



<b>Publication Year</b>	2020
<b>Acceptance in OA</b>	2021-01-21T16:20:31Z
<b>Title</b>	A comparative analysis of denoising algorithms for extragalactic imaging surveys
<b>Authors</b>	Roscani, V., Tozza, S., CASTELLANO, MARCO, MERLIN, Emiliano, Ottaviani, D., Falcone, M., FONTANA, Adriano
<b>Publisher's version (DOI)</b>	10.1051/0004-6361/201936278
<b>Handle</b>	<a href="http://hdl.handle.net/20.500.12386/29928">http://hdl.handle.net/20.500.12386/29928</a>
<b>Journal</b>	ASTRONOMY & ASTROPHYSICS
<b>Volume</b>	643

# A comparative analysis of denoising algorithms for extragalactic imaging surveys

V. Roscani<sup>1</sup>, S. Tozza<sup>3</sup>, M. Castellano<sup>1</sup>, E. Merlin<sup>1</sup>, D. Ottaviani<sup>1</sup>, M. Falcone<sup>2</sup>, and A. Fontana<sup>1</sup>

<sup>1</sup> INAF - Osservatorio Astronomico di Roma, Via Frascati 33, I-00040, Monteporzio, Italy. e-mail: valerio.roscani@inaf.it

<sup>2</sup> Department of Mathematics, Sapienza University of Rome, P. le Aldo Moro, 5, 00185 Rome, Italy

<sup>3</sup> Department of Mathematics and Applications “Renato Caccioppoli”, University of Naples Federico II, Via Cintia, Monte S. Angelo, I-80126 Naples, Italy

## ABSTRACT

**Aims.** We present a comprehensive analysis of the performance of noise-reduction (“denoising”) algorithms to determine whether they provide advantages in source detection, mitigating noise on extragalactic survey images.

**Methods.** The methods under analysis are representative of different algorithmic families: Perona-Malik filtering, Bilateral filter, Total Variation denoising, Structure-texture image decomposition, Non-local means, Wavelets, and Block-matching. We tested the algorithms on simulated images of extragalactic fields with resolution and depth typical of the Hubble, Spitzer, and Euclid Space Telescopes, and of ground-based instruments. After choosing their best internal parameters configuration, we assess their performance as a function of resolution, background level and image type, also testing their ability to preserve the objects fluxes and shapes. Finally, we analyze in terms of *completeness* and *purity* the catalogs extracted after applying denoising algorithms on a simulated Euclid Wide Survey VIS image, on real H160 (HST) and K-band (HAWK-I) observations of the CANDELS GOODS-South field.

**Results.** Denoising algorithms often outperform the standard approach of filtering with the Point Spread Function (PSF) of the image. Applying Structure-Texture image decomposition, Perona-Malik filtering, the Total Variation method by Chambolle, and Bilateral filtering on the Euclid-VIS image, we obtain catalogs that are both more pure and complete by 0.2 magnitudes than those based on the standard approach. The same result is achieved with the Structure-Texture image decomposition algorithm applied on the H160 image. The relative advantage of denoising techniques with respect to PSF filtering increases at increasing depth.

Moreover, these techniques better preserve the shape of the detected objects with respect to PSF smoothing.

**Conclusions.** Denoising algorithms provide significant improvements in the detection of faint objects and enhance the scientific return of current and future extragalactic surveys. We identify the most promising denoising algorithms among the 20 considered techniques.

**Key words.** Techniques: image processing – Methods: numerical, data analysis – Surveys

## 1. Introduction

Measuring the amount of photons that we receive from astronomical sources over a given range of wavelengths is the primary way to gather information about the Universe. From the advent of digital photography in the 1980’s, charge coupled device (CCD) imaging is one of the primary ways by which we do so. Currently, CCD devices can reach 100 million pixels, with read noise as low as one electron, almost 100% quantum efficiency, and sensitivity from the X-rays to the near-infrared (see e.g. Lesser 2015, for a review).

Before being ready for the extraction of meaningful scientific content, astronomical images must be processed to, for instance, combine different observations into a single mosaic, correct for flat-field, transients, artifacts, and defects, subtract a global or local background, etc. Once these preparatory steps are completed, the quality of the image mainly depends on its resolution capability (which is proportional to  $\lambda/D$ , the ratio between the observed wavelength and the diameter of the telescope, in the case of diffraction-limited instruments, e.g. space observatories; or from the atmospheric seeing for ground-based facilities), and on its depth (i.e. magnitude at a given reference signal-to-noise ratio (SNR), which mainly depends on the duration of the observations (exposure time). Since increasing the latter is often unfeasible or too demanding, searching for alternative methods

to increase the SNR is important. A possible solution can be the application of noise reduction (“denoising”) techniques.

Wavelet transforms are a standard and popular tool used for denoising and detection of sources on astronomical images. The technique is extremely versatile, as it can be applied on a wide range of scientific cases (e.g. X-ray images: XMM-LSS survey (Pierre et al. 2004) and Fermi catalog (Ackermann et al. 2013; Principe et al. 2018), Cosmic Microwave Background maps (Starck et al. 2004), N-body simulations (Romeo et al. 2003), etc.). Other interesting applications are summarized in Starck et al. (2006). Among the different possible implementations, the widely used is the so-called “à trous” algorithm (Shensa 1992). This algorithm is an undecimated wavelet transform (UWT), which is also isotropic, and because of this, it is very efficient for the detection of isotropic objects. It is therefore clear the reason of its popularity for astronomical image processing, where many objects are nearly isotropic (e.g. stars, galaxies, galaxy clusters) (Starck et al. 2014).

For what concerns extragalactic optical/near-infrared imaging, images are typically convolved with a PSF shaped kernel to enhance source detection (an application of the lemma by Neyman & Pearson 1933); this is the most standard example of a denoising algorithm, since filtering reduces the noise variance, allowing real sources to raise above the background. In many

familiar cases, the typical PSFs of telescopes are quite similar to 2D Gaussians, making the PSF filtering basically indistinguishable from a Gaussian filtering. However, in several non-astronomical applications of image analysis this approach is often outclassed by other, more refined methods, designed to be more efficient and to better preserve the borders and edges of the sources.

A special mention is required for machine learning and deep learning techniques, lately becoming popular in the image processing field and in astronomy. An interesting application is proposed by Beckouche et al. (2013), where a sparse dictionary learning algorithm has been tested on multiple astronomical images. On the other hand, autoencoder neural networks for image denoising seem to be very promising (Vincent et al. 2008, 2010; Xie et al. 2012). Some applications of autoencoders on different areas of astronomy can be found in the literature, even if they are primarily used for other purposes (e.g. spectral energy distribution recovery (Frontera-Pons et al. 2017), gravitational waves denoising (Shen et al. 2017), stellar cluster detection (Karmakar et al. 2018)). Although these techniques are undoubtedly highly appealing, the aim of this work is to perform a comprehensive comparison of "traditional" denoising algorithms based on non-linear partial differential equations (PDEs) and variational methods. The comparison with other approaches based on machine learning will be developed in future works.

We compared several classes of denoising techniques, in order to find which ones yield the best improvements in source detection. To this aim, we have performed an extended set of tests. We considered many different noise reduction algorithms, roughly belonging to the following families: Perona-Malik (PM) filtering, Bilateral filter, Total Variation (TV) denoising, Structure-texture image decomposition, Block-matching, Non-local means, and Wavelets. Note that the numerical methods employed ranges from variational methods to PDEs-based techniques, also including some statistical methods.

We tested them using two different datasets. First, we focused on simulated images, created by state-of-the-art codes and prescriptions in order to mimic different realistic cases. This simplified environment has the advantages to allow a detailed analysis of the results, since the "truth" is perfectly known. For real images, we applied the algorithms giving the best results obtained on the simulated dataset to check if the improvement is confirmed. To our knowledge, this is the first attempt to extensively compare a large number of denoising algorithms in an astrophysical context. In general, the performance of any of these methods depends on the kind of noise that affects the image. Here we are mainly interested in extragalactic imaging, and in particular we focus on the next generation of optical - near-infrared instruments and surveys such as Euclid (Laureijs et al. 2011), LSST (LSST Science Collaboration et al. 2009), DES (Dark Energy Survey Collaboration et al. 2016), and WFIRST (Spergel et al. 2015). The paper is organized as follows: in Sect. 2 we list and briefly describe all the denoising methods used in our tests, providing mathematical formulations and code information. In Sect. 3 we present our datasets. In Sect. 4 we describe and discuss all the tests we carried out, showing the results we obtained in Sect. 5. In Sect. 6 we apply the methods on real images from space and ground-based. Finally, Sect. 7 summarizes the main results, discussing also the possible future applications. Throughout this paper we adopt the AB magnitude system (Oke & Gunn 1983) and a  $\Lambda$ CDM cosmology with  $\Omega_m = 0.3$ ,  $\Omega_\Lambda = 0.7$ ,  $H_0 = 70 \text{ Kms}^{-1} \text{ Mpc}^{-1}$ .

## 2. Denoising techniques

The focus of this paper is on the comparison of different techniques proposed in the literature, when they are applied on astronomical images. As we said in the introduction, these images have specific features. So an efficient denoising method is crucial to extract the information contained in the image and could be used as a preliminary step for other image processing problems, like the image segmentation and/or deblurring.

In order to select the most efficient denoising approach for extragalactic survey images we have analyzed different classes of methods in order to cover the main families of noise reduction techniques, namely non-linear filtering (2.2), bilateral filter (2.3), TV denoising (2.4), image decomposition (2.5), wavelets (2.6) and non-local means (2.7).

We briefly summarize the mathematical formulations of these techniques and we provide information on the codes for reproducibility.

### 2.1. Gaussian smoothing

Let us consider the intensity function  $I(x, y)$  of a noisy image, with  $(x, y) \in \Omega$ , where  $\Omega \subset \mathbb{R}^2$  is the reconstruction domain. Let  $I_{clean}$  be the desired clean image. An image with a Gaussian noise component is

$$I(x, y) = I_{clean}(x, y) + \eta \quad (1)$$

where  $\eta \sim N(\mu, \sigma)$  is the additive noise component. Of course, we want to reconstruct  $I_{clean}$  from  $I$ .

This filter uses a Gaussian function for calculating the transformation to apply to each pixel in the image. Mathematically, applying a Gaussian filter to an image corresponds to convolve the image with a Gaussian function. Since the Fourier transform of a Gaussian is another Gaussian, applying a Gaussian smoothing has the effect of reducing the image's high-frequency components; a Gaussian filter is then a *low-pass filter*. In two dimensions, it is the product of two Gaussian functions, one in each dimension, so that the low-pass Gaussian filter is

$$G_\sigma(x, y) := \frac{1}{2\pi\sigma^2} \exp^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2)$$

where  $x$  is the distance from the origin in the horizontal axis,  $y$  is the distance from the origin in the vertical axis, and  $\sigma$  is the standard deviation of the Gaussian distribution.

Filtering the image  $I : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$  with a "low-pass" Gaussian filter corresponds to process it with the heat equation (Gabor 1965; Lindenbaum et al. 1994), that is solving the following linear PDE

$$\begin{cases} \frac{\partial I}{\partial t}(x, y, t) = \nabla I(x, y, t) & \forall (x, y, t) \in \Omega \times (0, T_C], \\ \frac{\partial I}{\partial \eta}(x, y, t) = 0, & \forall (x, y, t) \in \partial\Omega \times (0, T_C], \\ I(x, y, 0) = I_0(x, y), & \forall (x, y) \in \bar{\Omega}, \end{cases} \quad (3)$$

which has a diffusive effect on the initial datum  $I_0$ , for a small fixed time  $T_C > 0$ . The relation between the Gaussian filter (2) and the problem (3) is that the solution of the heat equation is a convolution with the Gaussian filter, i.e.

$$I(x, y, t) = (G_\sigma(x, y) * I_0)(x, y) \quad (4)$$

with  $\sigma = \sqrt{2}t$ .

It is well known that applying that filter does not preserve edges. This edge blurring is due to the isotropic diffusion.

We can get an improvement in three different ways:

1. Modifying the heat equation (see Sect. 2.2)
2. Making convolution "nonlinear" (see Sect. 2.3)
3. Defining an optimization problem (see Sect. 2.4).

#### Code information

In this paper we have used a simple Gaussian smoothing using a kernel that approximates a PSF of known Full Width at Half Maximum (FWHM) (Winkler 2011), referring to it as "PSF", whereas with "Gaussian" we refer to the Gaussian filter with internal parameter  $\sigma$ . We made use of the "gaussian\_filter" routine implemented in the Python package Scipy<sup>1</sup> (Jones et al. 2001), with  $\sigma \approx \frac{FWHM_{pixel}}{2.355}$ , easily obtained defining the Gaussian kernel radius  $r = x^2 + y^2$ , where the kernel maximum is at  $r = 0$  then  $FWHM = 2\sqrt{2\ln 2}\sigma$ , see also <https://brainder.org/2011/08/20/gaussian-kernels-convert-fwhm-to-sigma/> for further details.

## 2.2. Filters of Perona-Malik type

An improvement of the simple Gaussian filter is obtained by modifying the heat equation. Following the PM model (Perona & Malik 1990), we choose large values of  $|\nabla I|$  as an indicator of the edge points of the image, in order to stop the diffusion at these points. In this way we move from an isotropic to anisotropic diffusion as follows:

$$\frac{\partial I}{\partial t} = \text{div}(\nabla I) \Rightarrow \frac{\partial I}{\partial t} = \text{div}(g(|\nabla I|)\nabla I). \quad (5)$$

The equation (5) must be complemented with suitable boundary conditions (e.g. homogeneous Neumann boundary conditions) and an initial condition. Perona and Malik pioneered the idea of anisotropic diffusion and proposed two functions for the diffusion coefficient (also called *edge-stopping functions*):

$$g_1(|\nabla I|) := \frac{1}{1 + \left(\frac{|\nabla I|}{K}\right)^2} \quad (6)$$

$$g_2(|\nabla I|) := \exp\left(-\left(\frac{|\nabla I|}{K}\right)^2\right) \quad (7)$$

where  $K$  is the gradient magnitude threshold parameter that decides the amount of diffusion to take place.

We also consider other three edge-stopping functions that have been proposed after the original work by Perona and Malik:

Black et al. (1998) proposed an edge stopping function called Tukey's biweight function defined as:

$$g_3(|\nabla I|) := \begin{cases} \frac{1}{2} \left[ 1 - \left( \frac{|\nabla I|}{K\sqrt{2}} \right)^2 \right]^2 & \text{if } |\nabla I| \leq K\sqrt{2} \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

Guo et al. (2012) proposed the following function:

$$g_4(|\nabla I|) := \frac{1}{1 + \left(\frac{|\nabla I|}{K}\right)^{\alpha(|\nabla I|)}} \quad (9)$$

where

$$\alpha(|\nabla I|) := 2 - \frac{2}{1 + \left(\frac{|\nabla I|}{K}\right)^2}. \quad (10)$$

<sup>1</sup> [https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.gaussian\\_filter.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.gaussian_filter.html)

Weickert (1998) proposed:

$$g_s(|\nabla I|) := \begin{cases} 1 - \exp(-3.31488 * K^8 / (|\nabla I|)^8) & \text{if } |\nabla I| \neq 0 \\ 1 & \text{otherwise.} \end{cases} \quad (11)$$

#### Code information

This method has been implemented by us in C++ and it is available at: <https://github.com/valerioscani/perona-malik.git>.

## 2.3. Bilateral filter

The Bilateral filter is an edge-preserving denoising algorithm that was first introduced by Tomasi & Manduchi (1998). It is defined as (see also Banterle et al. 2012)

$$I(x) = \frac{1}{w} \sum_{x_i \in \Omega} I_0(x_i) f_r(\|I_0(x_i) - I_0(x)\|) g_s(\|x_i - x\|), \quad (12)$$

where

$$w := \sum_{x_i \in \Omega} f_r(\|I_0(x_i) - I_0(x)\|) g_s(\|x_i - x\|) \quad (13)$$

and

- $I$  is the filtered image
- $I_0$  is the original input image to be filtered
- $x$  are the coordinates of the current pixel to be filtered
- $\Omega$  is the window centered in  $x$ , so  $x_i \in \Omega$  is another pixel
- $f_r$  is the range kernel for smoothing differences in intensities (this function can be a Gaussian function)
- $g_s$  is the spatial (or domain) kernel for smoothing differences in coordinates (this function can be a Gaussian function).

It averages pixels based on their spatial closeness and on their radiometric similarity. Spatial closeness is measured by the Gaussian function of the Euclidean distance between two pixels and a certain standard deviation (`sigma_spatial`). Radiometric similarity is measured by the Gaussian function of the Euclidean distance between two color values and a certain standard deviation (`sigma_color`).

#### Code information

We used the Python routine "denoise\_bilateral" available in the Python package `scikit-image`<sup>2</sup>. We noticed that using our dataset, variations of the `sigma_spatial` were less effective than variations of `sigma_color`. We decided to set `sigma_spatial = 3` since it provides the best results.

## 2.4. Total Variation denoising

Total-variation denoising (also known as total-variation regularization) is based on the principle that images with excessive and possibly spurious detail have high TV, defined as

$$TV(u, \Omega) := \int_{\Omega} |\nabla u(x)| dx \quad (14)$$

for a function  $u \in C^1(\Omega)$  (note that a similar definition can be given also for  $L^1$  functions Kolmogorov & Fomin 1957). According to this principle, TV denoising tries to find an image with less TV under the constraint of being similar to the input image, which is controlled by the regularization parameter, i.e.

tries to minimize  $TV(I, \Omega)$ . This minimization problem leads to the Euler-Lagrangian equation, which can be solved via the following evolutive problem:

$$u_t = \frac{\partial}{\partial x} \left( \frac{u_x}{\sqrt{u_x^2 + u_y^2}} \right) + \frac{\partial}{\partial y} \left( \frac{u_y}{\sqrt{u_x^2 + u_y^2}} \right) - \lambda(u - u_0), \quad (15)$$

for  $t > 0$  and  $x, y \in \Omega$ , with homogeneous Neumann boundary conditions and a given initial condition. TV denoising tends to produce ‘‘cartoon-like’’ images, that is, piecewise-constant images. The concept was pioneered by Rudin, Osher, and Fatemi in Rudin et al. (1992) and is today known as the ROF model. TV denoising is remarkably effective at simultaneously preserving edges whilst smoothing away noise in flat regions, even at low SNRs.

#### Code information

We test the ROF method that was proposed by Chambolle in (Chambolle 2004) and the TV denoising using split-Bregman optimization (Goldstein & Osher 2009; Getreuer 2012; Bush 2011). For the implementation of the two aforementioned methods we have used the Python routines ‘‘denoise\_tv\_chambolle’’ and ‘‘denoise\_tv\_bregman’’ belonging to the Python package `SCIKIT-IMAGE`<sup>2</sup> (van der Walt et al. 2014).

### 2.5. Structure-texture image decomposition

A general approach to the denoising problem is based on the assumption that an image  $I$  can be regarded as composed of a structural part  $u$  (i.e. the objects in the image), and a textural part  $v$  which corresponds to finest details plus the noise. Following the approach described in Aujol et al. (2006), such image decomposition technique is based on the minimization of a functional with two terms, one based on the total variation and a second one on a different norm adapted to the texture component. Given an image  $I$  defined in a set  $\Omega$ , and let  $BV(\Omega)$  be the space of functions with limited total variation in  $\Omega$  we can decompose  $I$  into its two components by minimizing:

$$\inf \left( \int_{\Omega} |\nabla u(x)| + \lambda \|v(x)\|_X^p dx \right) \quad (16)$$

where  $\|\cdot\|_X^p$  denotes the norm of a given space  $X$  and the minimum is found among all functions  $(u, v) \in BV(\Omega) \times X$  such that  $u + v = I$ . The parameter  $p$  is a natural exponent, and  $\lambda$  is the so-called splitting parameter which modifies the relative weights. The best decomposition is found at the  $\lambda$  for which the correlation between  $u$  and  $v$  reaches a minimum.

#### Code information

In Castellano et al. (2015), the authors proposed a C++ code named Astro-Total Variation Denoiser (ATVD), which implements three versions of the technique, based respectively on the  $TV-L^2$  ( $X = L^2(\Omega)$ ),  $TV-L^1$  ( $X = L^1(\Omega)$ ) and TVG ( $X$  being a Banach space as defined in Aujol et al. 2006) norms. Two thresholds are defined and used in the stopping criteria of the algorithms, called  $\epsilon_{corr}$  and  $\epsilon_{sol}$ .

$\epsilon_{corr}$  defines the correlation algorithm precision, whereas  $\epsilon_{sol}$  defines the method precision (e.g. TVL2, TVG, TVL1). For all our tests, we will use  $\epsilon_{corr} = 10^{-4}$  and  $\epsilon_{sol} = 10^{-3}$ , as suggested by the authors of Castellano et al. (2015).

<sup>2</sup> <https://scikit-image.org/docs/dev/api/skimage.restoration.html>

#### Remark 1.

In the definition of the functional to be minimized one can also add a term taking into account some properties of the unknown image  $f$  corresponding to the available image  $g$ . This is a typical situation when the physical image is modeled as linear operator  $A$  acting from a Hilbert space  $X$  to a Hilbert space  $Y$ . In this approach  $X$  contains all the functions characterizing unknown objects and  $Y$  contains the functions describing the corresponding measurable images, so that

$$g = Af. \quad (17)$$

A typical choice is to choose  $X = Y = L^2(\Omega)$  and the inverse problem is then to minimize the functional

$$\int_{\Omega} \|Af - g\|_2^2 dx \quad (18)$$

over  $f \in X$ . This problem is often ill-posed so a popular Tikhonov regularization is obtained adding another term  $R(f)$  to the functional getting

$$\int_{\Omega} \|Af - g\|_2^2 dx + \mu R(f) \quad (19)$$

where  $\mu$  is a positive parameter to be tuned carefully. The term  $R(f)$  can also be used to introduce a prior, e.g. the regularity of  $f$  (based on Schwartz Theorem) or the sparsity of  $f$  (choosing  $R(f) = \|f\|_1$ ). Imposing a morphological prior on the shapes, such as penalizing shapes different from ellipses, would require an enormous number of parameters in the case of astronomical images that usually include several sky objects and is a very challenging problem which goes beyond the scopes of this paper. For the use of priors in other areas (e.g. in biomedical imaging), we refer the interested readers to Bertero & Piana (2006), and for a general introduction to inverse problems in imaging to the book Bertero & Boccacci (1998).

### 2.6. Wavelets

The wavelets transform is the counterpart for images of the Fourier transform and the wavelets domain, which is a sparse representation of the image that can be thought of similarly to the frequency domain of the Fourier transform (Valens 1999). Being a sparse representation means that most values are zero or near-zero and truly random noise is represented by many small values in the wavelet domain. Setting all values below some threshold to 0 reduces the noise in the image, but larger thresholds also decrease the detail present in the image.

Let us recall the relation introduced in Sect. 2.1

$$I = I_{clean} + \eta, \quad (20)$$

where  $\eta$  is the noise and  $I_{clean}$  is the clean image (signal). The components of  $\eta$  are independent and identically distributed (iid) as  $\mathcal{N}(0, \sigma^2)$  and independent of  $I_{clean}$ . The goal is again to remove the noise obtaining an approximation  $\widehat{I}$  of  $I_{clean}$  minimizing the mean square error (MSE)

$$MSE(\widehat{I}) := \frac{1}{N} \sum_{j=1}^N (\widehat{I}_j - I_j)^2, \quad (21)$$

where  $N$  is the number of pixels. Let us denote by  $Y = \mathcal{W}I$  the matrix of wavelet coefficients of the image  $I$  where  $\mathcal{W}$  is the orthogonal wavelet transform operator, similarly  $F = \mathcal{W}I_{clean}$

and  $E = \mathcal{W}\eta$  (see Vetterli & Kovačević (1995), Mallat (2001) for more details on  $\mathcal{W}$ ). The wavelet transform is based on the subbands (called details) at different scales usually indexed by  $k \in \mathcal{K}, \mathcal{K} \subset \mathbb{N}$ . The wavelet-thresholding method filters each coefficient  $Y_j$  from the detail subbands  $k \in \mathcal{K}$  with a threshold function to obtain  $\widehat{X}$ . The denoised approximation is  $\widehat{I} = \mathcal{W}^{-1}\widehat{X}$ , where  $\mathcal{W}^{-1}$  is the inverse wavelet transform. Two thresholding techniques are frequently used. The *soft-threshold* function

$$\varphi_T(x) := \text{sgn}(x) \max(|x| - T, 0) \quad (22)$$

which shrinks the argument  $x$  to 0 by the threshold  $T$ . The *hard-threshold* function

$$\psi_T(x) := x \mathbf{1}_{\{|x| > T\}} \quad (23)$$

which sets the input to 0 if is below (or equal) the threshold  $T$ . Note that the threshold procedure removes noise by thresholding *only* the wavelet coefficients of the corresponding subbands, keeping the low resolution coefficients unaltered.

#### Code information

We consider the two thresholding methods defined in the Python routine "denoise\_wavelet"<sup>2</sup> (Chang et al. 2000; Donoho & Johnstone 1994), the first applies BayesShrink, which is an adaptive thresholding method that computes separate thresholds for each wavelet subband as described in Chang et al. (2000), the second is "VisuShrink", in which a single "universal threshold" is applied to all wavelet detail coefficients as described in Donoho & Johnstone (1994). This threshold is designed to remove all Gaussian noise at a given  $\sigma$  with high probability, but tends to produce images that appear overly smooth.

In this work we decided to test several methods based on wavelet transforms. We selected the Meyer wavelet described in Meyer (1990) with VisuShrink thresholding method since, analyzing the application on our dataset, we found that it provides the best performance based on the analysis described in Sect. 4. The list from which we took the Meyer wavelet can be found in Lee et al. (2019). From now on, we will refer to this method as Orthogonal Wavelets.

We also consider other methods, based on multiscale wavelets decomposition (implemented in the C++ library called SPARSE2D, available at CosmoStat web page<sup>3</sup>). The first is an isotropic UWT, based on the *à trous* algorithm, better known as "Starlet transform", where 5 wavelets scales and an iterative hard thresholding technique are set. From now on, we will refer to this method as Starlet. The other two both use a biorthogonal UWT using a set of filters (introduced for the JPEG 2000 compression standard Starck et al. 2010) called "7/9 filters" and 5 wavelets scales. For the first method a  $3\sigma$  threshold is set, whereas for the second one a multi-resolution Wiener filter is performed. From now on, we will refer to these two methods as "b-UWT(7/9)" and "b-UWT(7/9)+Wiener", respectively. The optimal configurations for the methods implemented in SPARSE2D have been kindly suggested by the authors. We used the `mr_filter` program with the following options:

- Starlet: `t=default, f=3, n=5`
- b-UWT(7/9): `t=24, f=default, n=5`
- b-UWT(7/9)+Wiener: `t=24, f=6, n=5`

where `t` is the type of multi-resolution transform, `f` is the type of filtering, and `n` is the number of scales. For further details about these 3 methods and the UWTs in general, see Starck et al. (2010).

<sup>3</sup> <http://www.cosmostat.org/software/isap>

## 2.7. Non-local means

The non-local means algorithm averages the value of a given pixel with values of other pixels in a limited proximity, under the condition that the patches centered on the other pixels are similar enough to the patch centered on the pixel of interest. This algorithm is defined by the formula (Buades et al. 2005)

$$NL[I_0](x) = \frac{1}{C(x)} \int_{\Omega} \exp(-g_{\sigma}^h(x)) I_0(y) dy \quad (24)$$

where

$$g_{\sigma}^h(x) := \frac{G_{\sigma} * |I_0(x + \cdot) - I_0(y + \cdot)|^2(0)}{h^2}, \quad (25)$$

$I_0$  is the original image,  $x \in \Omega$ ,  $C(x)$  is a normalizing constant,  $G_{\sigma}$  is a Gaussian kernel with  $\sigma$  denoting the standard deviation, and  $h$  acts as a filtering parameter. The algorithm has been found to have excellent performance when used to denoise images with specific textures<sup>4</sup>.

We define by  $size_l$  the image size in pixels, by  $size_p$  the size of the patch in pixels, by  $d_p$  the maximal distance in pixels where to search patches, by  $n$  the image number of dimensions ( $n = 2, 3$  depends if we consider 2D or 3D images). In its original version the computational complexity of the algorithm is proportional to:  $size_l * (size_p * d_p)^n$  (Buades et al. 2005). A new "fast" version is now preferentially used since its actual complexity is proportional to:  $size_l * d_p^n$  (Darbon et al. 2008).

Compared to the classic algorithm, in the fast mode the distances are computed in a coarser way, indeed all pixels of a patch contribute to the distance to another patch with the same weight, no matter their distance to the center of the patch. This approach can result in a slightly poorer denoising performance.

When the standard deviation  $\sigma$  is given, the method gives a more robust computation of patch weights. A moderate improvement to denoising performance can be obtained subtracting the known noise variance from the computed patch distances, that improves the estimates of patch similarity (Buades et al. 2011).

#### Code information

In this work both the fast and slow version of the algorithm are tested. After a first selection of patch sizes and distances, through the analysis described in Sect. 4, we decided to set  $size_p = 5$  and  $d_p = 6$ . For our numerical tests we used the routine "denoise\_nl\_means"<sup>4</sup>, implemented in the Python package SKIMAGE.

## 2.8. Block-matching and 3D filtering

"Block-matching and 3D filtering" (BM3D) is a 3 dimensional block-matching algorithm which "groups" similar 2D fragments with a *Matching* method in the image. The *Matching* method finds similar fragments to a *reference* one, grouping fragments closer than a defined threshold. The matched fragments are then stored in 3D arrays called *groups*. A "collaborative filtering" is performed to each group, which consists in a 3D linear transform, a shrink to reduce noise and an invert linear transform which produces 2D estimates of all the fragments. Once the estimates are obtained, they are aggregated to form an estimate of the whole image. For further details see Dabov et al. (2007).

<sup>4</sup> [https://scikit-image.org/docs/dev/api/skimage.restoration.html#skimage.restoration.denoise\\_nl\\_means](https://scikit-image.org/docs/dev/api/skimage.restoration.html#skimage.restoration.denoise_nl_means)

### Code information

In this work we tested a C++ code of BM3D available at <https://github.com/gfaccio1/bm3d>. For performance improvements and memory reasons, the input images have been cut in smaller overlapping tiles, re-aggregated in a single output image at the end of the process, as suggested by the authors.

### 3. Test dataset

We first test the denoising algorithms on five different simulated images (Table 1), chosen as to reproduce the properties of a wide range of typical cases in terms of resolution, depth, pixel scale and wavelength:

- VIS: Euclid satellite visual band (wavelength: 550-900 nm)
- NIR H: Euclid satellite near-infrared H band (wavelength: 1372-2000 nm)
- EXT G: ground-based optical filter
- H160: Hubble Space Telescope (HST) near-infrared F160W band (e.g. CANDELS-wide Guo et al. 2013)
- IRAC: Irac-Spitzer 3.6 $\mu$ m channel.

From now on, we refer to the simulated images, provided as input to the algorithms, as "Original", whereas we refer to the simulated images representing the true sky, without noise included, as "Noiseless".

#### TEST IMAGES

Filter	PSF-FWHM arcsec	Pixel Scale arcsec	Mag Lim <sup>(a)</sup>
VIS	0.2	0.1	25.25
NIR H	0.3	0.3	24.01
EXT G	0.8	0.2	25.93
H160	0.15	0.06	27.23
IRAC	1.6	0.6	25.40
HUDF (H160)	0.15	0.06	28.16/29.74 <sup>(b)</sup>
Ks (HAWK-I)	0.4	0.1	24.45/26.3 <sup>(c)</sup>

**Table 1.** <sup>(a)</sup>: SNR=5; <sup>(b)</sup>: limiting magnitude at the CANDELS and at the full HUDF depth, respectively; <sup>(c)</sup>: images from Castellano et al. (2010), and from the HUGS survey (Fontana et al. 2014), respectively.

VIS and NIR H reproduce the expected features of the visual and near-infrared bands in the forthcoming ESA satellite Euclid (Laureijs et al. 2011), and EXT G simulates a typical ground-based complementary optical observation for the Euclid Wide Survey. H160 is modeled after the detection band in recent deep surveys such as CANDELS (Grogin et al. 2011; Koekemoer et al. 2011) and 3D-HST (Skelton et al. 2014), whereas IRAC simulates the features of the *Spitzer* Channel 1 band in the CANDELS GOODS-South field (Guo et al. 2013).

The images have been simulated with *SkyMaker* (Bertin 2009) on the basis of source catalogs generated by the Empirical Galaxy Generator (EGG) (Schreiber et al. 2017) and they have been perturbed by Gaussian noise in order to reach the limiting magnitudes reported in Table 1. All the PSFs are Gaussian except for the IRAC case where a real IRAC 3.6 $\mu$ m channel PSF has been used. The H160 and HAWK-I images are real observations whose tests are described in Sect. 6.1-6.2.

We can sort the simulated images in several different ways:

- Depth, from the deepest to the shallowest: *H160* > *EXT G* > *IRAC* > *VIS* > *NIR H*

- PSF, from the sharpest to the coarsest: *H160* > *VIS* > *NIR H* > *EXT G* > *IRAC*
- Pixscale, from the smallest to the largest: *H160* > *VIS* > *EXT G* > *NIR H* > *IRAC*.

For each simulated image, we cut three independent areas of the sky, which are the same for every band but differ in dimensions due to the different pixel scale. The regions are listed below:

- BG: centered on a big elliptical galaxy (see Fig. 1)
- CL: centered on a cluster of galaxies (see Fig. 2)
- CM: an average portion of the sky (see Fig. 3).

The three regions have a dimension of:

- VIS: 1000x1000 pixels
- NIR\_H: 333x333 pixels
- EXT\_G: 500x500 pixels
- H160: 1666x1666 pixels
- IRAC1: 166x166 pixels.

After the analysis described in Sect. 4, commenting the results obtained in Sect. 5, additional tests on real images (see Table 1 for details) ground-based (HAWK-I) and from space (HST) are reported and analyzed in Sect. 6.

### 4. Quality tests

The idea at the basis of the analysis is to first evaluate the algorithms through different tests, in order to apply only the most promising ones (with their best configurations) on real images. We organize our analysis on the five simulated images in different levels of testing. A brief description of each step is given below:

1. As a first step we compare the algorithms performance through three metrics: *mean square error (MSE)*, *structural similarity (SSIM)* Wang et al. 2004) and *CPU time*. The *MSE* is defined as:

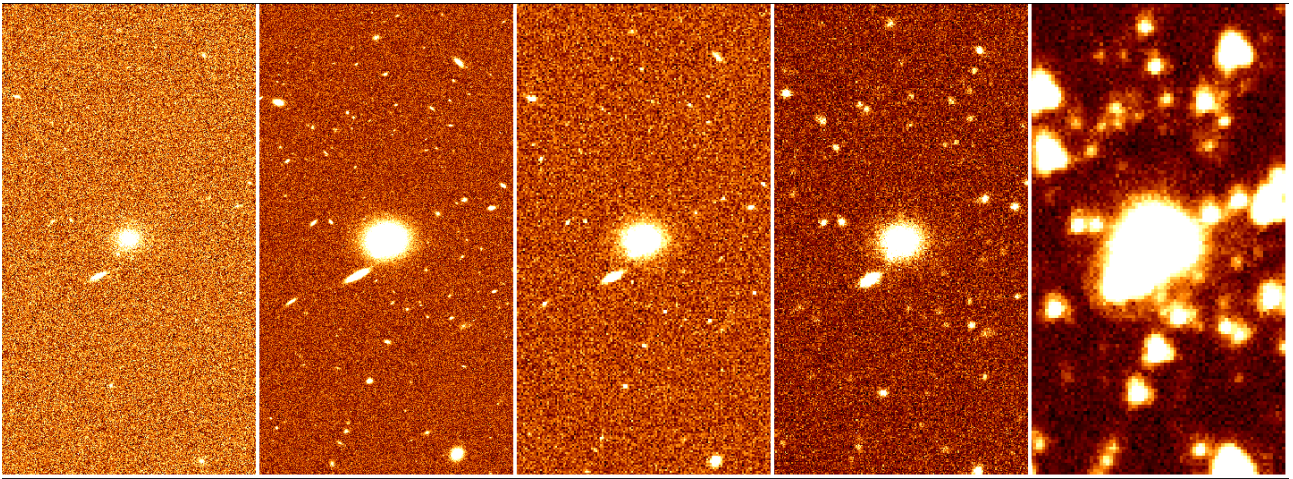
$$MSE := \frac{\sum_{i=1}^N (x_i - \widehat{x}_i)^2}{N} \quad (26)$$

where  $x_i$  is the  $i$ -th pixel in the denoised image and  $\widehat{x}_i$  is the  $i$ -th pixel in the original image (without noise). The *SSIM* is defined as:

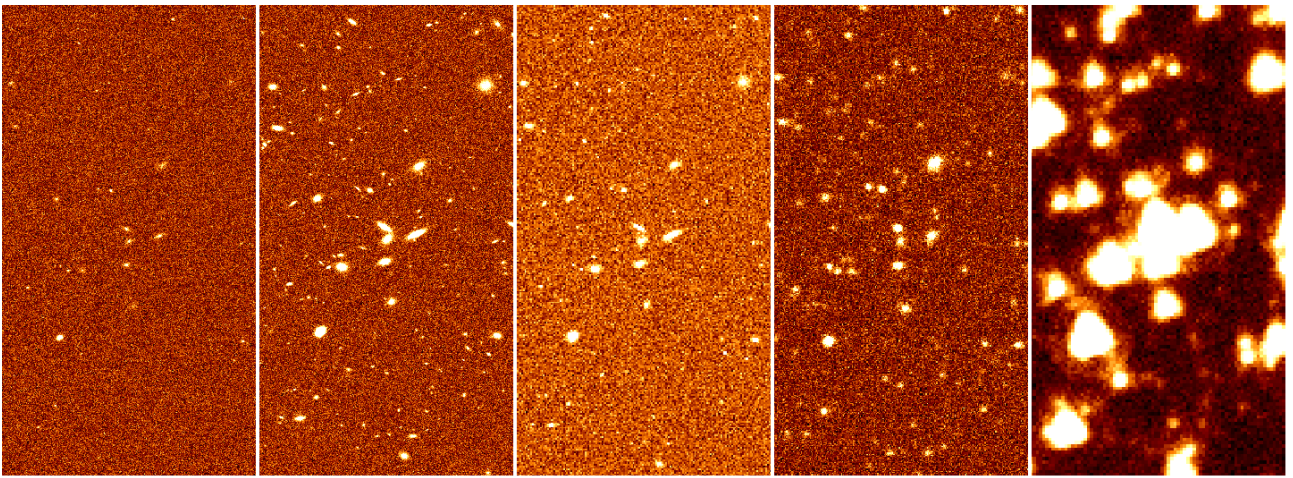
$$SSIM := \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (27)$$

where  $\mu_x$  is the average of  $x$ ,  $\mu_y$  is the average of  $y$ ,  $\sigma_x^2$  is the variance of  $x$ ,  $\sigma_y^2$  is the variance of  $y$ ,  $\sigma_{xy}$  is the covariance of  $x$  and  $y$ ,  $c_1$  and  $c_2$  are constants proportional to the dynamic range of the pixel values. The *CPU time* is the computational time required by the algorithms to filter the image. Through these tests we identify the main internal parameters of each algorithm and their ideal values.

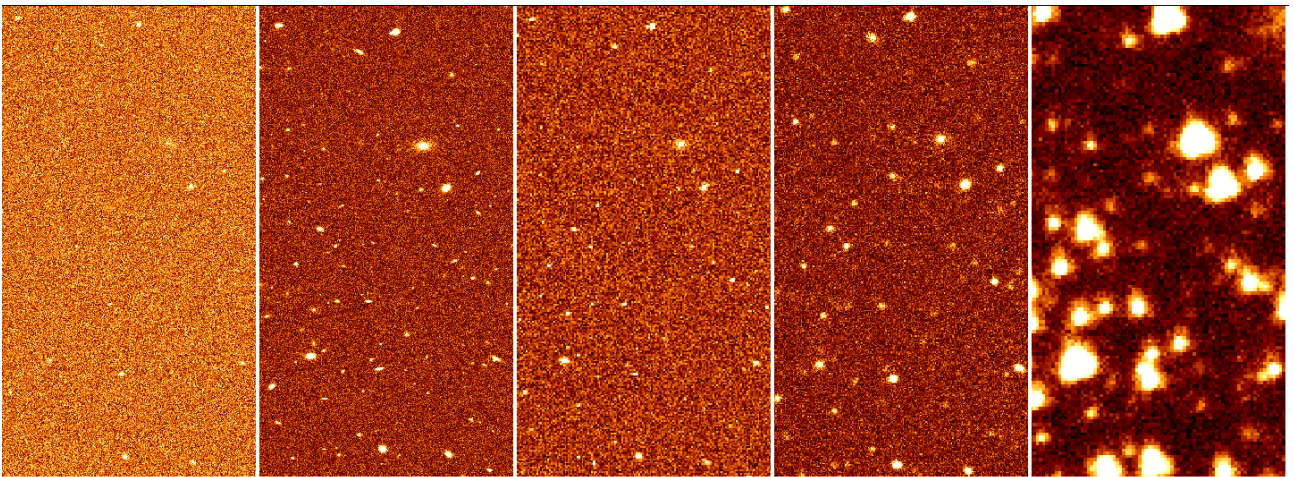
2. We test the stability of the algorithms selected in the previous step when faced with non-stationary Gaussian noise. Furthermore, we test their performance as a function of the FWHM of the PSF and as a function of the background noise level.
3. We test the stability of the algorithms selected in the previous steps against variations of the main internal parameter value (identified in Step 1), measuring how the MSE varies as a function of the parameter values.



**Fig. 1.** From left to right: Crops of the BG (Big Galaxy) image central area for VIS, H160, NIR H, EXT G and IRAC



**Fig. 2.** From left to right: Crops of the CL (Cluster) image central area for VIS, H160, NIR H, EXT G and IRAC



**Fig. 3.** From left to right: Crops of the CM (Average field) image central area for VIS, H160, NIR H, EXT G and IRAC

4. We test how the shapes of the objects is affected by the selected denoising algorithms checking if they preserve the FWHM of point-like objects, the *ellipticity* and the FWHM of the galaxies profiles.
5. We test the selected algorithms, studying two diagnostics, *completeness* and *purity*, which provide a quality estimate of the catalog produced after an ideal source detection, exploring a combination of SExtractor (Bertin & Arnouts 1996) detection parameters.
6. As last step, we test if the denoised images can be used also for photometry measurements, analyzing if the object fluxes are preserved after denoising.

Finally, we apply the best performing algorithms of our selection on real images acquired from space and ground-based telescope, as described in Sect. 6.

#### 4.1. Implementation details

We compare the different images, following always the same procedure here described:

- The Original image is scaled to the range  $[0, 1]$ <sup>5</sup>
- The Original image is filtered by the denoising algorithm providing the *denoised* image
- The *denoised* image is scaled back to  $[Original_{min}, Original_{max}]$ , where  $Original_{min}$  and  $Original_{max}$  are the minimum and maximum values in the Original image, using the following equation:

$$x_{Original}^i = (Original_{max} - Original_{min}) * x_{[0,1]}^i + Original_{min} \quad (28)$$

where  $x_{Original}^i$  is the  $i$ -th pixel in the Original image and  $x_{[0,1]}^i$  is the  $i$ -th pixel in the *denoised* image scaled to  $[0,1]$

- *MSE* and *SSIM* are computed by comparing the *denoised* image to the noiseless one.

In order to choose the best internal parameter for each denoising algorithm (a list of these parameters is in Sect. 5.1), we used different stopping criteria:

- **ATVD**: In this method the internal parameters are automatically optimized by the algorithm, and a stopping rule is already implemented, through a minimization problem, as described in Sect. 2.5
- **Perona-Malik**: In PM code we have a stopping rule composed by 3 conditions: in the first one we compare at each time step  $MSE_n$  with  $MSE_{n-1}$  where  $MSE_n$  is the *MSE* at the current time step, whereas  $MSE_{n-1}$  is the *MSE* at the previous time step. The code continues running as  $MSE_{n-1} - MSE_n > 0$ . The second condition concerns the number of iterations  $n$ : the code continues running until the number of iterations does not exceeds the maximum number of iterations NMAX, which is set to NMAX=500. The third condition is  $|\frac{MSE_{n-1} - MSE_n}{MSE_{n-1}}| \leq \epsilon$ , with  $\epsilon = 10^{-10}$
- **SPARSE2D**: Starlet and the two b-UWTs automatically estimate the noise background. For the optimal configurations of these three methods, we used the setting provided by the authors and reported at the end of Sect. 2.6. Further information can be found in the documentation file available with the code on the CosmoStat webpage<sup>6</sup>.
- **Other denoising algorithms**: For all the other denoising algorithms we get the optimal values of the main parameter(s) by minimizing the *MSE* with an iterative process. The stopping rule is reached when  $|MSE_{n-1} - MSE_n| \leq \epsilon$ , setting  $\epsilon = 10^{-10}$ , and the number of iterations is lower than the maximum number of iterations NMAX, which is in this case set to NMAX=100.

<sup>5</sup> The scaling is required only by PM methods which need values between 0 and 1 to work, and Bilateral, which needs only non-negative values for its use. The scaling step has been introduced for the two methods mentioned above and applied to all the methods only for comparison reasons. We verified that for the other algorithms the results do not significantly change if the scaling is not applied.

<sup>6</sup> [http://www.cosmostat.org/wp-content/uploads/2014/12/doc\\_iSAP.pdf](http://www.cosmostat.org/wp-content/uploads/2014/12/doc_iSAP.pdf)

## 5. Results

In this section we analyze and comment in a separate and sequential way the results related to the quality tests, following the same order of the steps used in Sect. 4.

### 5.1. Ranking with MSE and SSIM

In this test we use three metrics to constrain the performance of denoising methods: *MSE*, *SSIM* and *CPU time* (Sect. 4). We give priority to those algorithms that are able to minimize as much as possible the *MSE*, preferring the fastest method (in terms of *CPU time*) and the highest *SSIM* in case of comparable *MSE*. Following this criterion, in this step we identify the best configuration and the main parameters for every algorithm. These results are taken into account separately for all the simulated images presented in Sect. 3.

The main internal parameters identified for the different algorithms are listed below:

- Orthogonal Wavelets: *sigma* - The noise standard deviation used for compute threshold(s)
- NL means: *h* - Cut-off distance in grey levels
- TV Bregman: *weight* - Denoising weight, efficiency of denoising
- TV Chambolle: *weight* - Denoising weight, efficiency of denoising
- Gaussian: *sigma* - Standard deviation for Gaussian kernel
- Bilateral: *sigma\_color* - Standard deviation for gray value distance
- Perona-Malik: *T* - Number of iterations of the anisotropic diffusion
- ATVD (TVL1,TVL2,TVG): *lambda* - Structural-Texture splitting parameter
- BM3D: *sigma* - The noise standard deviation.

Further details for the algorithms implemented in Python and the measurement of *MSE* and *SSIM* can be found in the *scikit-image* documentation<sup>2</sup>. The method used to identify the best internal parameter for each algorithm is described in Sect. 4.1. In Appendix A we show the best *MSE* and *CPU time* values of every algorithm, for the different crops. The tables are organized to record the best *MSE* and *CPU time* values obtained with the algorithms. The columns represent the different image simulated filters and the value indicated in bold is the lowest of the column. Tables A.1-A.2-A.3 contain the *MSE* values for the crops BG, CM and CL, respectively. Table A.4 contains the *CPU time* values for the crop CM, after fixing the optimal internal parameters. We remind that in the following "PSF filtering" amounts to filtering with a Gaussian whose FWHM is the same as the PSF-FWHM, whereas in the case of the "Gaussian filtering" the  $\sigma$  (and thus the FWHM) is a free internal parameter. We shortly summarize here the main results:

- TVL2, BM3D, Starlet, the two b-UWTs, PM, NLmeans slow, TV Chambolle always yield good performance, typically providing the lowest values of *MSE*
- TVL2, BM3D, Starlet, the two b-UWTs, PM, NLmeans slow, TV Chambolle always perform better than Gaussian filtering, with the only exception of the IRAC image (we discuss the IRAC situation below in Sect. 5.2)
- the *MSE* of all the methods is proportional to the pixel scale of the image, so that low sampling implies worse results
- in most cases (with the exception of IRAC, which we discuss below), the PSF filtering provides a larger (i.e. worse) value of the *MSE* compared to the one provided by Gaussian filtering.

- in some cases, the *MSE* of the denoised image is larger (i.e. worse) than the one measured without denoising the image at all. Indeed some algorithms in the situations listed below tend to over-smooth the image, providing a worse *MSE*. This event occurs:
  - a) in VIS (CM) image, in the case of the PSF filtering
  - b) in all the H160 images for both the PSF filtering and TV Bregman
  - c) 2-4 times in NIR H images, for methods NLmeans fast, Orthogonal Wavelets, TV Bregman and PSF filtering, and only once for BM3D
  - d) only once in EXT G (CM), for the PSF filtering
  - e) 4 to 5 times in IRAC images, for NLmeans slow, NLmeans fast, TV Bregman, Orthogonal Wavelets, and PSF filtering.
- the *SSIM* ranking typically reflects the *MSE* ranking, pointing out the same group of best algorithms found in the *MSE* ranking; even if some positions are swapped in few cases, the *SSIM* values provided by the best algorithms are comparable ( $\Delta SSIM < 10^{-4}$ )
- the CPU time table (Table A.4) shows that Gaussian is the fastest algorithm among the ones we tested, followed by PSF and TV Bregman. The CPU time for the other algorithms differ from 1 to 4 order of magnitudes with respect to the Gaussian algorithm. However, the computational times are always manageable, at least for the cases of optimal performance.

If we focus on the algorithms belonging to the same classes of methods, we can note that:

1. all the PM methods yield similar performance, (see Fig. A.1), and therefore we choose to only keep  $g = g_1$  with the parameter  $k$  set to  $k = 10^{-3}$  in the following steps
2. TVL2 performs clearly better than TVG and TVL1. In fact, e.g. in BG,  $1 - \frac{mse}{mse_{original}}$  value is always within 5% from the value provided by the original image (no noise), with the exception of IRAC, where it drops to 0.2, which is still greater than the values provided by TVG and TVL1, as shown in Fig. A.2
3. NLmeans slow performs slightly better than NLmeans fast for H160, VIS, and EXT G, ( $1 - \frac{mse}{mse_{original}}$  differences are within 5% in favor of NLmeans slow) and much better for NIR H and IRAC (where NLmeans fast performs worse than Original). See Fig. A.3
4. TV Chambolle performs better than TV Bregman in H160, NIR H, and IRAC. TV Bregman performs worse than Original, whereas for VIS and EXT G it performs 14% and 3% worse than TV Chambolle, respectively (see Fig. A.3)
5. Bilateral is always within the best performing techniques (see Fig. A.3)
6. BM3D, the two b-UWTs and Starlet are always among the most efficient algorithms (see Fig. A.4 and Tables A.1-A.2-A.3 in Appendix A).

Nevertheless, we keep Gaussian and PSF filtering for reference since they are widely used. Hence, at the end of this first step we are left with 11 methods: PM with edge-stopping function  $g_1$  and  $k = 10^{-3}$ , TVL2, BM3D, Starlet, b-UWT(7/9), b-UWT(7/9)+Wiener, Gaussian, PSF, NL means slow, Bilateral, and TV Chambolle. Following our experiments analysis we decide to discard 9 algorithms: 4 PM methods, TVG and TVL1, NL-means fast, TV Bregman, and Orthogonal Wavelets.

## 5.2. The IRAC results

We note that the IRAC images do not follow the same trends as the other bands. In fact, whereas for all the other images there is always a small group of algorithms which perform better than all the others, for IRAC nearly all the denoising algorithms tend to have similar performance. We investigated the possibility that the number of pixels were not enough (166x166 pixels) to extract significant conclusions from these images and we tested the algorithms on an IRAC (CM) simulation with pixel scale 0.06 arcsec and size of  $1000 \times 1000$  pixels. We noticed that, TV Chambolle, NL means slow and Gaussian provide the best performance ( $MSE \propto 10^{-9}$ ), followed by BM3D, b-UWT(7/9), TV Bregman, TVL2, PM ( $g=g_1$   $k=0.01$ ) ( $MSE \propto 1 \times 10^{-8}$ ), PSF ( $MSE \propto 2 \times 10^{-8}$ ) and then Bilateral, PM ( $g=g_1$   $k=0.001$ ), b-UWT(7/9)+Wiener, Starlet and Orthogonal Wavelets ( $MSE > 2 \times 10^{-8}$ ). After this small test we point out that again the IRAC band does not follow the trend defined in the other bands (even if the *MSE* decreases for all the methods and the Original image), but with the increased number of pixels TV Chambolle, NL means slow and Gaussian are the algorithms which provide the best performance. The low resolution of IRAC here plays a fundamental role, impacting on most of the algorithms performance. This aspect of the algorithms will be described later on, in Sect. 5.4.

## 5.3. Stability against non-stationary Gaussian noise

In this section we discuss the results of tests on images with varying depths, i.e. obtained combining regions observed with different exposure times. To build the dataset, we used a noiseless simulated image  $I_{2exp}$ , which we mirrored along the  $x$ -axis to obtain a new image with two identical vertical halves; then, we added Gaussian noise with  $\sigma = \sigma_{VIS}$  on the lower half  $H_l$  and with  $\sigma = 2\sigma_{VIS}$  on the upper half  $H_u$ . In this way, the two halves of the image contain the very same objects with a different amount of observational noise, as if they had been observed with different exposure times. We applied the algorithms in their optimal configuration (see Sect. 5.1) on the mirrored version of the crops VIS (CM) and VIS (CL), and we calculated *MSE* and *SSIM* in  $H_l$  and  $H_u$  for both. We then compared these results with the ones obtained by the application of the algorithms on the mirrored image with stationary Gaussian noise, with  $\sigma_{VIS}$  and  $2\sigma_{VIS}$ , we refer to these images as  $I_\sigma$  and  $I_{2\sigma}$ , respectively. From the results reported in Tables B.1 and B.2 in Appendix B, obtained on both VIS (CM) and VIS (CL), respectively, we notice that:

- all the algorithms applied on  $H_l$  produce *MSE* values of the same order of magnitude of the ones obtained with  $I_\sigma$
- nearly all of them applied on  $H_u$  produce *MSE* values of the same order of magnitude of the ones obtained with  $I_\sigma$ , with the exception of Starlet and the two b-UWTs, for which the application on  $H_u$  produces a *MSE* which is around 1 order of magnitude larger than the respective *MSE* obtained with  $I_{2\sigma}$
- *MSE* relative variation for BM3D, PSF, Gaussian, PM and TVL2 is  $\propto 10^{-3}$
- For Bilateral, NL means, and TV Chambolle the *MSE* relative variation is  $\propto 10^{-2}$
- For Starlet, and the two b-UWTs methods, the *MSE* relative variation is  $\propto 10^{-1}$
- For all the algorithms, *SSIM* relative variation is  $< 10^{-2}$

We want to point out that *MSE* achieved by methods like Starlet and the two b-UWTs on  $H_l$  is  $\propto 10^{-6} - 10^{-7}$ , meaning

that *MSE* relative variation ( $\sim 10^{-1}$ ) still implies small absolute variations ( $\propto 10^{-6} - 10^{-7}$ ). Finally, we can conclude that the algorithms tested are not influenced (or negligibly influenced in few cases) by images with non-stationary Gaussian noise.

#### 5.4. Stability against FWHM and depth variations

In the second part of this test, we compare the performance of the 11 algorithms with respect to the variation of the FWHM and depth of the images. We consider two cases:

- A 1000x1000 pixels crop of the simulated VIS image convolved with different kernels, to degrade the resolution increasing the FWHM without changing the depth of the image (we considered the cases FWHM = 0.5, 1, 1.5 and 2.0", with the original FWHM being 0.2")
- We decreased the depth of a 1000x1000 pixels crop of the simulated H160 image without changing the FWHM, by adding Gaussian noise with increasing standard deviation  $\sigma$  ( $\times 1, 10, 20, 30$  and 40 times the original one) to the Noiseless image.

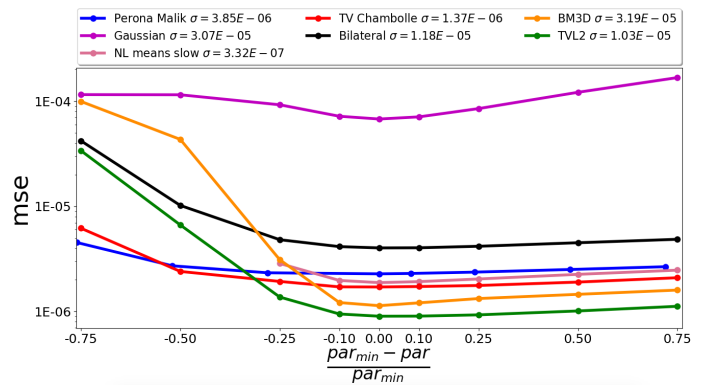
The plots summarizing the results are shown in Figs. C.1- C.6, in Appendix C. We can note that:

1. The MSE calculated on the original image alone decreases at increasing FWHM due to the loss of information (i.e. small objects and details). All the algorithms follow this trend while lowering the *MSE* even more due to the effect of filtering (see Fig. C.1)
2. The ratio between the MSE obtained by each algorithm and the MSE computed on the original image ( $\frac{mse}{mse_{Original}}$ ) increases at increasing FWHM, with the only exception of the Gaussian filtering which instead follows the opposite trend (see Fig. C.2)
3.  $\frac{mse}{mse_{PSF}}$  is weakly affected by variations of the FWHM for most of the denoising methods, with the exception of Gaussian (see Fig. C.3)
4. As expected, the MSE increases at increasing background level (due to the increasing of  $\sigma$  for the Gaussian noise) both in the original image and in the output denoised images for all the algorithms (see Fig. C.4)
5.  $\frac{mse}{mse_{PSF}}$  increases at increasing background level for all the methods (see Fig. C.5)
6.  $\frac{mse}{mse_{Original}}$  decreases at increasing background level for all the methods (see Fig. C.6).

Summarizing, we conclude that the best performance by any denoising algorithm are obtained on images with low SNR and high resolution (narrow FWHM). The best performance with respect to the PSF method are obtained by applying the denoising methods on image with high SNR, regardless of the PSF-FWHM. These results can be used to estimate the efficiency of the denoising algorithms in different situations, pointing out that when applied to high resolution images they provide the best improvements, whereas if applied on low SNR images (where there is the peak of performance), the improvements compared to the PSF are slightly less significant. From these results, it would be very interesting to apply these methods, as an alternative of the PSF filtering, on images with high resolution and high SNR.

#### 5.5. Stability against variations of the parameters

In this test we analyze the selected methods by varying the values of those internal parameters that had been kept fixed to the optimal ones in the previous tests. The goal is to understand whether



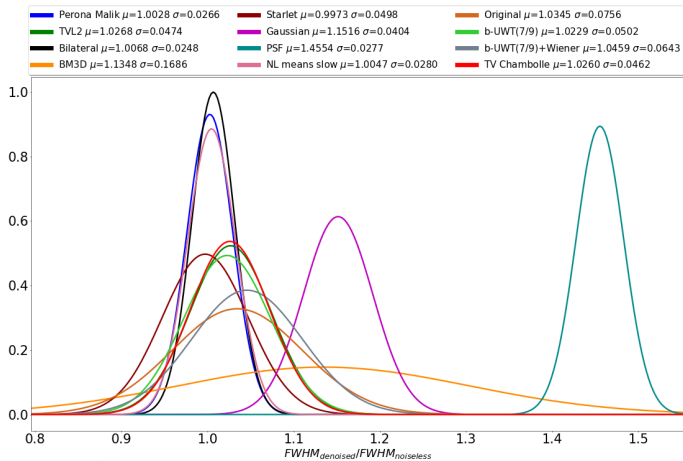
**Fig. 4.** Step 3: Stability against variations of the parameters. Each curve corresponds to a denoising algorithm. We plot the *MSE* against the relative variation of the parameters,  $\frac{par_{min} - par}{par_{min}}$ . Obviously the absolute minimum of the curves is reached in 0 on the x-axis, corresponding to the ideal value of the parameter. In the upper panel we report the standard deviations  $\sigma$  of the  $mse_{mean} - mse$  distribution for each method

the performance are stable against sub-optimal parameter settings. We exclude from this analysis the three denoising methods belonging to the SPARSE2D package, for which we simply used the configuration provided by the authors, reported at the end of Sect. 2.6, and the PSF filtering since it is just a particular case of the Gaussian filtering method. We perform the test on the VIS (CM) image and we change the main parameter value of each technique by  $\pm 10\%$ ,  $\pm 25\%$ ,  $\pm 50\%$  and  $\pm 75\%$  with respect to the value used for the *MSE* analysis (see Subsect. 5.1). The results are shown in Fig. 4. We notice that most of the techniques tend to have similar performance when over-estimating the parameters, remaining relatively stable; on the contrary, under-estimating it significantly worsens the performance. However, all the algorithms (with the exception of BM3D) have a lower dispersion in *MSE* compared to the Gaussian filtering (this is not evident in the plot because of the logarithmic y-axis scale, but we verified it numerically and we give the  $\sigma$  values in the upper panel of the plot), meaning that they are generally more stable against the variation of the parameters. In addition, they yield a  $\sim 1$  order of magnitude lower mean when the parameters are below the optimal value, and by  $\sim 2$  order of magnitudes when they are above it. BM3D like the other algorithms under-performs when its main parameter is under-estimated, producing in this case a large dispersion of the order reached by the Gaussian.

#### 5.6. Conservation of the FWHM and ellipticity

The optimal denoising approach should not significantly alter size and shape of the detected sources so to enable a meaningful scientific analysis. We thus tested the selected methods by measuring the FWHM of the detected sources with SEXTRACTOR, and comparing the measured values to the ones obtained on original, unfiltered images. We perform this test on the simulated VIS image described before, which is mainly populated by galaxies, and on a specific rendition of the simulated VIS image populated by stars distributed on a grid. The results are shown in Figs. 5- 6. Whereas for the stars in Fig. 5 the PSF filtering tends to smooth all the detected object as much as of  $\sim 50\%$  of the FWHM, the other algorithms have a much lower impact (the FWHM is degraded by less than 20% of the original value), even though BM3D has a significant dispersion. Similarly, for

the galaxies in Fig. 6, the PSF filtering causes again a small offset, whereas most of the other methods tend to better preserve the FWHM.



**Fig. 5.** Step 4: FWHM conservation test on stars. On the  $x$ -axis we plot the  $FWHM_{denoised}/FWHM_{noiseless}$ , where  $FWHM_{noiseless}$  is the FWHM of the objects measured on the Noiseless image.  $\mu$  and  $\sigma$  are the mean and the standard deviation of the distribution of  $FWHM_{denoised}/FWHM_{noiseless}$

For a comprehensive comparison, another quantity has been taken into consideration. The *ellipticity* of the galaxies has been measured before and after the application of the denoising algorithms, using the parameter ELLIPTICITY from SExtractor. Looking at Fig. 7, it is possible to notice that most of the algorithms do not modify the ellipticity at an alarming level (even if in some cases the dispersion is far from being optimal, e.g. for Bilateral and TV Chambolle). Nevertheless, the PSF is one of the most performing method for this test.

From these tests, we can conclude that most of the tested algorithms preserve the shape of the sources similarly (in the case of the *ellipticity*) or even better (in the case of the FWHM) than the PSF filtering.

### 5.7. Completeness and purity

In this test - perhaps the crucial one - we checked the quality of the catalogs of sources extracted from the denoised images. We analyze two diagnostics, both relevant to assess the performance of the detection process: namely, the *completeness* and the *purity* as defined below. We extract the catalogs running SExtractor in *dual* image mode using a denoised image as detection band and the original image as measurement band so to perform a cross-correlation between the extracted and the true catalogs of sources both in terms of position and flux.

We used the simulated VIS 5000x5000 pixels image, searching for the best SExtractor parameters configuration for every denoised image. We have thus tested a large number of possible combinations of the two parameters which control the detection, i.e. DETECT\_THRESH (from a minimum value of 0.2 to a maximum of 6.0, with steps of 0.1) and DETECT\_MINAREA (with values: 3,6,9,12,15,30), considering only the combinations for which the quantity  $DETECT\_THRESH * \sqrt{DETECT\_MINAREA} > 1$ , which provides a selection of objects with a global significance of at least  $1-\sigma$ . The number of detection parameters combinations which fulfill this requirement is  $\sim 350$ . We point out that

the best algorithms configurations, used for this test and obtained by *MSE* minimization, do not differ significantly from the best configurations found in the VIS (CM) image (Sect. 4).

We introduce some notations:

- $n_{detected}$  is the total number of detected objects, which includes both real and spurious detections indiscriminately
- $n_{simulated}$  is the number of simulated objects in the image
- $n_{spurious}$  is the number of spurious detections, as identified by the *spurious sources identification approach* described in the following.

The *spurious sources identification approach* that we define for this work is related to the SExtractor cross-correlation, when an association catalog is provided: we denote by  $C_{R_{assoc}}$  the circle centered on the simulated object original position with radius  $R_{assoc}$ , which is the maximal distance allowed for the association made by SExtractor. We set it to 6 pixels (i.e.  $3 \times FWHM$ ). Then, we tag an object as spurious if one of the following two conditions holds:

- is outside  $C_{R_{assoc}}$
- (is inside  $C_{R_{assoc}}$ ) AND  $(|mag_{measured} - mag_{simulated}| > 1.0)$  AND  $(mag_{aperture} - mag_{simulated} > 1.0)$ ,

where  $mag_{measured}$  is SExtractor MAG\_AUTO (an estimation of the total magnitude of the source),  $mag_{simulated}$  is the true magnitude of the simulated object, and  $mag_{aperture}$  is SExtractor MAG\_APER corresponding to the magnitude within a circular aperture with diameter of 6 pixels.

Finally, we can now define the two diagnostics:

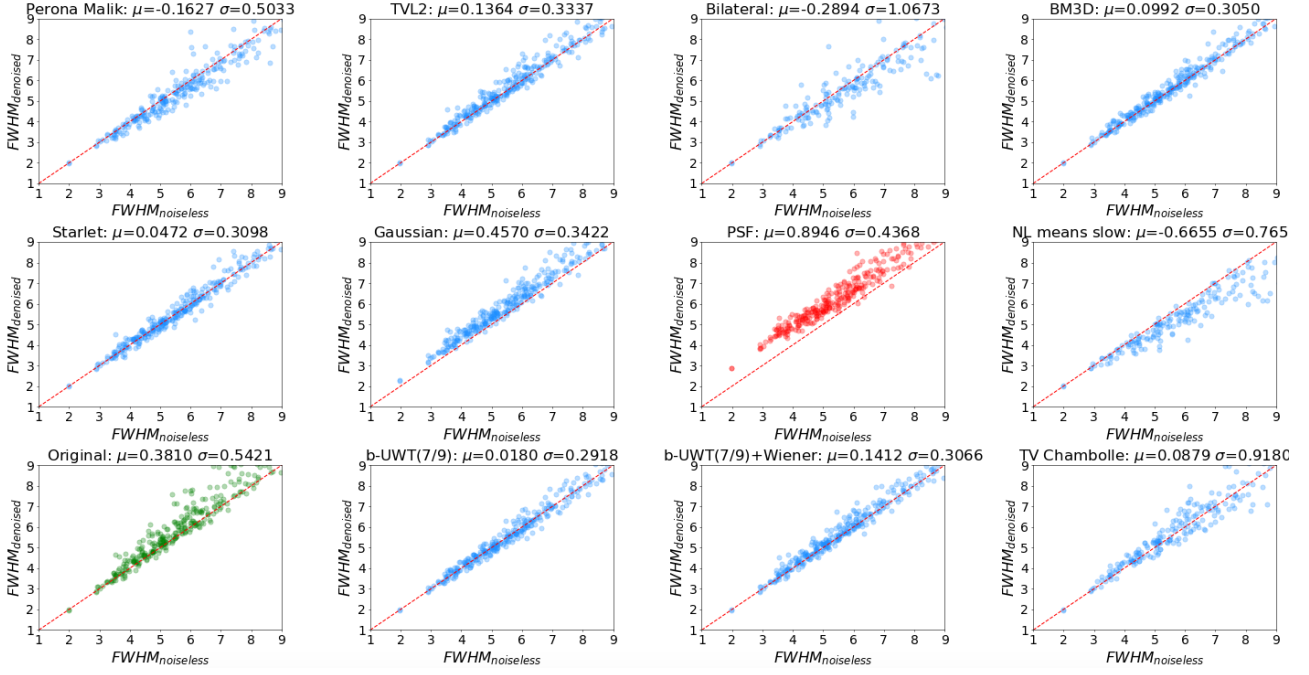
$$completeness := \frac{n_{detected} - n_{spurious}}{n_{simulated}} \quad (29)$$

$$purity := 1 - \frac{n_{spurious}}{n_{simulated}} \quad (30)$$

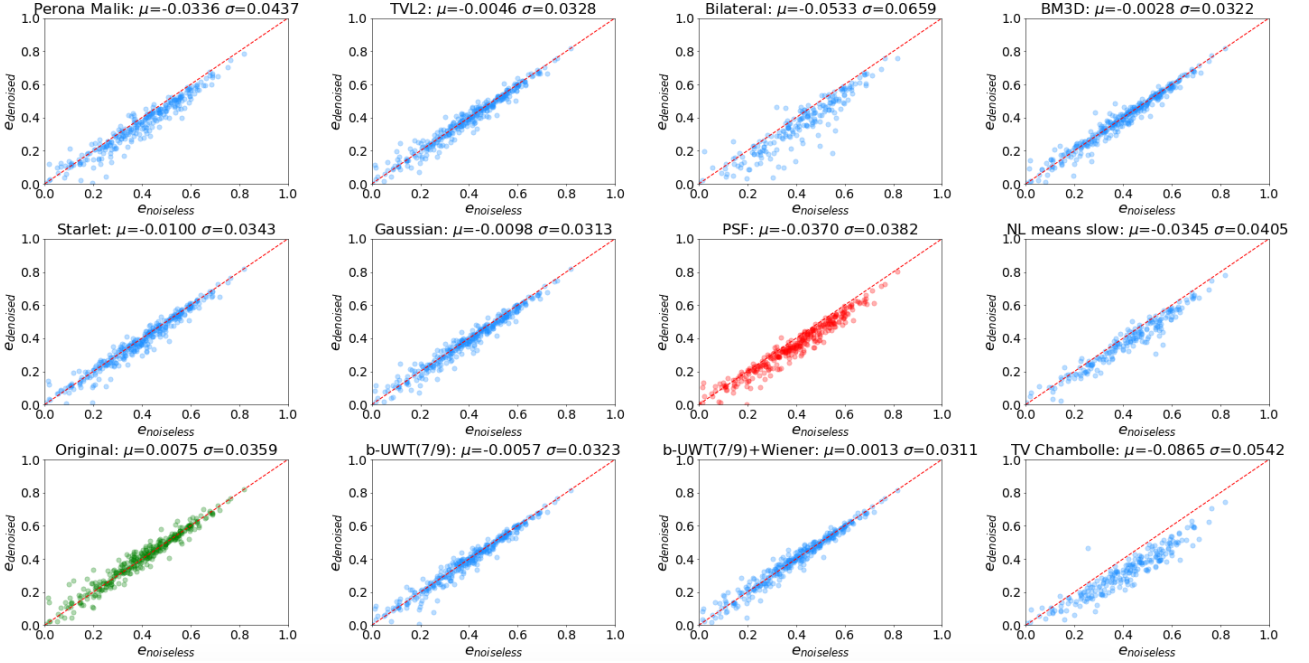
where  $purity = purity_{assoc}$ , determined by the association approach defined above. We measure completeness and purity in 0.2 magnitudes bins. In Fig. 8 we plot the magnitudes at which the completeness drops below 50% against the one at which the purity drops below 90%. Each symbol corresponds to a different denoising technique, and repetitions of the same symbols correspond to different combinations of the detection parameters for the same algorithm. For readability, a maximum of 5 combinations per algorithm (corresponding to the best ones) are shown in the figure.

We note that all the methods improve the detection. BM3D performs like the PSF, whereas methods like Starlet and b-UWT(7/9)+Wiener produce remarkable results, with a 0.2 increment in completeness with respect to the PSF, but the best performance are reached by TVL2, Perona-Malik, TV Chambolle, and Bilateral. Indeed, these 4 methods reach the completeness threshold 0.6 mag deeper, and the purity threshold 0.8 magnitude deeper than the non-denoised run. Moreover, they improve the detection compared to the PSF smoothing, reaching 0.2 magnitudes deeper in both completeness and purity.

It is tempting to consider the *MSE* and *SSIM* measured on the VIS images used for the completeness and purity analysis, searching for a possible correlation between the metrics and the diagnostics. In Fig. 9 the plots are produced using the results shown in Fig. 8. We find no or weak correlation between *MSE* (*SSIM*) and purity, whereas a stronger correlation exists between *MSE* (*SSIM*) and completeness.



**Fig. 6.** Step 4: FWHM conservation test on galaxies. On the  $x$ -axis we plot the FWHM of the objects measured on Noiseless image  $FWHM_{noiseless}$ , whereas on the  $y$ -axis we plot the FWHM measured on the Original image after the application of the denoising algorithms  $FWHM_{denoised}$ .  $\mu$  and  $\sigma$  are the mean and the standard deviation of the distribution of  $FWHM_{denoised} - FWHM_{noiseless}$ .

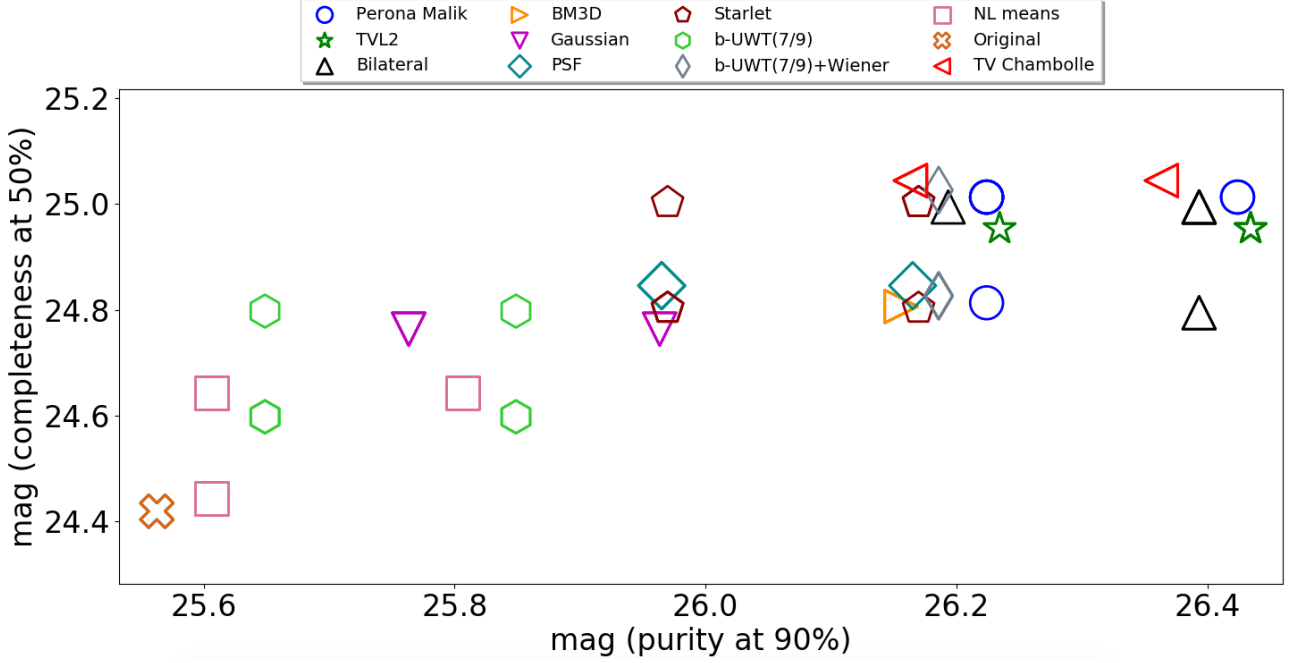


**Fig. 7.** Step 4: Ellipticity conservation test on galaxies. On the  $x$ -axis we plot the ellipticity  $e$  of the objects measured on Noiseless image  $e_{noiseless}$ , whereas on the  $y$ -axis we plot the  $e$  measured on the Original image after the application of the denoising algorithms  $e_{denoised}$ .  $\mu$  and  $\sigma$  are the mean and the standard deviation of the distribution of  $e_{denoised} - e_{noiseless}$ .

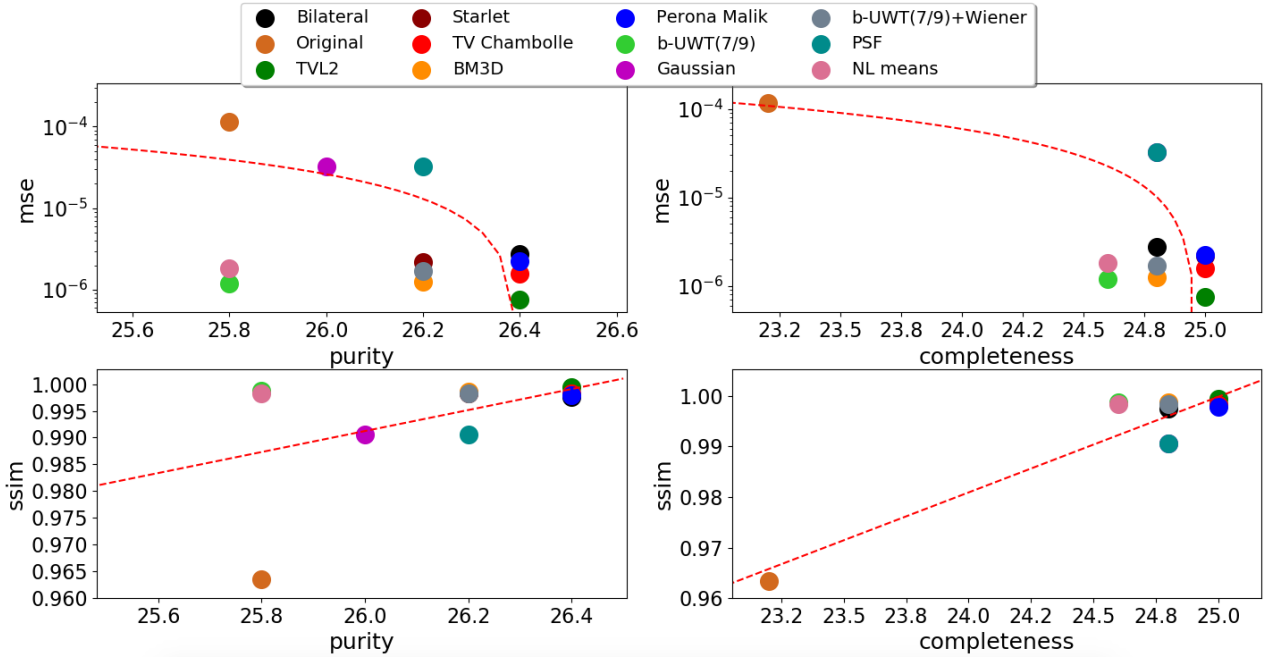
We show the snapshots of a sample of objects detected by the different methods in the VIS image in Appendix D. These snapshots give a visual match of objects detected in the denoised images. We only show the best performing algorithms results compared to the Original, PSF-filtered and the Noiseless images. For VIS the algorithms are: PM, TVL2, Bilateral, and TV Chambolle, followed by BM3D, Starlet, b-UWT(7/9), and b-UWT(7/9)+Wiener.

### 5.8. Conservation of the flux

In this final test, we compare the total fluxes (by using SExtractor MAG\_AUTO) measured on the simulated denoised images with the true input fluxes for objects with magnitude within 19 and 23. We do not consider here the Gaussian filtering but only the PSF filtering, since the last one is the reference method for this analysis. The results are shown in Figs. 10-11. The standard deviation of the difference between measured and true magnitudes is



**Fig. 8.** Step 5: Completeness and purity test. We extracted catalogs on the VIS simulated image processed with the denoising algorithms, using different configurations of SExtractor. We plot the magnitude at which the completeness drops below 50% against the magnitude at which the purity drops below 90%. Each symbol corresponds to a different denoising method, which can be present multiple times in the plot due to different combinations of detection parameters. The positions of the symbols are slightly randomized to improve readability.

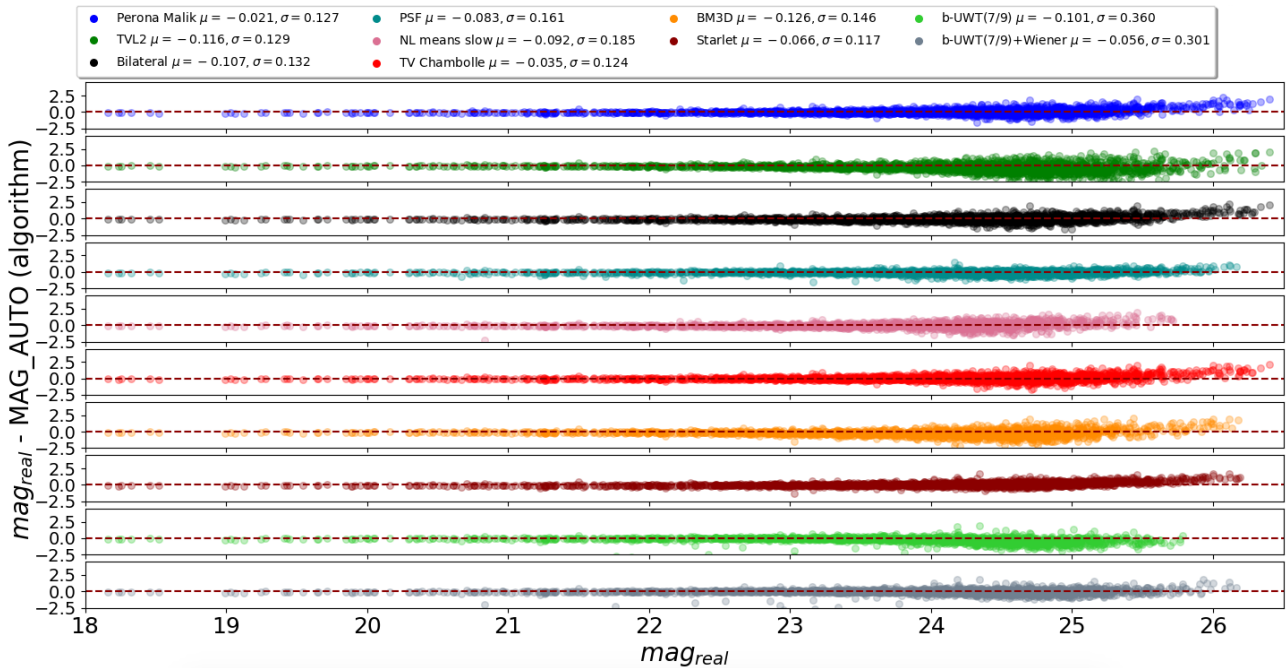


**Fig. 9.** Step 5: Correlation between *MSE* or *SSIM* and purity or completeness. On the *x*-axis we plot the magnitudes at which completeness (purity) reaches 50% (90%), whereas on the *y*-axis we plot the metrics *MSE* or *SSIM*. Dashed lines are the linear best-fitting.

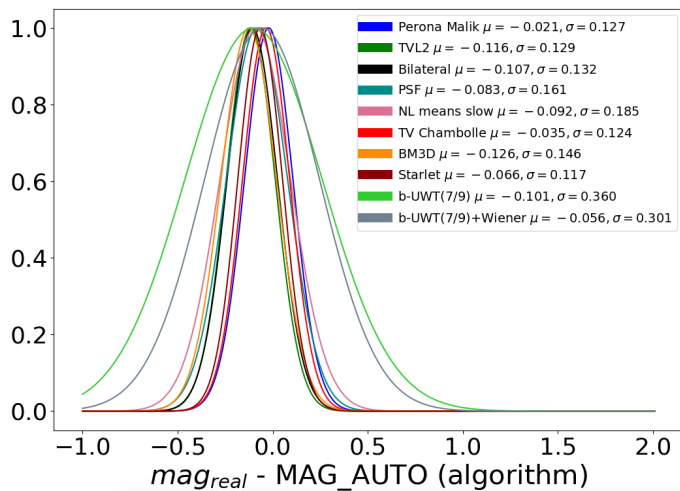
$\sim 0.16$  for PSF-filtered images. All denoising methods show similar performance with the exception of b-UWT(7/9) ( $\sigma_{mag}=0.36$ ) and b-UWT(7/9)+Wiener ( $\sigma_{mag}=0.30$ ). We conclude that denoising algorithms preserve the overall calibration of the input images and they enable a photometric accuracy comparable to the one usually achieved on images filtered with the PSF.

## 6. Test on real images

After having analyzed the performance of denoising techniques on a series of simulated images, testing their performance with stationary and non-stationary Gaussian noise, we test the algorithms on real images, using the HST *H160* observations of the GOODS-South Field and a crop of the HAWK-I survey. Notably, the analysis of real observations provides us with a straightfor-



**Fig. 10.** Step 6: Flux conservation distribution for objects with magnitude within 19 and 23. On the x-axis the real objects magnitude  $mag_{real}$ . On the y-axis, the difference between the magnitude measured  $MAG\_AUTO$  and  $mag_{real}$ . Only the detected objects within the purity and completeness thresholds (Sect. 5.7) are considered.  $\mu$  and  $\sigma$  are the distribution mean and the standard deviation values, respectively.



**Fig. 11.** Step 6: Flux conservation distribution for objects with magnitude within 19 and 23. On the x-axis, the difference between the magnitude measured  $MAG\_AUTO$  and the real objects magnitude from the catalog ( $mag_{real}$ ). On the y-axis the  $MAG\_AUTO - mag_{real}$  probability distribution function. Only the detected objects within the purity and completeness thresholds (Sect. 5.7) are considered.  $\mu$  and  $\sigma$  are the distribution mean and the standard deviation values.

ward test of more realistic situations, in particular concerning the presence of non-Gaussian noise and/or correlated noise.

### 6.1. Space telescope images

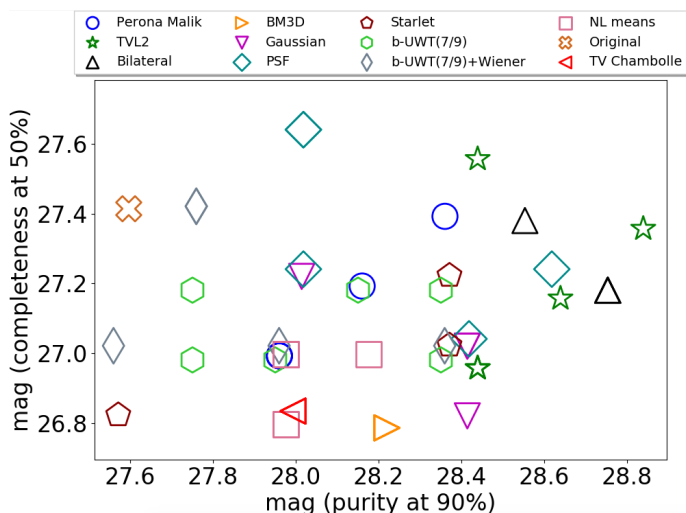
We use two images of the area of the Hubble Ultra Deep Field: one at the full depth released with the official CANDELS mosaics that includes all WFC3 observations of that region (HUDF09, reaching  $H160=29.74$  at  $SNR=5$ ), the second, shallower one at the depth obtained with WFC3 observations of

the CANDELS Treasury Program alone (GSDEEP,  $H160=28.16$  at  $SNR=5$ ) (Koekemoer et al. 2011; Grogin et al. 2011). We will use the former, deeper image as "true sky", against which we will compare the performance of denoising techniques on the shallower image. Using an analysis similar to that in Sect. 5.7, we take as reference catalog the one obtained running SExtractor on HUDF09 with conservative detection parameters. The goal is again to check completeness and purity. We use again an association radius  $R_{assoc}$  of  $3 \times FWHM$ , now corresponding to 7.5 pixels. We identify an object as spurious using the same criteria used in Sect. 5.7 with a  $mag_{aperture}$  within a circular aperture with diameter of 7.5 pixels. The resulting plot, visible in Fig. 12, is similar to the one obtained on the simulated image (Fig. 8). Clearly, TVL2 outperforms all the other algorithms, in particular the PSF by 0.2 mag in completeness and purity (or alternatively 0.4 magnitudes more complete and 0.2 magnitudes less pure); Bilateral performs better than the PSF filtering, by an amount of 0.2 magnitudes in completeness. Perona-Malik provides a 0.2 mag more complete and 0.2 less pure catalog, being in this way an alternative of the PSF filtering. Starlet and b-UWT(7/9) perform slightly worse than the PSF.

As done in Sect. 5.7, we show the snapshots of a sample of objects detected by the different methods in the GS-DEEP image in Appendix E. The algorithms reported are: PM, TVL2, Bilateral, NL means, BM3D, Starlet, b-UWT(7/9), and b-UWT(7/9)+Wiener.

### 6.2. Ground-based images

We repeat the same tests described above on two Ks-band observations of the Goods-South field acquired with the HAWK-I imager at the VLT: a shallower observation of the field presented in Castellano et al. (2010) and the  $\sim 2$  magnitude deeper observation released by the HUGS Survey (Fontana et al. 2014, see Table 1). As done before, we use the deepest image as "true sky" and we apply the algorithms to reduce the noise on the

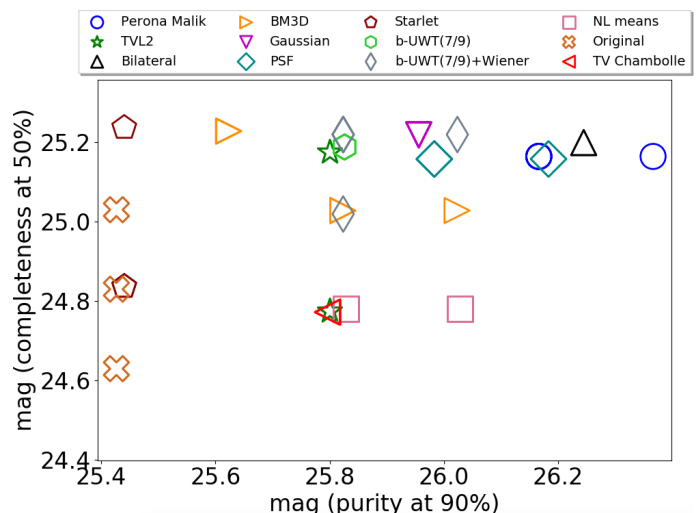


**Fig. 12.** Space telescope real Images Completeness & Purity (GS-DEEP and HUDF09). On the x-axis the magnitude at which the purity drops below 90%, on the y-axis the magnitude at which the completeness drops below 50%. Each symbol is referred to a different denoising method, which can be present multiple times in the plot due to different combinations of detection parameters, see text for details. The positions of the symbols are slightly randomized to improve readability.

shallow image. We use again the association radius  $R_{assoc}$  of  $3 \times FWHM$  corresponding to 11.25 pixels, with the relative  $mag_{aperture}$  (11.25 pixels diameter), identifying an object as spurious using again the same criteria already used in Sect. 5.7 and Sect. 6.1. The resulting plot (see Fig. 13) shows that these algorithms improve the image detection compared to not making denoising at all (same result obtained in Sect. 5.7 and Sect. 6.1), whereas they do not provide significant improvements compared to the PSF. Indeed, only PM creates a catalog of 0.2 magnitudes more pure at the same completeness. These results are in agreement with those in Sect. 5.4, where we noticed that all these methods give the best with high resolution images (see Fig. C.2), such as VIS and H160. Indeed the lower resolution of the ground-based images impacts the algorithms performance. In the same way the methods, and mainly the PSF, perform better for lower SNR images (e.g. HAWK-I compared to VIS and H160), as shown in Figs. C.5-C.6, resulting in less significant improvements from the methods compared to the PSF.

### 6.3. Artifacts and visual inspection

The analysis made in 5.7 only considers the objects SNR and magnitude to classify the detections as *correct* or *spurious*. The algorithms perform denoising following different strategies, and artifacts related to the family of these methods can be created. Several crops denoised by different tested methods are reported in Appendix D-E. By visually inspecting such snapshots, we can get an idea of the features and the artifacts produced. As a general comment, looking at the figures reported in Appendix D, TVL2 seems to be the best method, closer to the noiseless image reported in the last column of all the figures, followed by Starlet, b-UWT(7/9), and b-UWT(7/9)+Wiener, which also produce good results. PM, TV Chambolle, and BM3D are often very similar from a qualitative point of view. Most of the time the flux of the objects detected by TVL2 is highly concentrated (see e.g. Figs. D.2, D.5), but in some cases the objects flux is distributed on a larger area, as visible e.g. in Fig. D.6. Starlet produces in



**Fig. 13.** Ground-based real Images Completeness & Purity (HAWK-I and HAWK-I UDS). On the x-axis the magnitude at which the purity drops below 90%, on the y-axis the magnitude at which the completeness drops below 50%. Each symbol is referred to a different denoising method, which can be present multiple times in the plot due to different combinations of detection parameters, see text for details. The positions of the symbols are slightly randomized to improve readability.

some cases bright isotropic objects, which could lead to misleading morphological information (see e.g. Figs. D.14, D.16). Looking at Fig. D.10, PSF seems to detect wrongly two objects, differently from the other denoising methods. This qualitative analysis is confirmed by the corresponding quantitative analysis made, as visible in Fig. 8.

When more than one object is present in the figures, the different schemes sometimes are not able to recognize all the objects (see e.g. Figs. D.3, D.6, D.9, D.11, D.13). Moreover, fluctuations of the background around the objects are not completely removed, and the smoothing can create artifacts (see e.g. Fig. D.13, where an elliptical galaxy seems to appear as a spiral galaxy after denoising). b-UWT(7/9), and b-UWT(7/9)+Wiener are susceptible to background fluctuations and create visual artifacts (see Figs. D.17, D.18).

Inspecting the snapshots from real images, a qualitative analysis is inconclusive, so we have to stick to the quantitative analysis in Fig. 12. In fact, looking at the crops reported in Appendix E, PM and Bilateral are often so similar that it is difficult to distinguish them, although from Fig. 12 the differences between them are clear (PM performs worse than Bilateral). Analogous remark can be made for BM3D, which produces images similar to PM and Bilateral, as visible e.g. in Fig. E.12. TVL2 is again the most promising method, visually close to the HUDF09 also thanks to the automatic optimization of its internal parameters, and this is confirmed by the results reported in Fig. 12. Images from Starlet, b-UWT(7/9), and b-UWT(7/9)+Wiener are quite similar to each other, even though those from Starlet present less visual artifacts (see Figs. E.10, E.12, E.13, E.14). Differently from the other methods, NL means produces nearly squared patches of uniform flux, which effectively reduce noise fluctuations (see Figs. E.8, E.9), but also could lead to artifacts (see e.g. Fig. E.5, where the galaxy in the bottom-right corner assumes a nearly boxy-shape).

Even if artifacts are generated, and the morphology of the objects detected could be altered, we tested the efficiency of the algorithms in preserving the FWHM and the ellipticity of the

objects in Sect. 5.6, proving that on the average the shapes are preserved. A more detailed assessment of the effect on galaxy morphology is beyond the scope of the present work. However, we note that a promising improvement in this direction can be obtained by classifying objects using a convolutional neural network (e.g. one of the classifiers proposed in Tuccillo et al. 2017; Khalifa et al. 2017; Barchi et al. 2020), in order to assess whether their morphological class is preserved after denoising.

## 7. Summary and conclusions

The goal of this work is to make an extensive comparison between a number of denoising algorithms, aimed at identifying the best choice to improve the detection of faint objects in astronomical extragalactic images (e.g. considering the typical cases of HST and Euclid). To this purpose, we performed a large set of tests on simulated images. We also tested the methods on real images: from space and ground-based, collecting really interesting hints on many situations.

We chose to test a significant number of denoising algorithms based on traditional techniques (mainly, PDEs and variational methods), leaving a more complete comparison, including machine learning techniques, for future works. In particular, we point out that ATVD-TVL2, Bilateral, Perona-Malik, TV Chambolle, Starlet, and b-UWT(7/9)+Wiener are the most interesting to use among all, since they provide good performance in the different tests proposed, closely followed by BM3D. Even if most of these methods are quite unusual for the astronomical community, they are very well-known in many other fields. They are known to outperform a straightforward PSF/Gaussian filtering, which is the standard choice in astronomy. We therefore considered these techniques as the reference ones, against which we tested all the other methods.

As a first test, we considered the two metrics *MSE* and *SSIM* (defined in Sect. 4) and checked which methods yield the best performance with respect to them. We compared their performance again through *MSE* and *SSIM* in relation to depth, resolution and type of image. We tested the algorithms ability to preserve the FWHM, in order to understand if they can preserve the shape of the objects, useful in case photometric measurements on the denoised image are needed. We tested their stability using the *MSE*, varying the ideal parameter of a fixed percentage, with the goal of having a hint on their reliability, in case the best parameter is chosen wrongly. We have also tested possible detection improvements through two diagnostics, *completeness* and *purity*, which are used to measure the fraction of real detections on the total number of objects in the image. Finally, we applied these methods on real images (CANDELS-GS-deep and a crop of HAWK-I). We summarize below the key points of the analysis performed in this paper:

- From *MSE* and *SSIM* we noticed that 11 algorithms are always on top of rankings, especially for VIS and H160 images, which are of main interest for the detection in future surveys. These algorithms are: PM with edge-stopping function  $g_1$  and  $k = 10^{-3}$ , TVL2, BM3D, Starlet, b-UWT(7/9), b-UWT(7/9)+Wiener, Gaussian, PSF, NL means slow, Bilateral, and TV Chambolle
- From the stability test discussed in Sect. 5.3, the algorithms tested are not influenced (or negligibly influenced in few cases) by images with non-stationary Gaussian noise
- From the PSF and Depth variation test we noticed that most of the methods are performing better with narrower FWHM. Whereas all the methods perform better with the noise level

increasing (in terms of Gaussian noise standard deviation), their improvements are more significant compared to the PSF, with high SNR images

- From the FWHM conservation test, we noticed that most of the algorithms tend to not smooth the image, in terms of FWHM increments. On the contrary, the PSF smoothing provides an offset in the FWHM measurement, for both, stars-only image and galaxies-image. On the other hand, the ellipticity is well preserved by the PSF and most of the algorithms
- From the completeness and purity test we found a small number of algorithms which provide 0.2 magnitudes more pure and complete catalogs than the PSF filtering, these are TVL2, Perona-Malik, TV Chambolle and Bilateral (whereas Starlet and b-UWT(7/9)+Wiener provide a 0.2 increment only for completeness)
- From Flux conservation test we found that most of the algorithms have similar performance to the PSF filtering, preserving the overall calibration of the input images
- From real image test (H160) we found that TVL2 outperforms all the other algorithms, and it is the only one that performs better than the PSF of 0.2 magnitudes in completeness and purity, whereas Bilateral produces only a 0.2 more pure catalog
- From real image test (HAWK-I) we found that only Perona-Malik outperforms the PSF filtering, by 0.2 magnitudes in purity, the other methods performs worse/similarly to the PSF.
- From the visual inspection performed in Sect. 6.3 on simulated and real images, the best methods seems to be TVL2, PM, Bilateral, and Starlet, although they also generates artifacts like the other methods tested. The visual inspection confirms the results visible in Fig. 8 and Fig. 12.

The results we obtained demonstrate that denoising algorithms should be considered valuable tools in presence of Gaussian noise, which is a good approximation of the noise in optical and near-infrared extragalactic surveys, as they clearly improve the detection of faint objects. Structure-texture image decomposition, Total Variation denoising, Perona-Malik, Bilateral filtering, and Undecimated Wavelets Transform methods are of particular interest. While further specific tests are needed to define for each survey the optimal approach, and its parameter set, among the above mentioned ones, our investigation on a small but reasoned reference set of simulated and real extragalactic images shows that the scientific return of ongoing and future surveys can be significantly enhanced by the adoption of these denoising methods in place of standard filtering approaches. Moreover, in our opinion the use of the increasingly popular machine learning techniques, possibly combined with the best methods coming out from our analysis, has the potential to further improve on the performance of "traditional" denoising techniques described here. Also learning approaches for adding morphological priors (see e.g. Peyré et al. 2010) could be useful, subject of future works.

*Acknowledgements.* We would like to thank the referee, Prof. Jean-Luc Starck, for the many useful comments and suggestions that helped us improving our work. This research has been carried on within the INdAM-INAF project FOE 2015 "OTTICA ADATTIVA". The authors S. Tozza and M. Falcone are members of the INdAM Research Group GNCS, and gratefully acknowledge its financial support to this research.

## References

Ackermann, M., Ajello, M., Allafort, A., et al. 2013, ApJS, 209, 34

- Aujol, J.-F., Gilboa, G., Chan, T., & Osher, S. 2006, *International Journal of Computer Vision*, 67, 111
- Banterle, F., Corsini, M., Cignoni, P., & Scopigno, R. 2012, *Computer Graphics Forum*, 31, 19
- Barchi, P., de Carvalho, R., Rosa, R., et al. 2020, *Astronomy and Computing*, 30, 100334
- Beckouche, S., Starck, J.-L., & Fadili, J. 2013, *Astronomy and Astrophysics*, 556, 132
- Bertero, M. & Boccacci, P. 1998, *Introduction to inverse problems in imaging* (CRC press)
- Bertero, M. & Piana, M. 2006, *Inverse problems in biomedical imaging: modeling and methods of solution*, ed. A. Quarteroni, L. Formaggia, & A. Veneziani (Milano: Springer Milan), 1–33
- Bertin, E. 2009, *Mem. Soc. Astron. Italiana*, 80, 422
- Bertin, E. & Arnouts, S. 1996, *A&AS*, 117, 393
- Black, M. J., Sapiro, G., Marimont, D. H., & Heeger, D. 1998, *IEEE Transactions on Image Processing*, 7, 421
- Buades, A., Coll, B., & Morel, J.-M. 2005, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 2, 60–65 vol. 2
- Buades, A., Coll, B., & Morel, J.-M. 2011, *Image Processing On Line*, 1, 208
- Bush, J. 2011, Master's thesis, University of California, Santa Barbara, [http://www.math.ucsb.edu/~cgarcia/UGProjects/BregmanAlgorithms\\_JacquelineBush.pdf](http://www.math.ucsb.edu/~cgarcia/UGProjects/BregmanAlgorithms_JacquelineBush.pdf)
- Castellano, M., Fontana, A., Boutsia, K., et al. 2010, *A&A*, 511, A20
- Castellano, M., Ottaviani, D., Fontana, A., et al. 2015, in *Astronomical Society of the Pacific Conference Series*, Vol. 495, *Astronomical Data Analysis Software and Systems XXIV (ADASS XXIV)*, ed. A. R. Taylor & E. Rosolowsky, 257–260
- Chambolle, A. 2004, *Journal of Mathematical Imaging and Vision*, 20, 89
- Chang, S. G., Yu, B., & Vetterli, M. 2000, *IEEE Transactions on Image Processing*, 9, 1532
- Dabov, K., Foi, A., Katkovnik, V., & Egiazarian, K. 2007, *IEEE Transactions on Image Processing*, 16, 2080
- Darbon, J., Cunha, A., Chan, T. F., Osher, S., & Jensen, G. J. 2008, in *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 1331–1334
- Dark Energy Survey Collaboration, Abbott, T., Abdalla, F. B., et al. 2016, *MNRAS*, 460, 1270
- Donoho, D. L. & Johnstone, J. M. 1994, *Biometrika*, 81, 425
- Fontana, A., Dunlop, J. S., Paris, D., et al. 2014, *A&A*, 570, A11
- Frontera-Pons, J., Sureau, F., Bobin, J., & Le Floch, E. 2017, *A&A*, 603, A60
- Gabor, D. 1965, *Laboratory Investigation*, 14, 801
- Getreuer, P. 2012, *Image Processing On Line*, 2, 74
- Goldstein, T. & Osher, S. 2009, *SIAM Journal on Imaging Sciences*, 2, 323
- Grogin, N. A., Kocevski, D. D., Faber, S. M., et al. 2011, *ApJS*, 197, 35
- Guo, Y., Ferguson, H. C., Giavalisco, M., et al. 2013, *ApJS*, 207, 24
- Guo, Z., Sun, J., Zhang, D., & Wu, B. 2012, *IEEE Transactions on Image Processing*, 21, 958
- Jones, E., Oliphant, T., Peterson, P., et al. 2001, *SciPy: Open source scientific tools for Python*, [Online; accessed <today>]
- Karmakar, A., Mishra, D., & Tej, A. 2018, *arXiv e-prints*, arXiv:1809.01434
- Khalifa, N. E. M., Taha, M. H. N., Hassanien, A. E., & Selim, I. M. 2017, *arXiv e-prints*, arXiv:1709.02245
- Koekemoer, A. M., Faber, S. M., Ferguson, H. C., et al. 2011, *ApJS*, 197, 36
- Kolmogorov, A. M. & Fomin, S. V. 1957, *Elements of the theory of functions and functional analysis* (Dover Publications)
- Laureijs, R., Amiaux, J., Arduini, S., et al. 2011, *arXiv e-prints*, arXiv:1110.3193
- Lee, G. R., Gommers, R., Waselewski, F., Wohlfahrt, K., & O'Leary, A. 2019, *Journal of Open Source Software*, 4, 1237
- Lesser, M. 2015, *PASP*, 127, 1097
- Lindenbaum, M., Fischer, M., & Bruckstein, A. 1994, *Pattern Recognition*, 27, 1
- LSST Science Collaboration, Abell, P. A., Allison, J., et al. 2009, *arXiv e-prints*, arXiv:0912.0201
- Mallat, S. 2001, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 11, 674
- Meyer, Y. 1990, *Ondelettes et opérateurs: Ondelettes* (Hermann)
- Neyman, J. & Pearson, E. S. 1933, 231 (694706); 289337
- Oke, J. B. & Gunn, J. E. 1983, *ApJ*, 266, 713
- Perona, P. & Malik, J. 1990, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12, 629
- Peyré, G., Fadili, J., & Starck, J.-L. 2010, *SIAM Journal on Imaging Sciences*, 3, 646
- Pierre, M., Valtchanov, I., Altieri, B., et al. 2004, *J. Cosmology Astropart. Phys.*, 2004, 011
- Principe, G., Malyshev, D., Ballet, J., & Funk, S. 2018, *A&A*, 618, A22
- Romeo, A. B., Horellou, C., & Bergh, J. 2003, *MNRAS*, 342, 337
- Rudin, L. I., Osher, S., & Fatemi, E. 1992, *Physica D: Nonlinear Phenomena*, 60, 259
- Schreiber, C., Elbaz, D., Pannella, M., et al. 2017, *A&A*, 602, A96
- Shen, H., George, D., Huerta, E. A., & Zhao, Z. 2017, *arXiv e-prints*, arXiv:1711.09919
- Shensa, M. J. 1992, *IEEE Transactions on Signal Processing*, 40, 2464
- Skelton, R. E., Whitaker, K. E., Momcheva, I. G., et al. 2014, *ApJS*, 214, 24
- Spergel, D., Gehrels, N., Baltay, C., et al. 2015, *arXiv e-prints*, arXiv:1503.03757
- Starck, J. L., Aghanim, N., & Forni, O. 2004, *A&A*, 416, 9
- Starck, J. L., Moudden, Y., Abrial, P., & Nguyen, M. 2006, *A&A*, 446, 1191
- Starck, J.-L., Murtagh, F., & Bertero, M. 2014, *Starlet Transform in Astronomical Data Processing*, 1–40
- Starck, J.-L., Murtagh, F., & Fadili, J. 2010, *Sparse Image and Signal Processing: Wavelets and Related Geometric Multiscale Analysis* (Cambridge University Press)
- Tomasi, C. & Manduchi, R. 1998, in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, 839–846
- Tuccillo, D., Huertas-Company, M., Decenci re, E., & Velasco-Forero, S. 2017, *Proceedings of the International Astronomical Union*, 12
- Valens, C. 1999, *A Really Friendly Guide to Wavelets*
- van der Walt, S., Sch nberger, J. L., Nunez-Iglesias, J., et al. 2014, *PeerJ*, 2, e453
- Vetterli, M. & Kova evic, J. 1995, *Wavelets and Subband Coding* (Upper Saddle River, NJ, USA: Prentice-Hall, Inc.)
- Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.-A. 2008, in *Proceedings of the 25th International Conference on Machine Learning, ICML '08* (New York, NY, USA: Association for Computing Machinery), 1096–1103
- Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. 2010, *J. Mach. Learn. Res.*, 11, 3371
- Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. 2004, *IEEE Transactions on Image Processing*, 13, 600
- Weickert, J. 1998, *Anisotropic diffusion in image processing* (Teubner)
- Winkler, A. M. 2011, *Gaussian kernels: convert FWHM to sigma*
- Xie, J., Xu, L., & Chen, E. 2012, in *Advances in Neural Information Processing Systems 25*, ed. F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Curran Associates, Inc.), 341–349

Appendix A: MSE comparison tables and plots

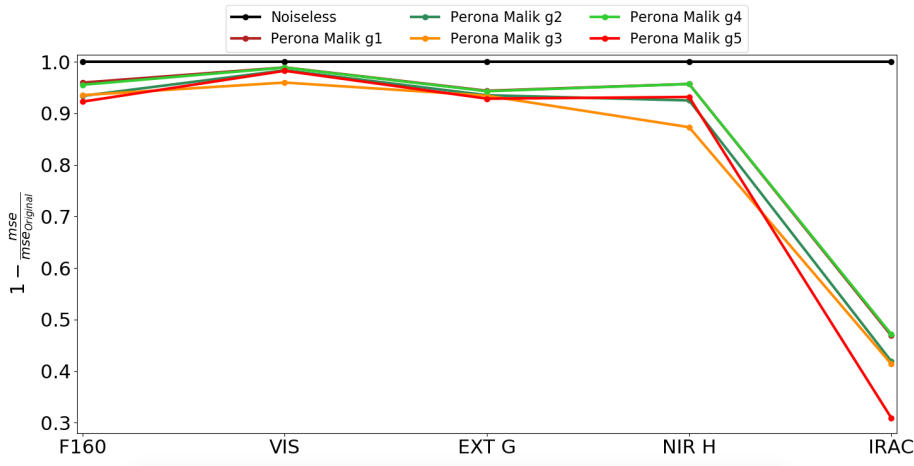


Fig. A.1. Step 1: MSE comparison between Perona-Malik functions on CM. On the x-axis, all the simulated CM crops in the different bands, whereas on the y-axis  $1 - \frac{mse}{mse_{Original}}$ .

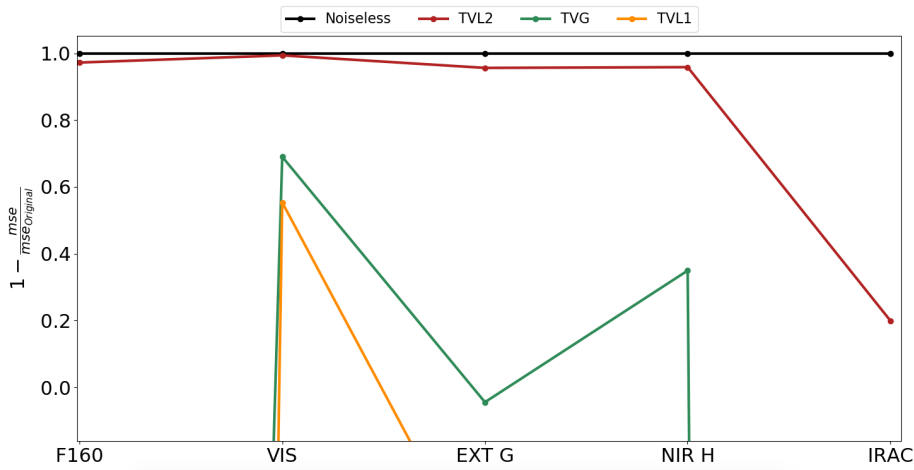


Fig. A.2. Step 1: MSE comparison between ATVD algorithms on BG. On the x-axis, all the simulated BG crops in the different bands, whereas on the y-axis  $1 - \frac{mse}{mse_{Original}}$ .

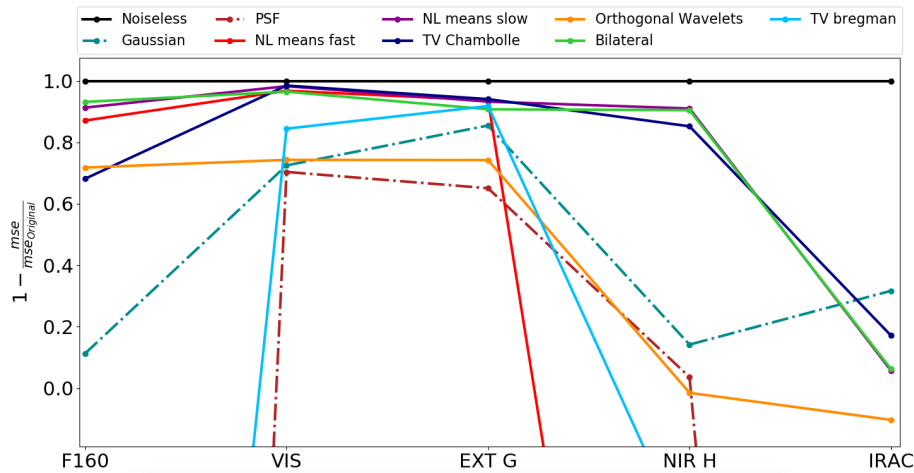
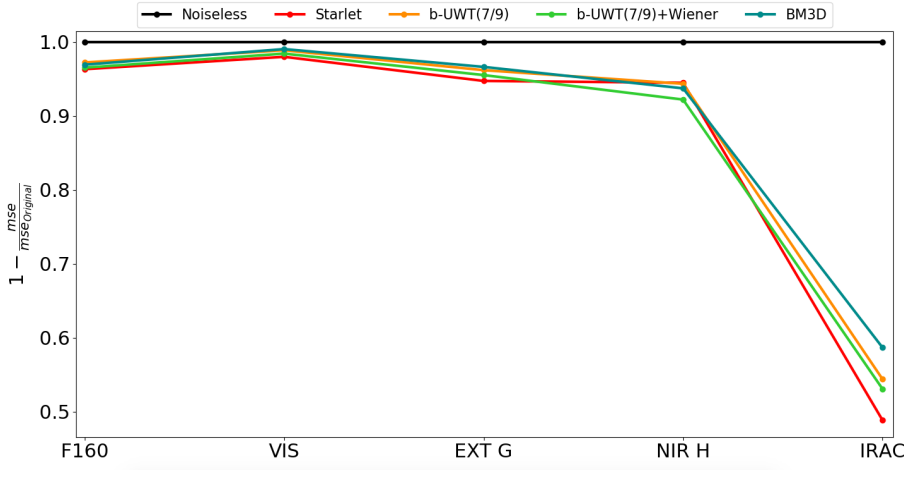


Fig. A.3. Step 1: MSE comparison between the other algorithms excluding ATVD and PM on CL. On the x-axis, all the simulated CL crops in the different bands, whereas on the y-axis  $1 - \frac{mse}{mse_{Original}}$ .



**Fig. A.4.** Step 1: MSE comparison between BM3D, Starlet and the two b-UWTs on CM. On the x-axis, all the simulated CM crops in the different bands, whereas on the y-axis  $1 - \frac{mse}{mse_{Original}}$ .

name	$MS E_{H160}$	$MS E_{VIS}$	$MS E_{EXTG}$	$MS E_{NIRH}$	$MS E_{IRAC}$
TVL2	$5.320 \times 10^{-8}$	<b><math>7.286 \times 10^{-7}</math></b>	$3.672 \times 10^{-7}$	<b><math>1.964 \times 10^{-4}</math></b>	$7.051 \times 10^{-5}$
TVL1	$7.297 \times 10^{-5}$	$5.155 \times 10^{-5}$	$1.325 \times 10^{-5}$	$4.409 \times 10^{-2}$	$4.429 \times 10^{-2}$
TVG	$3.922 \times 10^{-5}$	$3.576 \times 10^{-5}$	$8.760 \times 10^{-6}$	$3.071 \times 10^{-3}$	$6.370 \times 10^{-3}$
PM g=1 k=1 $\times 10^{-3}$	$9.490 \times 10^{-8}$	$1.108 \times 10^{-6}$	$6.776 \times 10^{-7}$	$2.528 \times 10^{-4}$	$5.473 \times 10^{-5}$
PM g=2 k=1 $\times 10^{-3}$	$9.790 \times 10^{-8}$	$8.941 \times 10^{-5}$	$8.147 \times 10^{-6}$	$1.926 \times 10^{-3}$	$7.073 \times 10^{-5}$
PM g=3 k=1 $\times 10^{-3}$	$9.930 \times 10^{-8}$	$1.122 \times 10^{-4}$	$8.390 \times 10^{-6}$	$3.640 \times 10^{-3}$	$7.855 \times 10^{-5}$
PM g=4 k=1 $\times 10^{-4}$	$7.960 \times 10^{-8}$	$8.291 \times 10^{-5}$	$7.641 \times 10^{-6}$	$2.760 \times 10^{-4}$	$6.288 \times 10^{-5}$
PM g=5 k=1 $\times 10^{-3}$	$1.077 \times 10^{-7}$	$8.843 \times 10^{-5}$	$8.131 \times 10^{-6}$	$1.775 \times 10^{-3}$	$7.157 \times 10^{-5}$
PSF	$8.557 \times 10^{-6}$	$1.833 \times 10^{-5}$	$1.615 \times 10^{-6}$	$3.492 \times 10^{-3}$	$1.174 \times 10^{-3}$
Original	$1.912 \times 10^{-6}$	$1.154 \times 10^{-4}$	$8.390 \times 10^{-6}$	$4.722 \times 10^{-3}$	$8.811 \times 10^{-5}$
TV Bregman	$3.778 \times 10^{-6}$	$5.908 \times 10^{-6}$	$6.147 \times 10^{-7}$	$1.960 \times 10^{-3}$	$2.620 \times 10^{-4}$
Gaussian	$1.301 \times 10^{-6}$	$1.724 \times 10^{-5}$	$9.434 \times 10^{-7}$	$3.469 \times 10^{-3}$	$6.955 \times 10^{-5}$
NL means slow	$8.940 \times 10^{-8}$	$1.513 \times 10^{-6}$	$5.992 \times 10^{-7}$	$3.066 \times 10^{-4}$	$6.454 \times 10^{-5}$
NL means fast	$1.019 \times 10^{-7}$	$1.628 \times 10^{-6}$	$5.750 \times 10^{-7}$	$5.551 \times 10^{-3}$	$1.389 \times 10^{-4}$
Bilateral	$1.109 \times 10^{-7}$	$3.754 \times 10^{-6}$	$7.972 \times 10^{-7}$	$4.612 \times 10^{-4}$	$6.297 \times 10^{-5}$
TV Chambolle	$1.876 \times 10^{-7}$	$2.112 \times 10^{-6}$	$4.914 \times 10^{-7}$	$2.894 \times 10^{-4}$	$5.444 \times 10^{-5}$
Orthogonal Wavelets	$5.119 \times 10^{-7}$	$2.962 \times 10^{-5}$	$2.117 \times 10^{-6}$	$4.776 \times 10^{-3}$	$9.583 \times 10^{-5}$
BM3D	<b><math>4.830 \times 10^{-8}</math></b>	$1.358 \times 10^{-6}$	<b><math>2.999 \times 10^{-7}</math></b>	$2.974 \times 10^{-4}$	<b><math>4.474 \times 10^{-5}</math></b>
Starlet	$6.760 \times 10^{-8}$	$2.224 \times 10^{-6}$	$5.061 \times 10^{-7}$	$2.978 \times 10^{-4}$	$5.211 \times 10^{-5}$
b-UWT(7/9)	$4.900 \times 10^{-8}$	$1.116 \times 10^{-6}$	$3.707 \times 10^{-7}$	$2.822 \times 10^{-4}$	$4.771 \times 10^{-5}$
b-UWT(7/9)+Wiener	$6.390 \times 10^{-8}$	$1.645 \times 10^{-6}$	$4.610 \times 10^{-7}$	$3.780 \times 10^{-4}$	$4.907 \times 10^{-5}$

**Table A.1.** MSE table of BG crops. The lowest MSE value per band is indicated in bold

Name	$MSE_{H160}$	$MSE_{VIS}$	$MSE_{EXTG}$	$MSE_{NIRH}$	$MSE_{IRAC}$
TVL2	$5.590 \times 10^{-8}$	<b><math>8.746 \times 10^{-7}</math></b>	$3.268 \times 10^{-7}$	<b><math>1.808 \times 10^{-4}</math></b>	$5.699 \times 10^{-5}$
TVL1	$5.878 \times 10^{-4}$	$1.075 \times 10^{-3}$	$5.485 \times 10^{-5}$	$2.390 \times 10^{-1}$	$8.910 \times 10^{-2}$
TVG	$2.977 \times 10^{-5}$	$3.147 \times 10^{-5}$	$4.477 \times 10^{-5}$	$3.672 \times 10^{-3}$	$5.752 \times 10^{-3}$
PM g=1 k=1 $\times 10^{-3}$	$1.248 \times 10^{-7}$	$2.277 \times 10^{-6}$	$4.646 \times 10^{-7}$	$2.625 \times 10^{-4}$	$3.886 \times 10^{-5}$
PM g=2 k=1 $\times 10^{-3}$	$1.257 \times 10^{-7}$	$1.889 \times 10^{-6}$	$4.310 \times 10^{-6}$	$3.503 \times 10^{-4}$	$4.969 \times 10^{-5}$
PM g=3 k=1 $\times 10^{-3}$	$1.228 \times 10^{-7}$	$8.549 \times 10^{-6}$	$7.173 \times 10^{-6}$	$1.386 \times 10^{-3}$	$5.755 \times 10^{-5}$
PM g=4 k=1 $\times 10^{-4}$	$8.470 \times 10^{-8}$	$1.281 \times 10^{-6}$	$3.914 \times 10^{-6}$	$2.011 \times 10^{-4}$	$4.697 \times 10^{-5}$
PM g=5 k=1 $\times 10^{-3}$	$1.468 \times 10^{-7}$	$2.006 \times 10^{-6}$	$8.287 \times 10^{-6}$	$3.186 \times 10^{-4}$	$5.055 \times 10^{-5}$
PSF	$4.853 \times 10^{-5}$	$1.532 \times 10^{-4}$	$9.444 \times 10^{-6}$	$6.024 \times 10^{-3}$	$5.052 \times 10^{-3}$
Original	$1.902 \times 10^{-6}$	$1.116 \times 10^{-4}$	$8.288 \times 10^{-6}$	$4.679 \times 10^{-3}$	$7.319 \times 10^{-5}$
TV Bregman	$2.170 \times 10^{-5}$	$1.084 \times 10^{-4}$	$2.264 \times 10^{-6}$	$1.023 \times 10^{-2}$	$6.225 \times 10^{-4}$
Gaussian	$1.835 \times 10^{-6}$	$6.773 \times 10^{-5}$	$1.913 \times 10^{-6}$	$4.373 \times 10^{-3}$	$7.094 \times 10^{-5}$
NL means slow	$1.201 \times 10^{-7}$	$1.883 \times 10^{-6}$	$4.885 \times 10^{-7}$	$7.916 \times 10^{-4}$	$1.559 \times 10^{-4}$
NL means fast	$1.990 \times 10^{-7}$	$5.729 \times 10^{-6}$	$4.655 \times 10^{-7}$	$6.982 \times 10^{-3}$	$1.814 \times 10^{-4}$
Bilateral	$1.104 \times 10^{-7}$	$4.027 \times 10^{-6}$	$7.348 \times 10^{-7}$	$3.630 \times 10^{-4}$	$4.779 \times 10^{-5}$
TV Chambolle	$4.964 \times 10^{-7}$	$1.698 \times 10^{-6}$	$4.723 \times 10^{-7}$	$6.008 \times 10^{-4}$	$5.398 \times 10^{-5}$
Orthogonal Wavelets	$5.303 \times 10^{-7}$	$2.999 \times 10^{-5}$	$2.128 \times 10^{-6}$	$4.732 \times 10^{-3}$	$7.795 \times 10^{-5}$
BM3D	$5.880 \times 10^{-8}$	$1.290 \times 10^{-6}$	<b><math>2.807 \times 10^{-7}</math></b>	$2.941 \times 10^{-4}$	<b><math>3.025 \times 10^{-5}</math></b>
Starlet	$7.040 \times 10^{-8}$	$2.343 \times 10^{-6}$	$4.380 \times 10^{-7}$	$2.589 \times 10^{-4}$	$3.744 \times 10^{-5}$
b-UWT(7/9)	<b><math>5.310 \times 10^{-8}</math></b>	$1.319 \times 10^{-6}$	$3.174 \times 10^{-7}$	$2.654 \times 10^{-4}$	$3.334 \times 10^{-5}$
b-UWT(7/9)+Wiener	$6.650 \times 10^{-8}$	$1.837 \times 10^{-6}$	$3.723 \times 10^{-7}$	$3.653 \times 10^{-4}$	$3.435 \times 10^{-5}$

Table A.2. MSE table of CM crops. The lowest MSE value per band is indicated in bold

Name	$MSE_{H160}$	$MSE_{VIS}$	$MSE_{EXTG}$	$MSE_{NIRH}$	$MSE_{IRAC}$
TVL2	$7.070 \times 10^{-8}$	<b><math>8.958 \times 10^{-7}</math></b>	$3.443 \times 10^{-7}$	<b><math>2.451 \times 10^{-4}</math></b>	$3.633 \times 10^{-4}$
TVL1	$2.754 \times 10^{-4}$	$1.707 \times 10^{-4}$	$1.471 \times 10^{-5}$	$1.121 \times 10^{-1}$	$5.820 \times 10^{-2}$
TVG	$4.424 \times 10^{-5}$	$3.336 \times 10^{-5}$	$1.387 \times 10^{-5}$	$4.519 \times 10^{-3}$	$1.013 \times 10^{-2}$
PM g=1 k=1 $\times 10^{-3}$	$1.463 \times 10^{-7}$	$1.752 \times 10^{-6}$	$5.923 \times 10^{-7}$	$3.433 \times 10^{-4}$	$3.312 \times 10^{-4}$
PM g=2 k=1 $\times 10^{-3}$	$1.437 \times 10^{-7}$	$1.865 \times 10^{-5}$	$7.553 \times 10^{-6}$	$4.436 \times 10^{-4}$	$3.433 \times 10^{-4}$
PM g=3 k=1 $\times 10^{-3}$	$1.431 \times 10^{-7}$	$6.034 \times 10^{-5}$	$8.342 \times 10^{-6}$	$1.413 \times 10^{-3}$	$3.501 \times 10^{-4}$
PM g=4 k=1 $\times 10^{-4}$	$1.029 \times 10^{-7}$	$1.076 \times 10^{-5}$	$6.809 \times 10^{-6}$	$2.893 \times 10^{-4}$	$3.397 \times 10^{-4}$
PM g=5 k=1 $\times 10^{-3}$	$1.668 \times 10^{-7}$	$1.432 \times 10^{-5}$	$7.471 \times 10^{-6}$	$4.163 \times 10^{-4}$	$3.439 \times 10^{-4}$
PSF	$2.728 \times 10^{-5}$	$3.421 \times 10^{-5}$	$2.913 \times 10^{-6}$	$4.576 \times 10^{-3}$	$2.952 \times 10^{-3}$
Original	$1.915 \times 10^{-6}$	$1.155 \times 10^{-4}$	$8.342 \times 10^{-6}$	$4.748 \times 10^{-3}$	$3.640 \times 10^{-4}$
TV Bregman	$1.233 \times 10^{-5}$	$1.795 \times 10^{-5}$	$6.815 \times 10^{-7}$	$7.211 \times 10^{-3}$	$6.886 \times 10^{-4}$
Gaussian	$1.699 \times 10^{-6}$	$3.175 \times 10^{-5}$	$1.211 \times 10^{-6}$	$4.074 \times 10^{-3}$	<b><math>2.487 \times 10^{-4}</math></b>
NL means slow	$1.664 \times 10^{-7}$	$1.976 \times 10^{-6}$	$5.627 \times 10^{-7}$	$4.264 \times 10^{-4}$	$3.432 \times 10^{-4}$
NL means fast	$2.470 \times 10^{-7}$	$3.702 \times 10^{-6}$	$5.260 \times 10^{-7}$	$1.975 \times 10^{-2}$	$4.581 \times 10^{-4}$
Bilateral	$1.307 \times 10^{-7}$	$4.055 \times 10^{-6}$	$7.697 \times 10^{-7}$	$4.521 \times 10^{-4}$	$3.413 \times 10^{-4}$
TV Chambolle	$6.098 \times 10^{-7}$	$1.740 \times 10^{-6}$	$4.938 \times 10^{-7}$	$7.015 \times 10^{-4}$	$3.017 \times 10^{-4}$
Orthogonal Wavelets	$5.399 \times 10^{-7}$	$2.968 \times 10^{-5}$	$2.149 \times 10^{-6}$	$4.821 \times 10^{-3}$	$4.016 \times 10^{-4}$
BM3D	<b><math>6.130 \times 10^{-8}</math></b>	$1.103 \times 10^{-6}$	<b><math>2.777 \times 10^{-7}</math></b>	$1.031 \times 10^{-1}$	$3.199 \times 10^{-4}$
Starlet	$8.400 \times 10^{-8}$	$2.387 \times 10^{-6}$	$4.358 \times 10^{-7}$	$3.596 \times 10^{-4}$	$3.311 \times 10^{-4}$
b-UWT(7/9)	$6.610 \times 10^{-8}$	$1.333 \times 10^{-6}$	$3.284 \times 10^{-7}$	$3.611 \times 10^{-4}$	$3.248 \times 10^{-4}$
b-UWT(7/9)+Wiener	$8.080 \times 10^{-8}$	$1.856 \times 10^{-6}$	$3.831 \times 10^{-7}$	$4.791 \times 10^{-4}$	$3.265 \times 10^{-4}$

Table A.3. MSE table of CL crops. The lowest MSE value per band is indicated in bold

Name	$T_{H160}$	$T_{VIS}$	$T_{EXTG}$	$T_{NIRH}$	$T_{IRAC}$
	s	s	s	s	s
TVL2	9.047	3.588	0.688	0.132	0.0938
TVL1	5.408	0.814	0.339	0.020	0.0031
TVG	8.910	5.177	0.334	0.134	0.0914
PM g=1 k=1 × 10 <sup>-3</sup>	6.353	5.283	2.127	0.362	0.063
PM g=2 k=1 × 10 <sup>-3</sup>	10.519	22.826	0.248	8.347	0.159
PM g=3 k=1 × 10 <sup>-3</sup>	6.847	30.578	12.136	5.544	0.142
PM g=4 k=1 × 10 <sup>-4</sup>	88.506	323.720	109-356	50.417	6.512
PM g=5 k=1 × 10 <sup>-3</sup>	6.778	16.001	51.941	11.955	0.125
PSF	0.071	0.019	0.008	0.008	0.001
Original	n.a.	n.a.	n.a.	n.a.	n.a.
TV Bregman	0.222	0.093	0.021	0.025	0.011
Gaussian	<b>0.055</b>	<b>0.017</b>	<b>0.005</b>	<b>0.002</b>	<b>0.001</b>
NL means slow	75.94	27.75	6.863	3.344	1.058
NL means fast	7.514	3.104	1.118	0.448	0.208
Bilateral	37.09	13.46	3.602	1.567	0.489
TV Chambolle	10.61	0.7668	0.109	0.587	0.034
Orthogonal Wavelets	0.827	0.329	0.091	0.038	0.012
BM3D	25.08	10.499	2.836	1.536	0.655
Starlet	7.671	2.752	0.702	0.314	0.083
b-UWT(7/9)	9.013	3.265	0.812	0.350	0.095
b-UWT(7/9)+Wiener	33.15	12.01	2.983	1.318	0.337

**Table A.4.** CPU Time table of CM crops after fixing the optimal internal parameters for each method. The lowest time value per band is indicated in bold

## Appendix B: Non-stationary Gaussian noise MSE comparison table

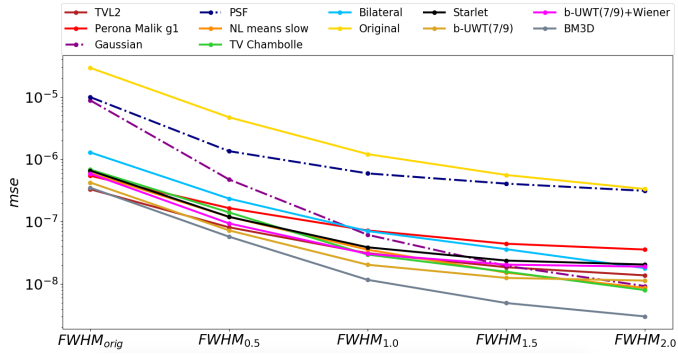
Name	$MSE_{H_l}$	$MSE_{I_\sigma}$	$MSE_{H_u}$	$MSE_{I_{2\sigma}}$
TVL2	7.622e-07	<b>7.654e-07</b>	2.477e-05	2.500e-05
PM g=1 k=1 $\times 10^{-3}$	9.277e-05	9.412e-05	4.352e-04	4.343e-04
PSF	1.398e-05	1.398e-05	5.239e-05	5.242e-05
Original	1.153e-04	1.152e-04	4.604e-04	4.604e-04
Gaussian	4.575e-05	4.574e-05	1.822e-04	1.823e-04
NL means slow	1.110e-04	1.114e-04	4.603e-04	4.426e-04
Bilateral	1.129e-04	1.130e-04	4.581e-04	4.505e-04
TV Chambolle	6.246e-05	6.387e-05	3.428e-04	3.411e-04
BM3D	1.141e-04	1.141e-04	4.592e-04	4.556e-04
Starlet	<b>5.647e-07</b>	8.924e-07	1.907e-05	2.571e-06
b-UWT(7/9)	8.316e-07	7.797e-07	<b>8.632e-06</b>	<b>2.255e-06</b>
b-UWT(7/9)+Wiener	7.954e-07	1.009e-06	1.957e-05	3.191e-06

**Table B.1.** MSE values for  $H_l, I_\sigma, H_u, I_{2\sigma}$  related to the VIS (CM) mirrored crop. See Sect. 5.3 for further details. The best results are in bold

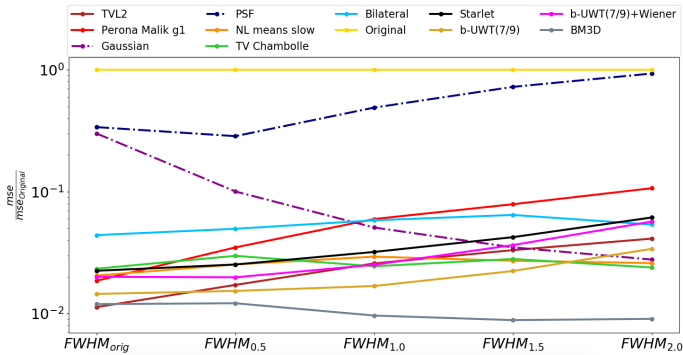
Name	$MSE_{H_l}$	$MSE_{I_\sigma}$	$MSE_{H_u}$	$MSE_{I_{2\sigma}}$
TVL2	8.105e-07	<b>7.933e-07</b>	2.463e-05	2.500e-05
PM g=1 k=1 $\times 10^{-3}$	9.174e-05	9.331e-05	4.339e-04	4.329e-04
PSF	1.429e-05	1.432e-05	5.246e-05	5.242e-05
Original	1.152e-04	1.151e-04	4.604e-04	4.597e-04
Gaussian	4.578e-05	4.577e-05	1.821e-04	1.818e-04
NL means slow	1.107e-04	1.111e-04	4.603e-04	4.411e-04
Bilateral	1.126e-04	1.127e-04	4.579e-04	4.495e-04
TV Chambolle	6.136e-05	6.303e-05	3.401e-04	3.388e-04
BM3D	1.140e-04	1.140e-04	4.591e-04	4.548e-04
Starlet	<b>7.482e-07</b>	2.170e-06	4.874e-05	7.682e-06
b-UWT(7/9)	1.048e-06	1.117e-06	<b>1.638e-05</b>	<b>3.472e-06</b>
b-UWT(7/9)+Wiener	1.038e-06	1.588e-06	3.805e-05	5.305e-06

**Table B.2.** MSE values for  $H_l, I_\sigma, H_u, I_{2\sigma}$  related to the VIS (CL) mirrored crop. See Sect. 5.3 for further details. The best results are in bold

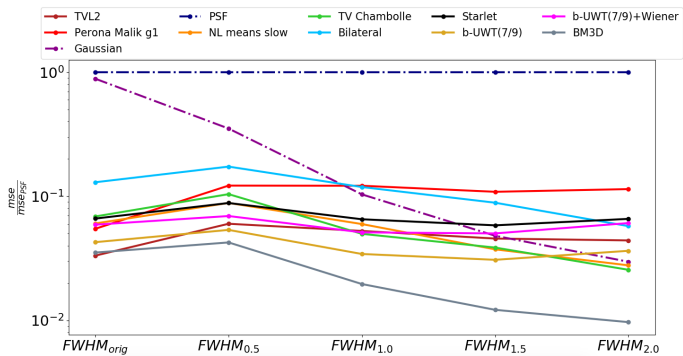
### Appendix C: PSF and depth comparison plots



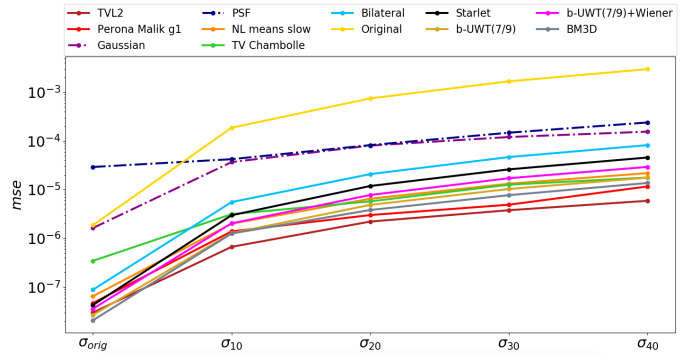
**Fig. C.1.** VIS FWHM variation comparison plot. On the x-axis the VIS images with FWHM equal to the original value, 0.5, 1.0, 1.5 and 2.0 arcsecs, whereas on the y-axis  $mse$ .



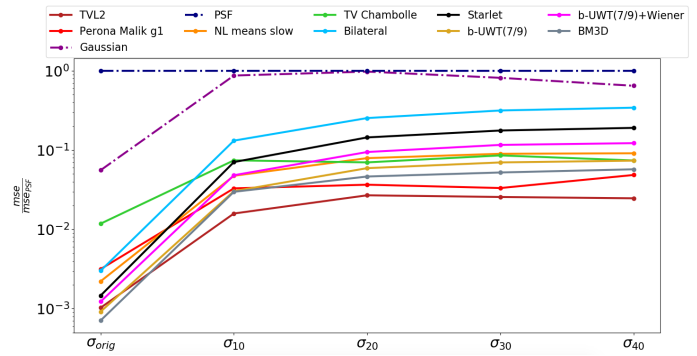
**Fig. C.2.** VIS FWHM variation comparison plot. On the x-axis the VIS images with FWHM equal to the original value, 0.5, 1.0, 1.5 and 2.0 arcsecs, whereas on the y-axis  $\frac{mse}{mse_{Original}}$ .



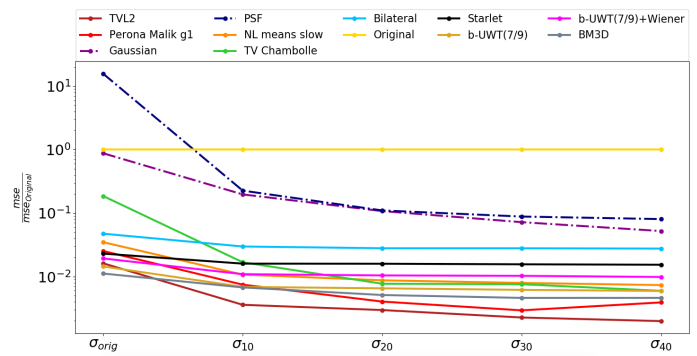
**Fig. C.3.** VIS FWHM variation comparison plot. On the x-axis the VIS images with FWHM equal to the original value, 0.5, 1.0, 1.5 and 2.0 arcsecs, whereas on the y-axis  $\frac{mse}{mse_{PSF}}$ .



**Fig. C.4.** H160 depth variation comparison plot. On the x-axis the H160 images with Gaussian noise standard deviation equal to 1, 10, 20, 30 and 40 times the original value, whereas on the y-axis  $mse$ .

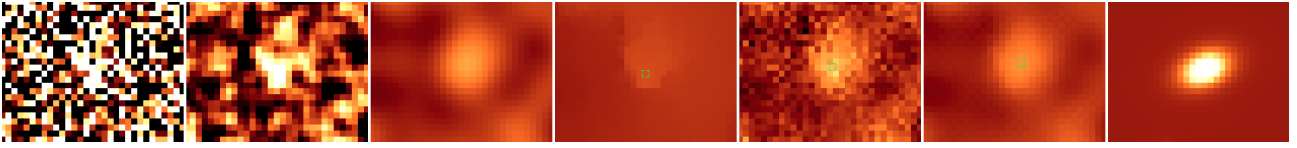


**Fig. C.5.** H160 depth variation comparison plot. On the x-axis the H160 images with Gaussian noise standard deviation equal to 1, 10, 20, 30 and 40 times the original value, whereas on the y-axis  $\frac{mse}{mse_{PSF}}$ .

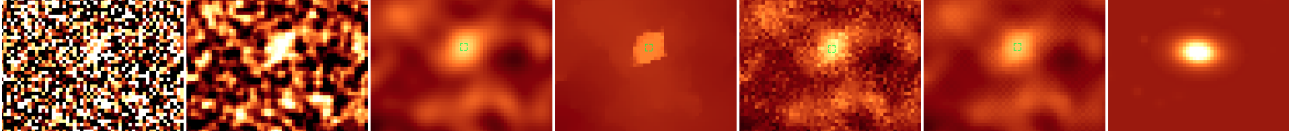


**Fig. C.6.** H160 depth variation comparison plot. On the x-axis the H160 images with Gaussian noise standard deviation equal to 1, 10, 20, 30 and 40 times the original value, whereas on the y-axis  $\frac{mse}{mse_{Original}}$ .

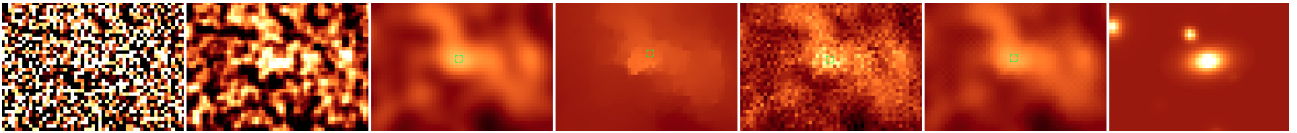
## Appendix D: VIS crops visual comparison



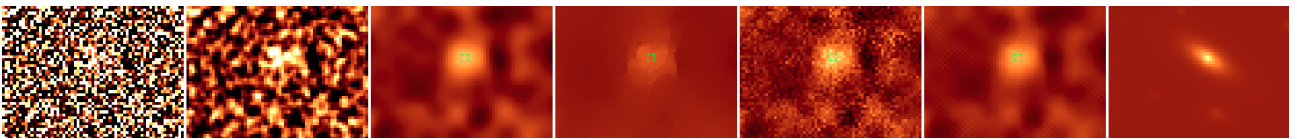
**Fig. D.1.** VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a  $SNR$  of 38.8 with magnitude of 25.79



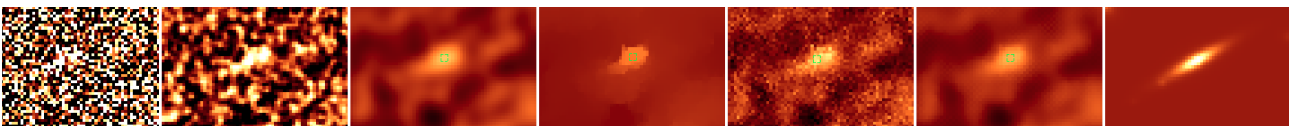
**Fig. D.2.** VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a  $SNR$  of 48.2 with magnitude of 24.76



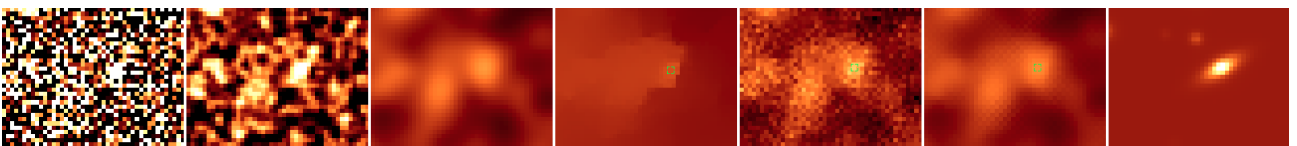
**Fig. D.3.** VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a  $SNR$  of 72.9 with magnitude of 23.82



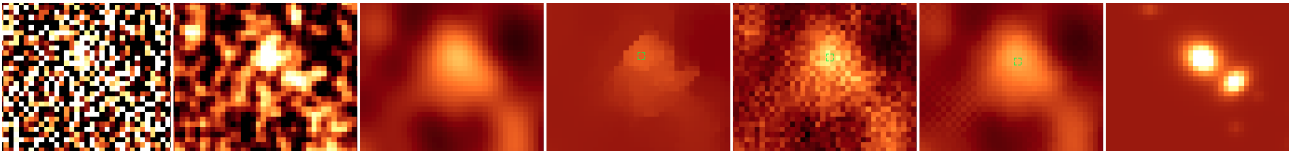
**Fig. D.4.** VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a  $SNR$  of 47.5 with magnitude of 25.01



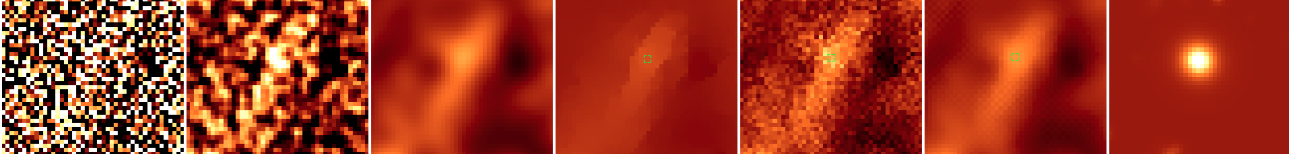
**Fig. D.5.** VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a  $SNR$  of 35.44 with magnitude of 25.39



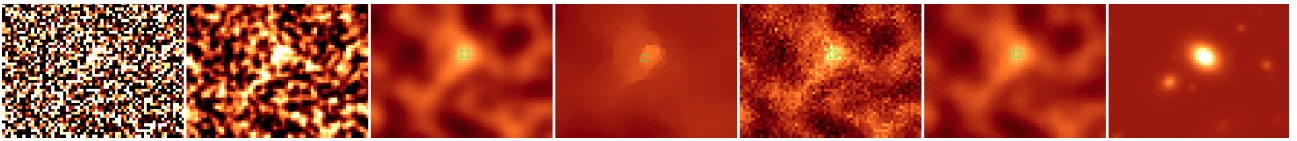
**Fig. D.6.** VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a  $SNR$  of 21.26 with magnitude of 26.48



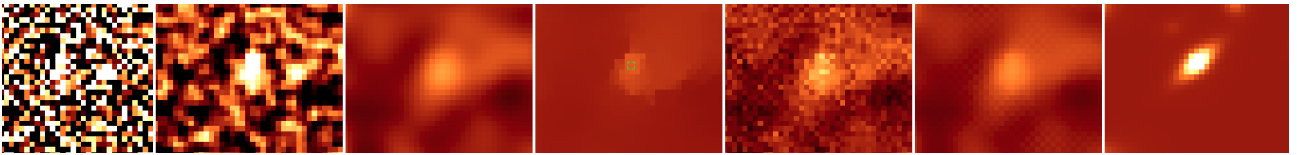
**Fig. D.7.** VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 27.70 with magnitude of 25.72



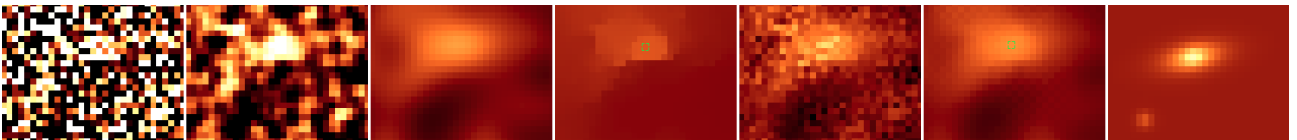
**Fig. D.8.** VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 45.57 with magnitude of 24.97



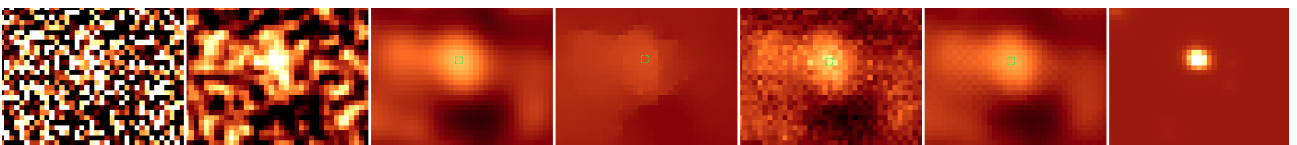
**Fig. D.9.** VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 68.29 with magnitude of 24.14



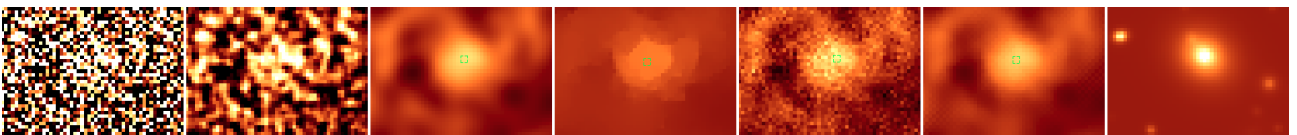
**Fig. D.10.** VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 26.74 with magnitude of 26.13



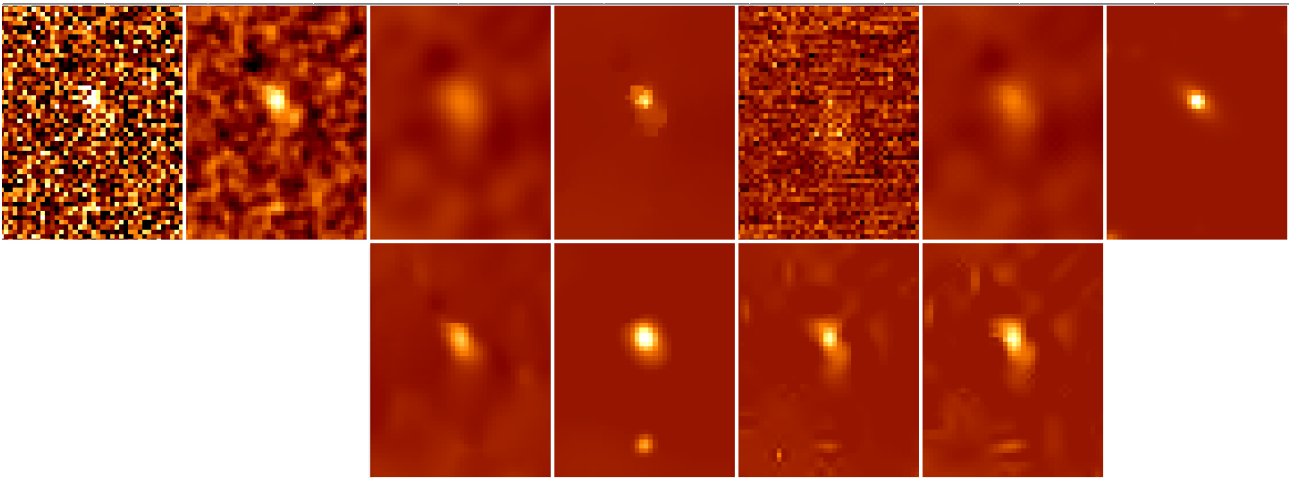
**Fig. D.11.** VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 25.44 with magnitude of 26.19



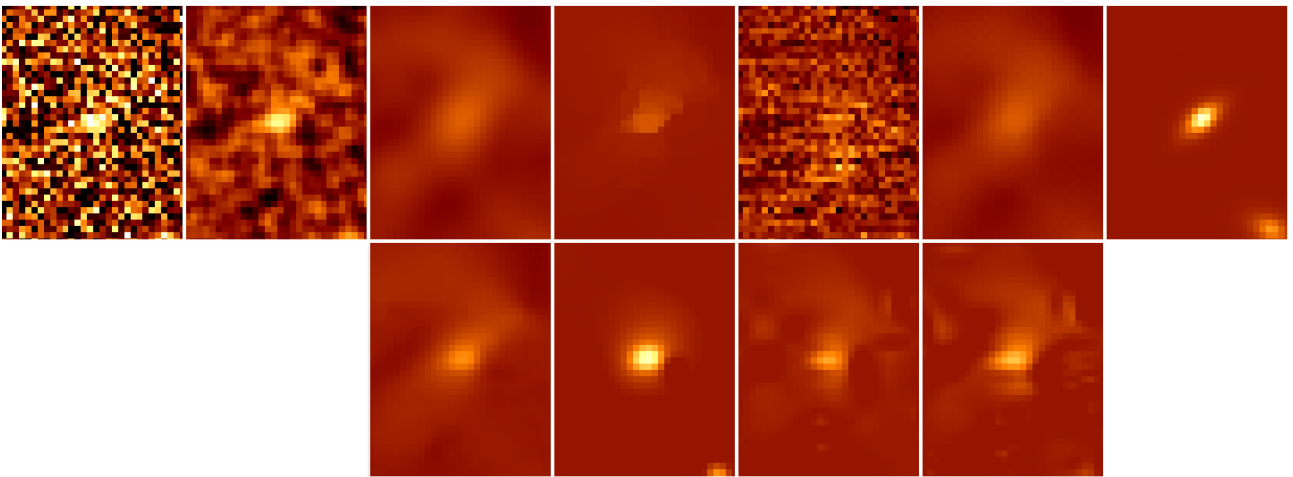
**Fig. D.12.** VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 36.99 with magnitude of 25.71



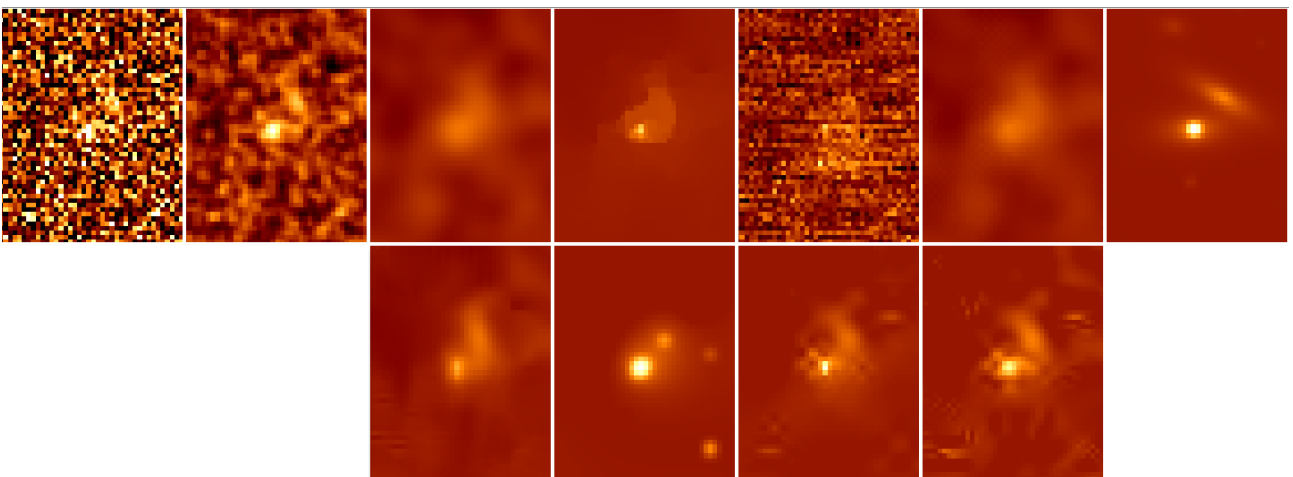
**Fig. D.13.** VIS crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 90.35 with magnitude of 23.34



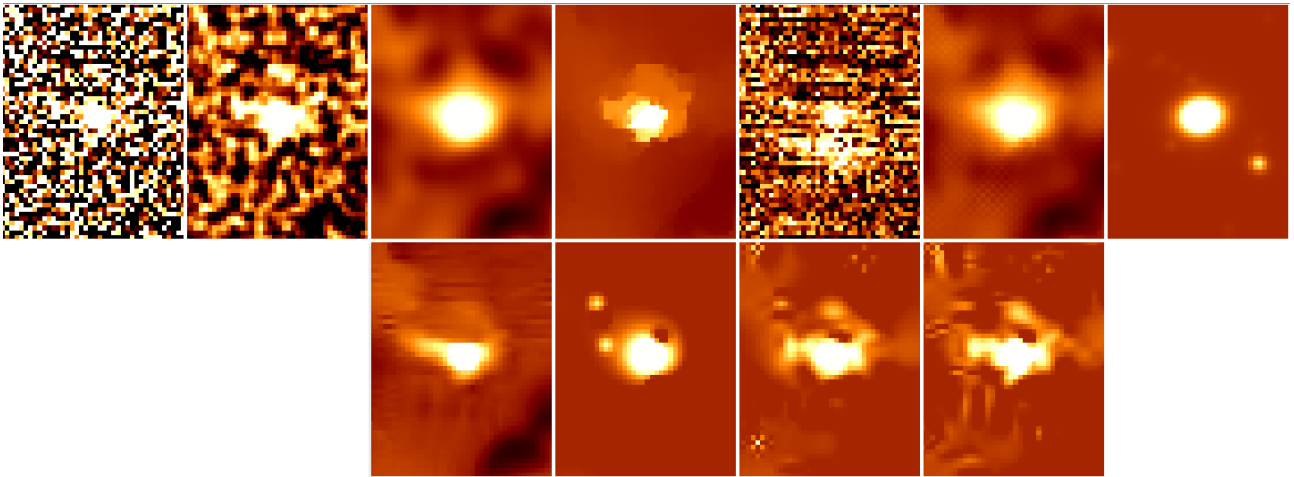
**Fig. D.14.** VIS crops visual comparison. On the first row: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless; On the second row: BM3D, Starlet, b-UWT(7/9), and b-UWT(7/9)+Wiener. The central object has been detected with a  $SNR$  of 7.99 with magnitude of 25.21



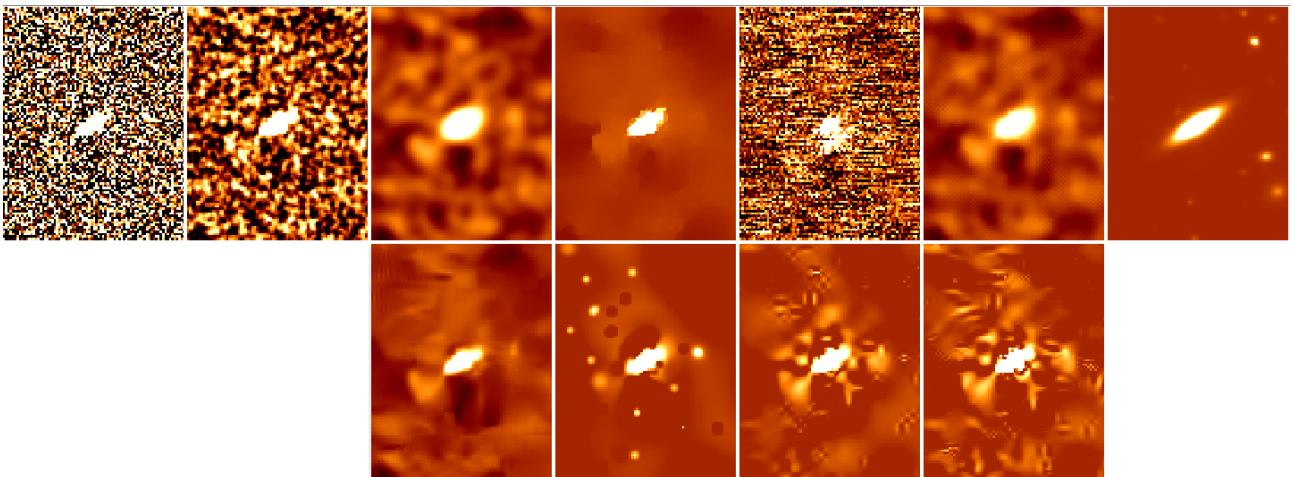
**Fig. D.15.** VIS crops visual comparison. On the first row: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless; On the second row: BM3D, Starlet, b-UWT(7/9), and b-UWT(7/9)+Wiener. The central object has been detected with a  $SNR$  of 3.80 with magnitude of 26.01



**Fig. D.16.** VIS crops visual comparison. On the first row: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless; On the second row: BM3D, Starlet, b-UWT(7/9), and b-UWT(7/9)+Wiener. The central object has been detected with a  $SNR$  of 2.23 with magnitude of 26.59

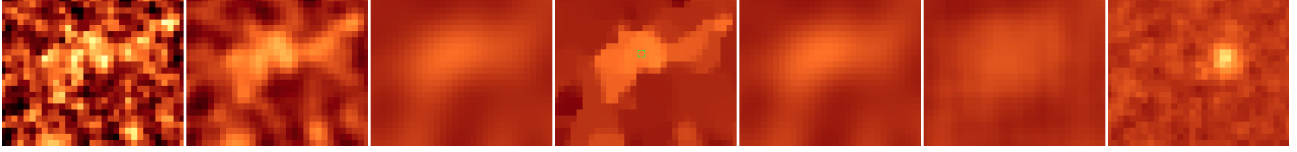


**Fig. D.17.** VIS crops visual comparison. On the first row: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless; On the second row: BM3D, Starlet, b-UWT(7/9), and b-UWT(7/9)+Wiener. The central object has been detected with a  $SNR$  of 15.73 with magnitude of 24.48

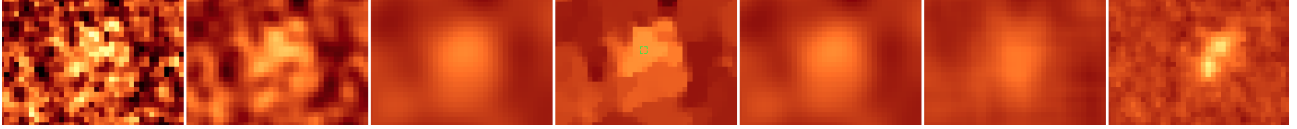


**Fig. D.18.** VIS crops visual comparison. On the first row: Original, PSF, Perona-Malik, TVL2, Bilateral, TV Chambolle, Noiseless; On the second row: BM3D, Starlet, b-UWT(7/9), and b-UWT(7/9)+Wiener. The central object has been detected with a  $SNR$  of 56.23 with magnitude of 23.09

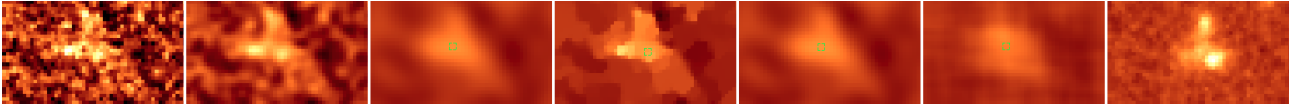
## Appendix E: GSDEEP crops visual comparison



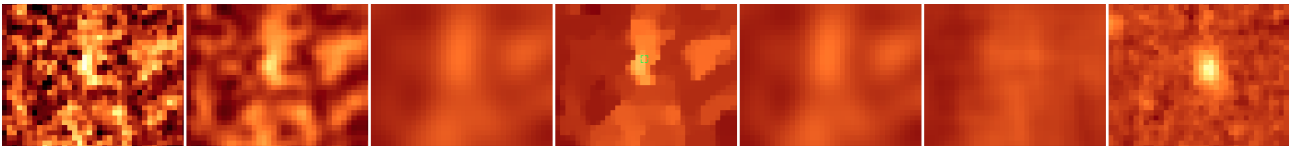
**Fig. E.1.** GSDEEP crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. The green boxes are the detected objects regions. The central object has been detected with a  $SNR$  of 6.71 with magnitude of 27.47



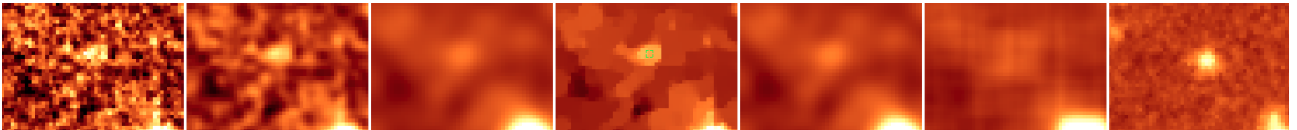
**Fig. E.2.** GSDEEP crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. The green boxes are the detected objects regions. The central object has been detected with a  $SNR$  of 7.37 with magnitude of 27.20



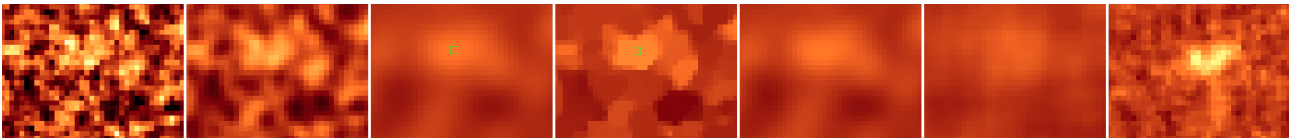
**Fig. E.3.** GSDEEP crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. The green boxes are the detected objects regions. The central object has been detected with a  $SNR$  of 8.44 with magnitude of 26.80



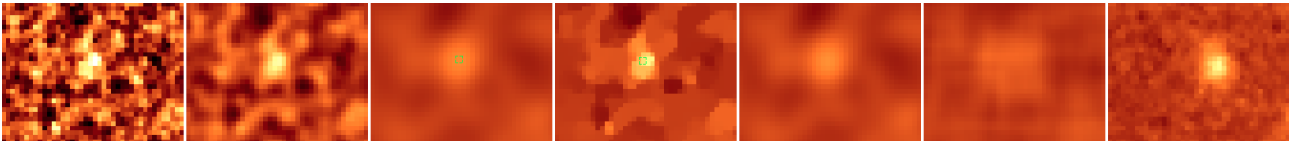
**Fig. E.4.** GSDEEP crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. The green boxes are the detected objects regions. The central object has been detected with a  $SNR$  of 6.88 with magnitude of 27.18



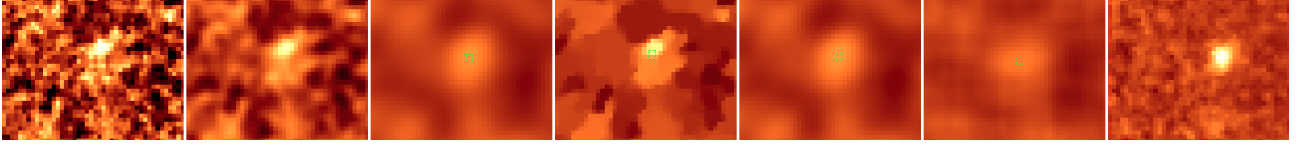
**Fig. E.5.** GSDEEP crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. The green boxes are the detected objects regions. The central object has been detected with a  $SNR$  of 5.82 with magnitude of 27.48



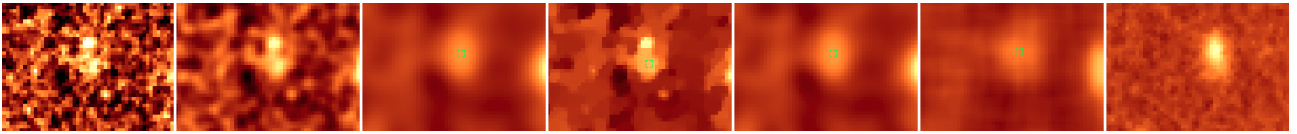
**Fig. E.6.** GSDEEP crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. The green boxes are the detected objects regions. The central object has been detected with a  $SNR$  of 5.08 with magnitude of 27.48



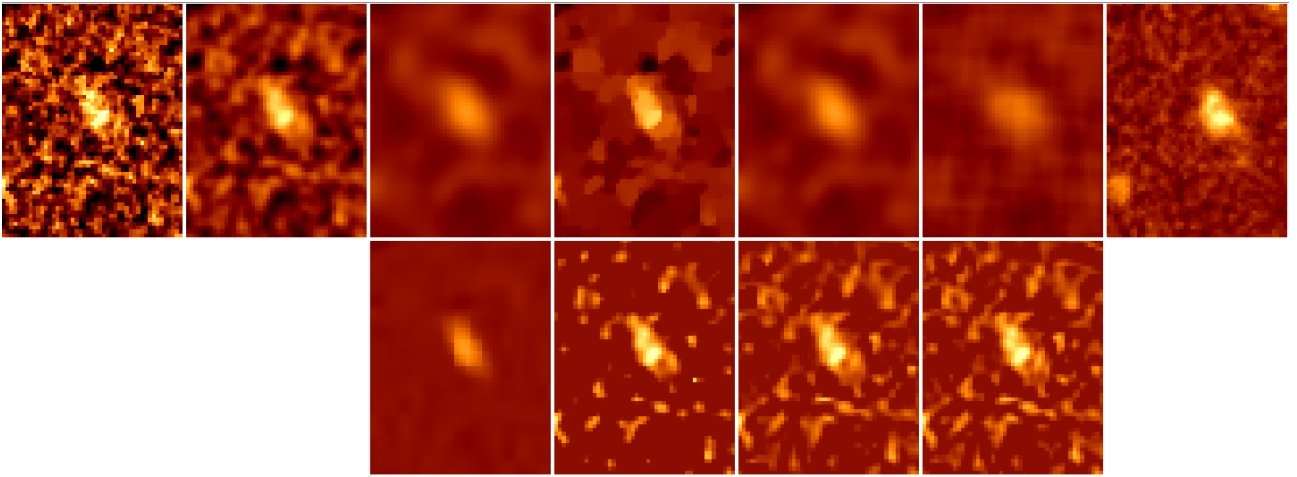
**Fig. E.7.** GSDEEP crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 6.51 with magnitude of 27.58



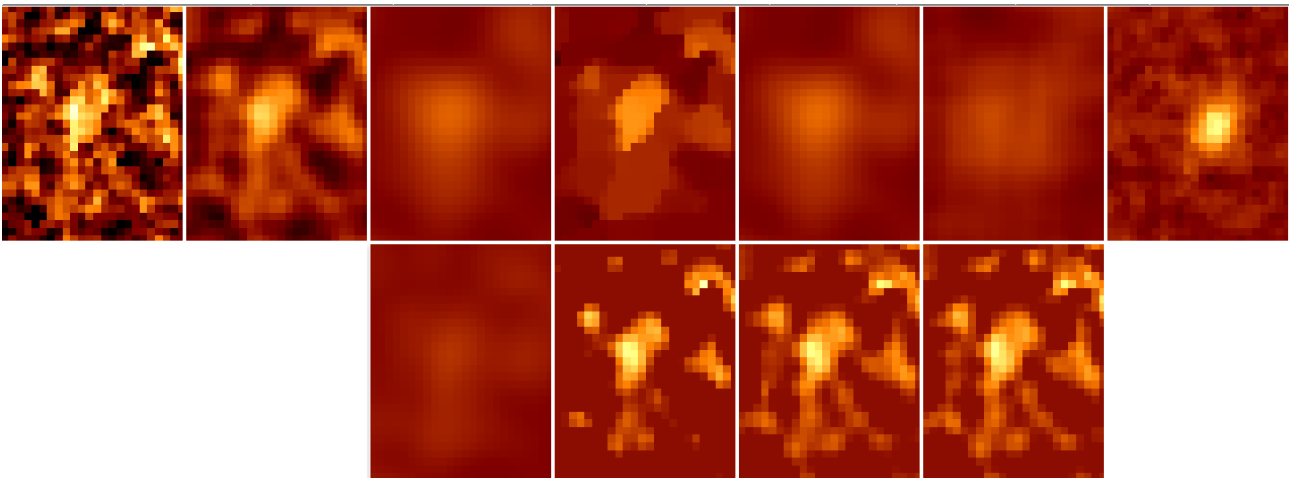
**Fig. E.8.** GSDEEP crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 12.48 with magnitude of 27.17



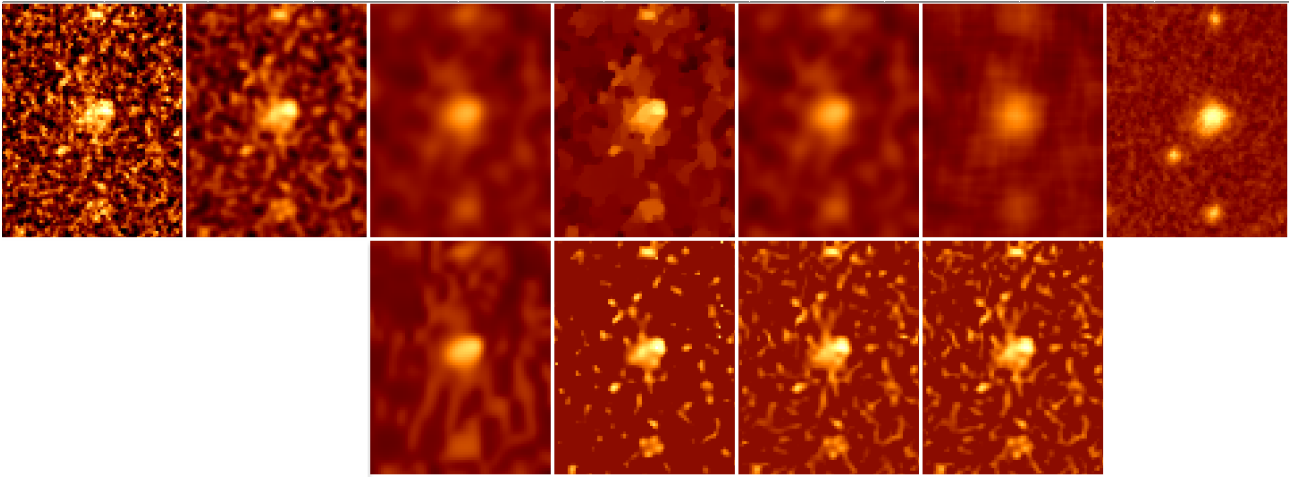
**Fig. E.9.** GSDEEP crops visual comparison: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. The green boxes are the detected objects regions. The central object has been detected with a *SNR* of 9.77 with magnitude of 27.36



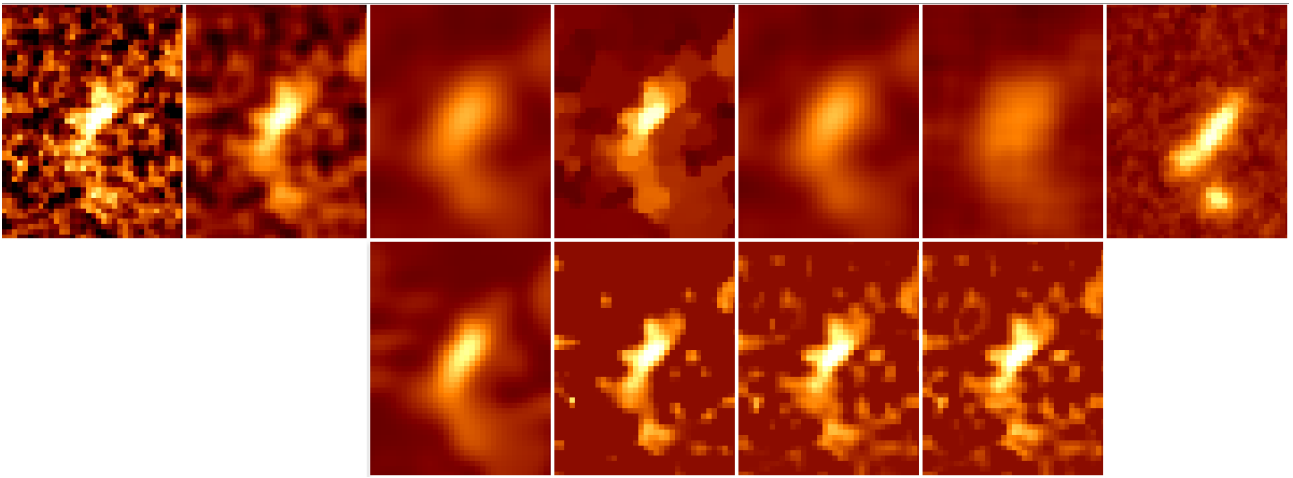
**Fig. E.10.** GSDEEP crops visual comparison. On the first row: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. On the second row: BM3D, Starlet, b-UWT(7/9), and b-UWT(7/9)+Wiener. The central object has been detected with a *SNR* of 11.98 with magnitude of 28.34



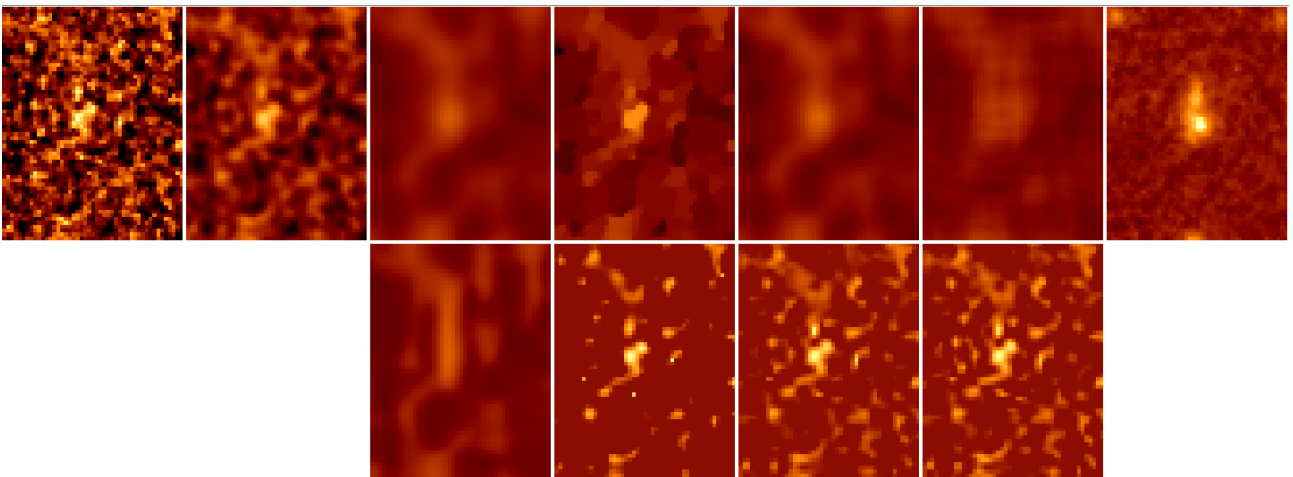
**Fig. E.11.** GSDEEP crops visual comparison. On the first row: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. On the second row: BM3D, Starlet, b-UWT(7/9), and b-UWT(7/9)+Wiener. The central object has been detected with a *SNR* of 9.85 with magnitude of 28.45



**Fig. E.12.** GSDEEP crops visual comparison. On the first row: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. On the second row: BM3D, Starlet, b-UWT(7/9), and b-UWT(7/9)+Wiener. The central object has been detected with a  $SNR$  of 11.26 with magnitude of 28.37



**Fig. E.13.** GSDEEP crops visual comparison. On the first row: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. On the second row: BM3D, Starlet, b-UWT(7/9), and b-UWT(7/9)+Wiener. The central object has been detected with a  $SNR$  of 17.70 with magnitude of 27.86



**Fig. E.14.** GSDEEP crops visual comparison. On the first row: Original, PSF, Perona-Malik, TVL2, Bilateral, NL means, HUDF09. On the second row: BM3D, Starlet, b-UWT(7/9), and b-UWT(7/9)+Wiener. The central object has been detected with a  $SNR$  of 10.40 with magnitude of 28.48