \hat{x}_p\|_0 = \sum_{p=1}^P \sum_{n=1}^N \delta(x_p(n) - \hat{x}_p(n))$$
where the Kronecker function $\delta(\cdot)$ is defined as

$$
\delta(x) = \begin{cases} 
1, & \text{if } x \neq 0; \\
0, & \text{if } x = 0.
\end{cases}
$$
Fig. 6. MADRAS data, scenario 4. 1st row: original image. 2nd row: corrupted image (left) and \( \ell_0 \)-norm errors for corrupted image (channel 6, right). 3rd to 5th rows: corrected image (left) and \( \ell_0 \)-norm errors for corrected image (right, channel 6) for the proposed, wFFT and TV algorithms, respectively. The multichannel images (left) are depicted using an arbitrary synthetic color composition.

Note that this \( \ell_0 \)-measure is particularly drastic since it penalizes all errors equally, whatever the absolute difference between the original and corrected samples. However, in the MADRAS applicative context, preserving integrity of uncorrupted samples is crucial, which can be assessed only by this sample-wise comparison before and after corruption. Finally, paragraph IV-A3
compares the correction performance of the proposed algorithm with those obtained by two destriping methods. In addition to the $\ell_0$-norm based quality measure, the peak signal-to-noise ratio (PSNR) which relies on a $\ell_2$-norm reconstruction error, is considered.

1) Visual inspection: Visual inspection of the corrupted and corrected images has been conducted for scenarios 1 and 4 (Figs. 5 and 6, respectively). The corrected images by the proposed algorithm as well as the wFFT and TV algorithms are visually compared with the original uncorrupted image: original data (1st row), corrupted data and $\ell_0$-norm of the error between the original and corrupted data (2nd row), corrected data by the proposed algorithm and corresponding $\ell_0$-norm of the error between the original and corrected data (3rd row), corrected data by the wFFT algorithm and corresponding $\ell_0$-norm of the error between the original and corrected data (4th row) and corrected data by the TV algorithm and corresponding $\ell_0$-norm of the error between the original and corrected data (3rd row). Note that the $\ell_0$-norm error takes two values (0 in green and 1 in dark red) that indicate which samples have been properly or wrongly corrected, respectively. Note also that some scans entirely appear as blue lines. This is due to another sensor anomaly that has been simulated, which is different from the one considered in this work but easily detectable (it consists of scans with all constant values and will not be further discussed here). Moreover, note that due to the insertion of glitches, the corrupted scans contain less than $N$ valid measurements. Thus, once these glitches have been detected and removed by the proposed algorithm, no additional measurements can be recovered at the end of the scan since these measurements are not contained in the corrupted data. These missing measurements appear as dark blue pixels in the corrected data of Figs. 5 and 6 (3rd row, left).

The corrected images obtained with the proposed method are visually of remarkably good quality, even in the most perturbed case. In particular, for less corrupted data (scenario 1), almost all the glitches have been properly detected and removed. For this scenario, only some scans (5 scans around scan #1600) appear as improperly corrected after frames/pixels #300. A thorough analysis of these scans allows two kinds of correction errors to be identified: First, for scan #1572 of scenario 1, an extra glitch has been detected, likely due to the fact that all channels contain similar values in the concerned frame. This leads to channel permutations of the remaining frames of the scan as illustrated by a color permutation after frame #358 in Fig. 7. Second, the algorithm sometimes removes an entire frame/pixel when several consecutive extra glitches have been detected in the data stream. This results in the deletion of a frame/pixel in all
the channels. Consequently, it has a small impact on the visual inspection since spatial coherence has been preserved. This behavior is illustrated in Fig. 8. The \( \ell_0 \)-norm of the error computed on scan #1627 of scenario 1, indicates that the proposed correction is completely wrong for all the frames/pixels after the frame #307. Even if the corrected scan (3rd plot) seems to be in very good agreement with the original data (1st plot) since there is no change in curve colors, the residual error (pixelwise distance in each channel) is non-zero for all the frames/pixels after frame #307 (4th plot). A simple explanation is that the original and corrected scans only differ by an entire
frame/pixel that has been incorrectly removed. Indeed, the residual error now computed with a shift of one frame/pixel is zero in all the frames/pixels for every channels (see the last plot).

Fig. 8. Scenario 1: end of scan #1627 appears as badly corrected due to extra detected glitches, that results in the deletion of an entire frame. Note that the overall behavior of the individual signals in each channel has been preserved after correction (no channel permutation).
In contrast, none of the alternative algorithms wFFT and TV yield satisfactory results: while for the less corrupted case (cf., Fig. 5), wFFT and TV yield only moderate visually artifacts, all of the measurements available in the corrupted data are altered by the algorithms. For a severe corruption level (cf., Fig. 6), the algorithms also fail to achieve visual improvements.

2) **Detection performance statistics:** A comprehensive quantitative performance analysis is conducted for evaluating the ability and accuracy of the proposed algorithm for detecting glitches. Two further scenarios with relatively severe corruptions (scenario 2 and 3) are simulated in addition to those used in the previous paragraph for visual inspection. Scenario 3 contains twice as many glitches as scenario 2, yet affecting exactly the same scans (thus resulting in a similar number of corrupted samples). The total number of glitches in scans 1 to 1000 for the four scenarios are reported in Table II (second column).

Columns 3 to 11 of Table II report the number of non-detected (related to the probability of detection) and incorrectly detected glitches (related to the probability of false alarm) within a local neighborhood ranging from $\Delta = 0$ (exact localization) to $\Delta = 8$ of the glitch locations. The results demonstrate that the proposed algorithm is capable of detecting and correcting a very large majority of the glitches originally present in the data stream at their precise location, and nearly all of them in a small neighborhood.

3) **Correction performance statistics:** The correction performance is further investigated by means of the PSNR (expressed in dB), which is a well-admitted quality measure for image processing applications. As in the previous paragraph, the number of corrupted samples in the corrected data has also been counted using the $\ell_0$-norm of the reconstruction error (expressed as percentages). Quantitative results are reported in Table III for all four scenarios. They demonstrate that the glitch detection and correction algorithm is highly effective and insensitive to the level of corruption, contrary to the two destriping methods. Even for scenario 4, for which initially almost 85% of samples were corrupted, less than 0.3% samples remain corrupted after correction with the proposed algorithm, i.e., more than 99.7% of the corrected samples are identical to the original measurements. Note also that both TV and wFFT methods slightly improve the PSNR measures when compared to the original image but are unable to keep the correct samples unaltered. As a consequence, these techniques cannot maintain the integrity of the data, since they slightly improve the (visual) quality of the corrupted images while failing to recover any of the valid measurements.
TABLE II
Algorithm performance as number of glitches not detected (top) and incorrectly detected (bottom).

<table>
<thead>
<tr>
<th>Sc.</th>
<th>#glitches</th>
<th>∆=0</th>
<th>∆=1</th>
<th>∆=2</th>
<th>∆=3</th>
<th>∆=4</th>
<th>∆=5</th>
<th>∆=6</th>
<th>∆=7</th>
<th>∆=8</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>267</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>#2</td>
<td>23970</td>
<td>106</td>
<td>31</td>
<td>13</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>109</td>
<td>34</td>
<td>17</td>
<td>9</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>#3</td>
<td>47894</td>
<td>192</td>
<td>128</td>
<td>78</td>
<td>44</td>
<td>21</td>
<td>12</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>236</td>
<td>172</td>
<td>122</td>
<td>88</td>
<td>65</td>
<td>56</td>
<td>48</td>
<td>46</td>
<td>44</td>
</tr>
<tr>
<td>#4</td>
<td>136378</td>
<td>611</td>
<td>398</td>
<td>270</td>
<td>116</td>
<td>66</td>
<td>30</td>
<td>13</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>853</td>
<td>640</td>
<td>512</td>
<td>358</td>
<td>308</td>
<td>272</td>
<td>255</td>
<td>248</td>
<td>247</td>
</tr>
</tbody>
</table>

TABLE III
Algorithm performance as percentage of corrupted samples and PSNR before and after correction.

<table>
<thead>
<tr>
<th>Sc.</th>
<th>corrupted data samples (%)</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>original proposed wFFT TV</td>
<td>original proposed wFFT TV</td>
</tr>
<tr>
<td>#1</td>
<td>7.81 0.20 100 100 23.5 44.3 33.3 36.0</td>
<td></td>
</tr>
<tr>
<td>#2</td>
<td>55.28 0.13 100 100 15.1 46.1 19.8 21.8</td>
<td></td>
</tr>
<tr>
<td>#3</td>
<td>55.07 0.14 100 100 14.3 46.1 18.9 20.3</td>
<td></td>
</tr>
<tr>
<td>#4</td>
<td>84.67 0.29 100 100 12.1 42.7 15.2 15.6</td>
<td></td>
</tr>
</tbody>
</table>

B. Real MADRAS data

The proposed algorithm has been applied on the real image #1391 – 1392 acquired by MADRAS, which is considered as being very strongly corrupted. Scans 1 – 1298 of the original image are plotted in Fig. 9 together with the corresponding corrected image (artificial color compositions). While the corrected image is not perfect, most of the glitches have been detected and corrected. A thorough analysis of the residual corruptions in the corrected image reveals that a large part of them are due to anomalies that are different from the one considered in this work. Fig. 10 summarizes as an histogram the number of detected and corrected glitches per scan. For this image, most of the scans are detected to be corrupted, and a large number of them severely. More precisely, the algorithm has detected 147820 glitches, which roughly corresponds to 57 glitches per scan on average, and a considerable number of scans with more than 100 glitches.
C. Simulated LANDSAT data

The proposed method has been also applied to a real LANDSAT dataset, corrupted by the glitch inclusion process described in paragraph II-B. More precisely, 8 bands of a $500 \times 1000$ real LANDSAT image have been corrupted according to the 4 scenarios considered in paragraph IV-A. The corrected images by the proposed algorithm as well as the wFFT and TV algorithms are visually compared with the original uncorrupted image in Fig. 11 and 12 for scenarios 1 and 4, respectively: original data (1st row), corrupted data and $\ell_0$-norm of the error between the original and corrupted data (2nd row), corrected data by the proposed algorithm and corresponding $\ell_0$-norm of the error between the original and corrected data (3rd row), corrected data by the wFFT.
algorithm and corresponding $\ell_0$-norm of the error between the original and corrected data (4th row) and corrected data by the TV algorithm and corresponding $\ell_0$-norm of the error between the original and corrected data (3rd row).

Fig. 11. LANDSAT data, scenario 1. 1st row: original image. 2nd row: corrupted image (left) and $\ell_0$-norm errors for corrupted image (channel 6, right). 3rd to 5th rows: corrected image (left) and $\ell_0$-norm errors for corrected image (right, channel 6) for the proposed, wFFT and TV algorithms, respectively. The multichannel images (left) are depicted using an arbitrary synthetic color composition.

Quantitative results obtained by the proposed detection and correction algorithm as well as the wFFT and TV methods are reported in Table IV. The conclusions are similar to those for the
Fig. 12. LANDSAT data, scenario 4. 1st row: original image. 2nd row: corrupted image (left) and $\ell_0$-norm errors for corrupted image (channel 6, right). 3rd to 5th rows: corrected image (left) and $\ell_0$-norm errors for corrected image (right, channel 6) for the proposed, wFFT and TV algorithms, respectively. The multichannel images (left) are depicted using an arbitrary synthetic color composition.

qualitative and quantitative results obtained on MADRAS data, reported in Sections IV-A1 and IV-A3, respectively: The proposed solution yields excellent detection and correction performance, regardless of the corruption level, while wFFT and TV yield, at best, slight visual improvements for lightly degraded images, yet completely fail to recover any of the valid measurements.
TABLE IV
ALGORITHM PERFORMANCE AS PERCENTAGE OF CORRUPTED SAMPLES AND PSNR BEFORE AND AFTER CORRECTION.

<table>
<thead>
<tr>
<th>Sc.</th>
<th>corrupted data samples (%)</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>original</td>
<td>proposed</td>
</tr>
<tr>
<td>#1</td>
<td>3.13</td>
<td>0.23</td>
</tr>
<tr>
<td>#2</td>
<td>50.17</td>
<td>0.24</td>
</tr>
<tr>
<td>#3</td>
<td>50.18</td>
<td>0.24</td>
</tr>
<tr>
<td>#4</td>
<td>79.22</td>
<td>0.20</td>
</tr>
</tbody>
</table>

V. CONCLUSION

We proposed a simple and efficient Viterbi-like dynamic programming algorithm for the detection and removal of glitches from multiplexed data streams of multi-channel measurements. Experiments conducted on simulated and real data provided by the MADRAS instrument demonstrated its excellent correction performance for both low and high corruption levels. The efficiency and performance of the algorithm were achieved by a concise modeling of the process corrupting the data. One of the specificities of the proposed correction algorithm was that no interpolation or approximation schemes were used and only valid original measurements were precisely recovered. The algorithm is operational [27] and has already been used to correct the data stream provided by the MADRAS instrument for its exploitation by the scientific community [28], [29]. Its versatility to correct data acquired by other modalities was further assessed by analyzing LANDSAT data. Future work will include other entities of geophysical multi-channel data, notably hyperspectral images, and will consider different types of anomalies.

ACKNOWLEDGMENTS

The authors would like to thank Prof. Huanfeng Shen, Wuhan University, for providing the ADMM code used to solve the variational problem in [26].

REFERENCES


