

# Poisson denoising on the sphere: application to the *Fermi* gamma ray space telescope

J. Schmitt<sup>1</sup>, J. L. Starck<sup>1</sup>, J. M. Casandjian<sup>1</sup>, J. Fadili<sup>2</sup>, and I. Grenier<sup>1</sup>

<sup>1</sup> CEA, Laboratoire AIM, CEA/DSM-CNRS-Université Paris Diderot, CEA, IRFU, Service d'Astrophysique, Centre de Saclay, 91191 Gif-Sur-Yvette Cedex, France  
e-mail: jeremy.schmitt@cea.fr

<sup>2</sup> GREYC CNRS-ENSICAEN-Université de Caen, 6 Bd du Maréchal Juin, 14050 Caen Cedex, France

Received 7 December 2009 / Accepted 8 March 2010

## ABSTRACT

The Large Area Telescope (LAT), the main instrument of the *Fermi* gamma-ray Space telescope, detects high energy gamma rays with energies from 20 MeV to more than 300 GeV. The two main scientific objectives, the study of the Milky Way diffuse background and the detection of point sources, are complicated by the lack of photons. That is why we need a powerful Poisson noise removal method on the sphere which is efficient on low count Poisson data. This paper presents a new multiscale decomposition on the sphere for data with Poisson noise, called multi-scale variance stabilizing transform on the sphere (MS-VSTS). This method is based on a variance stabilizing transform (VST), a transform which aims to stabilize a Poisson data set such that each stabilized sample has a quasi constant variance. In addition, for the VST used in the method, the transformed data are asymptotically Gaussian. MS-VSTS consists of decomposing the data into a sparse multi-scale dictionary like wavelets or curvelets, and then applying a VST on the coefficients in order to get almost Gaussian stabilized coefficients. In this work, we use the isotropic undecimated wavelet transform (IUWT) and the curvelet transform as spherical multi-scale transforms. Then, binary hypothesis testing is carried out to detect significant coefficients, and the denoised image is reconstructed with an iterative algorithm based on hybrid steepest descent (HSD). To detect point sources, we have to extract the Galactic diffuse background: an extension of the method to background separation is then proposed. In contrary, to study the Milky Way diffuse background, we remove point sources with a binary mask. The gaps have to be interpolated: an extension to inpainting is then proposed. The method, applied on simulated *Fermi* LAT data, proves to be adaptive, fast and easy to implement.

**Key words.** methods: data analysis – techniques: image processing – gamma rays: general

## 1. Introduction

The *Fermi* gamma-ray space telescope, which was launched by NASA in June 2008, is a powerful space observatory which studies the high-energy gamma-ray sky (Atwood et al. 2009). *Fermi*'s main instrument, the Large Area Telescope (LAT), detects photons in an energy range between 20 MeV to greater than 300 GeV. The LAT is much more sensitive than its predecessor, the EGRET telescope on the Compton Gamma Ray Observatory, and is expected to find several thousand gamma ray sources, which is an order of magnitude more than its predecessor EGRET (Hartman et al. 1999).

Even with its  $\sim 1 \text{ m}^2$  effective area, the number of photons detected by the LAT outside the Galactic plane and away from intense sources is expected to be low. Consequently, the spherical photon count images obtained by *Fermi* are degraded by the fluctuations on the number of detected photons. The basic photon-imaging model assumes that the number of detected photons at each pixel location is Poisson distributed. More specifically, the image is considered as a realization of an inhomogeneous Poisson process. This quantum noise makes the source detection more difficult, consequently it is better to have an efficient denoising method for spherical Poisson data.

Several techniques have been proposed in the literature to estimate Poisson intensity in 2D. A major class of methods

adopt a multiscale bayesian framework specifically tailored for Poisson data (Nowak & Kolaczyk 2000), independently initiated by Timmerman & Nowak (1999) and Kolaczyk (1999). Lefkimmiaits et al. (2009) proposed an improved bayesian framework for analyzing Poisson processes, based on a multi-scale representation of the Poisson process in which the ratios of the underlying Poisson intensities in adjacent scales are modeled as mixtures of conjugate parametric distributions. Another approach includes preprocessing the count data by a variance stabilizing transform (VST) such as the Anscombe (1948) and the Fisz (1955) transforms, applied respectively in the spatial (Donoho 1993) or in the wavelet domain (Fryźlewicz & Nason 2004). The transform reforms the data so that the noise approximately becomes Gaussian with a constant variance. Standard techniques for independent identically distributed Gaussian noise are then used for denoising. Zhang et al. (2008) proposed a powerful method called multi-scale variance stabilizing transform (MS-VST). It consists in combining a VST with a multiscale transform (wavelets, ridgelets or curvelets), yielding asymptotically normally distributed coefficients with known variances. The choice of the multi-scale method depends on the morphology of the data. Wavelets represent more efficiently regular structures and isotropic singularities, whereas ridgelets are designed to represent global lines in an image, and curvelets represent efficiently curvilinear contours. Significant coefficients

are then detected with binary hypothesis testing, and the final estimate is reconstructed with an iterative scheme. In [Starck et al. \(2009\)](#), it was shown that sources can be detected in 3D LAT data (2D+time or 2D+energy) using a specific 3D extension of the MS-VST.

There is, to our knowledge, no method for Poisson intensity estimation on spherical data. It is possible to decompose the spherical data into several 2D projections, denoise each projection and reconstitute the denoised spherical data, but the projection induces some caveats like visual artifacts on the borders or deformation of the sources.

In the scope of the *Fermi* mission, we have two main scientific objectives:

- detection of point sources to build the catalog of gamma ray sources;
- study of the Milky Way diffuse background, which is due to interaction between cosmic rays and interstellar gas and radiation.

The first objective implies the extraction of the galactic diffuse background. Consequently, we want a method to suppress Poisson noise while extracting a model of the diffuse background. The second objective implies the suppression of the point sources: we want to apply a binary mask on the data (equal to 0 on point sources, and to 1 everywhere else) and to denoise the data while interpolating the missing part. Both objectives are linked: a better knowledge of the Milky Way diffuse background enables us to improve our background model, which leads to a better source detection, while the detected sources are masked to study the diffuse background.

The aim of this paper is to introduce a Poisson denoising method on the sphere called multi-scale variance stabilizing transform on the sphere (MS-VSTS) in order to denoise the *Fermi* photon count maps. This method is based on the MS-VST ([Zhang et al. 2008](#)) and on recent on multi-scale transforms on the sphere ([Starck et al. 2006](#); [Abrial et al. 2007](#); [Abrial et al. 2008](#)). Section 2 recalls the multiscale transforms on the sphere which are used in this paper, and Gaussian denoising methods based on sparse representations. Section 3 introduces the MS-VSTS. Section 4 applies the MS-VSTS to spherical data restoration. Section 5 applies the MS-VSTS to inpainting. Section 6 applies the MS-VSTS to background extraction. Conclusions are drawn in Sect. 7. In this paper, all experiments were performed on HEALPix maps with  $n_{\text{side}} = 128$  ([Górski et al. 2005](#)), which corresponds to a good pixelisation choice for the GLAST/*FERMI* resolution. The performance of the method is not dependent on the  $n_{\text{side}}$  parameter. For a given data set, if  $n_{\text{side}}$  is small, it just means that we don't want to investigate the finest scales. If  $n_{\text{side}}$  is large, the number of counts per pixel will be very small, and we may not have enough statistics to get any information at the finest resolution levels. But it will not have any bad effect on the solution. Indeed, the finest scales will be smoothed, since our algorithm will not detect any significant wavelet coefficients in the finest scales. Hence, starting with a fine pixelisation (i.e. large  $n_{\text{side}}$ ), our method will provide a kind of automatic binning, by thresholding wavelets coefficients at scales and at spatial positions where the number of counts is not sufficient.

## 2. Multi-scale analysis on the sphere

New multi-scale transforms on the sphere were developed by [Starck et al. \(2006\)](#). These transforms can be inverted and are

easy to compute with the HEALPix pixelisation, and were used for denoising, deconvolution, morphological component analysis and inpainting applications ([Abrial et al. 2007](#)). In this paper, we use the isotropic undecimated wavelet transform (IUWT) and the curvelet transform.

### 2.1. Multi-scale transforms on the sphere

#### 2.1.1. Isotropic undecimated wavelet transform on the sphere

The isotropic undecimated wavelet transform on the sphere (IUWT) is a wavelet transform on the sphere based on the spherical harmonics transform and with a very simple reconstruction algorithm. At scale  $j$ , we denote  $a_j(\theta, \varphi)$  the scale coefficients, and  $d_j(\theta, \varphi)$  the wavelet coefficients, with  $\theta$  denoting the longitude and  $\varphi$  the latitude. Given a scale coefficient  $a_j$ , the smooth coefficient  $a_{j+1}$  is obtained by a convolution with a low pass filter  $h_j$ :  $a_{j+1} = a_j * h_j$ . The wavelet coefficients are defined by the difference between two consecutive resolutions:  $d_{j+1} = a_j - a_{j+1}$ . A straightforward reconstruction is then given by:

$$a_0(\theta, \varphi) = a_j(\theta, \varphi) + \sum_{j=1}^J d_j(\theta, \varphi). \quad (1)$$

Since this transform is redundant, the procedure for reconstructing an image from its coefficients is not unique and this can be profitably used to impose additional constraints on the synthesis functions (e.g. smoothness, positivity). A reconstruction algorithm based on a variety of filter banks is described in [Starck et al. \(2006\)](#).

#### 2.1.2. Curvelet transform on the sphere

The curvelet transform enables the directional analysis of an image in different scales. The data undergo an Isotropic Undecimated Wavelet Transform on the sphere. Each scale  $j$  is then decomposed into smoothly overlapping blocks of side-length  $B_j$  in such a way that the overlap between two vertically adjacent blocks is a rectangular array of size  $B_j \times B_j/2$ , using the HEALPix pixelisation. Finally, the ridgelet transform ([Candes & Donoho 1999](#)) is applied on each individual block. The method is best for the detection of anisotropic structures and smooth curves and edges of different lengths. More details can be found in [Starck et al. \(2006\)](#).

### 2.2. Application to Gaussian denoising on the sphere

Multiscale transforms on the sphere have been used successfully for Gaussian denoising via non-linear filtering or thresholding methods. Hard thresholding, for instance, consists of setting all insignificant coefficients (i.e. coefficients with an absolute value below a given threshold) to zero. In practice, we need to estimate the noise standard deviation  $\sigma_j$  in each band  $j$  and a coefficient  $w_j$  is significant if  $|w_j| > \kappa \sigma_j$ , where  $\kappa$  is a parameter typically chosen between 3 and 5. Denoting  $Y$  the noisy data and  $HT_\lambda$  the thresholding operator, the filtered data  $X$  are obtained by:

$$X = \Phi HT_\lambda(\Phi^T Y), \quad (2)$$

where  $\Phi^T$  is the multiscale transform (IUWT or curvelet) and  $\Phi$  is the multiscale reconstruction.  $\lambda$  is a vector which has the size of the number of bands in the used multiscale transform. The

thresholding operation thresholds all coefficients in band  $j$  with the threshold  $\lambda_j = \kappa\sigma_j$ .

### 3. Multi-scale variance stabilizing transform on the sphere (MS-VSTS)

#### 3.1. Principle of VST

##### 3.1.1. VST of a Poisson process

Given Poisson data  $\mathbf{Y} := (Y_i)_i$ , each sample  $Y_i \sim \mathcal{P}(\lambda_i)$  has a variance  $\text{Var}[Y_i] = \lambda_i$ . Thus, the variance of  $\mathbf{Y}$  is signal-dependant. The aim of a VST  $T$  is to stabilize the data such that each coefficient of  $T(\mathbf{Y})$  has an (asymptotically) constant variance, say 1, irrespective of the value of  $\lambda_i$ . In addition, for the VST used in this study,  $T(\mathbf{Y})$  is asymptotically normally distributed. Thus, the VST-transformed data are asymptotically stationary and Gaussian.

The [Anscombe \(1948\)](#) transform is a widely used VST which has a simple square-root form

$$T(Y) := 2\sqrt{Y+3/8}. \quad (3)$$

We can show that  $T(Y)$  is asymptotically normal as the intensity increases.

$$T(Y) - 2\sqrt{\lambda} \xrightarrow[\lambda \rightarrow +\infty]{\mathcal{D}} \mathcal{N}(0, 1). \quad (4)$$

It can be shown that the Anscombe VST requires a high underlying intensity to well stabilize the data (typically for  $\lambda \geq 10$ ) ([Zhang et al. 2008](#)).

##### 3.1.2. VST of a filtered Poisson process

Let  $Z_j := \sum_i h[i]Y_{j-i}$  be the filtered process obtained by convolving  $(Y_i)_i$  with a discrete filter  $h$ . We will use  $Z$  to denote any of the  $Z_j$ 's. Let us define  $\tau_k := \sum_i (h[i])^k$  for  $k = 1, 2, \dots$ . In addition, we adopt a local homogeneity assumption stating that  $\lambda_{j-i} = \lambda$  for all  $i$  within the support of  $h$ .

We define the square-root transform  $T$  as follows:

$$T(Z) := b \cdot \text{sign}(Z+c)|Z+c|^{1/2}, \quad (5)$$

where  $b$  is a normalizing factor. [Lemma 1](#) proves that  $T$  is a VST for a filtered Poisson process (with a nonzero-mean filter) in that  $T(Y)$  is asymptotically normally distributed with a stabilized variance as  $\lambda$  becomes large (see [Zhang et al. 2008](#), for a proof).

**Lemma 1 (Square root as VST)** *If  $\tau_1 \neq 0$ ,  $\|h\|_2, \|h\|_3 < \infty$ , then we have:*

$$\text{sign}(Z+c)\sqrt{|Z+c|} - \text{sign}(\tau_1)\sqrt{|\tau_1|\lambda} \xrightarrow[\lambda \rightarrow +\infty]{\mathcal{D}} \mathcal{N}\left(0, \frac{\tau_2}{4|\tau_1|}\right). \quad (6)$$

### 3.2. MS-VSTS

The MS-VSTS consists in combining the square-root VST with a multi-scale transform.

#### 3.2.1. MS-VSTS + IUWT

This section describes the MS-VSTS + IUWT, which is a combination of a square-root VST with the IUWT. The recursive scheme is:

$$\begin{aligned} \text{IUWT} \begin{cases} a_j = h_{j-1} * a_{j-1} \\ d_j = a_{j-1} - a_j \end{cases} \\ \implies \text{MS-VSTS} \begin{cases} a_j = h_{j-1} * a_{j-1} \\ d_j = T_{j-1}(a_{j-1}) - T_j(a_j). \end{cases} \end{aligned} \quad (7)$$

In (7), the filtering on  $a_{j-1}$  can be rewritten as a filtering on  $a_0 := \mathbf{Y}$ , i.e.,  $a_j = h^{(j)} * a_0$ , where  $h^{(j)} = h_{j-1} * \dots * h_1 * h_0$  for  $j \geq 1$  and  $h^{(0)} = \delta$ , where  $\delta$  is the Dirac pulse ( $\delta = 1$  on a single pixel and 0 everywhere else).  $T_j$  is the VST operator at scale  $j$ :

$$T_j(a_j) = b^{(j)} \text{sign}(a_j + c^{(j)}) \sqrt{|a_j + c^{(j)}|}. \quad (8)$$

Let us define  $\tau_k^{(j)} := \sum_i (h^{(j)}[i])^k$ . In [Zhang et al. \(2008\)](#), it has been shown that, to have an optimal convergence rate for the VST, the constant  $c^{(j)}$  associated to  $h^{(j)}$  should be set to:

$$c^{(j)} := \frac{7\tau_2^{(j)}}{8\tau_1^{(j)}} - \frac{\tau_3^{(j)}}{2\tau_2^{(j)}}. \quad (9)$$

The MS-VSTS+IUWT procedure is directly invertible as we have:

$$a_0(\theta, \varphi) = T_0^{-1} \left[ T_j(a_j) + \sum_{j=1}^J d_j \right](\theta, \varphi). \quad (10)$$

Setting  $b^{(j)} := \text{sgn}(\tau_1^{(j)})/\sqrt{|\tau_1^{(j)}|}$ , if  $\lambda$  is constant within the support of the filter.  $h^{(j)}$ , then we have ([Zhang et al. 2008](#)):

$$d_j(\theta, \varphi) \xrightarrow[\lambda \rightarrow +\infty]{\mathcal{D}} \mathcal{N} \left( 0, \frac{\tau_2^{(j-1)}}{4\tau_1^{(j-1)^2}} + \frac{\tau_2^{(j)}}{4\tau_1^{(j)^2}} - \frac{\langle h^{(j-1)}, h^{(j)} \rangle}{2\tau_1^{(j-1)}\tau_1^{(j)}} \right), \quad (11)$$

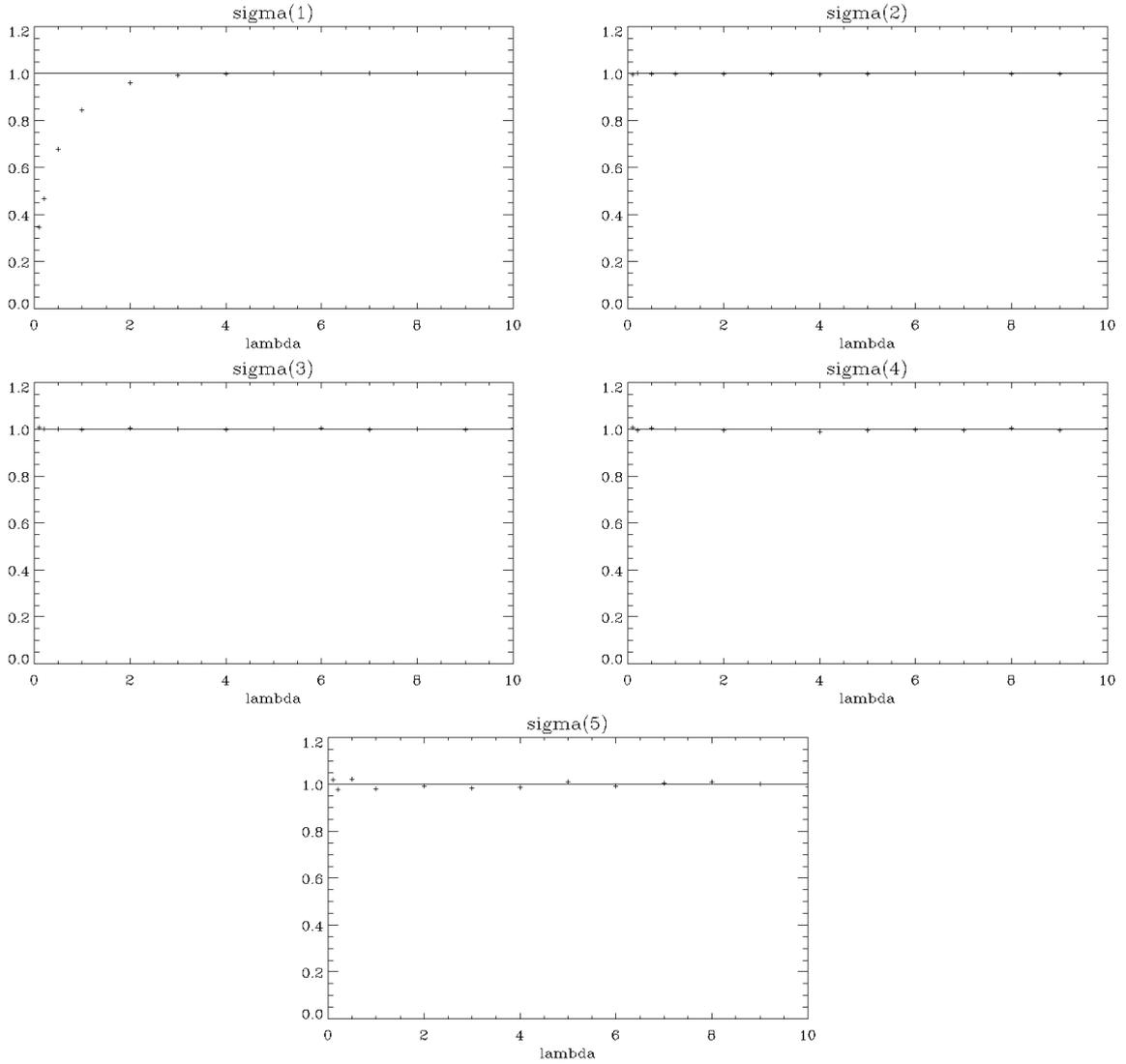
where  $\langle \cdot, \cdot \rangle$  denotes inner product.

It means that the detail coefficients issued from locally homogeneous parts of the signal follow asymptotically a central normal distribution with an intensity-independant variance which relies solely on the filter  $h$  and the current scale for a given filter  $h$ . Consequently, the stabilized variances and the constants  $b^{(j)}, c^{(j)}, \tau_k^{(j)}$  can all be pre-computed. Let us define  $\sigma_{(j)}^2$  the stabilized variance at scale  $j$  for a locally homogeneous part of the signal:

$$\sigma_{(j)}^2 = \frac{\tau_2^{(j-1)}}{4\tau_1^{(j-1)^2}} + \frac{\tau_2^{(j)}}{4\tau_1^{(j)^2}} - \frac{\langle h^{(j-1)}, h^{(j)} \rangle}{2\tau_1^{(j-1)}\tau_1^{(j)}}. \quad (12)$$

To compute the  $\sigma_{(j)}, b^{(j)}, c^{(j)}, \tau_k^{(j)}$ , we only have to know the filters  $h^{(j)}$ . We compute these filters thanks to the formula  $a_j = h^{(j)} * a_0$ , by applying the IUWT to a Dirac pulse  $a_0 = \delta$ . Then, the  $h^{(j)}$  are the scaling coefficients of the IUWT. The  $\sigma_{(j)}$  have been precomputed for a 6-scaled IUWT ([Table 1](#)).

We have simulated Poisson images of different constant intensities  $\lambda$ , computed the IUWT with MS-VSTS on each image and observed the variation of the normalized value of  $\sigma_{(j)}$  ( $(\sigma_{(j)})_{\text{simulated}}/(\sigma_{(j)})_{\text{theoretical}}$ ) as a function of  $\lambda$  for each scale  $j$  ([Fig. 1](#)). We see that the wavelet coefficients are stabilized when  $\lambda \geq 0.1$  except for the first wavelet scale, which is mostly constituted of noise. In [Fig. 2](#), we compare the result of MS-VSTS with



**Fig. 1.** Normalized value  $((\sigma_{(j)})_{\text{simulated}}/(\sigma_{(j)})_{\text{theoretical}})$  of the stabilized variances at each scale  $j$  as a function of  $\lambda$ .

**Table 1.** Precomputed values of the variances  $\sigma_j$  of the wavelet coefficients.

Wavelet scale $j$	Value of $\sigma_j$
1	0.484704
2	0.0552595
3	0.0236458
4	0.0114056
5	0.00567026

Anscombe + wavelet shrinkage, on sources of varying intensities. We see that MS-VSTS works well on sources of very low intensities, whereas Anscombe doesn't work when the intensity is too low.

### 3.2.2. MS-VSTS + curvelets

As the first step of the algorithm is an IUWT, we can stabilize each resolution level as in Eq. (7). We then apply the local ridgelet transform on each stabilized wavelet band.

It is not as straightforward as with the IUWT to derive the asymptotic noise variance in the stabilized curvelet domain.

In our experiments, we derived them using simulated Poisson data of stationary intensity level  $\lambda$ . After having checked that the standard deviation in the curvelet bands becomes stabilized as the intensity level increases (which means that the stabilization is working properly), we stored the standard deviation  $\sigma_{j,l}$  for each wavelet scale  $j$  and each ridgelet band  $l$  (Table 2).

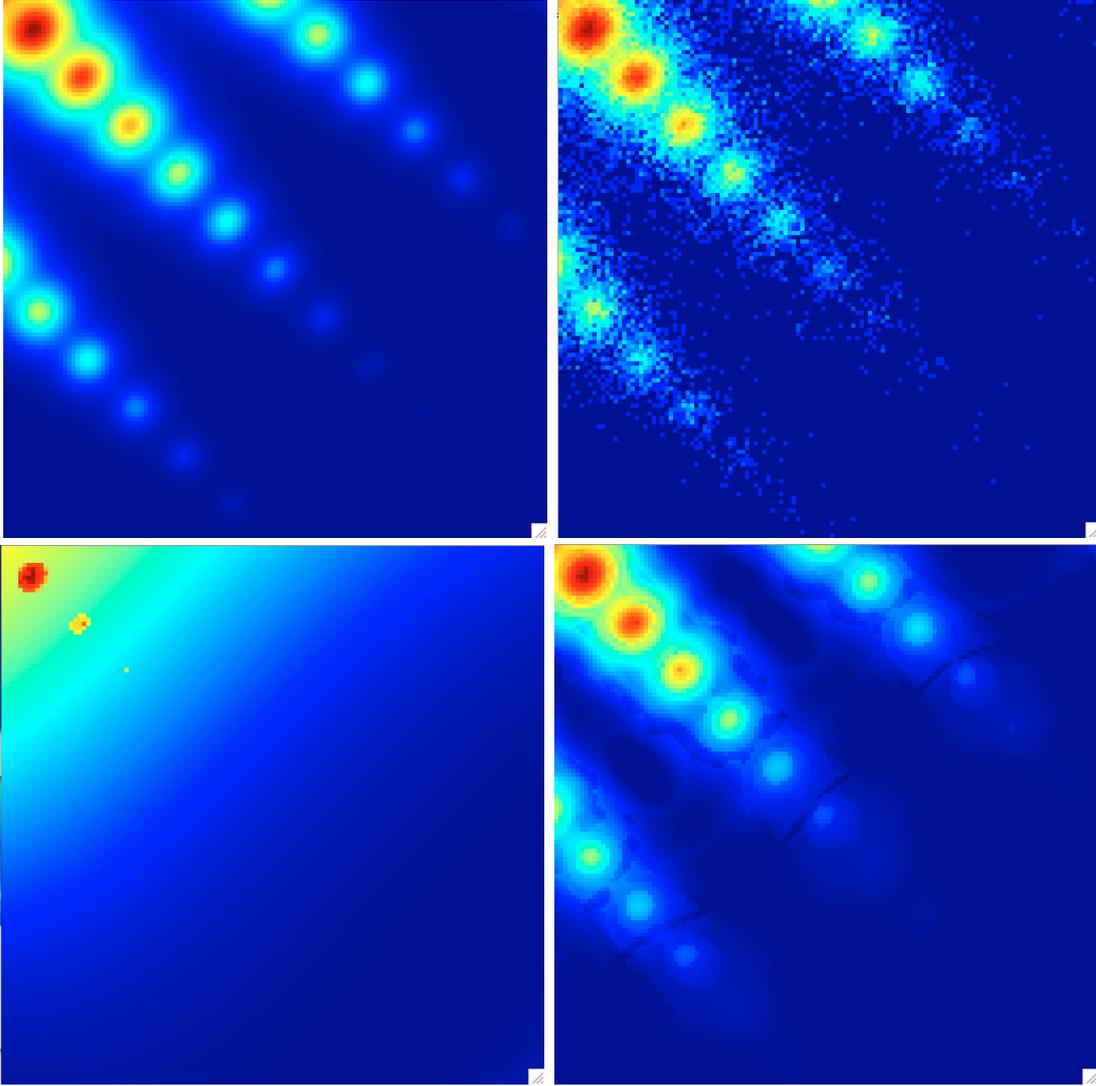
## 4. Poisson denoising

### 4.1. MS-VST + IUWT

Under the hypothesis of homogeneous Poisson intensity, the stabilized wavelet coefficients  $d_j$  behave like centered Gaussian variables of standard deviation  $\sigma_{(j)}$ . We can detect significant coefficients with binary hypothesis testing as in Gaussian denoising.

Under the null hypothesis  $\mathcal{H}_0$  of homogeneous Poisson intensity, the distribution of the stabilized wavelet coefficient  $d_j[k]$  at scale  $j$  and location index  $k$  can be written as:

$$p(d_j[k]) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp(-d_j[k]^2/2\sigma_j^2). \quad (13)$$



**Fig. 2.** Comparison of MS-VSTS with Anscombe + wavelet shrinkage on a single HEALPix face. *Top left:* sources of varying intensity. *Top right:* sources of varying intensity with Poisson noise. *Bottom left:* Poisson sources of varying intensity reconstructed with Anscombe + wavelet shrinkage. *Bottom right:* Poisson sources of varying intensity reconstructed with MS-VSTS.

**Table 2.** Asymptotic values of the variances  $\sigma_{jk}$  of the curvelet coefficients.

$j$	$l = 1$	$l = 2$	$l = 3$	$l = 4$
1	1.74550	0.348175		
2	0.230621	0.248233	0.196981	
3	0.0548140	0.0989918	0.219056	
4	0.0212912	0.0417454	0.0875663	0.20375
5	0.00989616	0.0158273	0.0352021	0.163248

The rejection of the hypothesis  $\mathcal{H}_0$  depends on the double-sided  $p$ -value:

$$p_j[k] = 2 \frac{1}{\sqrt{2\pi}\sigma_j} \int_{|d_j[k]|}^{+\infty} \exp(-x^2/2\sigma_j^2) dx. \quad (14)$$

Consequently, to accept or reject  $\mathcal{H}_0$ , we compare each  $|d_j[k]|$  with a critical threshold  $\kappa\sigma_j$ ,  $\kappa = 3, 4$  or  $5$  corresponding respectively to significance levels. This amounts to deciding that:

- if  $|d_j[k]| \geq \kappa\sigma_j$ ,  $d_j[k]$  is significant;
- if  $|d_j[k]| < \kappa\sigma_j$ ,  $d_j[k]$  is not significant.

Then we have to invert the MS-VSTS scheme to reconstruct the estimate. However, although the direct inversion is possible (Eq. (10)), it can not guarantee a positive intensity estimate, while the Poisson intensity is always nonnegative. A positivity projection can be applied, but important structures could be lost in the estimate. To tackle this problem, we reformulate the reconstruction as a convex optimisation problem and solve it iteratively with an algorithm based on hybrid steepest descent (HSD) (Yamada 2001).

We define the multiresolution support  $\mathcal{M}$ , which is determined by the set of detected significant coefficients after hypothesis testing:

$$\mathcal{M} := (j, k) \text{ if } d_j[k] \text{ is declared significant.} \quad (15)$$

We formulate the reconstruction problem as a convex constrained minimization problem:

$$\text{Arg min}_X \|\Phi^T X\|_1, \text{ s.t. } \begin{cases} X \geq 0, \\ \forall (j, k) \in \mathcal{M}, (\Phi^T X)_j[k] = (\Phi^T Y)_j[k], \end{cases} \quad (16)$$

where  $\Phi$  denotes the IUWT synthesis operator.

This problem is solved with the following iterative scheme: the image is initialised by  $X^{(0)} = 0$ , and the iteration scheme is, for  $n = 0$  to  $N_{\max} - 1$ :

$$\tilde{X} = P_+[X^{(n)} + \Phi P_{\mathcal{M}} \Phi^T (Y - X^{(n)})] \quad (17)$$

$$X^{(n+1)} = \Phi \text{ST}_{\lambda_n} [\Phi^T \tilde{X}] \quad (18)$$

where  $P_+$  denotes the projection on the positive orthant,  $P_{\mathcal{M}}$  denotes the projection on the multiresolution support  $\mathcal{M}$ :

$$P_{\mathcal{M}} d_j[k] = \begin{cases} d_j[k] & \text{if } (j, k) \in \mathcal{M}, \\ 0 & \text{otherwise.} \end{cases} \quad (19)$$

and  $\text{ST}_{\lambda_n}$  the soft-thresholding with threshold  $\lambda_n$ :

$$\text{ST}_{\lambda_n}[d] = \begin{cases} \text{sign}(d)(|d| - \lambda_n) & \text{if } |d| \geq \lambda_n, \\ 0 & \text{otherwise.} \end{cases} \quad (20)$$

We chose a decreasing threshold  $\lambda_n = \frac{N_{\max} - n}{N_{\max} - 1}$ ,  $n = 1, 2, \dots, N_{\max}$ .

The final estimate of the Poisson intensity is:  $\hat{\Lambda} = X^{(N_{\max})}$ . Algorithm 1 summarizes the main steps of the MS-VSTS + IUWT denoising algorithm.

---

#### Algorithm 1 MS-VSTS + IUWT Denoising

---

**Require:** data  $a_0 := Y$ , number of iterations  $N_{\max}$ , threshold  $\kappa$

##### Detection

- 1: **for**  $j = 1$  to  $J$  **do**
- 2:   Compute  $a_j$  and  $d_j$  using (7).
- 3:   Hard threshold  $|d_j[k]|$  with threshold  $\kappa\sigma_j$  and update  $\mathcal{M}$ .
- 4: **end for**

##### Estimation

- 5: Initialize  $X^{(0)} = 0$ ,  $\lambda_0 = 1$ .
  - 6: **for**  $n = 0$  to  $N_{\max} - 1$  **do**
  - 7:    $\tilde{X} = P_+[X^{(n)} + \Phi P_{\mathcal{M}} \Phi^T (Y - X^{(n)})]$ .
  - 8:    $X^{(n+1)} = \Phi \text{ST}_{\lambda_n} [\Phi^T \tilde{X}]$ .
  - 9:    $\lambda_{n+1} = \frac{N_{\max} - (n+1)}{N_{\max} - 1}$ .
  - 10: **end for**
  - 11: Get the estimate  $\hat{\Lambda} = X^{(N_{\max})}$ .
- 

#### 4.2. Multi-resolution support adaptation

When two sources are too close, the **less** intense source may not be detected because of the negative wavelet coefficients of the brightest source. To avoid such a drawback, we may update the multi-resolution support at each iteration. The idea is to withdraw the detected sources and to make a detection on the remaining residual, so as to detect the sources which may have been missed at the first detection.

At each iteration  $n$ , we compute the MS-VSTS of  $X^{(n)}$ . We denote  $d_j^{(n)}[k]$  the stabilised coefficients of  $X^{(n)}$ . We make a hard thresholding on  $(d_j[k] - d_j^{(n)}[k])$  with the same thresholds as in the detection step. Significant coefficients are added to the multiresolution support  $\mathcal{M}$ .

The main steps of the algorithm are summarized in Algorithm 2. In practice, we use Algorithm 2 instead of Algorithm 1 in our experiments.

#### 4.3. MS-VST + Curvelets

Insignificant coefficients are zeroed by using the same hypothesis testing framework as in the wavelet scale. At each wavelet

---

#### Algorithm 2 MS-VSTS + IUWT Denoising + Multiresolution Support Adaptation

---

**Require:** data  $a_0 := Y$ , number of iterations  $N_{\max}$ , threshold  $\kappa$

##### Detection

- 1: **for**  $j = 1$  to  $J$  **do**
- 2:   Compute  $a_j$  and  $d_j$  using (7).
- 3:   Hard threshold  $|d_j[k]|$  with threshold  $\kappa\sigma_j$  and update  $\mathcal{M}$ .
- 4: **end for**

##### Estimation

- 5: Initialize  $X^{(0)} = 0$ ,  $\lambda_0 = 1$ .
  - 6: **for**  $n = 0$  to  $N_{\max} - 1$  **do**
  - 7:    $\tilde{X} = P_+[X^{(n)} + \Phi P_{\mathcal{M}} \Phi^T (Y - X^{(n)})]$ .
  - 8:    $X^{(n+1)} = \Phi \text{ST}_{\lambda_n} [\Phi^T \tilde{X}]$ .
  - 9:   Compute the MS-VSTS on  $X^{(n)}$  to get the stabilised coefficients  $d_j^{(n)}$ .
  - 10:   Hard threshold  $|d_j[k] - d_j^{(n)}[k]|$  and update  $\mathcal{M}$ .
  - 11:    $\lambda_{n+1} = \frac{N_{\max} - (n+1)}{N_{\max} - 1}$ .
  - 12: **end for**
  - 13: Get the estimate  $\hat{\Lambda} = X^{(N_{\max})}$ .
- 

scale  $j$  and ridgelet band  $k$ , we make a hard thresholding on curvelet coefficients with threshold  $\kappa\sigma_{j,k}$ ,  $\kappa = 3, 4$  or  $5$ . Finally, a direct reconstruction can be performed by first inverting the local ridgelet transforms and then inverting the MS-VST + IUWT (Equation (10)). An iterative reconstruction may also be performed.

Algorithm 3 summarizes the main steps of the MS-VSTS + Curvelets denoising algorithm.

---

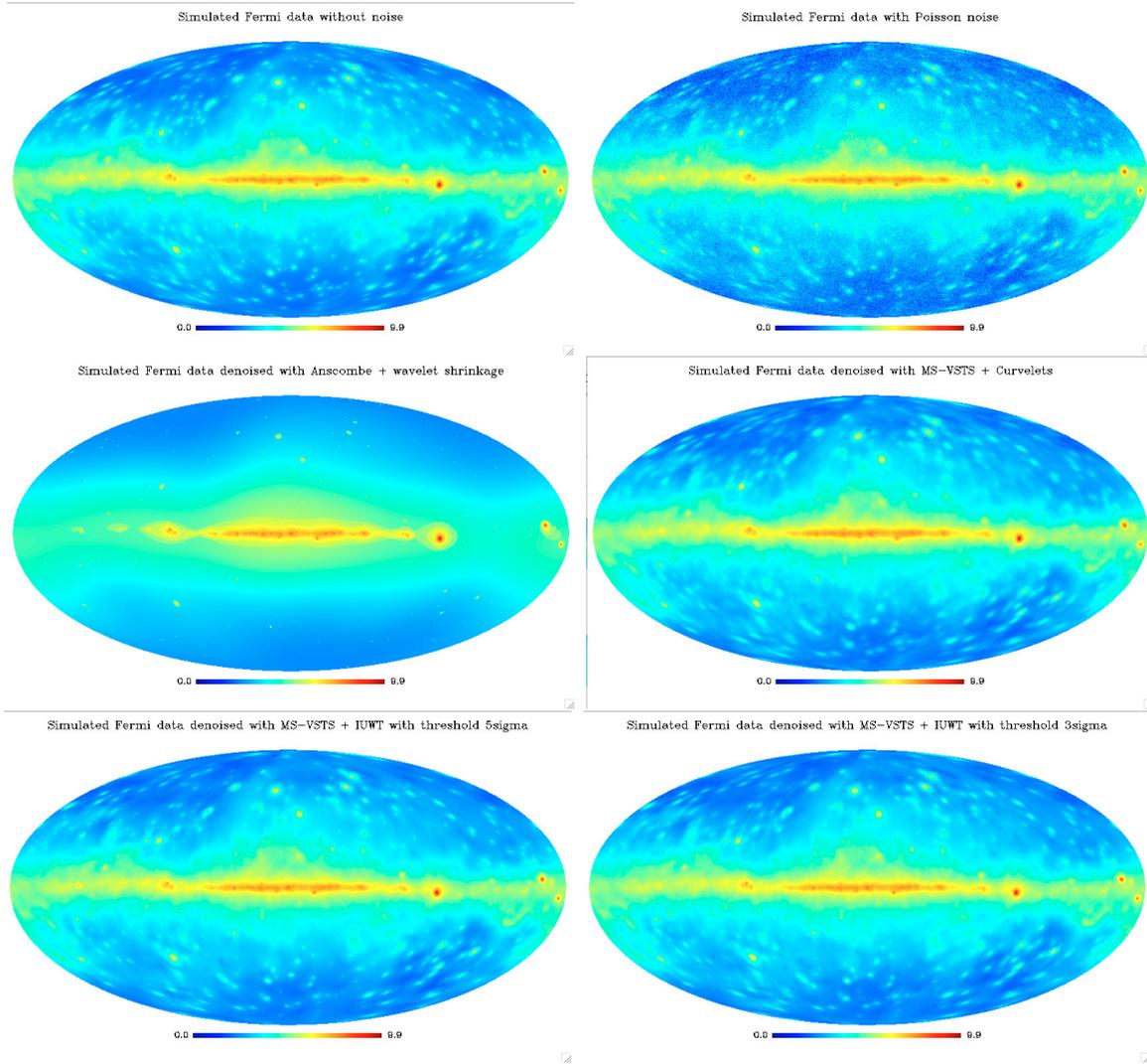
#### Algorithm 3 MS-VSTS + curvelets denoising

---

- 1: Apply the MS-VST + IUWT with  $J$  scales to get the stabilized wavelet subbands  $d_j$ .
  - 2: Set  $B_1 = B_{\min}$ .
  - 3: **for**  $j = 1$  to  $J$  **do**
  - 4:   Partition the subband  $d_j$  with blocks of side-length  $B_j$  and apply the digital ridgelet transform to each block to obtain the stabilized curvelets coefficients.
  - 5:   **if**  $j$  modulo  $2 = 1$  **then**
  - 6:      $B_{j+1} = 2B_j$
  - 7:   **else**
  - 8:      $B_{j+1} = B_j$
  - 9:   **end if**
  - 10:   HTs on the stabilized curvelet coefficients.
  - 11: **end for**
  - 12: Invert the ridgelet transform in each block before inverting the MS-VST + IUWT.
- 

#### 4.4. Experiments

The method was tested on simulated *Fermi* data. The simulated data are the sum of a Milky Way diffuse background model and 1000 gamma ray point sources. We based our Galactic diffuse emission model intensity on the model *gll\_iem\_v02* obtained at the *Fermi* Science Support Center (Myers 2009). This model results from a fit of the LAT photons with various gas templates as well as inverse Compton in several energy bands. We used a realistic point-spread function for the sources, based on Monte Carlo simulations of the LAT and accelerator tests, that scale approximately as  $0.8(E/1 \text{ GeV})^{-0.8}$  degrees. The position of the 205 brightest sources were taken from the *Fermi* 3-month source list (Abdo et al. 2009). The position of the 795 remaining sources follow the LAT 1-year Point Source Catalog



**Fig. 3.** *Top left:* Fermi simulated map without noise. *Top right:* Fermi simulated map with Poisson noise. *Middle left:* Fermi simulated map denoised with Anscombe VST + wavelet shrinkage. *Middle right:* Fermi simulated map denoised with MS-VSTS + curvelets (Algorithm 3). *Bottom left:* Fermi simulated map denoised with MS-VSTS + IUWT (Algorithm 1) with threshold  $5\sigma_j$ . *Bottom right:* Fermi simulated map denoised with MS-VSTS + IUWT (Algorithm 1) with threshold  $3\sigma_j$ . Pictures are in logarithmic scale.

(Myers 2010) sources distribution: each simulated source was randomly sorted in a box of  $\Delta l = 5^\circ$  and  $\Delta b = 1^\circ$  around a LAT 1-year catalog source. We simulated each source assuming a power-law dependence with its spectral index given by the 3-month source list and the first year catalog. We used an exposure of  $3 \times 10^{10}$  s cm<sup>2</sup> corresponding approximatively to one year of Fermi all-sky survey around 1 GeV. The simulated counts map shown here correspond to photons energy from 150 MeV to 20 GeV.

Figure 3 compares the result of denoising with MS-VST + IUWT (Algorithm 1), MS-VST + curvelets (Algorithm 3) and Anscombe VST + wavelet shrinkage on a simulated Fermi map. Figure 4 shows one HEALPix face of the results. As expected from theory, the Anscombe method produces poor results to denoise Fermi data, because the underlying intensity is too weak. Both wavelet and curvelet denoising on the sphere perform much better. For this application, wavelets are slightly better than curvelets ( $SNR_{\text{wavelets}} = 65.8$  dB,  $SNR_{\text{curvelets}} = 37.3$  dB,  $SNR(\text{dB}) = 20 \log(\sigma_{\text{signal}}/\sigma_{\text{noise}})$ ). As this image contains many point sources, this result is expected. Indeed wavelet are better than curvelets to represent isotropic objects.

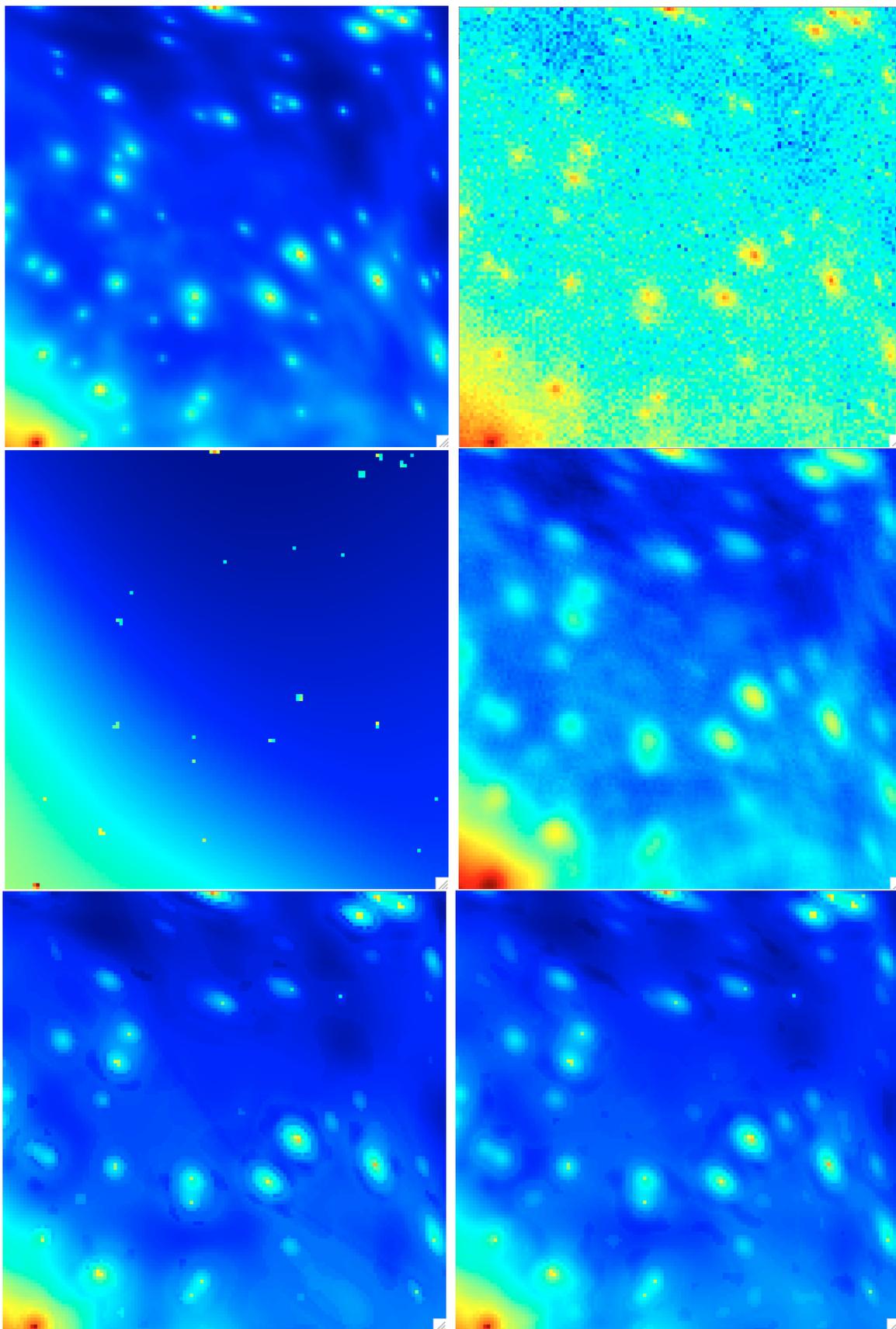
## 5. Milky Way diffuse background study: denoising and inpainting

In order to extract from the Fermi photon maps the galactic diffuse emission, we want to remove the point sources from the Fermi image. As our HSD algorithm is very close to the MCA algorithm (Starck et al. 2004), an idea is to mask the most intense sources and to modify our algorithm in order to interpolate through the gaps exactly as in the MCA-Inpainting algorithm (Abrial et al. 2007). This modified algorithm can be called MS-VSTS-Inpainting algorithm.

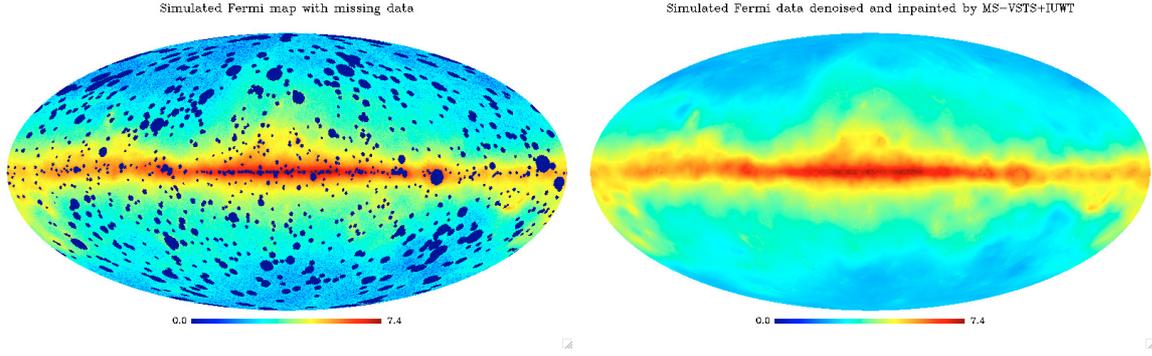
The problem can be reformulated as a convex constrained minimization problem:

$$\begin{aligned} \text{Arg min}_X \|\Phi^T X\|_1, \text{ s.t.} \\ \left\{ \begin{array}{l} X \geq 0, \\ \forall (j, k) \in \mathcal{M}, (\Phi^T \Pi X)_j[k] = (\Phi^T Y)_j[k], \end{array} \right. \end{aligned} \quad (21)$$

where  $\Pi$  is a binary mask (1 on valid data and 0 on invalid data).



**Fig. 4.** View of a single HEALPix face from the results of Fig. 3. *Top left:* Fermi simulated map without noise. *Top right:* Fermi simulated map with Poisson noise. *Middle left:* Fermi simulated map denoised with Anscombe VST + wavelet shrinkage. *Middle right:* Fermi simulated map denoised with MS-VSTS + curvelets (Algorithm 3). *Bottom left:* Fermi simulated map denoised with MS-VSTS + IUWT (Algorithm 1) with threshold  $5\sigma_j$ . *Bottom right:* Fermi simulated map denoised with MS-VSTS + IUWT (Algorithm 1) with threshold  $3\sigma_j$ . Pictures are in logarithmic scale.



**Fig. 5.** MS-VSTS – Inpainting. *Left:* *Fermi* simulated map with Poisson noise and the most luminous sources masked. *Right:* *Fermi* simulated map denoised and inpainted with wavelets (Algorithm 4). Pictures are in logarithmic scale.

The iterative scheme can be adapted to cope with a binary mask, which gives:

$$\tilde{X} = P_+[X^{(n)} + \Phi P_M \Phi^T \Pi(Y - X^{(n)})], \quad (22)$$

$$X^{(n+1)} = \Phi \text{ST}_{\lambda_n}[\Phi \tilde{X}]. \quad (23)$$

The thresholding strategy has to be adapted. Indeed, for the inpainting task we need to have a very large initial threshold in order to have a very smooth image in the beginning and to refine the details progressively. We chose an exponentially decreasing threshold:

$$\lambda_n = \lambda_{\max} \left( 2^{\left( \frac{N_{\max} - n}{N_{\max} - 1} \right)} - 1 \right), n = 1, 2, \dots, N_{\max}, \quad (24)$$

where  $\lambda_{\max} = \max(\Phi^T X)$ .

---

#### Algorithm 4 MS-VST + IUWT Denoising + Inpainting

---

**Require:** data  $a_0 := Y$ , mask  $\Pi$ , number of iterations  $N_{\max}$ , threshold  $\kappa$ .

##### Detection

- 1: **for**  $j = 1$  to  $J$  **do**
- 2:   Compute  $a_j$  and  $d_j$  using (7).
- 3:   Hard threshold  $|d_j[k]|$  with threshold  $\kappa \sigma_j$  and update  $\mathcal{M}$ .
- 4: **end for**

##### Estimation

- 5: Initialize  $\tilde{X}^{(0)} = 0$ ,  $\lambda_0 = \lambda_{\max}$ .
  - 6: **for**  $n = 0$  to  $N_{\max} - 1$  **do**
  - 7:    $\tilde{X} = P_+[X^{(n)} + \Phi P_M \Phi^T \Pi(Y - X^{(n)})]$ .
  - 8:    $X^{(n+1)} = \Phi \text{ST}_{\lambda_n}[\Phi \tilde{X}]$ .
  - 9:    $\lambda_{n+1} = \lambda_{\max} \left( 2^{\left( \frac{N_{\max} - (n+1)}{N_{\max} - 1} \right)} - 1 \right)$
  - 10: **end for**
  - 11: Get the estimate  $\hat{A} = X^{(N_{\max})}$ .
- 

## Experiment

We applied this method on simulated *Fermi* data where we masked the most luminous sources.

The results are in Fig. 5. The MS-VST + IUWT + Inpainting method (Algorithm 4) interpolates the missing data very well. Indeed, the missing part can not be seen anymore in the inpainted map, which shows that the diffuse emission component has been correctly reconstructed.

## 6. Source detection: denoising and background modeling

### 6.1. Method

In the case of *Fermi* data, the diffuse gamma-ray emission from the Milky Way, due to interaction between cosmic rays and interstellar gas and radiation, makes a relatively intense background. We have to extract this background in order to detect point sources. This diffuse interstellar emission can be modelled by a linear combination of gas templates and inverse compton map. We can use such a background model and incorporate a background removal in our denoising algorithm.

We note  $Y$  the data,  $B$  the background we want to remove, and  $d_j^{(b)}[k]$  the MS-VSTS coefficients of  $B$  at scale  $j$  and position  $k$ . We determine the multi-resolution support by comparing  $|d_j[k] - d_j^{(b)}[k]|$  with  $\kappa \sigma_j$ .

We formulate the reconstruction problem as a convex constrained minimization problem:

$$\text{Arg min}_X \|\Phi^T X\|_1, \text{ s.t.}$$

$$\begin{cases} X \geq 0, \\ \forall (j, k) \in \mathcal{M}, (\Phi^T X)_j[k] = (\Phi^T (Y - B))_j[k]. \end{cases} \quad (25)$$

Then, the reconstruction algorithm scheme becomes:

$$\tilde{X} = P_+[X^{(n)} + \Phi P_M \Phi^T (Y - B - X^{(n)})], \quad (26)$$

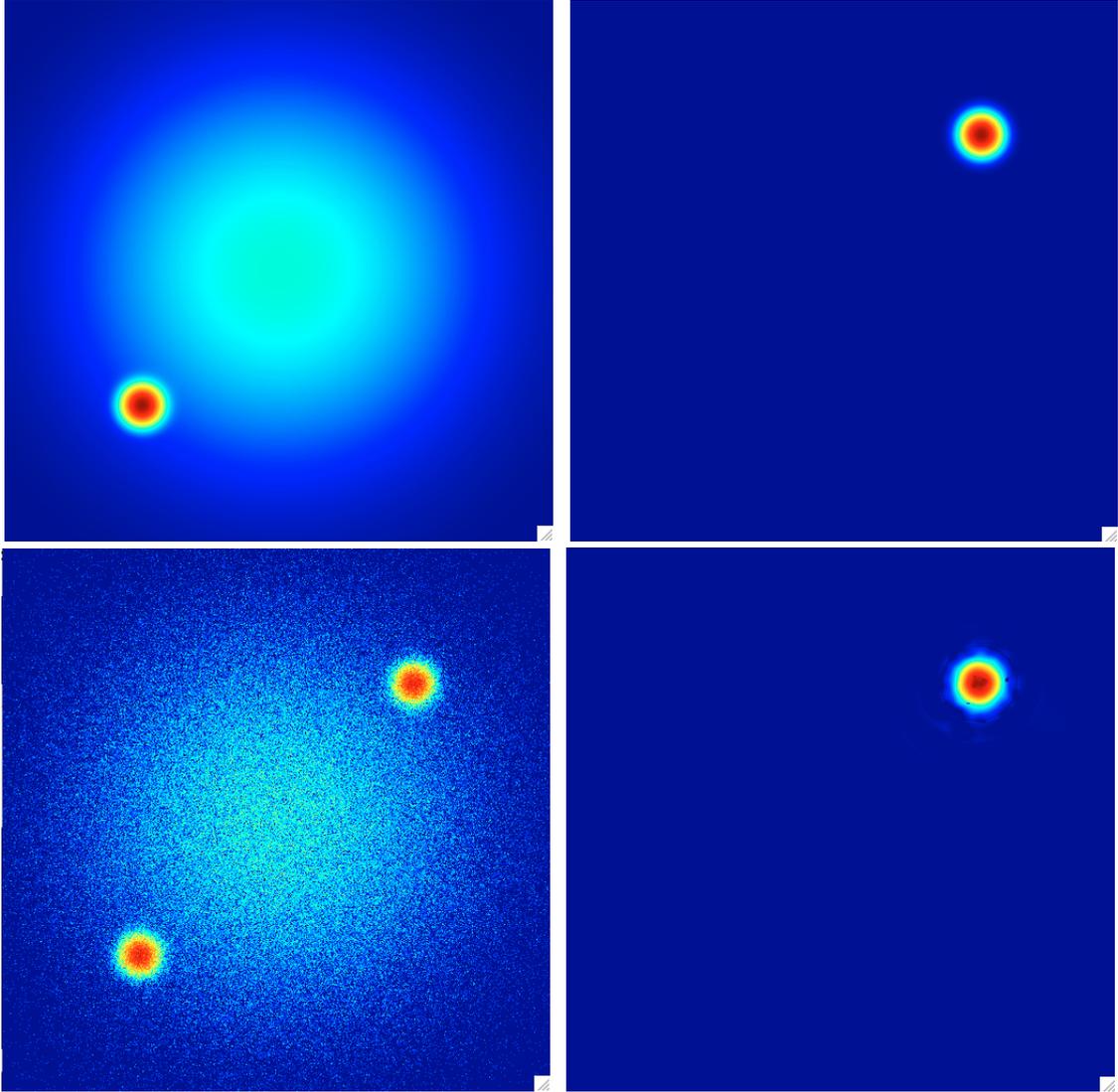
$$X^{(n+1)} = \Phi \text{ST}_{\lambda_n}[\Phi \tilde{X}]. \quad (27)$$

The algorithm is illustrated by the theoretical study in Fig. 6. We denoise Poisson data while separating a single source, which is a Gaussian of standard deviation equal to 0.01, from a background, which is a sum of two Gaussians of standard deviation equal to 0.1 and 0.01 respectively.

Like Algorithm 1, Algorithm 5 can be adapted to make multi-resolution support adaptation.

### 6.2. Experiment

We applied Algorithms 5 on simulated *Fermi* data. To test the efficiency of our method, we detect the sources with the SExtractor routine (Bertin & Arnouts 1996), and compare the detected sources with the theoretical sources catalog to get the number of true and false detections. Results are shown in Figs. 7 and 8. The SExtractor method was applied on the first wavelet scale of the reconstructed map, with a detection threshold equal to 1.



**Fig. 6.** Theoretical testing for MS-VSTS + IUWT denoising + background removal algorithm (Algorithm 5). View on a single HEALPix face. *Top left:* simulated background: sum of two Gaussians of standard deviation equal to 0.1 and 0.01 respectively. *Top right:* simulated source: Gaussian of standard deviation equal to 0.01. *Bottom left:* simulated poisson data. *Bottom right:* image denoised with MS-VSTS + IUWT and background removal.

---

**Algorithm 5** MS-VSTS + IUWT Denoising + Background extraction

---

**Require:** data  $a_0 := Y$ , background  $B$ , number of iterations  $N_{\max}$ , threshold  $\kappa$ .

**Detection**

- 1: **for**  $j = 1$  to  $J$  **do**
- 2:   Compute  $a_j$  and  $d_j$  using (7).
- 3:   Hard threshold  $(d_j[k] - d_j^{(b)}[k])$  with threshold  $\kappa\sigma_j$  and update  $\mathcal{M}$ .
- 4: **end for**

**Estimation**

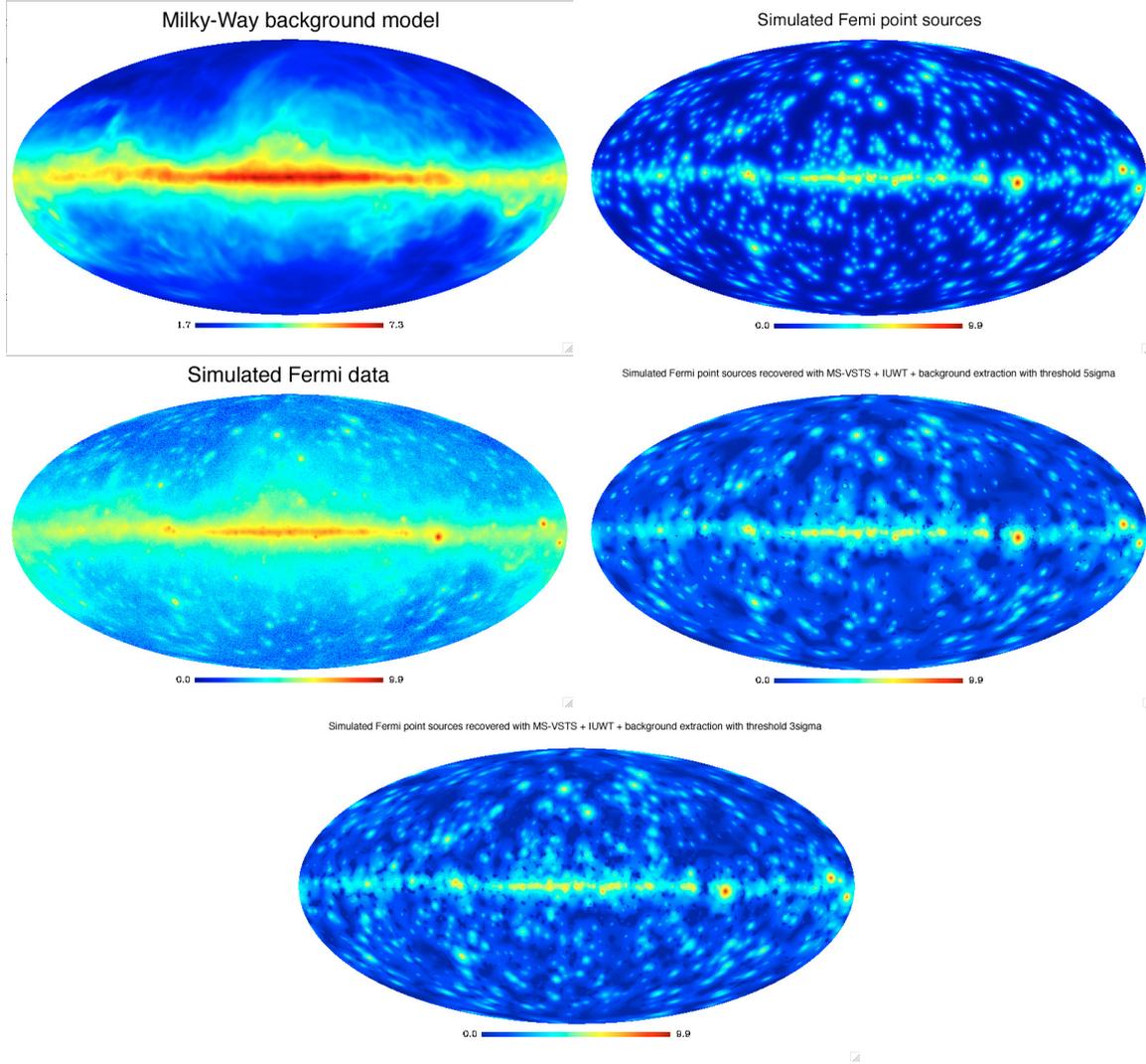
- 5: Initialize  $\tilde{X}^{(0)} = 0$ ,  $\lambda_0 = 1$ .
  - 6: **for**  $n = 0$  to  $N_{\max} - 1$  **do**
  - 7:    $\tilde{X} = P_+[X^{(n)} + \Phi P_{\mathcal{M}} \Phi^T (Y - B - X^{(n)})]$ .
  - 8:    $X^{(n+1)} = \Phi \text{ST}_{\lambda_n}[\Phi^T \tilde{X}]$ .
  - 9:    $\lambda_{n+1} = \frac{N_{\max} - (n+1)}{N_{\max} - 1}$ .
  - 10: **end for**
  - 11: Get the estimate  $\hat{\Lambda} = X^{(N_{\max})}$ .
- 

It has been chosen to optimise the number of true detections. SExtractor makes 593 true detections and 71 false detections on

the *Fermi* simulated map restored with Algorithm 2 among the 1000 sources of the simulation. On noisy data, many fluctuations due to Poisson noise are detected as sources by SExtractor, which leads to a big number of false detections (more than 2000 in the case of *Fermi* data).

### 6.2.1. Sensitivity to model errors

As it is difficult to model the background precisely, it is important to study the sensitivity of the method to model errors. We add a stationary Gaussian noise to the background model, we compute the MS-VSTS + IUWT with threshold  $3\sigma_j$  on the simulated *Fermi* Poisson data with extraction of the noisy background, and we study the percent of true and false detections with respect to the total number of sources of the simulation and the signal-noise ratio ( $SNR(\text{dB}) = 20 \log(\sigma_{\text{signal}}/\sigma_{\text{noise}})$ ) versus the standard deviation of the Gaussian perturbation. Table 3 shows that, when the standard deviation of the noise on the background model becomes of the same range as the mean of



**Fig. 7.** *Top left:* simulated background model. *Top right:* simulated gamma ray sources. *Middle left:* simulated *Fermi* data with Poisson noise. *Middle right:* reconstructed gamma ray sources with MS-VSTS + IUWT + background removal (Algorithm 5) with threshold  $5\sigma_j$ . *Bottom:* reconstructed gamma ray sources with MS-VSTS + IUWT + background removal (Algorithm 5) with threshold  $3\sigma_j$ . Pictures are in logarithmic scale.

the Poisson intensity distribution ( $\lambda_{\text{mean}} = 68.764$ ), the number of false detections increases, the number of true detections decreases and the signal noise ratio decreases. While the perturbation is not too strong (standard deviation  $<10$ ), the effect of the model error remains low.

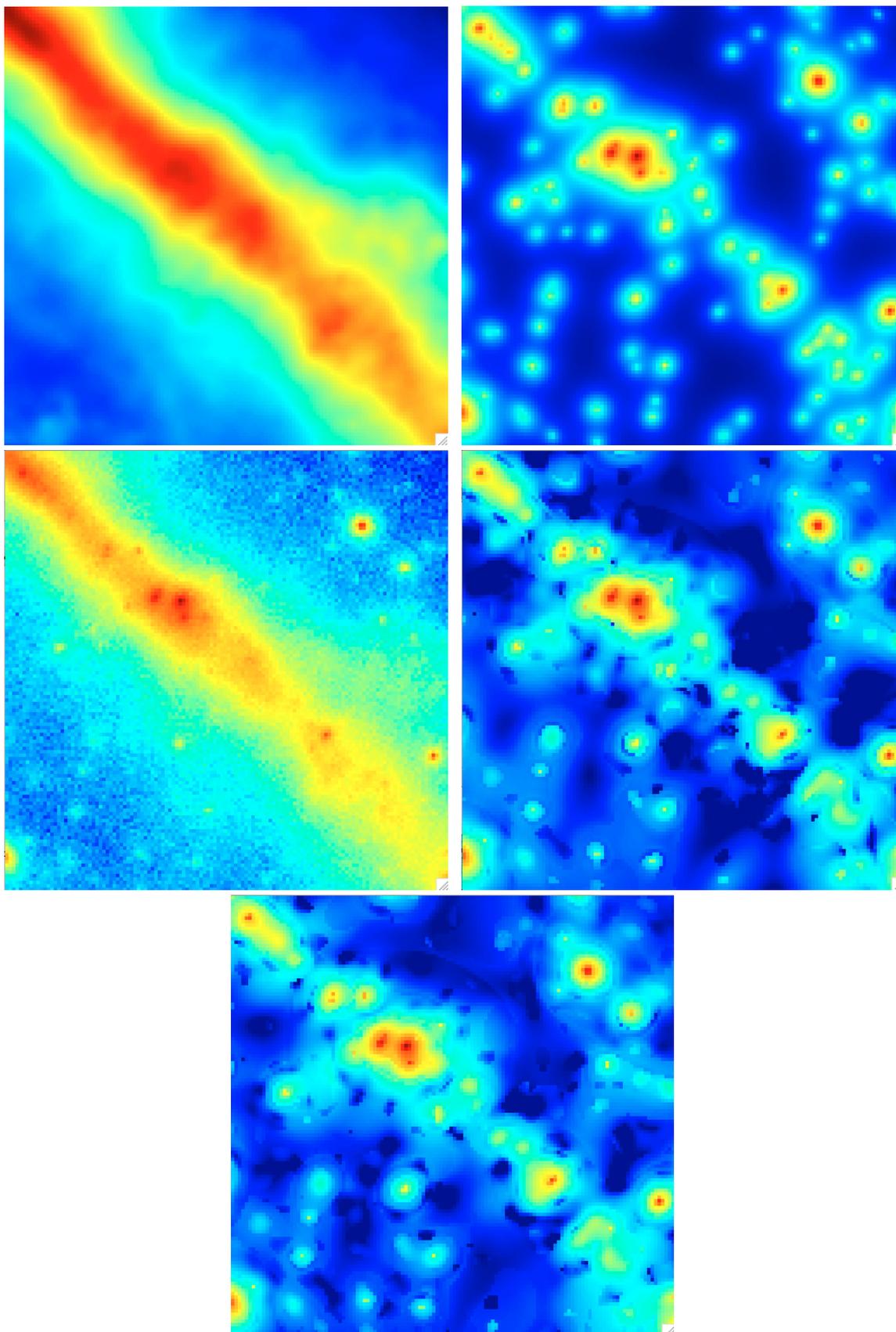
## 7. Conclusion

This paper presented new methods for restoration of spherical data with noise following a Poisson distribution. A denoising method was proposed, which used a variance stabilization method and multiscale transforms on the sphere. Experiments have shown it is very efficient for *Fermi* data denoising. Two spherical multiscale transforms, the wavelet and the curvelets, were used. Then, we have proposed an extension of the denoising method in order to take into account missing data, and we have shown that this inpainting method could be a useful tool to estimate the diffuse emission. Finally, we have introduced a new denoising method the sphere which takes into account a background model. The simulated data have shown

**Table 3.** Percent of true and false detection and signal-noise ratio versus the standard deviation of the Gaussian noise on the background model.

Model error std dev	% of true detect	% of false detect	SNR (dB)
0	59.3%	7.1%	23.8
10	57.0%	11.0%	23.2
20	53.2%	18.9%	22.6
30	49.1%	43.5%	21.7
40	42.3%	44.3%	21.0
50	34.9%	39.0%	20.3
60	30.3%	37.5%	19.5
70	25.0%	34.6%	18.9
80	23.0%	28.5%	18.7
90	23.6%	27.1%	18.3

that it is relatively robust to errors in the model, and can therefore be used for *Fermi* diffuse background modeling and source detection.



**Fig. 8.** View of a single HEALPix face from the results of Fig. 7. *Top left:* simulated background model. *Top right:* simulated gamma ray sources. *Middle left:* simulated *Fermi* data with Poisson noise. *Middle right:* reconstructed gamma ray sources with MS-VSTS + IUWT + background removal (Algorithm 5) with threshold  $5\sigma_j$ . *Bottom:* reconstructed gamma ray sources with MS-VSTS + IUWT + background removal (Algorithm 5) with threshold  $3\sigma_j$ . Pictures are in logarithmic scale.

*Acknowledgements.* This work was partially supported by the European Research Council grant ERC-228261.

## References

- Abdo, A. A., Ackermann, M., Ajello, M., et al. 2009, *ApJS*, 183, 46
- Abrial, P., Moudden, Y., Starck, J., et al. 2007, *Journal of Fourier Analysis and Applications*, 13, 729
- Abrial, P., Moudden, Y., Starck, J., et al. 2008, *Statistical Methodology*, 5, 289
- Anscombe, F. 1948, *Biometrika*, 35, 246
- Atwood, W. B., Abdo, A. A., Ackermann, M., et al. 2009, *ApJ*, 697, 1071
- Bertin, E., & Arnouts, S. 1996, *A&AS*, 117, 393
- Candes, E., & Donoho, D. 1999, *Phil. Trans. R. Soc. Lond. A*, 357, 2495
- Donoho, D. 1993, *Proc. Symp. Appl. Math.*, 47, 173
- Fisz, M. 1955, *Colloq. Math.*, 3, 138
- Fryźlewicz, P., & Nason, G. 2004, *J. Comp. Graph. Stat.*, 13, 621
- Górski, K. M., Hivon, E., Banday, A. J., et al. 2005, *ApJ*, 622, 759
- Hartman, R. C., Bertsch, D. L., Bloom, S. D., et al. 1999, *VizieR Online Data Catalog*, 212, 30079
- Kolaczyk, E. 1999, *J. Amer. Stat. Assoc.*, 94, 920
- Lefkimmiaits, S., Maragos, P., & Papandreou, G. 2009, *IEEE Transactions on Image Processing*, 20, 20
- Myers, J. 2009, *LAT Background Models*, <http://fermi.gsfc.nasa.gov/ssc/data/access/lat/BackgroundModels.html>
- Myers, J. 2010, *LAT 1-year Point Source Catalog*, [http://fermi.gsfc.nasa.gov/ssc/data/access/lat/1yr\\_catalog/](http://fermi.gsfc.nasa.gov/ssc/data/access/lat/1yr_catalog/)
- Nowak, R., & Kolaczyk, E. 2000, *IEEE Trans. Inf. Theory*, 45, 1811
- Starck, J., Fadili, J. M., Digel, S., Zhang, B., & Chiang, J. 2009, *A&A*, 504, 641
- Starck, J.-L., Elad, M., & Donoho, D. 2004, *Advances in Imaging and Electron Physics*, 132
- Starck, J.-L., Moudden, Y., Abrial, P., & Nguyen, M. 2006, *A&A*, 446, 1191
- Timmerman, K., & Nowak, R. 1999, *IEEE Trans. Inf. Theory*, 45, 846
- Yamada, I. 2001, in *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications* (Elsevier), 473
- Zhang, B., Fadili, J., & Starck, J.-L. 2008, *IEEE Transactions on Image Processing*, 11, 1093