Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **447,** 1471–1483 (2015) doi:10.1093/mnras/stu2453

UNIMAP: a generalized least-squares map maker for *Herschel* data

Lorenzo Piazzo,[1⋆] Luca Calzoletti,[2,3] Fabiana Faustini,[2] Michele Pestalozzi,[4] Stefano Pezzuto,[4] Davide Elia,[4] Anna di Giorgio[4] and Sergio Molinari[4†]

[1]*DIET Department, Sapienza University, Rome, Italy*
[2]*ASDC Centre, Italian Space Agency, Rome, Italy*
[3]*ESAC Centre, Herschel Science Centre, Madrid, Spain*
[4]*IAPS Institute, National Institute of Astrophysics, Rome, Italy*

Accepted 2014 November 19. Received 2014 November 19; in original form 2014 May 23

ABSTRACT

The *Herschel* space telescope hosts two infrared photometers having an unprecedented resolution, sensitivity and dynamic range. The map making, i.e. the formation of sky images from the instruments' data, is critical for the full exploitation of the satellite and is a difficult task, since the readouts are affected by several disturbances, most notably by correlated noise. An effective map making approach is based on generalized least squares (GLS). However, when applied to *Herschel* data this approach poses several challenges and requires a specific pre- and post-processing. In the paper, we describe these challenges and introduce a set of algorithms and procedures which successfully address the issues. We also describe the implementation of the procedures and how these are integrated into an image formation software called UNIMAP, which is the first GLS map maker capable of automatically producing quality *Herschel* images with manageable memory and complexity requirements.

Key words: instrumentation: photometers – methods: data analysis – techniques: image processing.

1 INTRODUCTION

Herschel is a space telescope operated by the European Space Agency (ESA) from year 2009 to 2013 (Pilbratt et al. 2010). The satellite hosted three instruments, namely the Photodetector Array Camera and Spectrometer (PACS; Poglitsch et al. 2010), the Spectral and Photometric Imaging Receiver (SPIRE; Griffin et al. 2010) and the Heterodyne Instrument for the Far-Infrared (De Graauw et al. 2010). In this paper, we are interested in the *Herschel*'s photometers, namely PACS and SPIRE, which were equipped with imaging cameras having an unprecedented resolution, sensitivity and dynamic range. These are therefore key instruments, the data of which will be exploited in the years to come. The formation of sky images from the photometers' readouts, a process also called data reduction or map making, is clearly critical for the full exploitation of the satellite and is a difficult task, due to the many disturbances affecting the output. One of the most annoying problems is that the detectors' noise spectrum is not flat but rises at low frequencies. This noise, also called the $1/f$ noise, is temporally correlated, in the sense that there is statistical dependency between successive readouts, and causes long lasting departures of the data from the baseline level, complicating the image formation.

In the presence of $1/f$ noise, an effective map making technique is based on generalized least squares (GLS; Aitken 1935). The first application of this approach to astronomical imaging was the reduction of the *Cosmic Background Explorer* data (Janssen & Gulkis 1992). The technique has been studied and exploited during the 1990s, when both its theoretical framework was clarified, e.g. Tegmark (1997), and efficient implementations, suitable for large data sets, were introduced, e.g. Wright, Hinshaw & Bennett (1996). Recently, the GLS approach has been exploited for the reduction of *Herschel* data. However, this application turned out to be a difficult one and posed several challenges. Besides the large data volume, which requires efficient implementations, and the presence of discontinuities in the data stream due to cosmic rays, which requires a specific preprocessing, there are two major points that need to be addressed. First, the noise is spatially correlated too (Traficante et al. 2011), in the sense that adjacent detectors experience similar noise. This spatial correlation makes the GLS implementation more involved, as we will see. Secondly, the GLS approach introduces a distortion, i.e. a signal dependent error (Traficante et al. 2011; Piazzo et al. 2012). This distortion may be severe for *Herschel*, due to the high dynamic range and resolution of the instruments, and may drastically reduce the image quality.

⋆E-mail: lorenz@infocom.uniroma1.it
†This work was partially funded by the Sapienza University Research Projects no. C26A11825Y/2011 and C26A1324B9/2013 and by the Italian Space Agency within the Hi-GAL project, contracts I/005/11/00 and I/029/12/0.

Many GLS map makers exist for *Herschel* data and several were evaluated and compared during a workshop recently organized by the National Aeronautics and Space Administration and the ESA, see the reports of Paladini et al. (2013) and Xu et al. (2013). The first is MADMAP (Cantalupo et al. 2010), which is included in the *Herschel* Interactive Processing Environment (HIPE), the standard data reduction pipeline. Unfortunately, MADMAP was found to be among the worst performers (Paladini et al. 2013), a fact likely due to a poor pre-processing and not to the GLS implementation itself. A second GLS map maker is SANEPIC (Patanchon et al. 2008), which is important because it satisfactorily addresses the spatial correlation issue, by accounting for the noise cross-spectrum. A third map maker is TAMASIS (Chanial & Panuzzo 2013) which, in addition to dealing effectively with the spatial correlation, implements an accurate data model, allowing to deconvolve the instrument point spread function (PSF) thereby increasing quality and resolution. However, both SANEPIC and TAMASIS do not address the distortion problem and the resulting images may be affected by artefacts, see Mecina et al. (2014) for an example. Yet another GLS map maker is ROMAGAL (Traficante et al. 2011), which, however, requires human intervention in the pre-processing, making it not practical.

An additional GLS map maker for *Herschel* data, described in this paper, is UNIMAP,[1] which was the first mapper to successfully address both the spatial correlation and the distortion issues. Moreover, it has the lowest memory requirements among all existing *Herschel* map makers, which is a key feature, given the large size of the data, and moderate computational requirements. As a result, UNIMAP found widespread diffusion in the Herschel community and was used to reduce the data of several *Herschel* key programmes, e.g. HerMES (Viero et al. 2013) and Hi-GAL (Elia et al. 2013). An interface towards UNIMAP will be included in HIPE release 13.

Besides the GLS core, UNIMAP is based on two major building blocks, namely a spatial correlation removal method, exploiting an alternating least squares (ALS) algorithm, and a distortion removal method, exploiting the post-processing for GLS (PGLS) algorithm.[2] To keep the presentation manageable, the full analysis of these algorithms has been carried out in two companion papers, namely Piazzo, Panuzzo & Pestalozzi (2015) and Piazzo et al. (2012). In this paper, we present the GLS implementation and an overall description of the software, explaining how the ALS and PGLS are integrated with the GLS to realize a quality map maker. We also substantially expand Piazzo et al. (2012) by better analysing the GLS distortion, introducing tools to evaluate its impact and giving a new version of the weighted GLS (WGLS) approach, which is another useful post-processing step. Moreover, we present a rich set of examples[3] and experiments, useful to understand the impact of the parameters on the processing. As a result the paper may be of interest both to UNIMAP users and to imaging systems designers.

The paper is organized as follows. In Section 2, we introduce some basic image formation concepts. In Section 3, we describe the instruments and develop the data model. In Section 4, we present an overview of UNIMAP and describe the pre-processing. In Section 5, we present the reduction strategy and the ALS algorithm. In Section 6, we discuss the GLS map making and distortion. In Section 7, we describe the PGLS distortion removal. In Section 8, we discuss the performance and in Section 9, we give the conclusions.

Notation. We use uppercase letters to denote matrices and boldface lowercase letters to denote vectors, e.g. A and $\mathbf{b}$. We use subscript to denote the elements of vectors and matrices, e.g. $A_{i,j}$ and b_k. We use a superscript T to denote matrix or vector transposition, e.g. A^{T}. We use $|\mathbf{x}|^2 = \mathbf{x}^{\mathrm{T}}\mathbf{x}$ to denote the squared magnitude of the column vector $\mathbf{x}$. We use $E\{.\}$ to denote expectation. We use brackets to denote functions of a continuous variable, e.g. $f(x)$, and square brackets to denote functions of a discrete variable (sequences), e.g. $f[k]$. In particular, $\delta[k]$ is the discrete unit pulse, such that $\delta[k] = 1$ for $k = 0$ and $\delta[k] = 0$ elsewhere.

2 BASIC CONCEPTS

We review some basic models and techniques needed in the paper.

2.1 Image synthesis in the presence of noise

Consider an image of M pixels, represented by an $M \times 1$ vector $\mathbf{m}$. The image is observed with an instrument producing $D \gg M$ readouts, represented by a $D \times 1$ *data* vector $\mathbf{d}$. Assuming a linear, noisy instrument, the data vector can be written as

$$\mathbf{d} = P\mathbf{m} + \mathbf{n} = \mathbf{s} + \mathbf{n}, \tag{1}$$

where P is a $D \times M$, full-rank matrix, which depends on the instrument and on the observation protocol, $\mathbf{n}$ is a zero mean, random *noise* vector and we introduced the *signal* vector $\mathbf{s} = P\mathbf{m}$ representing the ideal, noiseless output. The image formation problem is that of producing an estimate of $\mathbf{m}$, denoted by $\bar{\mathbf{m}}$, knowing $\mathbf{d}$, P and the statistics of $\mathbf{n}$. This is a classical problem having several established solutions. A simple and effective one is based on least squares (LS) and is summarized below.

Since for a generic image $\mathbf{x}$ the ideal output is $P\mathbf{x}$, we can minimize $|\mathbf{d} - P\mathbf{x}|^2$ for varying $\mathbf{x}$ and obtain the image producing the output closest to the actual data vector. This image is the LS estimate of $\mathbf{m}$ and is given by

$$\bar{\mathbf{m}} = (P^{\mathrm{T}}P)^{-1}P^{\mathrm{T}}\mathbf{d}. \tag{2}$$

The corresponding LS fit to the signal is given by

$$\bar{\mathbf{s}} = P\bar{\mathbf{m}} = P(P^{\mathrm{T}}P)^{-1}P^{\mathrm{T}}\mathbf{d}. \tag{3}$$

LS estimation is an effective technique when the noise is white. Indeed, in white Gaussian noise it yields the maximum-likelihood (ML) estimate. On the contrary, when the noise is not white, LS performs poorly. In this case, by denoting by $N = E\{\mathbf{n}\mathbf{n}^{\mathrm{T}}\}$ the noise covariance matrix, we can use a GLS approach and minimize $|N^{-1/2}(\mathbf{d} - P\mathbf{x})|^2$ for varying $\mathbf{x}$. The solution is the GLS estimate given by

$$\bar{\mathbf{m}} = (P^{\mathrm{T}}N^{-1}P)^{-1}P^{\mathrm{T}}N^{-1}\mathbf{d}, \tag{4}$$

which, in Gaussian noise, is also the ML estimate (Cantalupo et al. 2010).

2.2 Image synthesis in the presence of drift

We now assume that the instrument introduces a second, deterministic disturbance that we call a drift and that is represented by a $D \times 1$ *drift* vector $\mathbf{y} = X\mathbf{a}$, where X is a given, full rank, $D \times K$

[1] UNIMAP is an evolution of ROMAGAL. The name is a shorthand for 'University of Rome Map Maker'. The release described in the paper is 5.5.0. The software is available at http://infocom.uniroma1.it/unimap.

[2] The PGLS has been implemented in MADMAP and ROMAGAL too.

[3] In the examples, we concentrate on PACS, because this is the most challenging instrument. Moreover, we only present selected parts of the reduction process, for the sake of brevity. However, the full reductions can be found in the online appendix, together with an example covering SPIRE data.

matrix, termed the *drift* matrix, and $\mathbf{a}$ is a $K \times 1$ unknown coefficients vector, with $K \ll D$. Then the data vector is

$$\mathbf{d} = P\mathbf{m} + X\mathbf{a} + \mathbf{n} = \mathbf{s} + \mathbf{y} + \mathbf{n}. \quad (5)$$

In the presence of the drift, both LS and GLS can be extended to the joint estimation of the image and coefficients. Indeed, by introducing the $D \times (M + K)$ matrix $A = [P, X]$ and the $(M + K) \times 1$ vector $\mathbf{z} = [\mathbf{m}^{\mathrm{T}}, \mathbf{a}^{\mathrm{T}}]^{\mathrm{T}}$, the data vector can be written as

$$\mathbf{d} = A\mathbf{z} + \mathbf{n}$$

and the problem becomes that of estimating $\mathbf{z}$. Then, by exploiting the LS approach and assuming that A is full rank, we obtain

$$\bar{\mathbf{z}} = \begin{bmatrix} \bar{\mathbf{m}} \\ \bar{\mathbf{a}} \end{bmatrix} = (A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}\mathbf{d}, \quad (6)$$

which will be called the joint LS (JLS) estimate. Similarly, by exploiting the GLS approach, we obtain

$$\bar{\mathbf{z}} = \begin{bmatrix} \bar{\mathbf{m}} \\ \bar{\mathbf{a}} \end{bmatrix} = (A^{\mathrm{T}}N^{-1}A)^{-1}A^{\mathrm{T}}N^{-1}\mathbf{d}, \quad (7)$$

which will be called the joint GLS (JGLS) estimate.

2.3 Filtering and other basic operations

Given n numbers organized into a vector $\mathbf{x} = [x_1, \ldots, x_n]$, their (arithmetic) mean will be denoted by $\mathrm{mea}(\mathbf{x}) = (\sum_{k=1}^{n} x_k)/n$ and the median by $\mathrm{med}(x_1, \ldots, x_n)$ or by $\mathrm{med}(\mathbf{x})$. With a terminology abuse we also introduce the variance, given by $\mathrm{var}(\mathbf{x}) = (\sum_{k=1}^{n} x_k^2)/n - [\mathrm{mea}(\mathbf{x})]^2$.

A basic operation used in the paper is the linear filtering of a sequence $x[k]$ with an impulse response $h[k]$, producing an output sequence $y[k] = x[k] * h[k] = \sum_{n=-\infty}^{\infty} x[n]h[k-n]$, where $\star$ denotes convolution. Moreover, given a sequence $x[k]$, we can compute the median over a sliding window of $2w + 1$ samples and subtract it from $x[k]$ to produce $y[k] = x[k] - \mathrm{med}(x[k-w], \ldots, x[k+w])$. This operation is a form of high-pass filtering and will be called the median high-pass filtering. Naturally, the filters can be applied to finite length sequences too, by padding with zeros or mirroring.

Finally, note that the data vector $\mathbf{d}$ is obtained by stitching several different sequences of data and we need to carry out operations separately on these sequences or on parts of them. In this case, we simply say that a sequence is 'extracted' from the vector $\mathbf{d}$ and do not introduce a formal description of this operation, in order to keep the notation simple. In particular, we can extract the individual sequences from $\mathbf{d}$, filter each of them and stitch the output into a $D \times 1$ vector $\mathbf{h}$. When this is done using a median high-pass filter, we will denote the operation by $\mathbf{h} = \mathrm{high}(\mathbf{d})$.

3 *HERSCHEL* DATA

The photometers of PACS and SPIRE consist of two and three arrays of bolometers, respectively. The first PACS array operates in the 70 (blue) μm or 100 (green) μm band and is divided into eight subarrays. The second operates in the 160 (red) μm band and is divided into two subarrays. The SPIRE arrays operate in the 250, 350 and 500 μm bands. The field of view of the arrays is of the order of some square arcminutes, but a typical *Herschel* observation covers a much larger sky area, up to some square degrees wide. To observe the area, the telescope is moved along a set of parallel scan lines covering the area. During the scan the bolometers are sampled at frequency f_s, producing a sequence of readouts for each bolometer which is termed a *timeline*. Redundancy is guaranteed by observing the area at least twice, along two different scan directions, so that each bolometer normally produces two or more timelines. The timelines, together with the corresponding pointing information, constitute the observation output and are collectively referred as the time ordered data (TOD).

In the observation process, the true sky is convolved with the telescope PSF and projected on to the focal plane. Each bolometer integrates the emission hitting a small square of the focal plane, called the detector's pixel. This is the emission coming from a corresponding sky area centred on the direction into which the bolometer is pointed and, ideally, each readout gives the emission received at the sampling time. In practice, there are a number of deviations from this assumption, the most relevant of which are reviewed in the next subsection.

3.1 Noise and disturbances

Each timeline is affected by a thermal and electronic noise which can be modelled as a stationary random sequence having a power spectrum given by (Griffin et al. 2010; Poglitsch et al. 2010)

$$P_n(f) = \left[\left(\frac{f_0}{f}\right)^{\alpha} + 1\right] N_0 \qquad \text{for } |f| < \frac{f_s}{2}. \quad (8)$$

The spectrum can be regarded as the sum of two parts, namely a white part with flat spectrum $P_w(f) = N_0$ plus an $1/f$ part with spectrum $P_c(f) = (f_0/f)^{\alpha}N_0$, where f_0 is called the knee frequency and α the frequency exponent. Since the spectrum is not flat, the noise is temporally correlated, in the sense that there is statistical dependency between successive samples. A second observable fact, e.g. Traficante et al. (2011), is that the noise is spatially correlated too, in the sense that there is statistical dependency between the noise sequences of adjacent bolometers.

The spatial correlation complicates the implementation of the GLS synthesis, as we will see in Section 6.1. In order to overcome this problem, we note that the noise can be considered as the sum of two components. The first component, that will be called the electronic *noise*, is a fast varying, small amplitude signal, responsible for the white part of the spectrum and for a fraction of the $1/f$ part. The second component, that will be called the *drift*, is a slowly varying, large amplitude signal, with a spectrum concentrated at low frequencies. An example of timeline affected by drift is shown in Fig. 1(A). Moreover, we assume that the electronic noise is spatially uncorrelated while the drift is spatially correlated. This decomposition is justified by the physics of the instrument: loosely speaking, the electronic noise can be attributed to the readout electronics of each bolometer, while the drift can be attributed to slow temperature variations of the focal plane[4] and to the common electronics. However, it should be noted that the border is not sharp and the separation into these two components is somewhat arbitrary.

Based on the decomposition just introduced, we now specify the models that will be used in the paper. In particular, we consider the electronic noise as a zero mean, stationary, Gaussian random sequence with a power spectrum given by equation (8). Most important, we assume that the electronic noise is spatially

[4] For SPIRE, this is confirmed by the fact that a large part of the drift can be compensated using the thermometers output. These sensors are lacking in PACS, where the case is less clear and the drift a worse problem.

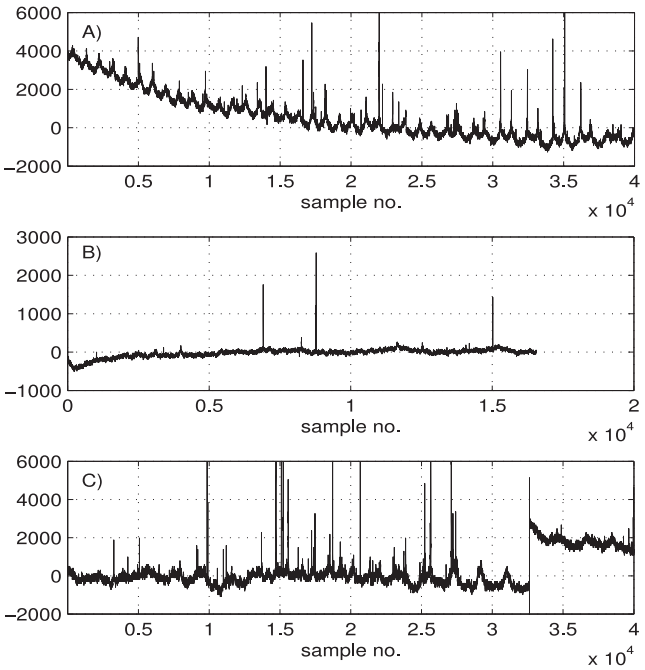

Figure 1. Plot (A) reports a PACS blue timeline with a strong drift in the first half. Plot (B) reports a PACS red timeline where an initial onset and three glitches are visible. Plot (C) reports a PACS blue timeline with a jump.

uncorrelated. For the drift, since it is a slowly varying signal, we abandon the stochastic model and consider it as a parametric, deterministic signal. In particular, a low-order polynomial is normally an adequate representation (Traficante et al. 2011). Specifically, we will consider two drift models. In the first, we assume that there is a common polynomial drift, equal for all the timelines of an array (SPIRE) or subarray (PACS). In the second, we assume that there is a specific polynomial drift, different for each timeline.

Besides noise and drift, there are other disturbances affecting the timelines. The first one is an *offset*, i.e. an unknown additive term, due to the fact that the bolometers are differential detectors. Moreover, the PACS instrument starts the observation by a calibration sequence consisting in the observation of two blackbodies, which may cause a deviation from the baseline of the initial part of the timeline. This effect will be called an *onset* and can be roughly modelled as a decaying exponential. An example is shown in Fig. 1(B). The timelines are also affected by impulsive disturbances, due to cosmic rays. Specifically, when a ray hits the instrument, it may provoke a *glitch* or a *jump*, depending on where it hits. A glitch is a high spike in the timeline, lasting one or two samples, examples of which can be seen in Fig. 1(B). A jump is an abrupt and long lasting deviation from the baseline, shown in Fig. 1(C). When the ray hits the readout electronics, the jump may affect a whole row of the array and be seen in all the corresponding timelines. Another problem is the *saturation* which occurs when a bolometer is pointed towards a very bright source, which may affect several consecutive samples.

There are several additional effects that should be accounted for in an accurate instrument model, including the quantization noise, the bolometer impulse response, the interference from the solar panels, the relative pointing error (RPE) affecting the pointing information and, for PACS, the onboard compression. While these effects do degrade the image, they have less impact and are not considered in this paper.

3.2 Data models

We now develop a model for the *Herschel* data. In the model, we do not account for impulsive disturbances because the readouts affected can be detected and discarded. This can be done with a negligible impact on the performance as long as the percentage of excluded readouts is low.

Initially, we neglect the drift and cast the *Herschel* data into the model of equation (1). To this end, we need to specify the data vector $\mathbf{d}$, the noise vector $\mathbf{n}$ and the signal $\mathbf{s}$. Assuming that the TOD is composed of N_t timelines and that each timeline is composed of N_r readouts,[5] the data vector $\mathbf{d}$ is obtained by stitching the timelines and is a $D \times 1$ vector, where $D = N_t N_r$ is the total number of readouts. The noise vector $\mathbf{n}$ represents the electronic noise and is specified by the covariance matrix, which is dictated by the spectrum and will be discussed in Section 6.1. Concerning the signal $\mathbf{s} = P\mathbf{m}$, we assume that the observed sky is pixellized, i.e. that it is partitioned into a grid of M non-overlapping squares (sky pixels) where the flux is constant, and is represented by the $M \times 1$ vector $\mathbf{m}$. Next, we assume that the signal component of each readout is equal to the value of the sky pixel where the bolometer was pointed at the sampling time. Then the matrix P, which is termed the *pointing matrix*, is a $D \times M$, sparse, binary matrix, constructed from the pointing information and such that $P_{k,i}$ is 1 if the kth readout falls into the ith sky pixel and is 0 otherwise. In other words, we use a nearest-neighbour rule, where each readout is entirely attributed to the closest sky pixel.

Clearly, the model is an approximation. In fact, we do not map the detector's pixel on to the sky pixels neither we account in any way for the telescope PSF, implying that our image estimate will be blurred by the PSF. We could solve these problems by using an appropriate pointing matrix (Cantalupo et al. 2010), but at the price of a sharp increase in the computational complexity, which is undesirable for *Herschel* data. On the other hand, the choice of a sparse and binary pointing matrix greatly simplifies the processing. Indeed the matrix can be efficiently stored. Moreover, the LS estimate of equation (2), which is also called the *simple projection* or the *naive map*, can be computed with D sums and M divisions, since the value of each pixel is just equal to the average of the readouts falling into the pixel. Finally, the production of the signal estimate of equation (3), namely $\bar{\mathbf{s}} = P\bar{\mathbf{m}}$, termed the *back-projection* of $\bar{\mathbf{m}}$, only requires D memory access operations.

A second approximation implicit in the model is that the sky is continuous in nature and not pixelized. However, the approximation is good when the sky pixel size is smaller than the PSF size. Naturally, the sky pixel also has to be large enough that several readouts fall into each pixel, so that $D \gg M$ and the redundancy needed for the map making is guaranteed. Both conditions can be satisfied for *Herschel* data.

A third problem with equation (1) is that the timelines are affected by the drift too, in addition to the electronic noise. In order to account for the drift, we can use the model of equation (5), which requires to specify the drift vector $\mathbf{y} = X\mathbf{a}$. To fix the ideas, suppose that there is a single timeline. Assuming that the drift is a polynomial of order N_a, specified by $K = N_a + 1$ coefficients, the drift vector can be written as $\mathbf{y} = X\mathbf{a}$, where $\mathbf{a}$ is a $K \times 1$ coefficient's vector

[5] The extension to the case when each timeline has a specific number of readouts is immediate.

and $X = \tilde{X}$, where $\tilde{X}$ is a $N_r \times (N_a + 1)$ Vandermonde matrix of the form

$$\tilde{X} = \begin{pmatrix} 1 & 1 & 1^2 & 1^3 & \dots & 1^{N_a} \\ 1 & 2 & 2^2 & 2^3 & \dots & 2^{N_a} \\ 1 & 3 & 3^2 & 3^3 & \dots & 3^{N_a} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & N_r & N_r^2 & N_r^3 & \dots & N_r^{N_a} \end{pmatrix}.$$

When there are many timelines we have two options. One is to use the specific drift model and consider a polynomial for each timeline. In this case, there are $K = N_t(N_a + 1)$ coefficients, the vector $\mathbf{a}$ is obtained by stitching all the coefficients and the drift matrix is block diagonal with N_t blocks, each block being equal to $\tilde{X}$ (Piazzo et al. 2015). A second option is to use the common drift model. In this case, assuming a single observation, for SPIRE we consider a single polynomial for the whole array, so that $K = N_a + 1$ and the drift matrix is obtained by stitching N_t copies of $\tilde{X}$ (Piazzo 2013). For PACS blue and red, we consider eight and two polynomials, respectively, i.e. one for each subarray. The corresponding drift matrix is easy to obtain and can be found in Piazzo (2013). Also the extension to the case of multiple observations is not difficult.

4 OVERVIEW AND PRE-PROCESSING

We now proceed to describe the UNIMAP processing, which is conveniently broken into four logical steps, namely pre-processing, drift removal, GLS synthesis and post-processing. The pre-processing is described in this section while the other steps are presented in Sections 5, 6 and 7, respectively.

4.1 Data formatting and offset compensation

The input is a set of *Herschel* observations[6] containing readouts organized into timelines and the corresponding pointing information. For each readout also an *input flag* must be there, which is a binary value indicating whether the readout is valid or has to be excluded from the synthesis. It is assumed that all the readouts affected by disturbances not treated by UNIMAP, like saturation, are flagged.

The first processing step is to load the input and build the TOD $\mathbf{d}$. In this step, the initial samples of each timeline may be discarded, which is a brute force approach to remove the onset. Next, given a user-specified pixellization of the observed sky area, each readout is assigned to a pixel and a $D \times 1$ vector $\mathbf{p}$ is produced, such that p_k is the index of the pixel where the kth readout falls. This vector is called the time ordered pixels and is an efficient way to store the pointing matrix. Finally, a rough offset compensation is performed, by subtracting to each timeline its median, thereby producing an updated TOD that, in order to keep the notation simple, is still denoted by $\mathbf{d}$.

At this stage, for evaluation purposes, two images are produced: the naive map obtained by projecting the TOD and a noise map, giving the corresponding standard deviation. That is, for each pixel, if $\mathbf{x}$ is the vector of the readouts belonging to the pixel, the naive map gives mea($\mathbf{x}$) and the noise map $\sqrt{\text{var}(\mathbf{x})}$. Since in the following, we will encounter several other naive maps, we number these maps and refer to the current one as the Naive 0 (N0) map. An example is given in Fig. 2. As can be seen, at this stage the image is of low quality and the disturbances affecting the data are clearly visible.

[6] Typically, but not necessarily, the input is obtained from HIPE using a tool called UniHIPE (Calzoletti & Faustini 2013), which extracts the so-called level 1 products.

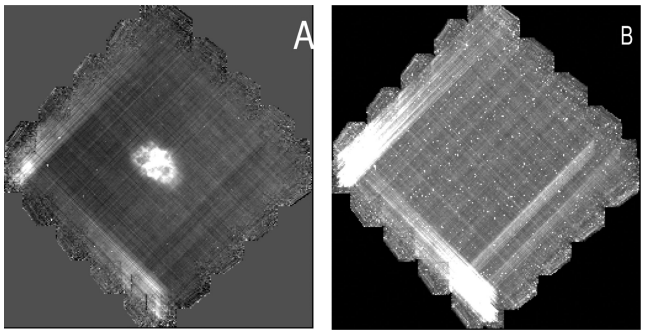

Figure 2. Images of the Crab nebula in the red band. The image is obtained by two, orthogonal scans. Plot (A) shows the N0 map and plot (B) shows the corresponding noise map. Several disturbances are clearly seen. The onset at the beginning of each scan is visible in the left-hand and lower corners. A jump involving an entire array row, starting near the right-hand corner, is seen in the noise map and, to a lesser extent, also in the naive map. Several glitches can be seen in the naive map as bright, isolated pixels and even more can be seen in the noise map. The stripes following the scan lines are due to the $1/f$ noise. The drift is almost absent.

4.2 Jumps and onset

The jumps are detected using a procedure described in Appendix A1. After the detection, a set of *jump flags* is produced, by flagging 100 readouts starting from the position of any detected jump. Typically, the percentage of flagged readouts is less than 0.5 per cent for PACS, while the jumps are almost absent in SPIRE data. Next, the input and the jump flags are merged and fixed. Specifically, the vector $\mathbf{d}$ is updated by replacing the flagged readouts with a linear interpolation between the preceding and succeeding valid readouts. This is a rough approximation, useful to regularize the timelines during the pre-processing. In a later stage the flagged data will be removed.

The next step is the onset removal, which is controlled by a parameter λ, specifying the length of the onset. An exponential curve is fitted to the first λ samples of each timeline and the TOD $\mathbf{d}$ is updated by subtracting the fit from the original timeline. The onset removal has to be used with caution since the exponential model is not always adequate.

At this stage, for evaluation, a flag mask, i.e. an image where each pixel gives the number of flagged (input or jump) readouts falling into it, is produced, together with a Naive 1 (N1) map, obtained by projecting the updated TOD, and the corresponding noise map. Examples are shown in Figs 3(A) and 4.

4.3 Glitches

The glitches are detected by identifying the outliers, as described in Appendix A2, and a *glitch flag* is set for every readout affected. After the detection, the TOD $\mathbf{d}$ is updated by linearly interpolating the flagged readouts. A sane glitch rate is no more than 0.5 per cent for both PACS and SPIRE data. A known problem is that the procedure tends to overflag pixels containing point sources or strong signal

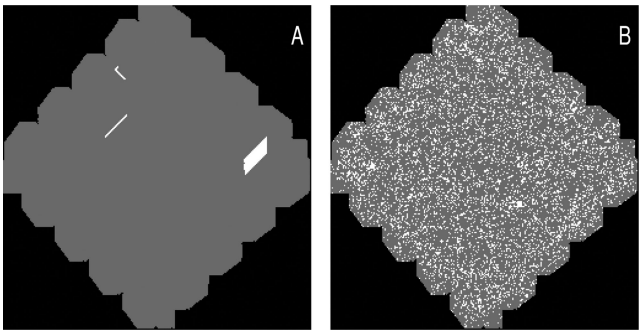

Figure 3. Plot (A) shows the jump and input flag mask for the Crab nebula. The jumps can be recognized as long sequences of flags. We see that the jump cluster observable in the right-hand corner of Fig. 2 is detected and flagged. Moreover, two isolated jumps are detected. There are no input flags in this case, because there are no saturated pixels. Plot (B) shows the glitch flag mask. Note the glitch clusters, indicating overglitching. Checking the values we verified that the overglitching is moderate and can be ignored.

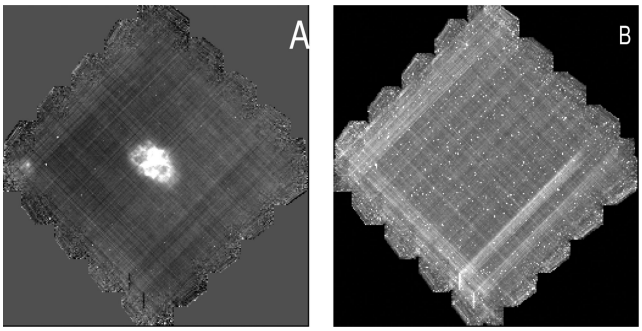

Figure 4. Images of the Crab nebula in the red band after the jumps and onset removal. Plot (A) shows the N1 map and plot (B) the corresponding noise map. By comparing with Fig. 2 we see that the onset has been substantially reduced. Moreover, the initial part of the jumps has been interpolated but the jumps are longer and are still clearly visible. This is not a problem because the jumps have been detected and will be removed by the GLS image synthesis.

gradients,[7] since in these pixels the assumption of a constant signal is less valid.

For evaluation, a glitch flag mask is produced, together with a Naive 2 (N2) map, obtained by projecting the updated TOD, and the corresponding noise map. Examples are shown in Figs 3(B) and 5.

5 REDUCTION STRATEGY AND DRIFT REMOVAL

At this stage, since the impulsive disturbances have been removed, the data are affected by noise plus drift and equation (5) applies. Then we could compute the optimal solution, namely the JGLS map of equation (7). However, the computation of this map is not easy. Besides the fact that the programming is more involved, preliminary tests indicate that the convergence is extremely slow and that there are numerical stability issues. A more practical approach, followed by UNIMAP, is to break the image estimation in two steps. In the first step, called the drift removal and described in this section, the JLS

[7] This problem can be mitigated by correctly mapping the detectors' pixels on to the sky pixels.

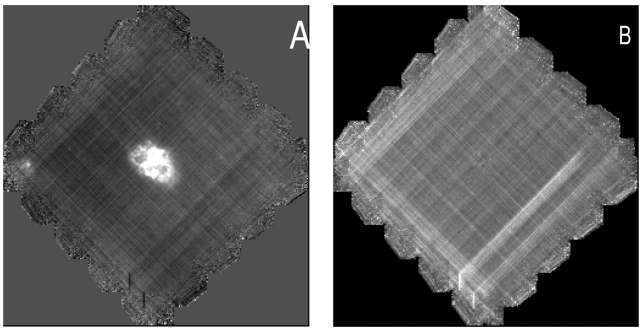

Figure 5. Images of the Crab nebula after the glitch detection. Plot (A) shows the N2 map and plot (B) the corresponding noise map. By comparing with Fig. 4 we see that the bright pixels due to the glitches are no more there. This is better seen in the noise map.

approach is used in order to produce an estimate of the drift, denoted by $\bar{\mathbf{y}}$ and given by $\bar{\mathbf{y}} = X\bar{\mathbf{a}}$, where $\bar{\mathbf{a}}$ is the coefficient's subvector extracted from $\bar{\mathbf{z}}$ of equation (6). The estimate is then subtracted from the data, to obtain an updated TOD given by $\tilde{\mathbf{d}} = \mathbf{d} - \bar{\mathbf{y}}$, which is, ideally, drift free. In the second step, described in the next section, the image is estimated from the updated TOD using GLS.

The two steps approach replaces the JGLS estimation with JLS and GLS, thereby achieving a reduction in the computational and implementation complexity. However, the approach is suboptimal, in the sense that it will not produce exactly the JGLS map. Nevertheless, since the drift depends on a few parameters (the polynomial coefficients), the JLS noise rejection is good and the drift estimate is accurate. In turn, since the drift is well removed, the GLS approach yields a solution close to the JGLS one, making the two steps approach nearly optimal. This fact has been verified by means of simulations (Piazzo et al. 2015).

In practice, the updated TOD $\tilde{\mathbf{d}}$ can be computed using an efficient iterative algorithm,[8] based on ALS, consisting of the following steps.

Set $\tilde{\mathbf{d}} = \mathbf{d}$ and repeat 1–6 until convergence.

1. Make a naive map: $\bar{\mathbf{m}} = (P^{\mathrm{T}}P)^{-1}P^{\mathrm{T}}\tilde{\mathbf{d}}$.
2. Estimate signal: $\bar{\mathbf{s}} = P\bar{\mathbf{m}}$.
3. Remove signal: $\mathbf{w} = \tilde{\mathbf{d}} - \bar{\mathbf{s}}$.
4. Estimate coefficients: $\bar{\mathbf{a}} = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}\mathbf{w}$.
5. Estimate drift: $\bar{\mathbf{y}} = X\bar{\mathbf{a}}$.
6. Remove drift: $\tilde{\mathbf{d}} = \tilde{\mathbf{d}} - \bar{\mathbf{y}}$.

The algorithm performs a sequence of alternating signal and drift LS estimations. Indeed, steps 1 and 2 implement the signal estimation and amount at computing a naive map followed by a back projection, while steps 4 and 5 implement the drift estimation and amount at a polynomial fit. A key point is that the drift is estimated from the vector $\mathbf{w}$, from which the signal has been removed, preventing the signal to bias the drift estimate.

The ALS algorithm is analysed in Piazzo et al. (2015), showing that it converges and produces the same updated TOD that would be obtained using the JLS. In particular, the updated TOD is given by

$$\tilde{\mathbf{d}} = \mathbf{s} + \mathbf{c} + \tilde{\mathbf{n}}, \tag{9}$$

where $\mathbf{c}$ is a constant vector and $\tilde{\mathbf{n}}$ is a zero mean random vector. Comparing the latter expression with equation (5) we see that the

[8] Similar algorithms are used in TAMASIS and in the SPIRE standard pipeline.

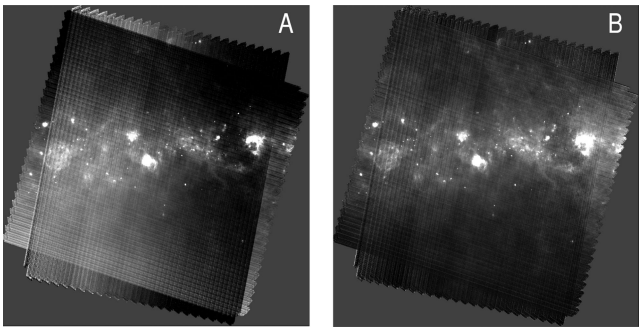

Figure 6. Image of a 2 × 2 degree sky area centred at a Galactic latitude of 4° (L004). Data are PACS blue. The area has been observed twice, with two orthogonal scans. Plot (A) shows the N2 map. Along the image borders, where the two observations do not overlap, there is a strong signal gradient which is due to the drift. Also a non-flat background can be observed. Plot (B) shows the N3 map, after that the common drift has been removed using ALS with a polynomial order of 3. It can be seen that the gradient on the image borders has been corrected and that the background is flat. The image is still affected by the $1/f$ noise, causing the stripes following the scan lines.

algorithm preserves the signal and removes the drift except for a constant. However, this is irrelevant, since $\mathbf{c}$ can be absorbed into the offset affecting the timelines. The convergence can be controlled by checking the mean square error (MSE), given by $|\mathbf{w}|^2/D$. In particular, the MSE decreases across the iterations and tends to a stable lower limit. Therefore, we stop the iterations when the MSE variation becomes negligible.

A point worth discussing is the selection of the polynomial order N_a. This can be performed empirically, by running the ALS with an increasing order and studying the MSE. In particular, the MSE decreases but, past a certain order, the improvement becomes negligible and further increasing N_a is useless. In practice, a third order polynomial is normally an adequate choice.

As a final comment note that, depending on the drift matrix, the ALS can be used to remove the specific or the common drift, see Section 3.2. The second option is safer, because the common drift depends on less parameters, which makes the estimation more reliable. However, the first option is more powerful, because the specific drift is a general model comprising the common drift as a special case, and may speed up the convergence of the iterative solver used to compute the GLS map. In practice, this is not a critical choice, because essentially the same GLS map is obtained using either drift model.

For evaluation, after the ALS run, a Naive 3 (N3) map, obtained by projecting the updated TOD, and the corresponding noise map are produced. An example of the results is discussed in Fig. 6.

6 GLS IMAGE SYNTHESIS

At this stage, by neglecting the constant and dropping the tilde in equation (9), we obtain equation (1). Then, we can use equation (4) to obtain the GLS estimate. However, the direct computation of $\bar{\mathbf{m}}$ is infeasible, due to the size of the matrices involved. Nevertheless, the estimate can be approximately computed, as described in this section.

We start by rewriting equation (4) as

$$B\bar{\mathbf{m}} = \mathbf{m}^*,$$

where $B = P^{\mathrm{T}}N^{-1}P$ and $\mathbf{m}* = P^{\mathrm{T}}N^{-1}\mathbf{d}$, and note that the latter is a linear system in the unknown vector $\bar{\mathbf{m}}$, which can be solved using a standard method. Typically, e.g. Patanchon et al. (2008), Cantalupo et al. (2010) and Traficante et al. (2011), the system is solved using the parallel conjugate gradient (PCG; Vetterling et al. 1992) which is an iterative solver that, to be implemented, only requires to compute the vector $\mathbf{m}^*$ and to perform a sequence of multiplications of the matrix B with a vector. Both these operations can be broken into submultiplications with the matrices P, P^{T} and N^{-1}. The multiplication by P or P^{T} is a projection and can be implemented efficiently since P is sparse. The multiplication by N^{-1} can be implemented by means of linear filtering. This result is well known and is normally obtained by approximating N as a circulant matrix and working in the frequency domain, e.g. Cantalupo et al. (2010). In the following we give an equivalent proof working in the time domain.

6.1 Multiplication by the inverse covariance matrix

We start by assuming that there is a single timeline. Then the vector $\mathbf{n}$ is a window of D samples of a noise sequence $n[k]$ for $k = -\infty, \ldots, \infty$. The noise has an autocorrelation sequence given by $R[k] = E\{n[i]n[i+k]\}$, which is the inverse Fourier transform (IFT) of the noise spectrum $P_n(f)$, denoted as $R[k] = \mathrm{IFT}\{P_n(f)\}$. An example is shown in Fig. 7. The generic element of the covariance matrix is $N_{i,j} = E\{n[i]n[j]\} = R[j-i]$, so that N is a Toeplitz matrix of the form

$$N = \begin{pmatrix} R[0] & R[1] & R[2] & \ldots \\ R[-1] & R[0] & R[1] & \ldots \\ R[-2] & R[-1] & R[0] & \ldots \\ \ldots & \ldots & \ldots & \ldots \end{pmatrix}.$$

Since $P_n(f) > 0$, we introduce $H(f) = 1/P_n(f)$ and the sequence $h[k] = \mathrm{IFT}\{H(f)\}$ which is called the *noise filter* response and has the important property that $R[k]*h[k] = \delta[k]$, where $\delta[k]$ is the unit pulse. Moreover, since $P_n(f)$ is real and symmetric, the same is true for $h[k]$, as seen in Fig. 7. Furthermore, in practice, the sequence $h[k]$ tends to zero and there is an index L such that $h[k] \approx 0$ for $|k| > L$.

Consider now the multiplication of the inverse of the covariance matrix, N^{-1}, by a generic $D \times 1$ vector $\mathbf{v}$, to produce $\mathbf{w} = N^{-1}\mathbf{v}$. In Appendix A3 we show that, when $L \ll D$, the vector $\mathbf{w}$ can be

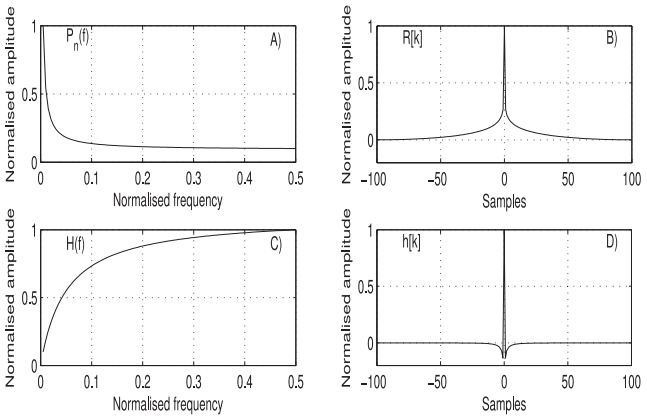

Figure 7. Plot (A) reports an example of noise spectrum $P_n(f)$ following the model of equation (8). Plot (B) the corresponding autocorrelation sequence $R[k]$. Plot (C) is the reciprocal of the spectrum $H(f)$ and plot (D) reports the noise filter response $h[k]$. Since $P_n(0) = \infty$, reflecting the fact that the timeline is affected by an unknown offset, we set $H(0) = 0$ and the correlation base to zero.

approximated as follows. First, from **v** we construct a sequence $x[k]$ by padding and mirroring the vector

$$\begin{aligned} x[k] &= v_{k+1} \quad \text{for } k = 0, \ldots, D-1 \\ x[k] &= v_{-k} \quad \text{for } k = -L, \ldots, -1 \\ x[k] &= v_{2D-k} \quad \text{for } k = D, \ldots, D+L-1 \\ x[k] &= 0 \quad \text{elsewhere.} \end{aligned} \tag{10}$$

Next, we filter the sequence $x[k]$ with the response $h[k]$, to produce $z[k] = x[k] * h[k]$. Finally, the vector **w** is obtained by extracting the first D samples from the sequence $z[k]$, i.e. $w_k = z[k-1]$ for $k = 1, \ldots, D$.

The extension to the case of $N_t > 1$ timelines is straightforward. Indeed, since the vector **n** represents the electronic noise, we have that the noise sequences of the timelines are uncorrelated. Then, the matrix N is block diagonal with N_t blocks and the reasoning can be repeated for each block. As a result, the multiplication $\mathbf{w} = N^{-1}\mathbf{v}$ can be approximated by breaking the vector **v** into timelines, separately filtering the N_t timelines and stitching the results into **w**. Moreover, now we see why the spatial correlation is so detrimental for the GLS implementation: indeed, in the presence of spatially correlated noise, the matrix N is not block diagonal and the multiplication by N^{-1} is much more involved.

6.2 Estimation of the noise filter response

The noise filter responses are estimated from the TOD. To this end, an estimate of the noise vector is produced, given by $\bar{\mathbf{n}} = \mathbf{d} - \bar{\mathbf{s}}$, where $\bar{\mathbf{s}} = P^{\mathrm{T}}(P^{\mathrm{T}}P)^{-1}P\mathbf{d}$ is the signal estimate. Next, the filter responses are computed, separately for each timeline. In particular, given a timeline, we compute only the non-zero part of $h[k]$, which is a $T = 2L + 1$ samples long sequence.[9] This sequence can be obtained as the inverse discrete Fourier transform of $F[i] = 1/S[i]$, where $S[i] = P_n(if_s/T)$ for $i = 0, \ldots, T-1$ is a sequence of T samples of the noise spectrum. In turn, the sequence $S[i]$ can be measured from the noise estimate $\bar{\mathbf{n}}$. This is done by extracting from $\bar{\mathbf{n}}$ the subvector corresponding to the timeline, by segmenting the subvector into blocks of T samples with an overlap of L, by computing the discrete Fourier transform (DFT) of each block and by averaging the squared magnitude of all[10] the DFTs. Note that, due to the $1/f$ noise, we have $P_n(0) = \infty$ and correspondingly we set $F[0] = 0$, implying that the filter response is zero mean. This guarantees that when the timelines are filtered to produce the map $\mathbf{m}^*$, any residual offset is eliminated.

One problem is that the signal may leak into $\bar{\mathbf{n}}$ and bias the spectrum measurement. A first way to mitigate this problem is to separate the measurement of the spectrum shape and amplitude. To this end, we normalize the DFT of each block before the averaging and produce a normalized response $\tilde{h}[k]$, enforcing $\tilde{h}[0] = 1$. Next, we estimate the noise power P as the median of the variances of the blocks. Finally, we compute the response as $h[k] = \tilde{h}[k]/P$. As a second improvement, we fit the measured spectrum $S[i]$ to the model of equation (8) and obtain a sequence $\tilde{S}[i]$ which follows the theoretical spectrum and is used to produce the filter response instead of the measured spectrum. The fit must be used with caution because the model is not always applicable.[11]

For evaluation, the measured noise spectra and the filter responses are saved. An example is discussed in Fig. 8.

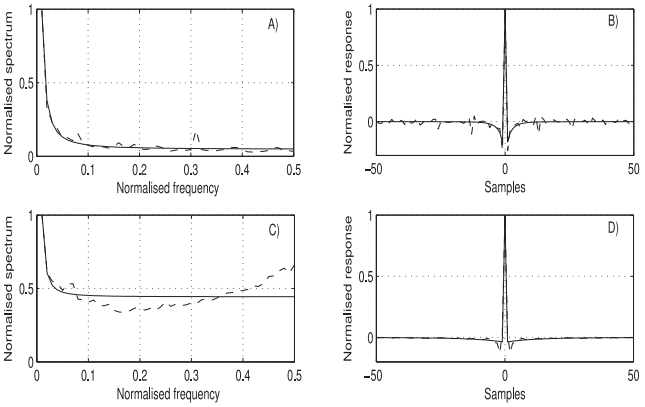

Figure 8. Examples of spectra and filter responses. Plot (A) shows the measured spectrum (dashed line) and the fit (continuous line) for a PACS blue timeline from an observation of the Galactic plane (L004). The spikes in the measurement are due to the periodic crossing of the orthogonal scan. The spectrum follows the theoretical model of equation (8) well and the fit is useful to combat the spikes. Plot (B) shows the corresponding filter response and we see that the fit produces a smoother response. Plot (C) shows the measured spectrum (dashed line) and the fit (continuous line) for a SPIRE 500 μm timeline. The spectrum rises at high frequencies (see footnote 11) and does not follow the model of equation (8) used in the fit. In this case, it is better to use the measurement directly. Plot (D) shows the corresponding filter response.

6.3 Flagged data removal

The flagged readouts have to be excluded from the image synthesis. In principle this is simple: to exclude a flagged readout it is sufficient to remove the entry from the vector **d**, the corresponding row from the matrix P and the corresponding row and column from the matrix N; then, the GLS map is still given by equation (4). However, this is not easy to implement when the multiplication by N^{-1} is realized as a filtering. Therefore, an approximate approach is used, which is described in the following.

As a first step, the flagged readouts (input, jump and glitch) are removed from the TOD **d**, shifting the other elements downwards to fill the gaps. In this way an updated, shorter TOD is produced, where the time continuity of the noise is not preserved. However, this is not a serious problem as long as the flagged readouts are isolated samples. On the contrary, when a sequence is removed, the problem is exacerbated. For this reason, when a sequence of samples is removed, the corresponding timeline is broken into two segments, which are treated from now on as independent timelines, albeit with the same noise filter. In particular, this guarantees that the timelines are broken in correspondence to all the detected jumps. This is important, because in this way the jump translates into two different offsets for the two segments, which are eliminated by the zero mean noise filter.

[9] The filter length L is a parameter that can be set by means of a trial and error procedure. For example, to verify that the length is adequate, one can check that doubling it does not change the GLS map appreciably.

[10] Blocks containing flagged data are skipped.

[11] For example, the SPIRE noise does not follow the model, because the standard pipeline performs a high-pass filtering prior to level 1.

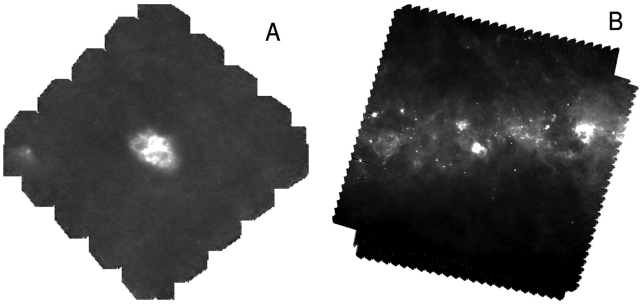

Figure 9. Plot (A): GLS image of the Crab nebula (red band). Plot (B): GLS image of L004 (blue band). Comparing with Figs 5 and 6 we see that the image quality is improved and the stripes due to the $1/f$ noise are absent.

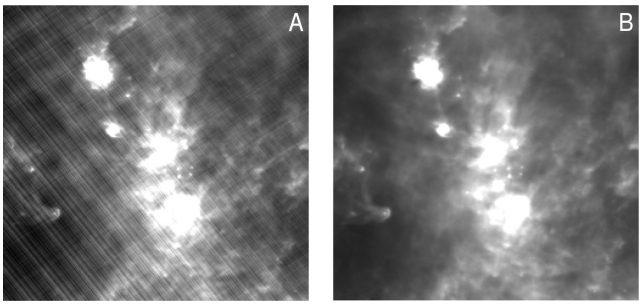

Figure 10. Naive ($\mathbf{m}_n$, plot A) and GLS ($\mathbf{m}_g$, plot B) maps of a patch of the Rosette nebula in the PACS red band. Stripes due to the $1/f$ noise are clearly visible in the naive map but do not affect the GLS map.

6.4 GLS image synthesis and distortion

The GLS image is produced by running the PCG, which is an iterative solver and needs an initial guess to start with. We can choose between the naive map and an all zero map. In principle, the PCG will produce the same solution, irrespective of the start image. However, the iterations required do depend on the initial guess. Moreover, the convergence may be difficult to achieve if the start image is not properly selected. Typically, when the image is signal rich it is better to start from the naive map while when the image is dominated by the background it is better to start from the zero map.

The PCG run produces the GLS map, which, from now on, will be denoted by $\mathbf{m}_g$, and several by products, namely a final[12] naive map $\mathbf{m}_n$, the corresponding noise map $\mathbf{m}_\sigma$ and a coverage image $\mathbf{m}_c$, giving the number of readouts falling into each pixel. Examples are given in Figs 9 and 10 and show that the GLS approach is effective against the $1/f$ noise. Unfortunately, the approach also has a serious drawback, namely it may introduce distortion, i.e. a signal dependent error. An example is given in Fig. 11 where the distortion is evident and takes the form of a cross-like artefact placed on a bright source. However, as we will soon see, the distortion can take other forms and is not always so obvious. The distortion is due to the approximations of the signal model and to the uncompensated disturbances, e.g. the RPE (Piazzo et al. 2012). The distortion is not present in the naive map, which is instead affected by the $1/f$ noise.

In order to evaluate the presence of distortion, a useful tool is the difference of the GLS and the naive maps, denoted by $\mathbf{e}_g = \mathbf{m}_g - \mathbf{m}_n$. In fact, assuming that $\mathbf{m}_g$ is constituted by the signal plus distortion

[12] The difference with the N3 map is that the flagged data are not used in the final naive map.

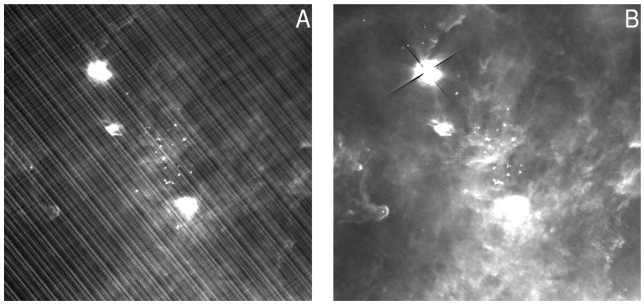

Figure 11. Naive ($\mathbf{m}_n$, plot A) and GLS ($\mathbf{m}_g$, plot B) maps of a patch of the Rosette nebula in the PACS blue band. Note the cross-like artefact affecting the GLS map.

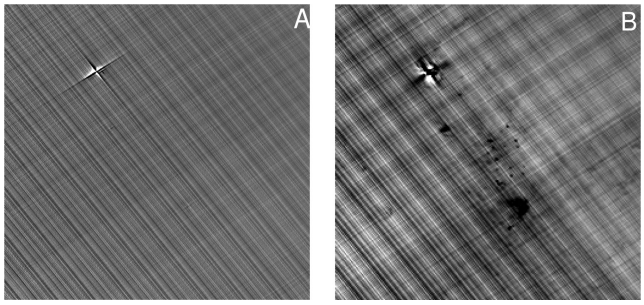

Figure 12. Difference of GLS and naive maps, $\mathbf{e}_g$, for the Rosette patch in the blue band (plot A) and red band (plot B). The $1/f$ noise is seen as the stripes following the scan lines. Comparing with Fig. 11 we see that the cross-like artefact present in the GLS map is clearly visible. Signal dependent distortion is also there in the red band but it is diffuse and not cross-like.

and $\mathbf{m}_n$ by the signal plus $1/f$ noise, by taking the difference we remove the signal and expect $\mathbf{e}_g$ to be constituted by the $1/f$ noise (with negative polarity) plus the distortion. This is indeed seen in Fig. 12, where the distortion is clearly visible, embedded in the $1/f$ noise. Moreover, the figure shows that in the red band there is a diffuse distortion, which is difficult to spot in the GLS map.

By inspection of hundreds of images we verified that the distortion is a serious problem, more pronounced for PACS but not uncommon in SPIRE. Cross-like distortion is due to bright, narrow sources and typically affects PACS blue. Diffuse distortion is due to broad sources or to diffuse emission and is normally found in PACS red or SPIRE. We also verified that the sky pixel has an impact on the distortion and that, specifically, a larger size exacerbates the problem.

7 GLS DISTORTION REMOVAL

The distortion introduced by the GLS synthesis is removed by means of the PGLS algorithm (Piazzo et al. 2012), which produces the PGLS map $\mathbf{m}_p$ starting from the GLS map $\mathbf{m}_g$ and the TOD $\mathbf{d}$. The algorithm steps are reported below and are followed by an explanation of the functioning.

Set $\mathbf{m}_p = \mathbf{m}_g$ and repeat 1–5 until convergence.

1. Estimate signal plus distortion: $\mathbf{t} = P\mathbf{m}_p$.
2. Remove signal (and add noise): $\mathbf{r} = \mathbf{t} - \mathbf{d}$.
3. Remove $1/f$ noise: $\mathbf{w} = \text{high}(\mathbf{r})$.
4. Average white noise and estimate distortion: $\mathbf{m}_d = (P^T P)^{-1} P^T \mathbf{w}$.
5. Subtract distortion from the map: $\mathbf{m}_p = \mathbf{m}_p - \mathbf{m}_d$.

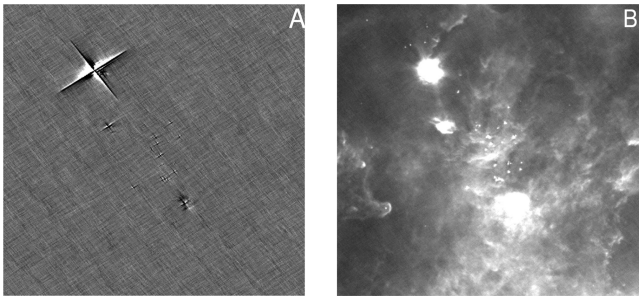

Figure 13. Estimated distortion (**e**, plot A) and PGLS map ($\mathbf{m}_p$, plot B) for the Rosette patch in the blue band. By comparing with Fig. 12 we see that the distortion is well estimated and by comparing with Fig. 11 we see the image improvement achieved.

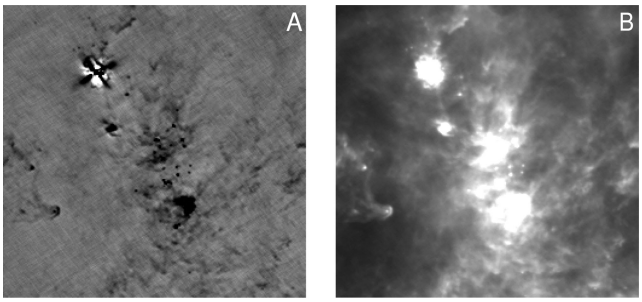

Figure 14. Estimated distortion (**e**, plot A) and PGLS map ($\mathbf{m}_p$, plot B) for the Rosette patch in the red band. By comparing with Fig. 12 we see that also diffuse distortion is well estimated by the PGLS.

Assuming that the GLS map $\mathbf{m}_g$ contains signal plus distortion, the same is true for the back-projected TOD **t**. Since the original TOD **d** contains signal, $1/f$ noise and white noise, by subtracting it from **t** in step 2 we produce a TOD **r** which contains the noise (with changed polarity) and the distortion. The $1/f$ noise is removed by the median high-pass filtering of step 3 and the white noise is averaged out by the naive projection of step 4. As a result, the map $\mathbf{m}_d$ is an estimate of the distortion, which is removed by subtraction, in step 5. The removal is improved by repeating the process iteratively. Convergence is achieved when the map $\mathbf{m}_p$ does not change significantly across two iterations. At convergence, the total distortion can be computed as $\mathbf{e} = \mathbf{m}_g - \mathbf{m}_p$.

The PGLS was analysed in Piazzo et al. (2012), showing that it effectively removes the distortion, provided that there is enough redundancy in the data. Examples are given in Figs 13 and 14. The only drawback is an increase in the background noise which is due to the fact that the distortion estimate $\mathbf{m}_d$ is noisy and this noise is injected into the PGLS map when $\mathbf{m}_d$ is subtracted from $\mathbf{m}_p$ in step 5. However, when the signal is strong the noise increase is negligible. The PGLS has a single parameter, namely the window length of the median high-pass filter, which can be set using a trial and error procedure. In particular, wider filters improve the distortion removal but also inject more noise. A useful tool is the difference of the naive and PGLS maps, denoted by $\mathbf{e}_p = \mathbf{m}_p - \mathbf{m}_n$, which can be used to evaluate the results and to set the window of the median filter. Examples are shown in Fig. 15.

The last processing step aims at minimizing the PGLS noise injection. Observing Figs 13 and 14 we see that the distortion only affects a fraction of the pixels. Then, the noise injection can be reduced by subtracting the distortion only in these pixels. To this end, we produce a mask $\mathbf{m}_e$ which is one in the pixels affected by distortion and zero elsewhere, as described in Appendix A4. Next, we compute a WGLS map given by $\mathbf{m}_w = \mathbf{m}_g - \mathbf{e} \odot \mathbf{m}_e$, where $\odot$ denotes the element-wise multiplication of two vectors. This is the final output map. An example is given in Fig. 16.

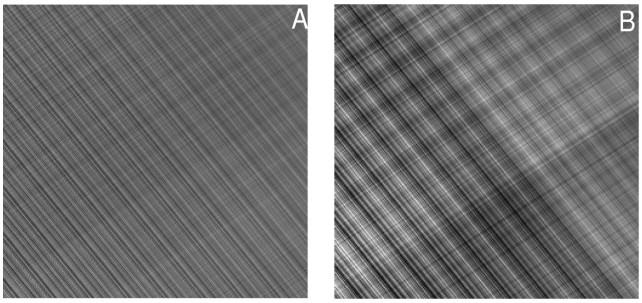

Figure 15. Difference of PGLS and naive map, $\mathbf{e}_p$, for the Rosette patch. Plot (A): blue band. Plot (B): red band. Note that the difference maps only contain $1/f$ noise, indicating that the PGLS filter length is adequate and that distortion has been removed.

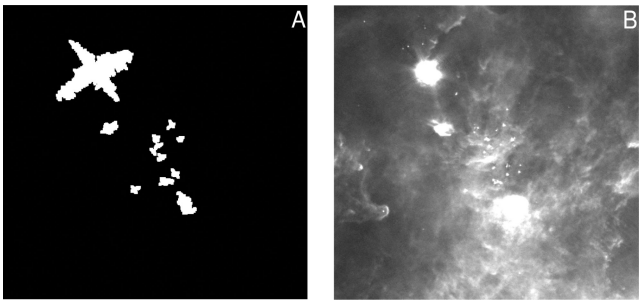

Figure 16. The WGLS mask ($\mathbf{m}_e$, plot A) and the corresponding WGLS map ($\mathbf{m}_w$, plot B) for the Rosette patch in the blue band. By comparing with Fig. 13 we see that the WGLS mask correctly identifies pixels where appreciable distortion is present.

As a comment, note that the noise affecting the WGLS map has different statistics in the pixels included in the mask, where it is the PGLS noise, and in those excluded from the mask, where it is the GLS noise. The latter fact may be annoying for some applications. In these cases, it may be preferable to use the PGLS or the GLS maps directly.

8 PERFORMANCE EVALUATION

To evaluate the performance, we developed a simple simulator of the PACS instrument. The simulator is based on a synthetic but realistic sky image, which is used as the true map. In order to account for the continuous nature of the sky, the true map has a higher resolution than the map to be formed, i.e. a smaller pixel size. The pointing information is obtained from a true *Herschel* observation. In order to produce the simulated data vector, we perturb the pointing information with an RPE and sample the synthetic sky, to produce a vector **x**. In particular, we shift each sampling point along the scan direction, by a fraction (one fifth) of the sampling spacing, which is a form of systematic RPE encountered in practice in PACS data. Next, the signal vector **s** is obtained from **x** by averaging every second sample, which mimics the data compression scheme exploited by PACS (co-addition). Then, a realistic example of the *Herschel* disturbances is obtained by picking a true TOD,

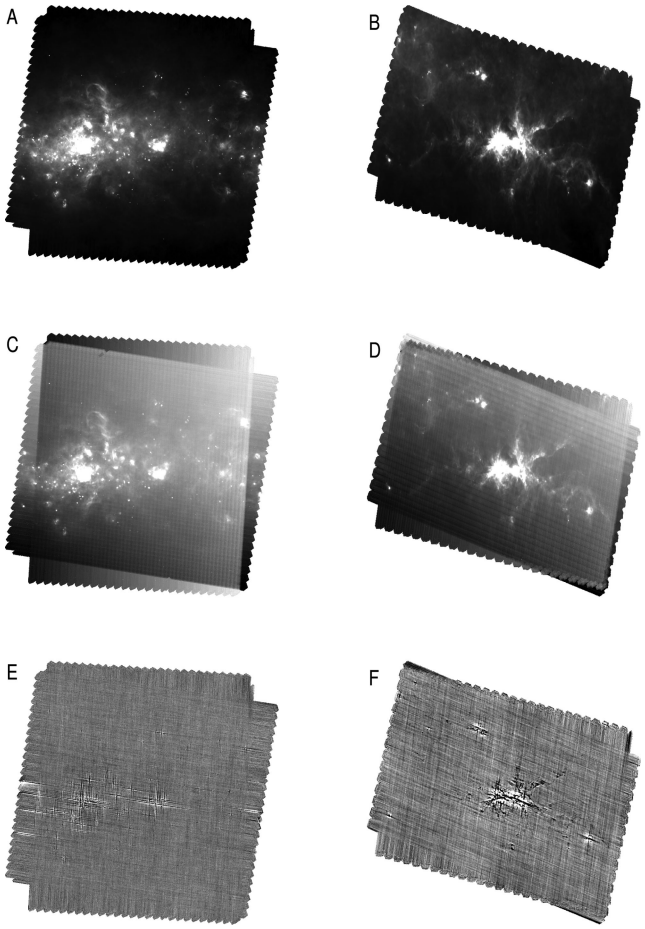

Figure 17. Examples of simulated data. Top row displays the true maps for the blue (plot A) and red (plot B) bands. Middle row (plots C and D) displays the N0 maps. Bottom row (plots E and F) displays the PGLS estimated distortion.

denoted by **w**, produced by observing a sky area free of emission.[13] Finally, the simulated TOD is constructed by adding the signal and the disturbances, as $\mathbf{d} = \mathbf{s} + g\mathbf{w}$, where g is a gain factor that is used to set the signal-to-disturbance ratio (SDR), defined as $\mathrm{SDR} = \mathrm{var}(\mathbf{s})/[g^2\mathrm{var}(\mathbf{w})]$.

The simulator is clearly an approximation of the real instrument, the development of an accurate one being beyond the scope of the paper. Most notably, the simulator does not account for the PSF and the detector pixel and only implements a simple form of RPE. Nevertheless, the results are realistic and mimic several of the effects that can be observed in true data. Indeed, we set up two simulations, one in the blue band and one in the red, starting from the true maps shown in Figs 17(A) and (B), which have a pixel size of 3.2 and 6.4 arcsec, respectively. To give an idea of the reduction results, plots C and D show the N0 maps, with a pixel size of 12.8 arcsec, when the input is simulated data with an SDR of 0 dB. We note that both the drift and the $1/f$ noise are clearly seen in the N0 maps, confirming that the simulated data are realistic. Moreover, plots (E) and (F) show the PGLS distortion estimates and we see that the distortion is cross-like in the blue band while it is diffuse in the red band, indicating that both these types of distortion can be simulated.

[13] We used a sky area at 14° right ascension and 0° declination, called the Atlas field. Clearly, there are some extragalactic sources which, however, are below the noise level.

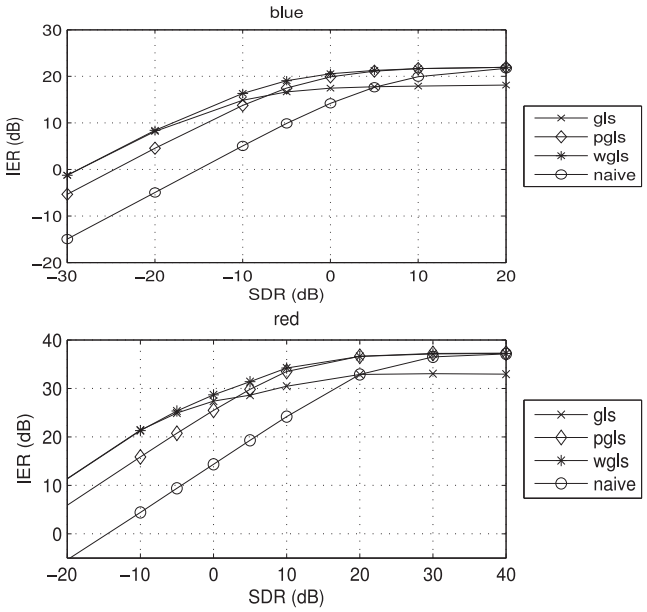

Figure 18. IER (dB) of the various UNIMAP output maps as a function of the SDR (dB) for the blue (top) and red (bottom) data.

In order to evaluate the quality, we produce a target map $\mathbf{m}_\mathrm{t}$ with a pixel size of 12.8 arcsec by averaging the true map over 16 (blue) or 4 pixels (red). Then, to evaluate, say, the GLS map $\mathbf{m}_\mathrm{g}$, we compute the image-to-error ratio (IER) that is the ratio of the target image variance to the error variance, i.e. $\mathrm{IER} = \mathrm{var}(\mathbf{m}_\mathrm{t})/\mathrm{var}(\mathbf{m}_\mathrm{g} - \mathbf{m}_\mathrm{t})$. The IER is computed similarly for the other maps. In practice, we compute the IER excluding the pixels at the borders. This is because at the borders the image is more noisy due to the shallower coverage. Moreover, we exclude the strong point sources, because the simulation set-up is not sophisticated enough to give reliable results for these objects.[14] Indeed, point sources are mainly affected by the PSF, the co-addition and the RPE, three effects that are not accounted for or roughly modelled in the simulator.

The IER is plotted in Fig. 18 as a function of the SDR, for both the blue and red bands. Observing the figure we see that, at low SDR, the GLS yields the highest IER, indicating that the distortion is negligible. The PGLS is inferior, indicating that it introduces background noise. However, the WGLS map attains the same IER of GLS. The naive map is plagued by the $1/f$ noise and is the worst one. When the SDR is increased, all the maps get closer to the target. However, the GLS performance saturates at some point, due to the distortion. At this point, the PGLS yields a higher IER, indicating that it successfully removes the distortion. Moreover, the WGLS attains a further IER increase and yields the best map. At high SDR the WGLS and PGLS performance is the same, indicating that the noise injected by the PGLS is negligible. At very high SDR, even the naive map becomes a good choice. We conclude that the WGLS map is the best one across the whole range of SDR, but at high or low SDR other maps can have a comparable quality.

[14] Several tests involving sources can be found in Paladini et al. (2013) and Xu et al. (2013).

9 CONCLUSION

We discussed the GLS map making for PACS and SPIRE data, noting that this application poses several challenges. One is the presence of spatial correlation in the data. To tackle this problem, we introduced a data model, namely equation (5), where the spatial correlation affecting the data is represented as a deterministic, parametric drift. Based on that model, an expression for the optimal map, namely the JGLS map of equation (7), can be developed. However, the direct computation of the JGLS map is difficult. Therefore, in Section 5, we introduced a simpler, two steps reduction strategy, producing nearly optimal results, where in the first step the drift is removed and in the second step the image is synthesized using GLS. We also developed an efficient implementation of the drift removal, using the ALS algorithm.

As a second contribution we discussed the distortion affecting the GLS maps and presented a post-processing, constituted by the PGLS and WGLS algorithms, which is capable of largely correcting the distortion. Moreover, we described a pre-processing that can be used to combat the cosmic ray hits and other disturbances. Finally, we considered the integration of the pre- and post-processing with the GLS core into a full reduction pipeline, called UNIMAP, which stands out among the existing GLS map makers for its efficiency, quality and robustness.

ACKNOWLEDGEMENTS

We would like to thank David Ikhenaode (DIET) and Luca De Nardis (DIET) for helping with the compiler and Eugenio Schisano (IAPS) for performing initial performance verification. Moreover UNIMAP is indebted to its users, who often gave suggestions or spotted problems, and in particular with Kazi Kljrygl (ESTEC) and Bruno Altieri (ESAC). Moreover, we are deeply indebted to the anonymous reviewer for his precious suggestions.

REFERENCES

Aitken A. C., 1935, Proc. R. Soc. Edinburgh, 55, 42
Calzoletti L., Faustini F., 2013, UniHIPE (http://herschel.asdc.asi.it/index.php?page=unimap.html)
Cantalupo C. M., Borrill J. D., Jaffe A. H., Kisner T. S., Stompor R., 2010, ApJS, 187, 212
Chanial P., Panuzzo P., 2013, Tamasis (http://herschel.esac.esa.int/2013MapmakingWorkshop/presentations/TAMASIS_PChanial_MapMaking2013_DPWS.pdf)
De Graauw T. et al., 2010, A&A, 518, L6
Elia D. et al., 2013, ApJ, 772, 45
Griffin M. J. et al., 2010, A&A, 518, L3
Janssen M. A., Gulkis S., 1992, NATO ASI Ser., 359, 391
Mecina M., Mayer A., Ottensamer R., Luntzer A., Kerschbaum F., 2014, Proc. SPIE, 9152, 91522T
Paladini R. et al., 2013, PACS Map-making Tools: Analysis and Benchmarking (http://herschel.esac.esa.int/twiki/pub/Public/PacsCalibrationWeb/pacs_mapmaking_report_ex_sum_v3.pdf)
Patanchon G. et al., 2008, ApJ, 681, 708
Piazzo L., 2013, preprint (arXiv:1301.1246)
Piazzo L., Ikhenaode D., Natoli P., Pestalozzi M., Piacentini F., Traficante A., 2012, IEEE Trans. Image Process., 21, 3687
Piazzo L., Panuzzo P., Pestalozzi M., 2015, Signal Process., 108, 430
Pilbratt G. et al., 2010, A&A, 518, L1
Poglitsch A. et al., 2010, A&A, 518, L2
Tegmark M., 1997, Phys. Rev. D, 56, 4514
Traficante A. et al., 2011, MNRAS, 416, 2932
Vetterling W. T., Press W. H., Teukolsky S. A., Flannery B. P., 1992, Numerical Recipes in C. Cambridge Univ. Press, Cambridge
Viero M. P. et al., 2013, ApJ, 779, 32
Wright E. L., Hinshaw G., Bennett C. L., 1996, ApJ, 458, L53
Xu C. K. et al., 2013, preprint (arXiv:1401.2109)

APPENDIX A

A1 Jump detection

The jump detection is divided in three phases and is controlled by two parameters, the detection window ν and the detection threshold τ. In the first phase, each timeline is extracted from the TOD $\mathbf{d}$ and segmented into a sequence of blocks of 2ν samples with an overlap of ν samples. Then, by denoting by K the number of blocks, the kth block median μ_k and variance σ_k^2 are computed and a reference standard deviation is obtained as $\sigma = \text{med}(\sigma_1, \ldots, \sigma_K)$. Finally, candidate jumps are detected and approximately located by checking for jumps in the median sequence. Specifically, for $k = 1, \ldots, K-1$, if $|\mu_k - \mu_{k+1}| > \sigma\tau$ a candidate jump is detected, in blocks k and $k+1$.

In the second phase, the position of each candidate is found, by checking the differences of consecutive samples. Specifically, given two blocks containing a candidate and assuming that the first starts at index i and the second ends at index j, we evaluate $w_k = |d_k - d_{k+1}|$ for $k = i, \ldots, j-1$ and locate the candidate where w_k attains its maximum, i.e. at index k^* such that $w_{k^*} \geq w_k$.

In the third phase, each candidate undergoes a set of morphological tests and is discarded if it does not pass all of them. One test is a comparison with an estimate of the signal, $\bar{\mathbf{s}} = P\bar{\mathbf{m}} = P(P^{\mathrm{T}}P)^{-1}P\mathbf{d}$. The test is carried out for each candidate, by extracting from $\mathbf{d}$ a column vector $\mathbf{v}_{\mathrm{d}}$ of $2\nu + 1$ samples centred on the candidate position. A corresponding vector, $\mathbf{v}_{\mathrm{s}}$, is extracted from $\bar{\mathbf{s}}$. Next the vectors' mean, $\mu_{\mathbf{d}}$ and $\mu_{\mathbf{s}}$, variances, $\sigma_{\mathbf{d}}^2$ and $\sigma_{\mathbf{s}}^2$, and correlation coefficient[15] $\rho = (\mathbf{v}_{\mathrm{d}} - \mu_{\mathbf{d}})^{\mathrm{T}}(\mathbf{v}_{\mathrm{s}} - \mu_{\mathbf{s}})/(\sigma_{\mathbf{d}}\sigma_{\mathbf{s}})$ are computed. Finally, if $\rho > 0.7$, indicating that the two vectors have a similar shape, and $\min(\sigma_{\mathbf{d}}, \sigma_{\mathbf{s}})/\max(\sigma_{\mathbf{d}}, \sigma_{\mathbf{s}}) > 0.8$, indicating that the two vectors have a similar magnitude, the jump is deemed a signal feature and the candidate is discarded. The other tests are less important and we omit the details for the sake of brevity.

A2 Glitch detection

As a preliminary step, in order to remove drift and diffuse emission, the TOD is median high-pass filtered, to produce $\mathbf{h} = \text{high}(\mathbf{d})$. Next, like done in the PACS standard pipeline, we run a median absolute deviation filter, which is known to be robust in the presence of outliers. Specifically, for each pixel, the corresponding valid readouts are extracted from $\mathbf{h}$ and stored into a vector $\mathbf{v}$. An estimate of the pixel mean value is computed as $\mu = \text{med}(\mathbf{v})$ and an estimate of the standard deviation is obtained as $\sigma = \text{med}(|v_1 - \mu|, |v_2 - \mu|, |v_3 - \mu|, \ldots)$. Finally, given a threshold β, if $|v_k - \mu| > \sigma\beta$ the kth element of $\mathbf{v}$ is considered a glitch and the corresponding readout is flagged.

The detection is reliable only if a sufficient number of readouts, some tens, falls into each pixel. If this condition is not met, a coarser pixellization can be used in the detection. This is controlled by a parameter η that is an integer specifying a multiplicative factor for the pixel edge.

[15] Here and in the following it is understood that when a scalar is subtracted from a vector, it is subtracted from each of the vector elements.

A3 Multiplication by the covariance matrix

As a preliminary remark, note that the multiplication of a Toeplitz matrix with a vector can be implemented by means of linear convolution. Indeed, consider the multiplication of N with a vector $\mathbf{v}$ to produce $\mathbf{w} = N\mathbf{v}$. From $\mathbf{v}$ we can construct a sequence $x[k]$ for $k = -\infty, \ldots, \infty$ by padding the vector with zeros. Specifically, we set $x[k] = v_{k+1}$ for $k = 0, \ldots, D-1$ and set $x[k] = 0$ elsewhere. Then, it is not difficult to show that $\mathbf{w}$ can be obtained by extracting the first D samples from the sequence $z[k] = R[k]*x[k]$. Specifically, we have $w_k = z[k-1]$ for $k = 1, \ldots, D$.

We now show that, when $L \ll D$, an approximation to N^{-1} is the Toeplitz matrix Y obtained by setting $Y_{i,j} = h[i-j]$, where $h[k]$ is the noise filter response. To verify this fact, we consider the product $Z = NY$ and show that Z is approximately equal to the identity matrix. The jth column of Z, denoted by $\mathbf{w}$, is the product of the jth column of Y, denoted by $\mathbf{v}$, with N. As we have seen, $\mathbf{w}$ can be obtained by extracting D samples from the sequence $R[k]*x[k]$, where $x[k]$ is the sequence obtained by padding $\mathbf{v}$. By construction, $\mathbf{v}$ is a window of D samples taken from the sequence $h[k]$. Moreover, when $j > L$ and $j < D - L$, the vector $\mathbf{v}$ contains the whole non-zero part of $h[k]$ and we can write $x[k] = h[k-j]$, showing that $x[k]$ is a shifted copy of the noise filter response. Since $R[k]*h[k] = \delta[k]$ we have that $R[k]*x[k] = R[k]*h[k-j] = \delta[k-j]$ is a shifted copy of the unit pulse and the vector $\mathbf{w}$ is indeed the jth column of the identity matrix. On the contrary, when $j < L$ or $j > D - L$, vector $\mathbf{w}$ is only an approximation to the corresponding column of the identity matrix. This is a border effect that becomes negligible when $L \ll D$.

Since Y is an approximation of N^{-1} we can write $\mathbf{w} = N^{-1}\mathbf{v} \approx Y\mathbf{v}$. Moreover, since Y is Toeplitz, we can implement the multiplication as the convolution of the zero padded sequence $x[k]$ obtained from vector $\mathbf{v}$ with the noise filter response $h[k]$. Finally, as done in equation (10), we can improve the padding by using a mirror copy of the last L samples at each end of the vector $\mathbf{v}$, in order to reduce the border effect.

A4 Construction of the WGLS mask

First, an estimate of the variance of the noise affecting the distortion map, denoted by σ^2, is produced to serve as a reference. This is done by detecting the background in the PGLS map $\mathbf{m}_\mathrm{p}$, by extracting the background pixels from $\mathbf{e}$ and by computing their variance. Next, the mask is initialized using a threshold ϵ, by setting to 1 the kth element of $\mathbf{m}_\mathrm{e}$ if $|e_k| > \epsilon\sigma$, i.e. if the distortion in the kth pixel is higher than the standard deviation times the threshold. Finally, the mask is enlarged using a threshold $\gamma < \epsilon$, by adding to the mask the kth pixel if $|e_k| > \gamma\sigma$ and the pixel is adjacent to a pixel already belonging to the mask. The last step is repeated until no new pixels are added to the mask. The thresholds γ and ϵ can be set by means of a trial and error procedure, guided by the comparison of the WGLS mask with the PGLS distortion $\mathbf{e}$.

SUPPORTING INFORMATION

Additional Supporting Information may be found in the online version of this article:

Appendix.
(http://mnras.oxfordjournals.org/lookup/suppl/doi:10.1093/mnras/stu2453/-/DC1).

Please note: Oxford University Press are not responsible for the content or functionality of any supporting materials supplied by the authors. Any queries (other than missing material) should be directed to the corresponding author for the paper.

This paper has been typeset from a TEX/LATEX file prepared by the author.