



Poisson Noise Removal in Spherical Multichannel Images: Application to Fermi data

Jeremy Schmitt, Jean-Luc Starck, Jalal M. Fadili, Seth Digel

► **To cite this version:**

Jeremy Schmitt, Jean-Luc Starck, Jalal M. Fadili, Seth Digel. Poisson Noise Removal in Spherical Multichannel Images: Application to Fermi data. Way, Michael J. and Scargle, Jeffrey D. and Ali, Kamal M. and Srivastava, Ashok N. Advances in Machine Learning and Data Mining for Astronomy, Chapman and Hall, pp.183-212, 2012, 9781439841730. <10.1201/b11822-14>. <hal-00812518>

HAL Id: hal-00812518

<https://hal.archives-ouvertes.fr/hal-00812518>

Submitted on 17 Mar 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Author Name

Data Mining and Knowledge Discovery

Contents

I	Data Mining and Machine Learning for Astronomical Applications	1
1	Poisson Noise Removal in Spherical Multichannel Images : Application to Fermi data	3
	<i>Jérémy Schmitt, Jean-Luc Starck, Jalal Fadili, and Seth Digel</i>	
1.1	Introduction	4
1.2	Wavelets and Curvelets on the Sphere	6
1.2.1	The HEALPix Pixellisation for Spherical Data	6
1.2.2	Isotropic Undecimated Wavelet Transform on the sphere	6
1.2.3	Curvelet Transform on the sphere	7
1.2.4	Application to Gaussian denoising on the sphere	7
1.3	Multi-Scale Transforms on the Sphere and Poisson Noise	10
1.3.1	Principle of the Multi-Scale Variance Stabilizing Transform on the Sphere (MS-VSTS)	10
1.3.2	Wavelets and Poisson Noise	11
1.3.3	Curvelets and Poisson Noise	15
1.4	Application to Poisson Denoising on the Sphere	15
1.4.1	MS-VSTS + IUWT	15
1.4.2	Multi-resolution support adaptation	17
1.4.3	MS-VSTS + Curvelets	17
1.4.4	Experiments	19
1.5	Application to Inpainting and Source Extraction	19
1.5.1	Milky Way diffuse background study: denoising and inpainting	19
1.5.2	Source detection: denoising and background modeling	23
1.5.2.1	Method	23
1.5.2.2	Experiment	24
1.6	Extension to multichannel data	28
1.6.1	Gaussian Noise	28
1.6.1.1	2D-1D Wavelet Transform on the Sphere	28
1.6.1.2	Fast Undecimated 2D-1D decomposition/reconstruction	29
1.6.1.3	Multichannel Gaussian denoising	30
1.6.2	Poisson Noise	31

1.6.2.1	Multi-Scale Variance Stabilizing Transform	31
1.6.2.2	Detection-Reconstruction	31
1.7	Conclusion	33
	Bibliography	35

Part I

Data Mining and Machine
Learning for Astronomical
Applications



Chapter 1

Poisson Noise Removal in Spherical Multichannel Images : Application to Fermi data

Jérémy Schmitt

Laboratoire AIM, CEA/DSM-CNRS-Université Paris Diderot, IRFU/SEDI-SAP, CEA Saclay, Orme des Merisiers, 91191 Gif-sur-Yvette, France

Jean-Luc Starck

Laboratoire AIM, CEA/DSM-CNRS-Université Paris Diderot, IRFU/SEDI-SAP, CEA Saclay, Orme des Merisiers, 91191 Gif-sur-Yvette, France

Jalal Fadili

GREYC CNRS-ENSICAEN-Université de Caen, 6, Bd du Maréchal Juin, 14050 Caen Cedex, France

Seth Digel

Kavli Institute for Particle Astrophysics and Cosmology, SLAC National Accelerator Laboratory, Menlo Park, CA 94075, USA

1.1	Introduction	4
1.2	Wavelets and Curvelets on the Sphere	5
1.2.1	The HEALPix Pixelisation for Spherical Data	6
1.2.2	Isotropic Undecimated Wavelet Transform on the sphere	6
1.2.3	Curvelet Transform on the sphere	7
1.2.4	Application to Gaussian denoising on the sphere	7
1.3	Multi-Scale Transforms on the Sphere and Poisson Noise	10
1.3.1	Principle of the Multi-Scale Variance Stabilizing Transform on the Sphere (MS-VSTS)	10
1.3.2	Wavelets and Poisson Noise	11
1.3.3	Curvelets and Poisson Noise	12
1.4	Application to Poisson Denoising on the Sphere	15
1.4.1	MS-VSTS + IUWT	15
1.4.2	Multi-resolution support adaptation	16
1.4.3	MS-VSTS + Curvelets	17
1.4.4	Experiments	17
1.5	Application to Inpainting and Source Extraction	19
1.5.1	Milky Way diffuse background study: denoising and inpainting . Experiment	19 22
1.5.2	Source detection: denoising and background modeling	23
1.5.2.1	Method	23
1.5.2.2	Experiment	24
1.6	Extension to multichannel data	28

1.6.1	Gaussian Noise	28
1.6.1.1	2D-1D Wavelet Transform on the Sphere	28
1.6.1.2	Fast Undecimated 2D-1D decomposition/reconstruction	29
1.6.1.3	Multichannel Gaussian denoising	30
1.6.2	Poisson Noise	30
1.6.2.1	Multi-Scale Variance Stabilizing Transform	31
1.6.2.2	Detection-Reconstruction	31
1.7	Conclusion	32
	Acknowledgement	33

1.1 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 [5]. 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 point sources, which is an order of magnitude more than its predecessor EGRET [13].

Even with its relatively large acceptance ($\sim 2 \text{ m}^2 \text{ sr}$), the number of photons detected by the LAT outside the Galactic plane and away from intense sources is relatively low and the sky overall has a diffuse glow from cosmic-ray interactions with interstellar gas and low-energy photons that makes a background against which point sources need to be detected. In addition, the per-photon angular resolution of the LAT is relatively poor and strongly energy dependent, ranging from more than 10° at 20 MeV to $\sim 0.1^\circ$ above 100 GeV. 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 statistical noise makes the source detection more difficult, consequently it is highly desirable 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 [19], independently initiated by [23] and [15]. [16] proposed an improved bayesian framework for analyzing Poisson processes, based on a multiscale 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 [4] and the Fisz [10] transforms, applied respectively in the spatial [8] or in the wavelet domain [11]. 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. [25] 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 [21], 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.

To denoise Fermi maps, we need a 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, two of the main scientific objectives are in a sense complementary:

- Detection of point sources to build the catalog of gamma ray sources,
- Study of the Milky Way diffuse background.

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 chapter is to present a multi-scale representation for spherical data with Poisson noise called Multi-Scale Variance Stabilizing Transform on the Sphere (MS-VSTS) [14], combining the MS-VST [25] with various multi-scale transforms on the sphere (wavelets and curvelets) [22, 2, 3]. Section 1.2 presents some multi-scale transforms on the sphere. Section 1.3 introduces a new multi-scale representation for data with Poisson noise called MS-VSTS. Section 1.4 applies this representation to Poisson noise removal on Fermi data. Section 1.5 presents applications to missing data interpolation and source extraction. Section 1.6 extends the method to multichannel data.

All experiments were performed on HEALPix maps with $n_{side} = 128$ [12], which corresponds to a good pixelisation choice for the GLAST/FERMI resolution.

1.2 Wavelets and Curvelets on the Sphere

New multi-scale transforms on the sphere were developed by [22]. These transforms can be inverted and are easy to compute with the HEALPix pixellisation, and were used for denoising, deconvolution, morphological component analysis and inpainting applications [2]. In this chapter, here we use the Isotropic Undecimated Wavelet Transform (IUWT) and the Curvelet Transform.

1.2.1 The HEALPix Pixellisation for Spherical Data

Different kinds of pixellization scheme exist for data on the sphere. For Fermi data, we use the HEALPix representation (Hierarchical Equal Area isoLatitude Pixellization of a sphere) [12], a curvilinear hierarchical partition of the sphere into quadrilateral pixels of exactly equal area but with varying shape. The base resolution divides the sphere into 12 quadrilateral faces of equal area placed on three rings around the poles and equator. Each face is subsequently divided into n_{side}^2 pixels following a quadrilateral multiscale tree structure. (Fig. 1.1) The pixel centers are located on iso-latitude rings, and pixels from the same ring are equispaced in azimuth. This is critical for computational speed of all operations involving the evaluation of spherical harmonic transforms, including standard numerical analysis operations such as convolution, power spectrum estimation, etc. HEALPix is a standard pixellization scheme in astronomy.

1.2.2 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 θ denoting the longitude and φ 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:

$$a_0(\theta, \varphi) = a_J(\theta, \varphi) + \sum_{j=1}^J d_j(\theta, \varphi) \quad (1.1)$$

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).

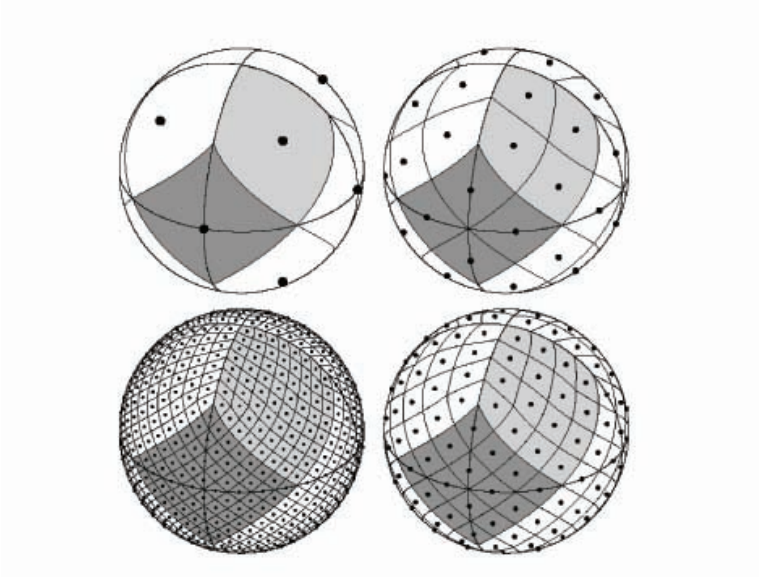


Figure 1.1: The HEALPix sampling grid for four different resolutions.

A reconstruction algorithm based on a variety of filter banks is described in [22]. Figure 1.2 shows the result of the IUWT on WMAP data (Cosmic Microwave Background).

1.2.3 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 pixellisation. Finally, the ridgelet transform [7] is applied on each individual block. The method is best for the detection of anisotropic structures and smooth curves and edges of different lengths. The principle of the curvelet transform is schematized on Figure 1.3. More details can be found in [22].

1.2.4 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,

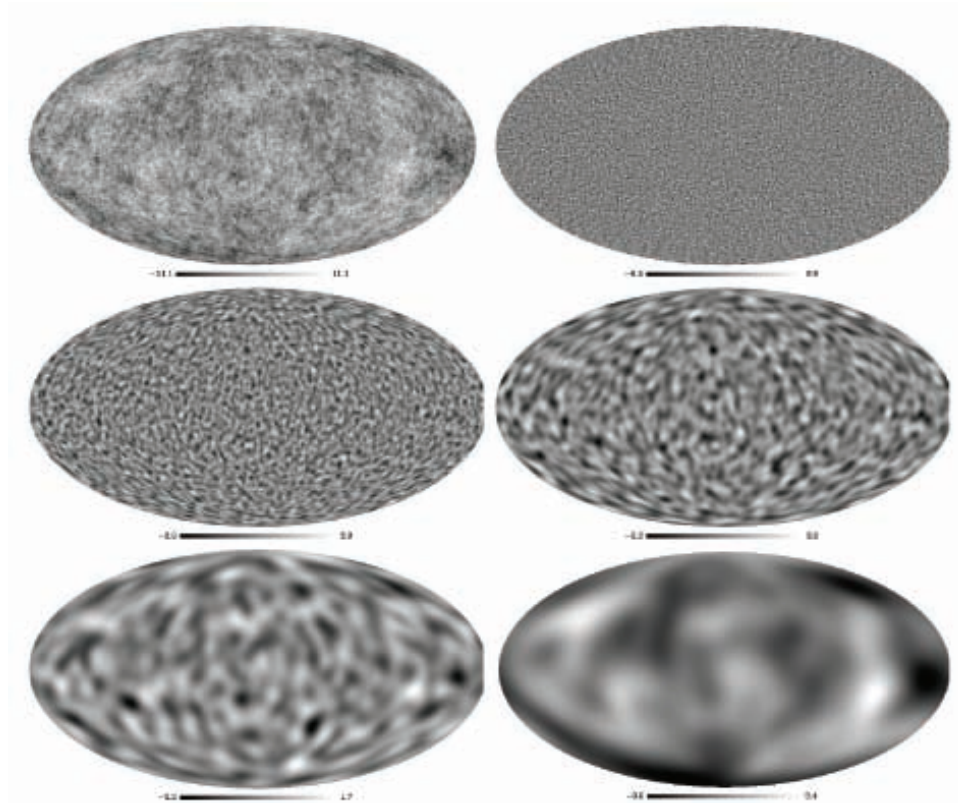


Figure 1.2: WMAP data and its wavelet transform on the sphere using five resolution levels (4 wavelet scales and the coarse scale). The sum of these five maps reproduces exactly the original data (top left). Top: original data and the first wavelet scale. Middle: the second and third wavelet scales. Bottom: the fourth wavelet scale and the last smoothed array.

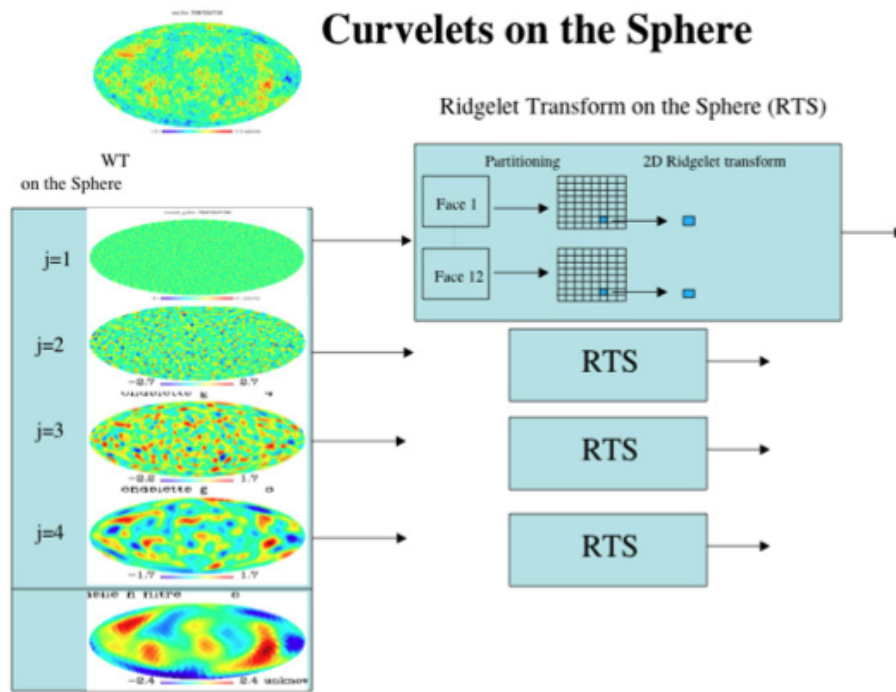


Figure 1.3: Principle of curvelets transform on the sphere.

we need to estimate the noise standard deviation σ_j in each band j and a coefficient w_j is significant if $|w_j| > \kappa\sigma_j$, where κ is a parameter typically chosen between 3 and 5. Denoting \mathbf{Y} the noisy data and HT_λ the thresholding operator, the filtered data \mathbf{X} are obtained by:

$$\mathbf{X} = \Phi HT_\lambda(\Phi^T \mathbf{Y}), \quad (1.2)$$

where Φ^T is the multiscale transform (IUWT or curvelet) and Φ is the multiscale reconstruction. λ 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 $\lambda_j = \kappa\sigma_j$.

1.3 Multi-Scale Transforms on the Sphere and Poisson Noise

1.3.1 Principle of the Multi-Scale Variance Stabilizing Transform on the Sphere (MS-VSTS)

In this section, we propose a multi-scale representation designed for data with Poisson noise. The idea is to combine the spherical multi-scale transforms with a variance stabilizing transform (VST), in order to have a multi-scale representation of the data where the noise on multi-scale coefficients behaves like Gaussian noise of known variance. With this representation, it is easy to denoise the data using standard Gaussian denoising methods.

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 \mathbf{T} is to stabilize the data such that each coefficient of $\mathbf{T}(\mathbf{Y})$ has an (asymptotically) constant variance, say 1, irrespective of the value of λ_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 [4] transform is a widely used VST which has a simple square-root form

$$\mathbf{T}(Y) := 2\sqrt{Y + 3/8}. \quad (1.3)$$

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

$$\mathbf{T}(Y) - 2\sqrt{\lambda} \xrightarrow[\lambda \rightarrow +\infty]{\mathcal{D}} \mathcal{N}(0, 1) \quad (1.4)$$

It can be shown that the Anscombe VST requires a high underlying intensity to well stabilize the data (typically for $\lambda \geq 10$) [25].

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:

$$T(Z) := b \cdot \text{sign}(Z + c)|Z + c|^{1/2}, \tag{1.5}$$

where b is a normalizing factor. It is proven in [25] 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 λ becomes large.

The Multi-Scale Variance Stabilizing Transform on the Sphere (MS-VSTS) consists in combining the square-root VST with a spherical multi-scale transform (wavelets, curvelets...).

1.3.2 Wavelets and Poisson Noise

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

$$\begin{aligned} \text{IUWT} \begin{cases} a_j &= h_{j-1} * a_{j-1} \\ d_j &= a_{j-1} - a_j \end{cases} \\ \implies \text{MS-VSTS} \begin{cases} a_j &= h_{j-1} * a_{j-1} \\ d_j &= T_{j-1}(a_{j-1}) - T_j(a_j) \end{cases} \end{aligned} \tag{1.6}$$

In (1.6), 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 δ is the Dirac pulse ($\delta = 1$ on a single pixel and 0 everywhere else). T_j is the VST operator at scale j :

$$T_j(a_j) = b^{(j)} \text{sign}(a_j + c^{(j)}) \sqrt{|a_j + c^{(j)}|}. \tag{1.7}$$

Let us define $\tau_k^{(j)} := \sum_i (h^{(j)}[i])^k$. In [25], 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:

$$c^{(j)} := \frac{7\tau_2^{(j)}}{8\tau_1^{(j)}} - \frac{\tau_3^{(j)}}{2\tau_2^{(j)}}. \tag{1.8}$$

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

$$a_0(\theta, \varphi) = T_0^{-1} \left[T_J(a_J) + \sum_{j=1}^J d_j \right] (\theta, \varphi). \tag{1.9}$$

Setting $b^{(j)} := \text{sign}(\tau_1^{(j)})/\sqrt{|\tau_1^{(j)}|}$, if λ is constant within the support of the

filter. $h^{(j)}$, then we have [25]:

$$d_j(\theta, \varphi) \xrightarrow[\lambda \rightarrow +\infty]{\mathcal{D}} \mathcal{N} \left(0, \frac{\tau_2^{(j-1)}}{4\tau_1^{(j-1)^2}} + \frac{\tau_2^{(j)}}{4\tau_1^{(j)^2}} - \frac{\langle h^{(j-1)}, h^{(j)} \rangle}{2\tau_1^{(j-1)}\tau_1^{(j)}} \right), \quad (1.10)$$

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

This means that the detail coefficients issued from locally homogeneous parts of the signal follow asymptotically a central normal distribution with an intensity-independent variance which relies solely on the filter h and the current scale for a given filter h . Let us define $\sigma_{(j)}^2$ the stabilized variance at scale j for a locally homogeneous part of the signal:

$$\sigma_{(j)}^2 = \frac{\tau_2^{(j-1)}}{4\tau_1^{(j-1)^2}} + \frac{\tau_2^{(j)}}{4\tau_1^{(j)^2}} - \frac{\langle h^{(j-1)}, h^{(j)} \rangle}{2\tau_1^{(j-1)}\tau_1^{(j)}}. \quad (1.11)$$

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 1.1).

TABLE 1.1: Precomputed values of the variances σ_j of the wavelet coefficients.

Wavelet scale j	Value of σ_j
1	0.484704
2	0.0552595
3	0.0236458
4	0.0114056
5	0.00567026

We have simulated Poisson images of different constant intensities λ , 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 λ for each scale j (Fig. 1.4). We see that the wavelet coefficients are stabilized when $\lambda \gtrsim 0.1$ except for the first wavelet scale, which is largely noise. In Fig. 1.5, we compare the result of MS-VSTS with Anscombe + wavelet shrinkage, on sources of varying intensities. We see that MS-VSTS works well on sources of very low intensities, whereas Anscombe does not work when the intensity is too low.

Poisson Noise Removal in Spherical Multichannel Images : Application to Fermi data 13

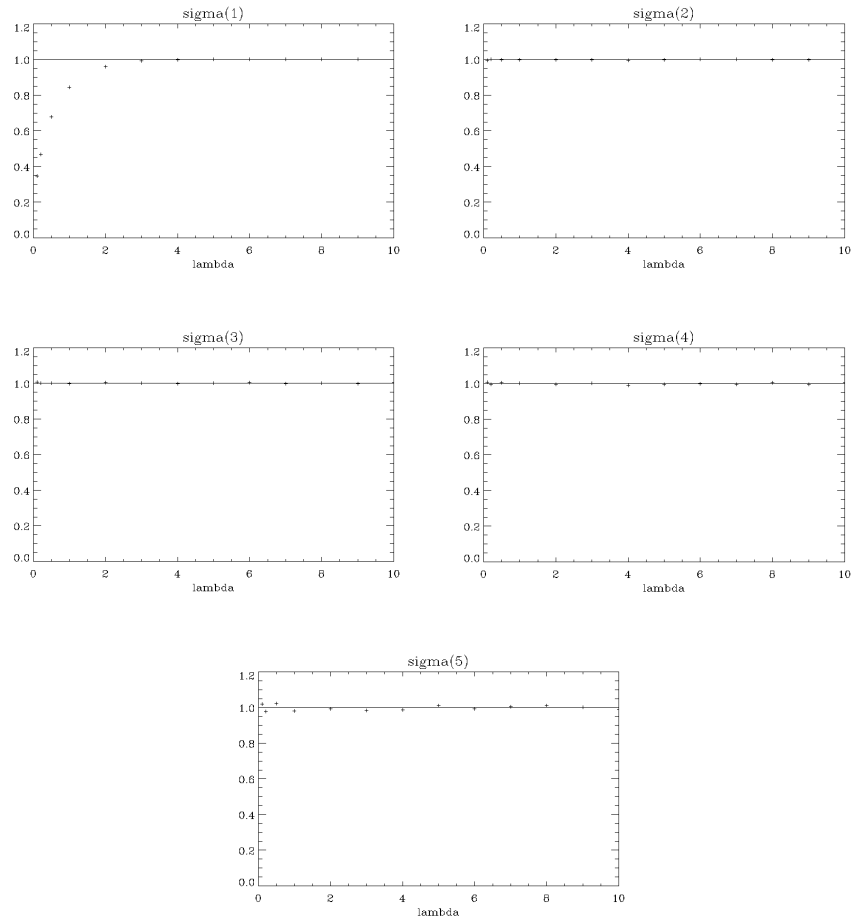


Figure 1.4: Normalized value $((\sigma_j)_{\text{simulated}}/(\sigma_j)_{\text{theoretical}})$ of the stabilized variances at each scale j as a function of λ .

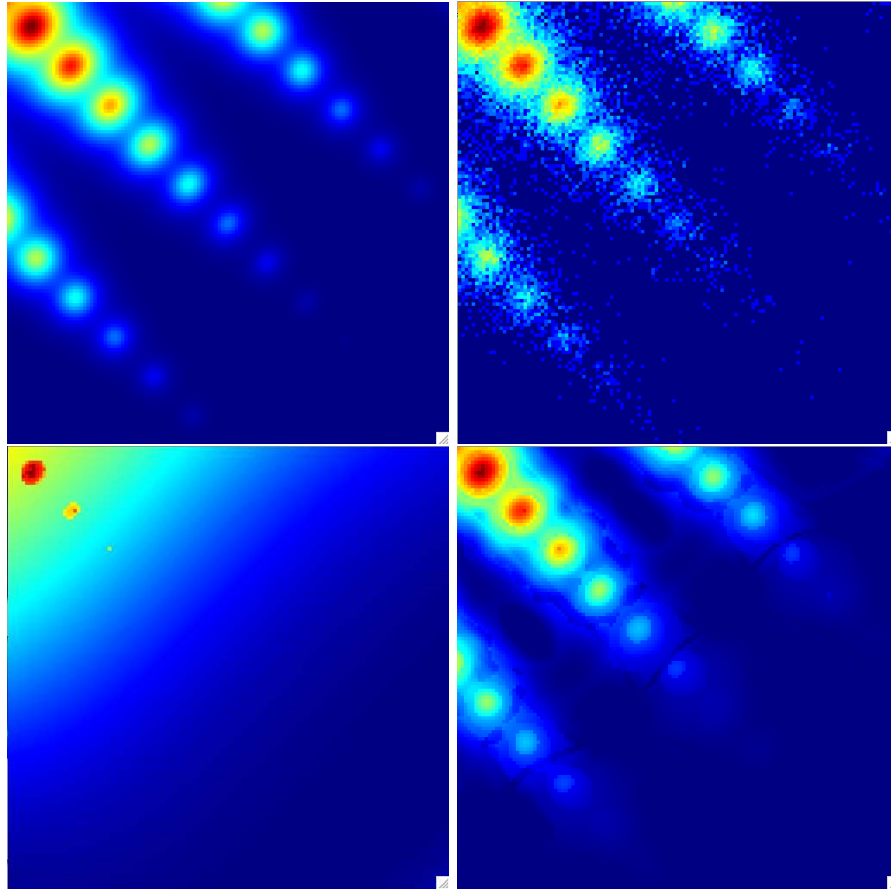


Figure 1.5: Comparison of MS-VSTS with Anscombe + wavelet shrinkage on a single face of the first scale of the HEALPix pixelization (angular extent: $\pi/3sr$). Top Left : Sources of varying intensity. Top Right : Sources of varying intensity with Poisson noise. Bottom Left : Poisson sources of varying intensity reconstructed with Anscombe + wavelet shrinkage. Bottom Right : Poisson sources of varying intensity reconstructed with MS-VSTS.

1.3.3 Curvelets and Poisson Noise

As the first step of the algorithm is an IUWT, we can stabilize each resolution level as in Equation 1.6. 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 λ . 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 1.2).

TABLE 1.2: Asymptotic values of the variances $\sigma_{j,k}$ 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

1.4 Application to Poisson Denoising on the Sphere

1.4.1 MS-VSTS + 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:

$$p(d_j[k]) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp(-d_j[k]^2/2\sigma_j^2). \quad (1.12)$$

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

$$p_j[k] = 2 \frac{1}{\sqrt{2\pi}\sigma_j} \int_{|d_j[k]|}^{+\infty} \exp(-x^2/2\sigma_j^2) dx. \quad (1.13)$$

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. (??)), 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) [24].

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

$$\mathcal{M} := \{(j, k) | d_j[k] \text{ is declared significant}\}. \quad (1.14)$$

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

$$\begin{aligned} & \underset{\mathbf{X}}{\text{Arg min}} \|\Phi^T \mathbf{X}\|_1, \text{ s.t.} \\ & \begin{cases} \mathbf{X} \geq 0, \\ \forall (j, k) \in \mathcal{M}, (\Phi^T \mathbf{X})_j[k] = (\Phi^T \mathbf{Y})_j[k], \end{cases} \end{aligned} \quad (1.15)$$

where Φ denotes the IUWT synthesis operator.

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

$$\tilde{\mathbf{X}} = P_+[\mathbf{X}^{(n)} + \Phi P_{\mathcal{M}} \Phi^T (\mathbf{Y} - \mathbf{X}^{(n)})] \quad (1.16)$$

$$\mathbf{X}^{(n+1)} = \Phi \text{ST}_{\lambda_n} [\Phi^T \tilde{\mathbf{X}}] \quad (1.17)$$

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

$$P_{\mathcal{M}} d_j[k] = \begin{cases} d_j[k] & \text{if } (j, k) \in \mathcal{M}, \\ 0 & \text{otherwise} \end{cases}. \quad (1.18)$$

and ST_{λ_n} the soft-thresholding with threshold λ_n :

$$\text{ST}_{\lambda_n}[d] = \begin{cases} \text{sign}(d)(|d| - \lambda_n) & \text{if } |d| \geq \lambda_n, \\ 0 & \text{otherwise} \end{cases}. \quad (1.19)$$

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: $\hat{\mathbf{A}} = \mathbf{X}^{(N_{\max})}$. Algorithm 1 summarizes the main steps of the MS-VSTS + IUWT denoising algorithm.

Algorithm 1 MS-VSTS + IUWT Denoising

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

Detection

- 1: **for** $j = 1$ to J **do**
- 2: Compute a_j and d_j using (1.6).
- 3: Hard threshold $|d_j[k]|$ with threshold $\kappa\sigma_j$ and update \mathcal{M} .
- 4: **end for**

Estimation

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

1.4.2 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 $\mathbf{X}^{(n)}$. We denote $d_j^{(n)}[k]$ the stabilised coefficients of $\mathbf{X}^{(n)}$. 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.

1.4.3 MS-VSTS + Curvelets

Insignificant coefficients are zeroed by using the same hypothesis testing framework as in the wavelet scale. At each wavelet 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 (1.9)). An iterative reconstruction may also be performed.

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

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

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

Detection

- 1: **for** $j = 1$ to J **do**
- 2: Compute a_j and d_j using (1.6).
- 3: Hard threshold $|d_j[k]|$ with threshold $\kappa\sigma_j$ and update \mathcal{M} .
- 4: **end for**

Estimation

- 5: Initialize $\mathbf{X}^{(0)} = 0$, $\lambda_0 = 1$.
 - 6: **for** $n = 0$ to $N_{\max} - 1$ **do**
 - 7: $\tilde{\mathbf{X}} = P_+[\mathbf{X}^{(n)} + \Phi P_{\mathcal{M}} \Phi^T (\mathbf{Y} - \mathbf{X}^{(n)})]$.
 - 8: $\mathbf{X}^{(n+1)} = \Phi \text{ST}_{\lambda_n}[\Phi^T \tilde{\mathbf{X}}]$.
 - 9: Compute the MS-VSTS on $\mathbf{X}^{(n)}$ 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 $\hat{\Lambda} = \mathbf{X}^{(N_{\max})}$.
-

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 curvelet 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.
-

1.4.4 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 [17]. 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 with energy approximately as $0.8(E/1\text{GeV})^{-0.8}$ degrees (68% containment angle). The positions of the 205 brightest sources were taken from the Fermi 3-month source list [1]. The positions of the 795 remaining sources follow the LAT 1-year Point Source Catalog [18] source distribution: each simulated source was randomly sorted in a box in Galactic coordinates 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 $3.10^{10} s.cm^2$ corresponding approximatively to one year of Fermi all-sky survey around 1 GeV. The simulated counts map shown in this section correspond to photons energy from 150 MeV to 20 GeV.

Fig. 1.6 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. Fig. 1.7 shows the results on one single face of the first scale of the HEALPix pixelization (angular extent: $\pi/3sr$). 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 ($SNR_{wavelets} = 65.8dB$, $SNR_{curvelets} = 37.3dB$, $SNR(dB) = 20 \log(\sigma_{signal}/\sigma_{noise})$). As this image contains many point sources, this result is expected. Indeed wavelets are better than curvelets to represent isotropic objects.

1.5 Application to Inpainting and Source Extraction

1.5.1 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 (Morphological Component Analysis) algorithm [20], 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-

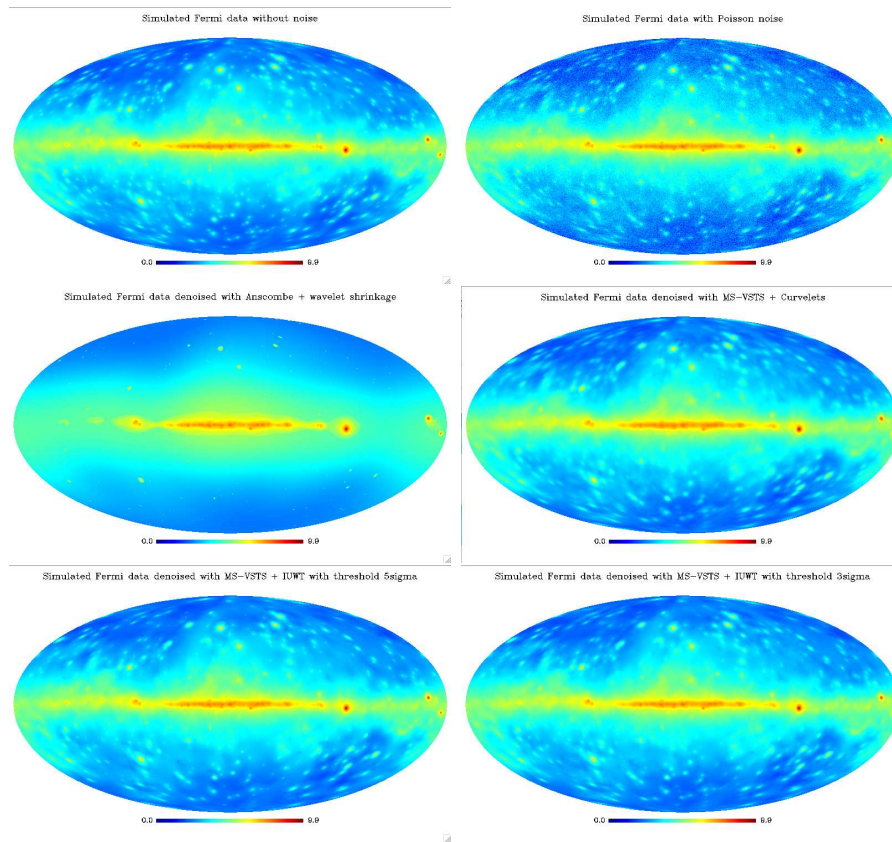


Figure 1.6: Top Left: Fermi simulated map without noise. Top Right: Fermi simulated map with Poisson noise. Middle Left: Fermi simulated map denoised with Anscombe VST + wavelet shrinkage. Middle Right: Fermi simulated map denoised with MS-VSTS + curvelets (Algorithm 3). Bottom Left: Fermi simulated 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.

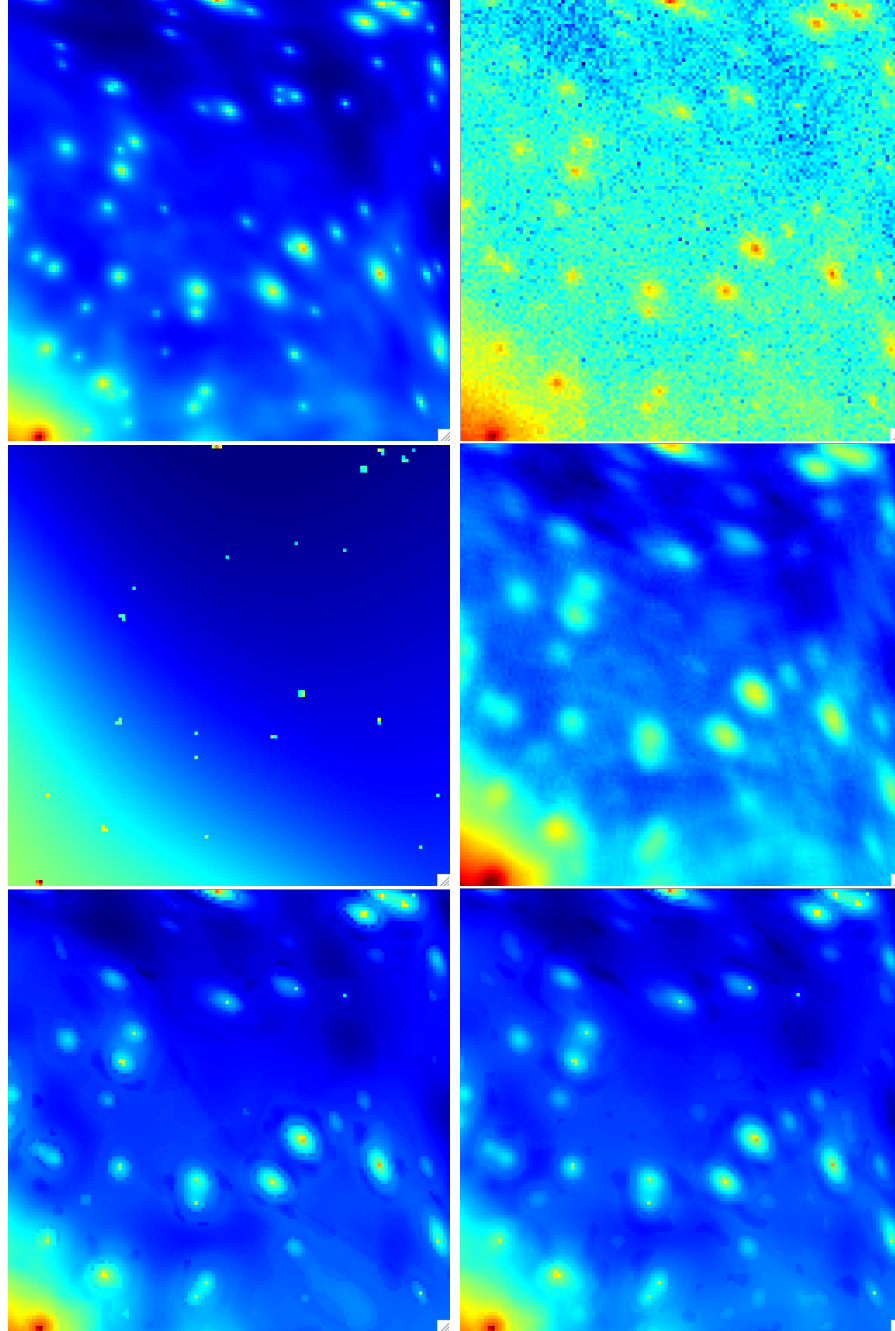


Figure 1.7: View of a single HEALPix face (angular extent: $\pi/3sr$) from the results of Figure 1.6. Top Left: Fermi simulated map without noise. Top Right: Fermi simulated map with Poisson noise. Middle Left: Fermi simulated map denoised with Anscombe VST + wavelet shrinkage. Middle Right: Fermi simulated 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.

Inpainting algorithm [2]. This modified algorithm can be called MS-VSTS-Inpainting algorithm.

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

$$\begin{aligned} & \text{Arg min}_{\mathbf{X}} \|\Phi^T \mathbf{X}\|_1, \text{ s.t.} \\ & \left\{ \begin{array}{l} \mathbf{X} \geq 0, \\ \forall (j, k) \in \mathcal{M}, (\Phi^T \Pi \mathbf{X})_j[k] = (\Phi^T \mathbf{Y})_j[k], \end{array} \right. \end{aligned} \quad (1.20)$$

where Π is a binary mask (1 on valid data and 0 on invalid data).

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

$$\tilde{\mathbf{X}} = P_+[\mathbf{X}^{(n)} + \Phi P_{\mathcal{M}} \Phi^T \Pi (\mathbf{Y} - \mathbf{X}^{(n)})], \quad (1.21)$$

$$\mathbf{X}^{(n+1)} = \Phi \text{ST}_{\lambda_n}[\Phi \tilde{\mathbf{X}}]. \quad (1.22)$$

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:

$$\lambda_n = \lambda_{\max} (2^{(\frac{N_{\max}-n}{N_{\max}-1})} - 1), n = 1, 2, \dots, N_{\max}, \quad (1.23)$$

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

Algorithm 4 MS-VST + IUWT Denoising + Inpainting

Require: data $a_0 := \mathbf{Y}$, mask Π , number of iterations N_{\max} , threshold κ .

Detection

- 1: **for** $j = 1$ to J **do**
- 2: Compute a_j and d_j using (1.6).
- 3: Hard threshold $|d_j[k]|$ with threshold $\kappa \sigma_j$ and update \mathcal{M} .
- 4: **end for**

Estimation

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

Experiment

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

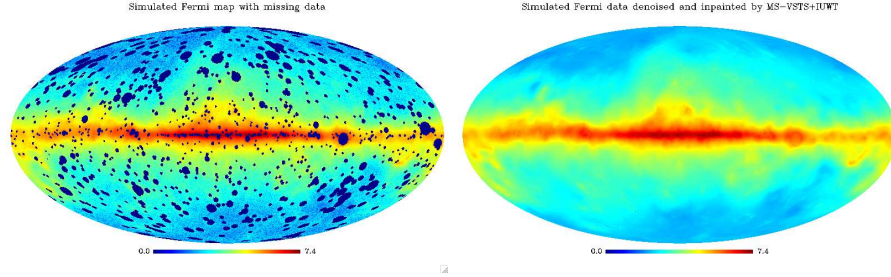


Figure 1.8: MS-VSTS - Inpainting. Left: Fermi simulated map with Poisson noise and the most luminous sources masked. Right: Fermi simulated map denoised and inpainted with wavelets (Algorithm 4). Pictures are in logarithmic scale.

The results are on Figure 1.8. 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.

1.5.2 Source detection: denoising and background modeling

1.5.2.1 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 denote \mathbf{Y} the data, \mathbf{B} the background we want to remove, and $d_j^{(b)}[k]$ the MS-VSTS coefficients of \mathbf{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:

$$\begin{aligned} & \underset{\mathbf{X}}{\text{Arg min}} \|\Phi^T \mathbf{X}\|_1, \text{ s.t.} \\ & \begin{cases} \mathbf{X} \geq 0, \\ \forall (j, k) \in \mathcal{M}, (\Phi^T \mathbf{X})_j[k] = (\Phi^T (\mathbf{Y} - \mathbf{B}))_j[k], \end{cases} \end{aligned} \quad (1.24)$$

Then, the reconstruction algorithm scheme becomes:

$$\tilde{\mathbf{X}} = P_+[\mathbf{X}^{(n)} + \Phi P_{\mathcal{M}} \Phi^T (\mathbf{Y} - \mathbf{B} - \mathbf{X}^{(n)})], \quad (1.25)$$

$$\mathbf{X}^{(n+1)} = \Phi \text{ST}_{\lambda_n} [\Phi^T \tilde{\mathbf{X}}]. \quad (1.26)$$

The algorithm is illustrated by the theoretical study in Figure 1.9. 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.

Algorithm 5 MS-VSTS + IUWT Denoising + Background extraction

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

Detection

- 1: **for** $j = 1$ to J **do**
- 2: Compute a_j and d_j using (1.6).
- 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 $\mathbf{X}^{(0)} = 0$, $\lambda_0 = 1$.
 - 6: **for** $n = 0$ to $N_{\max} - 1$ **do**
 - 7: $\tilde{\mathbf{X}} = P_+[\mathbf{X}^{(n)} + \Phi P_{\mathcal{M}} \Phi^T (\mathbf{Y} - \mathbf{B} - \mathbf{X}^{(n)})]$.
 - 8: $\mathbf{X}^{(n+1)} = \Phi \text{ST}_{\lambda_n} [\Phi^T \tilde{\mathbf{X}}]$.
 - 9: $\lambda_{n+1} = \frac{N_{\max} - (n+1)}{N_{\max} - 1}$.
 - 10: **end for**
 - 11: Get the estimate $\hat{\mathbf{A}} = \mathbf{X}^{(N_{\max})}$.
-

Like Algorithm 1, Algorithm 5 can be adapted to make multiresolution support adaptation.

1.5.2.2 Experiment

We applied Algorithms 5 on simulated Fermi data. To test the efficiency of our method, we detect the sources with the SExtractor routine [6], and compare the detected sources with the input source list to get the number of true and false detections. Results are shown on Figures 1.10 and 1.11. The SExtractor method was applied on the first wavelet scale of the reconstructed map, with a detection threshold equal to 1. 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).

Sensitivity to model errors

As it is difficult to model the background precisely, it is important to study

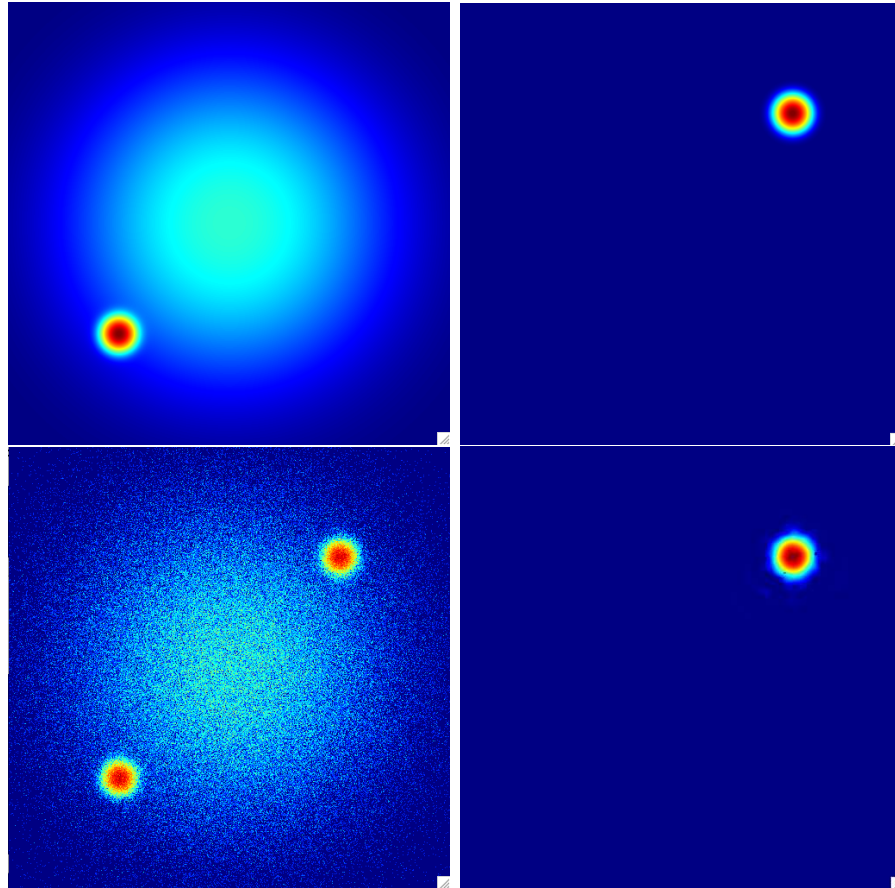


Figure 1.9: Theoretical testing for MS-VSTS + IUWT denoising + background removal algorithm (Algorithm 5). View on a single HEALPix face. Top Left: Simulated background : sum of two Gaussians of standard deviation equal to 0.1 and 0.01 respectively. Top Right: Simulated source: Gaussian of standard deviation equal to 0.01. Bottom Left: Simulated poisson data. Bottom Right: Image denoised with MS-VSTS + IUWT and background removal.

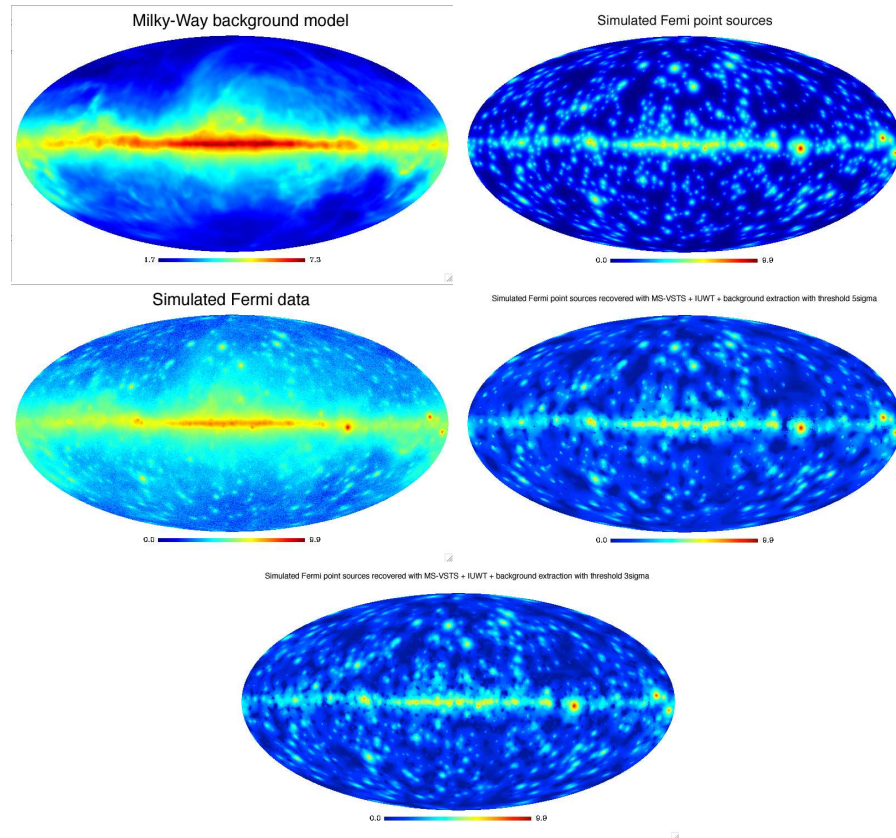


Figure 1.10: Top Left: Simulated background model. Top Right: Simulated Gamma Ray sources. Middle Left: Simulated Fermi data with Poisson noise. Middle Right: Reconstructed Gamma Ray Sources with MS-VSTS + IUWT + background removal (Algorithm 5) with threshold $5\sigma_j$. Bottom: Reconstructed Gamma Ray Sources with MS-VSTS + IUWT + background removal (Algorithm 5) with threshold $3\sigma_j$. Pictures are in logarithmic scale.

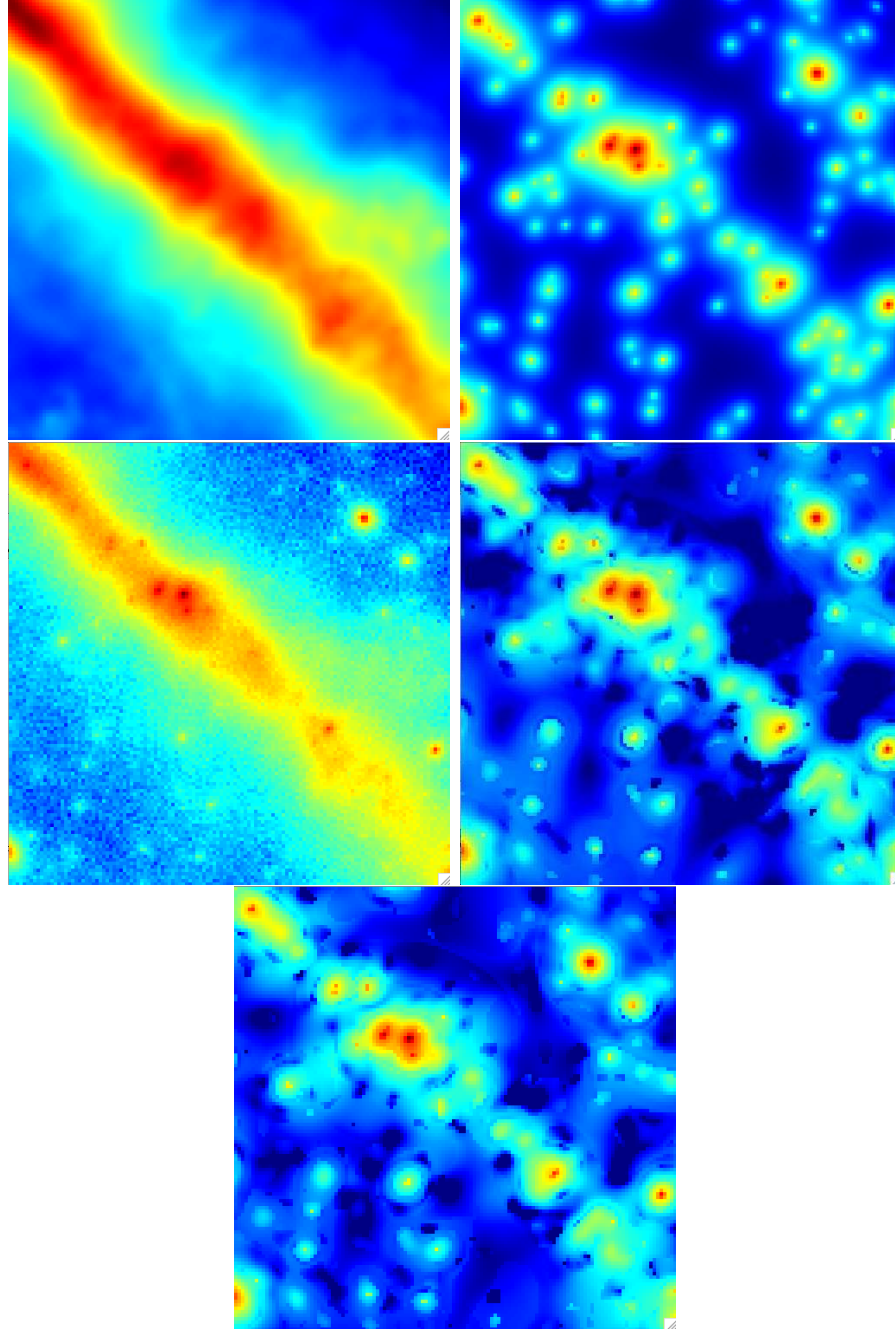


Figure 1.11: View of a single HEALPix face (angular extent: $\pi/3sr$) from the results of Figure 1.10. Top Left: Simulated background model. Top Right: Simulated Gamma Ray sources. Middle Left: Simulated Fermi data with Poisson noise. Middle Right: Reconstructed Gamma Ray Sources with MS-VSTS + IUWT + 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.

TABLE 1.3: 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

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 ($\text{SNR}(dB) = 20 \log(\sigma_{\text{signal}}/\sigma_{\text{noise}})$) versus the standard deviation of the Gaussian perturbation. Table 1.3 shows that, when the standard deviation of the noise on the background model becomes of the same range as the mean of 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.

1.6 Extension to multichannel data

1.6.1 Gaussian Noise

1.6.1.1 2D-1D Wavelet Transform on the Sphere

We propose a denoising method for 2D - 1D data on the sphere, where the two first dimensions are spatial (longitude and latitude) and the third dimension is either the time or the energy. We need to analyze the data with a non-isotropic wavelet, where the time or energy scale is not connected to the spatial scale. An ideal wavelet function would be defined by:

$$\psi(\theta, \varphi, t) = \psi^{(\theta, \varphi)}(\theta, \varphi) \psi^{(t)}(t) \quad (1.27)$$

where $\psi^{(\theta, \varphi)}$ is the spatial wavelet and $\psi^{(t)}$ is the temporal (or energy) wavelet. In the following, we will consider only isotropic and dyadic spatial scales, and we denote j_1 the spatial resolution index (i.e. scale 2^{j_1}), j_2 the time (or energy) resolution index. We thus define the scaled spatial and temporal (or energy) wavelets $\psi_{j_1}^{(\theta, \varphi)}(\theta, \varphi) = \frac{1}{2^{j_1}} \psi^{(\theta, \varphi)}(\frac{\theta}{2^{j_1}}, \frac{\varphi}{2^{j_1}})$ and $\psi_{j_2}^{(t)} = \frac{1}{2^{j_2}} \psi^{(t)}(\frac{t}{2^{j_2}})$.

Hence, we derive the wavelet coefficients $w_{j_1, j_2}[k_\theta, k_\varphi, k_t]$ from a given data set D (k_θ and k_φ are spatial index and k_z a time (or energy) index. In continuous coordinates, this amounts to the formula

$$w_{j_1, j_2}[k_\theta, k_\varphi, k_t] = \frac{1}{2^{j_1}} \frac{1}{\sqrt{2^{j_2}}} \int \int \int_{-\infty}^{+\infty} D(\theta, \varphi, t) \times \psi^{(\theta, \varphi)}(\frac{\theta - k_\theta}{2^{j_1}}, \frac{\varphi - k_\varphi}{2^{j_1}}) \psi^{(t)}(\frac{t - k_t}{2^{j_2}}) dx dy dz = D * \bar{\psi}_{j_1}^{(\theta, \varphi)} * \bar{\psi}_{j_2}^{(t)}(\theta, \varphi, t) \quad (1.28)$$

where $*$ is the convolution and $\bar{\psi}(t) = \psi(-t)$.

1.6.1.2 Fast Undecimated 2D-1D decomposition/reconstruction

In order to have a fast algorithm for discrete data, we use wavelet functions associated to filter banks. Hence, our wavelet decomposition consists in applying first a IUWT on the sphere for each frame k_z . Using the spherical IUWT, we have the reconstruction formula:

$$D[k_\theta, k_\varphi, k_t] = a_{J_1}[k_\theta, k_\varphi] + \sum_{j_1=1}^{J_1} w_{j_1}[k_\theta, k_\varphi, k_t], \forall k_t \quad (1.29)$$

where J_1 is the number of spatial scales. To have simpler notations, we replace the two spatial indexes by a single index k_r which corresponds to the pixel index:

$$D[k_r, k_t] = a_{J_1}[k_r] + \sum_{j_1=1}^{J_1} w_{j_1}[k_r, k_t], \forall k_t \quad (1.30)$$

Then, for each spatial location k_r and for each 2D wavelet scale j_1 , we apply a 1D wavelet transform along t on the spatial wavelet coefficients $w_{j_1}[k_r, k_t]$ such that

$$w_{j_1}[k_r, k_t] = w_{j_1, J_2}[k_r, k_t] + \sum_{j_2=1}^{J_2} w_{j_1, j_2}[k_r, k_t], \forall (k_r, k_t) \quad (1.31)$$

where j_2 is the number of scales along t . The same processing is also applied on the coarse spatial scale $a_{J_1}[k_r, k_t]$ and we have

$$a_{J_1}[k_r, k_t] = a_{J_1, J_2}[k_r, k_t] + \sum_{j_2=1}^{J_2} w_{J_1, j_2}[k_r, k_t], \forall (k_r, k_t) \quad (1.32)$$

Hence, we have a 2D-1D spherical undecimated wavelet representation of the input data D :

$$D[k_r, k_t] = a_{J_1, J_2}[k_r, k_t] + \sum_{j_1=1}^{J_1} w_{j_1, J_2}[k_r, k_t] + \sum_{j_2=1}^{J_2} w_{J_1, j_2}[k_r, k_t] + \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} w_{j_1, j_2}[k_r, k_t] \quad (1.33)$$

From this expression, we distinguish four kinds of coefficients:

- Detail-Detail coefficients ($j_1 \leq J_1$ and $j_2 \leq J_2$):

$$w_{j_1, j_2}[k_r, k_t] = (\delta - \bar{h}_{1D}) \star (\bar{h}_{1D}^{(j_2-1)} \star a_{j_1-1}[k_r, \cdot] - h_{1D}^{(j_2-1)} \star a_{j_1}[k_r, \cdot]) \quad (1.34)$$

- Approximation-Detail coefficients ($j_1 = J_1$ and $j_2 \leq J_2$):

$$w_{J_1, j_2}[k_r, k_t] = h_{1D}^{(j_2-1)} \star a_{J_1}[k_r, \cdot] - h_{1D}^{(j_2)} \star a_{J_1}[k_r, \cdot] \quad (1.35)$$

- Detail-Approximation coefficients ($j_1 \leq J_1$ and $j_2 = J_2$):

$$w_{j_1, J_2}[k_r, k_t] = h_{1D}^{(J_2)} \star a_{j_1-1}[k_r, \cdot] - h_{1D}^{(J_2)} \star a_{j_1}[k_r, \cdot] \quad (1.36)$$

- Approximation-Approximation coefficients ($j_1 = J_1$ and $j_2 = J_2$):

$$a_{J_1, J_2}[k_r, k_t] = h_{1D}^{(J_2)} \star a_{J_1}[k_r, \cdot] \quad (1.37)$$

1.6.1.3 Multichannel Gaussian denoising

As the spherical 2D-1D undecimated wavelet transform just described is fully linear, a Gaussian noise remains Gaussian after transformation. Therefore, all thresholding strategies which have been developed for wavelet Gaussian denoising are still valid with the spherical 2D-1D wavelet transform. Denoting TH the thresholding operator, the denoised cube in the case of additive white Gaussian noise is obtained by:

$$\tilde{D}[k_r, k_t] = a_{J_1, J_2}[k_r, k_t] + \sum_{j_1=1}^{J_1} \text{TH}(w_{j_1, J_2}[k_r, k_t]) + \sum_{j_2=1}^{J_2} \text{TH}(w_{J_1, j_2}[k_r, k_t]) + \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} \text{TH}(w_{j_1, j_2}[k_r, k_t]) \quad (1.38)$$

A typical choice of TH is the hard thresholding operator, i.e.

$$\text{TH}(x) = \begin{cases} 0 & \text{if } |x| < \tau \\ x & \text{if } |x| \geq \tau \end{cases} \quad (1.39)$$

The threshold τ is generally chosen between 3 and 5 times the noise standard deviation.

1.6.2 Poisson Noise

1.6.2.1 Multi-Scale Variance Stabilizing Transform

To perform a Poisson denoising, we have to plug the MS-VST into the spherical 2D-1D undecimated wavelet transform. Again, we distinguish four kinds of coefficients that take the following forms:

- Detail-Detail coefficients ($j_1 \leq J_1$ and $j_2 \leq J_2$):

$$w_{j_1, j_2}[k_r, k_t] = (\delta - \bar{h}_{1D}) \star (T_{j_1-1, j_2-1}[\bar{h}_{1D}^{(j_2-1)} \star a_{j_1-1}[k_r, \cdot]] - T_{j_1, j_2-1}[h_{1D}^{(j_2-1)} \star a_{j_1}[k_r, \cdot]]) \quad (1.40)$$

- Approximation-Detail coefficients ($j_1 = J_1$ and $j_2 \leq J_2$):

$$w_{J_1, j_2}[k_r, k_t] = T_{J_1, j_2-1}[h_{1D}^{(j_2-1)} \star a_{J_1}[k_r, \cdot]] - T_{J_1, j_2}[h_{1D}^{(j_2)} \star a_{J_1}[k_r, \cdot]] \quad (1.41)$$

- Detail-Approximation coefficients ($j_1 \leq J_1$ and $j_2 = J_2$):

$$w_{j_1, J_2}[k_r, k_t] = T_{j_1-1, J_2}[h_{1D}^{(J_2)} \star a_{j_1-1}[k_r, \cdot]] - T_{j_1, J_2}[h_{1D}^{(J_2)} \star a_{j_1}[k_r, \cdot]] \quad (1.42)$$

- Approximation-Approximation coefficients ($j_1 = J_1$ and $j_2 = J_2$):

$$a_{J_1, J_2}[k_r, k_t] = h_{1D}^{(J_2)} \star a_{J_1}[k_r, \cdot] \quad (1.43)$$

Hence, all 2D-1D wavelet coefficients w_{j_1, j_2} are now stabilized, and the noise on all these wavelet coefficients is Gaussian with known scale-dependent variance that depends solely on h . Denoising is however not straightforward because there is no explicit reconstruction formula available because of the form of the stabilization equations above. Formally, the stabilizing operators T_{j_1, j_2} and the convolution operators along the spatial and temporal dimensions do not commute, even though the filter bank satisfies the exact reconstruction formula. To circumvent this difficulty, we propose to solve this reconstruction problem by using an iterative reconstruction scheme.

1.6.2.2 Detection-Reconstruction

As the noise on the stabilized coefficients is Gaussian, and without loss of generality, we let its standard deviation equal to 1, we consider that a wavelet coefficient $w_{j_1, j_2}[k_r, k_t]$ is significant, i.e., not due to noise, if its absolute value is larger than a critical threshold τ , where τ is typically between 3 and 5.

The multiresolution support will be obtained by detecting at each scale the significant coefficients. The multiresolution support for $j_1 \leq J_1$ and $j_2 \leq J_2$ is defined as:

$$\mathcal{M}_{j_1, j_2}[k_r, k_t] = \begin{cases} 1 & \text{if } w_{j_1, j_2}[k_r, k_t] \text{ is significant} \\ 0 & \text{otherwise} \end{cases} \quad (1.44)$$

We denote \mathcal{W} the spherical 2D-1D undecimated wavelet transform described above, and \mathcal{R} the inverse wavelet transform. We want our solution X to preserve the significant structures of the original data by reproducing exactly the same coefficients as the wavelet coefficients of the input data Y , but only at scales and positions where significant signal has been detected. At other scales and positions, we want the smoothest solution with the lowest budget in terms of wavelet coefficients. Furthermore, as Poisson intensity functions are positive by nature, a positivity constraint is imposed on the solution. It is clear that there are many solutions satisfying the positivity and multiresolution support consistency requirements, e.g. Y itself. Thus, our reconstruction problem based solely on these constraints is an ill-posed inverse problem that must be regularized. Typically, the solution in which we are interested must be sparse by involving the lowest budget of wavelet coefficients. Therefore our reconstruction is formulated as a constrained sparsity-promoting minimization problem that can be written as follows

$$\min_{\mathbf{X}} \|\mathcal{W}\mathbf{X}\|_1 \text{ subject to } \begin{cases} \mathcal{M}\mathcal{W}\mathbf{X} = \mathcal{M}\mathcal{W}\mathbf{Y} \\ \mathbf{X} \geq 0 \end{cases} \quad (1.45)$$

where $\|\cdot\|$ is the L_1 -norm playing the role of regularization and is well known to promote sparsity[9]. This problem can be solved efficiently using the hybrid steepest descent algorithm [24][25], and requires about 10 iterations in practice. Transposed into our context, its main steps can be summarized as follows:

Require: Input noisy data \mathbf{Y} , a low-pass filter h , multiresolution support \mathcal{M} from the detection step, number of iterations N_{\max}

- 1: Initialize $\mathbf{X}^{(0)} = \mathcal{M}\mathcal{W}\mathbf{Y} = \mathcal{M}w_Y$,
- 2: **for** $n = 1$ to N_{\max} **do**
- 3: $\tilde{d} = \mathcal{M}w_Y + (1 - \mathcal{M})\mathcal{W}X^{(n-1)}$,
- 4: $\mathbf{X}^{(n)} = P_+(\mathcal{R}\text{ST}_{\beta_n}[\tilde{d}])$,
- 5: Update the step $\beta_n = (N_{\max} - n)/(N_{\max} - 1)$
- 6: **end for**

where P_+ is the projector onto the positive orthant, i.e. $P_+(x) = \max(x, 0)$, ST_{β_n} is the soft-thresholding operator with threshold β_n , i.e. $\text{ST}_{\beta_n}[x] = x - \beta_n \text{sign}(x)$ if $|x| \geq \beta_n$, and 0 otherwise.

The final spherical MSVST 2D-1D wavelet denoising algorithm is the following:

Require: Input noisy data \mathbf{Y} , a low-pass filter h , threshold level τ

- 1: **Spherical 2D-1D MSVST:** Apply the spherical 2D-1D-MSVST to the data using (1.40)-(1.43).
 - 2: **Detection:** Detect the significant wavelet coefficients that are above τ , and compute the multiresolution support \mathcal{M} .
 - 3: **Reconstruction:** Reconstruct the denoised data using the algorithm above.
-

1.7 Conclusion

This chapter 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. Then, we have introduced a new denoising method on the sphere which takes into account a background model. The simulated data have shown that it is relatively robust to errors in the model, and can therefore be used for Fermi diffuse background modeling and source detection. Finally, we introduced an extension for multichannel data.

Acknowledgement

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

Bibliography

- [1] A. A. Abdo, M. Ackermann, and al. Fermi/large area telescope bright gamma-ray source list. ApJS, 183:46–66, 2009.
- [2] P. Abrial, Y. Moudden, J.L. Starck, J. Bobin, M.J. Fadili, B. Afeyan, and M.K. Nguyen. Morphological component analysis and inpainting on the sphere: Application in physics and astrophysics. Journal of Fourier Analysis and Applications, 13(6), 2007. 729–748.
- [3] P. Abrial, Y. Moudden, J.L. Starck, M. J. Fadili, J. Delabrouille, and M. Nguyen. CMB data analysis and sparsity. Statistical Methodology, 5(4), 2008. 289-298.
- [4] F.J. Anscombe. The transformation of poisson, binomial and negative-binomial data. Biometrika, vol. 35, no. 3:pp. 246–254, 1948.
- [5] W. B. et. al. Atwood. The Large Area Telescope on the Fermi Gamma-Ray Space Telescope Mission. ApJ, 697:1071–1102, June 2009.
- [6] E. Bertin and S. Arnouts. SExtractor: Software for source extraction. Astronomy and Astrophysics Supplement, 117:393–404, 1996.
- [7] E. Candes and D.L. Donoho. Ridgelets: the key to high dimensional intermittency? Philosophical Transactions of the Royal Society of London, A 357:2495, 1999.
- [8] D.L. Donoho. Nonlinear wavelet methodes for recovery of signals, den-sitites, and spectra from indirect and noisy data. Proc. Symposia in Applied Mathematics, vol. 47:pp. 173–205, 1993.
- [9] D.L. Donoho. For most large underdetermined systems of linear equations, the minimal ℓ^1 -norm near-solution approximates the sparsest near-solution. Technical report, Department of Statistics of Stanford University, 2004.
- [10] M. Fisz. The limiting distribution of a function of two independant random variables and its statistical application. Colloquium Mathematicum, vol. 3:pp. 138–146, 1955.
- [11] P. Fryźlewicz and G.P. Nason. A haar-fisz algorithm for poisson intensity estimation. J. Comp. Graph. Stat., vol. 13:pp. 621–638, 2004.

- [12] K. M. Górski, E. Hivon, A. J. Banday, B. D. Wandelt, F. K. Hansen, M. Reinecke, and M. Bartelmann. HEALPix: A framework for high-resolution discretization and fast analysis of data distributed on the sphere. *Astrophysical Journal*, 622, April 2005. 759–771.
- [13] R. C. Hartman, D. L. Bertsch, S. D. Bloom, A. W. Chen, P. Deines-Jones, J. A. Esposito, C. E. Fichtel, D. P. Friedlander, S. D. Hunter, L. M. McDonald, P. Sreekumar, D. J. Thompson, B. B. Jones, Y. C. Lin, P. F. Michelson, P. L. Nolan, W. F. Tompkins, G. Kanbach, H. A. Mayer-Hasselwander, A. Mucke, M. Pohl, O. Reimer, D. A. Kniffen, E. J. Schneid, C. von Montigny, R. Mukherjee, and B. L. Dingus. Third EGRET catalog (3EG) (Hartman+, 1999). *VizieR Online Data Catalog*, 212:30079+, August 1999.
- [14] J. Schmitt, J. L. Starck, J. M. Casandjian, J. Fadili, and I. Grenier. Poisson denoising on the sphere: Application to the Fermi gamma ray space telescope. *Astronomy and Astrophysics*, 517, 2010.
- [15] E. Kolaczyk. Bayesian multiscale models for Poisson processes. *J. Amer. Stat. Assoc.*, vol. 94, no. 447:pp. 920–933, 1999.
- [16] S. Lefkimmiatis, P. Maragos, and G. Papandreou. Bayesian inference on multiscale models for Poisson intensity estimation: Applications to photon-limited image denoising. *IEEE Transactions on Image Processing*, vol. 20, no. 20, January 2009.
- [17] J. D. Myers. Lat background models, 2009. <http://fermi.gsfc.nasa.gov/ssc/data/access/lat/BackgroundModels.html>.
- [18] J. D. Myers. Lat 1-year point source catalog, 2010. http://fermi.gsfc.nasa.gov/ssc/data/access/lat/1yr_catalog/.
- [19] R. Nowak and E. Kolaczyk. A statistical multiscale framework for Poisson inverse problems. *IEEE Trans. Inf. Theory*, vol. 45, no. 5:pp. 1811–1825, 2000.
- [20] J.-L. Starck, M. Elad, and D. L. Donoho. Redundant multiscale transforms and their application for morphological component analysis. *Advances in Imaging and Electron Physics*, 132, 2004.
- [21] J.-L. Starck, J. M. Fadili, S. Digel, B. Zhang, and J. Chiang. Source detection using a 3D sparse representation: application to the Fermi gamma-ray space telescope. *A&A*, 504:641–652, September 2009.
- [22] J.-L. Starck, Y. Moudou, P. Abrial, and M. Nguyen. Wavelets, ridgelets and curvelets on the sphere. *Astronomy and Astrophysics*, 446:1191–1204, 2006.

- [23] K. Timmerman and R. Nowak. Multiscale modeling and estimation of poisson processes with application to photon-limited imaging. IEEE Trans. Inf. Theory, vol. 45, no. 3:pp. 846–862, 1999.
- [24] I. Yamada. The hybrid steepest descent method for the variational inequality problem over the intersection of fixed point sets of nonexpansive mappings. In Inherently Parallel Algorithm in Feasibility and Optimization and their Applications, pages pp. 473–504. Elsevier, 2001.
- [25] B. Zhang, J. Fadili, and J.-L. Starck. Wavelets, ridgelets and curvelets for poisson noise removal. IEEE Transactions on Image Processing, vol. 11, no. 6:pp. 1093–1108, 2008.