PARIGI: a Patch-based Approach to Remove Impulse-Gaussian Noise from Images

Julie Delon[1], Agnès Desolneux[2], Thierry Guillemot[3]

[1] MAP5, Université Paris Descartes, France (julie.delon@parisdescartes.fr)
[2] CNRS, CMLA, ENS Cachan, France (agnes.desolneux@cmla.ens-cachan.fr)
[3] MAP5, Université Paris Descartes, France (thierry.guillemot.work@gmail.com)

This IPOL article is related to a companion publication in the SIAM Journal on Imaging Sciences:
J. Delon and A. Desolneux, "A patch-based approach for removing mixed Gaussian-impulse noise", *SIAM Journal on Imaging Sciences*, 6(2), pp. 1140-1174, 2013.

Abstract

In this paper, we present an implementation of the PARIGI method that addresses the problem of the restoration of images affected by impulse noise or by a mixture of Gaussian and impulse noise. The method relies on a patch-based approach, which requires careful choices for both the distance between patches and for the statistical estimator of the original patch. Experiments are performed in the case of pure impulse noise and in the case of a mixture of Gaussian and impulse noise.

Source Code

The source code and the code documentation are available as supplementary material. The documentation is included in the archive. Basic compilation and usage instructions are included in the *README.txt* file. All those files and the online demo are accessible at the IPOL web page of this article[1].

Keywords: impulse noise; Gaussian noise; denoising; image patches

---

[1]http://dx.doi.org/10.5201/ipol.2016.161

1   Introduction

The aim of this paper is to precisely describe an implementation of the PARIGI denoising method introduced in [3]. PARIGI stands for Patch-based Approach to Remove Impulse-Gaussian noise from Images. Its goal is to address the problem of the restoration of images that have been affected by impulse noise or by a mixture of Gaussian and impulse noise, relying on a patch-based approach. It works on gray scale images.

Gaussian noise generally appears during the acquisition of an image and impulse noise generally models the fact that some data is lost during the transmission of the image. It also may be caused by hot or dead pixels in the camera sensor. At the end of the acquisition process, a user may have to deal with an image that is damaged by a mixture of Gaussian and impulse noise (it means that the image is first corrupted by Gaussian noise, and then by impulse noise). Removing this kind of noise while preserving image details and textures is of great importance before most image analysis tasks (edge detection, segmentation, etc). The Gaussian noise is an additive noise, and it is very different in nature from the impulse noise. Two models of impulse noise are generally used in the literature. In the first one, called *salt-and-pepper noise*, each gray level is replaced with a given probability by 0 or M, where M is the maximal gray level value ($M = 255$, in general). In this paper, we will focus on the second model of impulse noise, called *random-valued impulse noise*, where each gray level value is replaced with probability p, called *noise ratio* or *impulse noise intensity*, by a random value in the set $\{0, 1, \ldots, M\}$. In all the following, M is set to the value 255.

The discrete damaged image is denoted by u and the original gray level image is denoted by u^0. These images are defined on a discrete domain Ω, assumed to be a bounded rectangle of $\mathbb{Z}^2$. We consider in this paper that the damaged image u is obtained from the realization of a random image U which can be written

$$U = (1 - T) \cdot (u^0 + W) + T \cdot V, \tag{1}$$

where T, V and W are independent random images such that

- the values $T(x)$, $x \in \Omega$ are i.i.d. random variables with a Bernoulli distribution of parameter p (in the sense that $\mathbb{P}[T(x) = 1] = p$); p is called the *noise ratio* or *impulse noise intensity*;

- the values $V(x)$, $x \in \Omega$ are i.i.d. random variables with a discrete uniform distribution in the range $[0, M]$;

- the values $W(x)$, $x \in \Omega$ are independent centered Gaussian random variables with variance σ^2: $W(x) \sim \mathcal{N}(0, \sigma^2)$.

In other words, the value $u^0(x)$ of a pixel x is first modified by adding a centered Gaussian noise and then, with probability p, replaced by a uniform value on $[0, M]$. Pure impulse noise can be considered as a particular case of the previous model, obtained when $\sigma = 0$. The final observed noisy image u is a realization of the random image U, after quantization on 8 bits, that is on the set $\{0, 1, \ldots, M\}$.

The most efficient approaches for impulse noise removal usually combine a well chosen impulse noise detector (see Section 2.2) with a restoration method restricted to the set of corrupted pixels. The restoration step can for instance rely on filtering [6], variationnal principles [9, 5], on patch redundancy [12, 7] or combine the two last approaches via dictionnary learning [11], to name just a few. The most recent approaches are often dedicated to the more realistic mixture of Gaussian and impulse noise [8, 12, 7, 11].

This paper focuses on the recent patch-based denoising algorithm called PARIGI [3]. The idea of patch-based approaches is to take advantage of the redundancy of patches (also sometimes called the self-similarity) in images. The central hypothesis of these approaches is that for each patch $\mathbf{P}_1$ in

u, there exist other patches $\mathbf{P}_2, \ldots, \mathbf{P}_n$ in u such that all the $\mathbf{P}_i$ are realizations of the same random patch, obtained from an underlying non-noisy patch $\mathbf{P}^0$ of the original image u^0.

The PARIGI algorithm is composed of three main steps:

1. First, the impulse noise parameter p is estimated (see Section 2.2).

2. Then, for each patch $\mathbf{P}$ in the noisy image u, we look for the most similar replicas of $\mathbf{P}$ in u; this first step boils down to finding the n most similar patches according to a robust distance between patches. This robust distance, described in Section 2.3, can be seen as a way to avoid the explicit detection of noise locations, although it necessitates a rough estimate of the impulse noise parameter p (described in Section 2.2).

3. Finally, each pixel x of Ω is denoised by applying a maximum likelihood estimator to the set of all patches containing x and their corresponding similar patches. The derivation of this estimator is described in Section 2.4.

We first recall the theory behind PARIGI in Section 2, then the algorithm itself in Section 3. For a more detailed analysis, see our SIIMS paper [3].

2 Theory

2.1 Notations

The following notations are used throughout the paper:

- Ω is the image domain

- $\Omega_f = [-f, f] \times [-f, f]$ is the patch domain

- $\mathbf{P}_x$ is the $(2f + 1) \times (2f + 1)$ square image patch defined on Ω_f by

$$\forall \delta \in \Omega_f, \quad \mathbf{P}_x(\delta) = u(x + \delta),$$

- if $\mathbf{P}$ is a patch in u, $\mathbf{P}^0$ is the patch at the same position in u^0.

2.2 Estimation of the Impulse Noise Intensity p

Different methods are possible to estimate p globally on the damaged image. In this implementation of PARIGI, for the sake of simplicity, we propose to rely on an impulse noise detector in order to estimate p. However, note that the information that a pixel is detected is not used as such in the algorithm.

Many impulse noise detectors are proposed in the literature, such as the detectors ROAD [6], ACWMF [1], ROLD [4] or the pixel-wise MAD [2], to cite only a few. The goal of these schemes is to yield a map of noisy (impulse affected) pixels in an image damaged by impulse noise, or by a mixture of impulse and Gaussian noise. An estimation of p can naturally be derived from this estimation by computing the ratio between the number of noisy pixels and the image size. As explained in [3], the estimation of p in the implementation of PARIGI is done with the detector ROAD.

The detector ROAD (for "Rank Ordered Absolute Differences"), proposed in [6], can be described as follows: for each pixel x, the absolute differences between $u(x)$ and $u(y)$ are computed for all $y \neq x$

in a centered 3×3 patch around x. These differences are ordered. The value ROAD(x) is obtained by computing the sum of the 4 smallest differences

$$\text{ROAD}(x) = \sum_{i=1}^{4} r_i(x),$$

where $r_i(x)$ is the ith smallest absolute distance between $u(x)$ and its 8 neighboring values $u(y)$. This value measures how close $u(x)$ is from its neighbors. When ROAD(x) is above a given threshold τ, set as 70, x is considered as noisy. The proportion p of corrupted pixels is then estimated as

$$p = \frac{1}{|\Omega|} \sum_{x \in \Omega} \mathbf{1}_{\text{ROAD}(x) > 70}.$$

Figure 1 shows the quality of this estimator for different values of p on the 512×512 image *Lena* in the case of pure impulse noise (on the left) and in the case of a mixture of impulse noise and Gaussian noise with $\sigma = 10$ (on the right). For each value of p between 0.1 and 0.9 with a step of 0.1, we show the box-plot of the estimation of p for 100 different noise samples. The ROAD-based estimation of p, shown in Figure 1, is quite accurate in both cases and fast to compute.
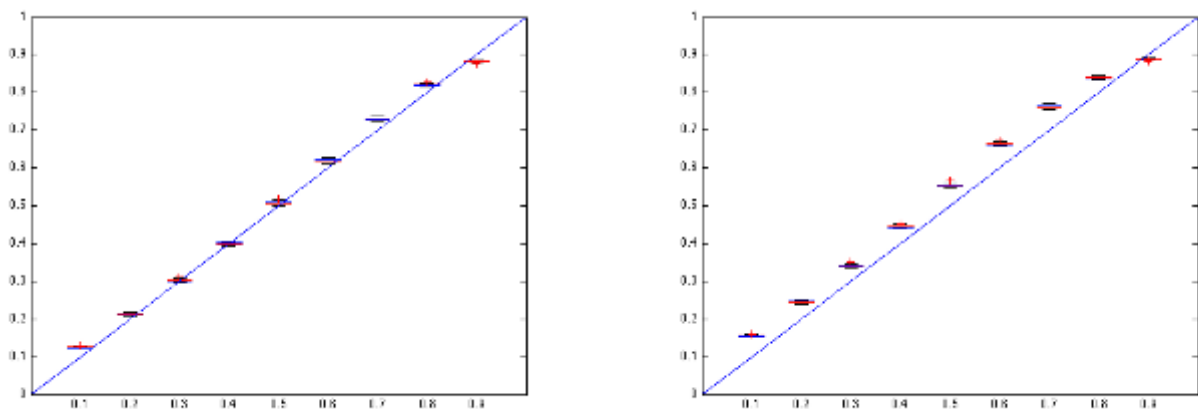

Figure 1: ROAD estimation of p. On the left, statistical results of the estimation of the impulse noise parameter p on the *Lena* image when there is no Gaussian noise. The results $\overline{p}$ are shown as box-plot graphics, where the horizontal axis represents the tested values of p: from 0.1 to 0.9. Boxes are the statistics obtained from 100 samples for each value of p. On the right, same experiment made on the *Lena* image in the case of a mixed noise: impulse and Gaussian with $\sigma = 10$.

2.3 Robust Distance between Patches

The search for similar patches in the noisy image u relies on a robust weighted distance. Let $\mathbf{P}$ and $\mathbf{Q}$ be two patches of the noisy image u. We define a distance between these two patches that is able to measure their similarity despite the presence of noise. It is known that, for non-noisy patches, the L^2 distance is a good distance to measure the similarity between the patches. But when impulse noise is present, many large values of $(P(\delta) - Q(\delta))^2$, $\delta \in \Omega_f$, are observed that are due to the noise and the L^2 distance then contains many outliers. Our distance is built to be similar to the Euclidean distance but robust to large values created by the noise. Let $|P - Q|_{(1)} \leq |P - Q|_{(2)} \leq \cdots \leq |P - Q|_{(2f+1)^2}$ be

the values obtained by ordering the $(2f+1)^2$ values of the differences $|\mathbf{P}(\delta) - \mathbf{Q}(\delta)|$. We then define a robust distance between $\mathbf{P}$ and $\mathbf{Q}$ by

$$D_w(\mathbf{P}, \mathbf{Q})^2 = \sum_{k=1}^{(2f+1)^2} w_k |P - Q|^2_{(k)}, \tag{2}$$

where $w_1, \ldots, w_n$ are positive weights, defined by

$$w_k = \mathcal{B}((2f+1)^2, k, (1-p)^2),$$

where p is the impulse noise intensity estimated on the noisy image u (by the method described in the previous Section 2.2) and $\mathcal{B}$ denotes the tail of the binomial distribution given for all $0 \leq k \leq n$ integers and $q \in [0, 1]$ by

$$\mathcal{B}(n, k, q) = \sum_{i=k}^{n} \binom{n}{i} q^i (1-q)^{n-i}.$$

The idea underlying this choice is the following property: if $\mathbf{P}$ and $\mathbf{Q}$ are two independent random patches obtained from the same original patch $\mathbf{P}^0 = \mathbf{Q}^0$, the probability that the kth difference $|P - Q|_{(k)}$ stems from two untouched pixels is $\mathcal{B}((2f+1)^2, k, (1-p)^2)$ (with the approximation that the smallest distances correspond to untouched pixels). More properties of this distance and some comparisons with other possible robust distances were given in [3]. Notice in particular that when $p = 0$, then the weighted distance D_w is simply the L^2 distance.

2.4 Maximum Likelihood Estimator for the Mixture of Impulse and Gaussian noise

The second step of the PARIGI algorithm consists in computing an estimator of $u^0(x)$ at each point x of Ω. We will explain at the end of the Section on which values this estimator is computed exactly, but first we present the main formula used for this estimator.

Maximum Likelihood Estimator. Assume that we have n values $z_1, \ldots, z_n$, that are independent and identically distributed, following a mixture distribution

$$f_{\mu,\sigma}(z) = \frac{p}{M} + (1-p)g_{\mu,\sigma}(z),$$

where $g_{\mu,\sigma}$ is the Gaussian probability density of mean μ and variance σ^2, defined for all $z \in \mathbb{R}$ by

$$g_{\mu,\sigma}(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2}.$$

Assume that we know the parameter p (see Section 2.2) and that we wish to estimate both the mean μ and the variance σ^2 of the Gaussian part. In this continuous setting, the Maximum Likelihood Estimator (MLE) of (μ, σ) is defined as

$$\begin{aligned}
(\widehat{\mu}, \widehat{\sigma}) &= \arg\max_{\mu',\sigma'} \log \mathbb{P}[z_1, \ldots, z_n | \mu', \sigma'] \\
&= \arg\max_{\mu',\sigma'} \sum_{i=1}^{n} \log\left(\frac{p}{M} + (1-p)g_{\mu',\sigma'}(z_i)\right),
\end{aligned} \tag{3}$$

where $g_{\mu',\sigma'}$ is the Gaussian probability density of mean μ' and variance σ'^2. By introducing the function $f_{\sigma'} : m \mapsto \log\left(\frac{p}{M} + \frac{1-p}{\sqrt{2\pi}\sigma}e^{-\frac{m^2}{2\sigma'^2}}\right)$, this can be rewritten

$$
\begin{aligned}
(\widehat{\mu}, \widehat{\sigma}) &= \arg\max_{\mu',\sigma'} \sum_{i=1}^{n} f_{\sigma'}(\mu' - z_i), \\
&= \arg\max_{\mu',\sigma'} h * f_{\sigma'}(\mu'),
\end{aligned}
\tag{4}
$$

with $*$ the convolution operator and $h : m \mapsto \sum_{i=1}^{n} \delta_{\{z_i=m\}}$ the empirical distribution of the values $\{z_1, \ldots, z_n\}$ on $[0, M]$.

The previous maximum likelihood is written in the continuous setting for the sake of simplicity. In the discrete setting, we approximate the computation of $h * f_{\sigma'}$ by a discrete convolution between the discrete histogram of $\{z_1, \ldots, z_n\}$ and a discretized version of $f_{\sigma'}$.

Estimation in practice. The previous estimator is used at each pixel x in order to estimate $u^0(x)$ and $\sigma(x)$. If $\mathbf{P}_x$ is the patch centered at x and $\mathcal{V}_x^n$ the set of its similar patches in Ω, then we compute $\widehat{u^0(x)}, \widehat{\sigma(x)}$ from the set of observations $\{\mathbf{Q}(0), \mathbf{Q} \in \mathcal{V}_x^n\}$. Observe that in practice, the mixture case is a realistic model for the estimation problem even in presence of pure impulse noise. Indeed, similar patches in an image are never exactly equal and it is a sound hypothesis to assume that their inner variability can be modeled with a standard deviation $\sigma(x)$ that is pixel-dependent.

3   Implementation

3.1   Algorithm

The estimation of p is the first step of PARIGI and is done as described in Section 2.2. In a second step, for each point x in Ω, we seek the n nearest neighbors $\mathbf{P}_y$ of $\mathbf{P}_x$ for the distance D_w introduced in Section 2.3. We restrict this investigation to the points y spanning a $(2t+1) \times (2t+1)$ square $\mathcal{V}_x$ centered at x (see Figure 2)

$$
\mathcal{V}_x = \{x + \delta; \quad \delta \in [-t, t] \times [-t, t]\}.
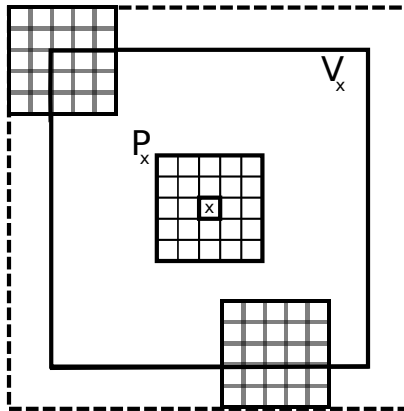\tag{5}
$$

Figure 2: Neighborhood $\mathcal{V}_x$ and possible corresponding patches.

To avoid any computation problem, the image boundaries are extended by symmetry. This allows to define $\mathcal{V}_x^n$, the subset of $\mathcal{V}_x$ corresponding to the centers of the n nearest neighbors of

$\mathbf{P}_x$. At this point, the MLE $(\widehat{u(x)}, \widehat{\sigma(x)})$ could be directly computed at each x from the n-tuple $(u(y), y \in \mathcal{V}_x^n)$. In practice, following the idea of [10], we observe better results by using all patches $\mathbf{P}_{x+\delta}, \delta \in [-f, f] \times [-f, f]$ containing x and all of their replicas in the estimation. The MLE is then computed at each x from the set of values (with possible repetitions)

$$\mathcal{L}_x = \left\{ u(y - \delta); \; \delta \in [-f, f] \times [-f, f] \text{ and } y \in \mathcal{V}_{x+\delta}^n \right\}. \tag{6}$$

The image $\widehat{u}$ constitutes a first denoised version of u. In practice, this restored image is sometimes a little bit too smooth. At the same time, some impulse pixels can remain. In order to recover the "grain" of the original image and to eliminate the remaining impulses, we take into account the estimated standard deviation $\widehat{\sigma}(x)$ at each point x for defining a map of noisy pixels

$$\mathcal{M} = \{x \in \Omega; \; |\widehat{u}(x) - u(x)| > \widehat{\sigma}(x)\}.$$

We then compute the mixture of $\widehat{u}$ and u: $u_2(x) = \widehat{u}(x).\mathbf{1}_{x \in \mathcal{M}} + u(x).(1 - \mathbf{1}_{x \in \mathcal{M}})$, and eventually apply the previous denoising steps to u_2 in order to obtain $\widehat{u_2}$. These steps can be repeated two or three times to improve the final result.

The whole algorithm is described in Algorithm 1.

Algorithm 1: Denoising algorithm PARIGI

input : Image u, number of iterations N_{it} ($N_{it} = 2$ by default)

output: Denoised image $u_{N_{it}+1}$

Initialize $u_1 = u$;

for i from 1 to N_{it} **do**

    Estimate p on u_i; // *See Section 2.2*

    Compute $n = NumberNearestNeighbors(p)$; // *see Table 1 in Section 2.4.*

    for each pixel x **do**

        $\mathcal{L}_x = \emptyset$;

    for each pixel x **do**

        Find $\mathcal{V}_x^n$, the subset of $\mathcal{V}_x$ corresponding to the n nearest neighbors of $\mathbf{P}_x$ for D_w in u_i;

        for each $\delta \in [-f, f] \times [-f, f]$ **do**

            $\mathcal{L}_{x+\delta} = \mathcal{L}_{x+\delta} \cup \{u_i(y + \delta), y \in \mathcal{V}_x^n\}$;

    for each pixel x **do**

        Obtain $(\widehat{u(x)}, \widehat{\sigma(x)})$ as the MLE from the set $\mathcal{L}_x$; // *see Section 2.4.*

    Compute

$$u_{i+1} = \widehat{u}.\mathbf{1}_{\mathcal{M}} + u.(1 - \mathbf{1}_{\mathcal{M}})$$

    where

$$\mathcal{M} = \{x \in \Omega; \; |\widehat{u(x)} - u(x)| > \widehat{\sigma(x)}\}.$$

Return $u_{N_{it}+1}$

3.2   Choice of the Parameters n, f and t

The number n of acceptable neighbors for each patch must be chosen carefully. An empirical study of the quadratic risk for the estimation of $\widehat{u}$ was made in [3]. Obviously, the risk of the estimator in an ideal case, where the number of samples of a noisy patch is potentially infinite, decreases as n gets larger. Therefore, n should in theory be chosen as large as possible. However, in images, the

number of patches that are similar to a given patch is usually limited, and therefore choosing the value of n in practice is a trade-off between a high value (to ensure having enough samples for the estimation), and a small one (to ensure these samples are indeed samples of the same patch). This choice can also be interpreted as a bias-variance trade off, the bias of the estimator increasing with the number of samples, while the variance decreases. The value of n used in practice will depend on the intensity p of the impulse noise, on the standard deviation σ, but also on the values of f and t.

The choices of the half patch size f and of the half search window size t also impact the quality and the performance of the algorithm.

After extensive experiments on several images with different noise levels, we observe that using $f = 3$ and $t = 5$ as default values is a good compromise between computational time and PSNR values. However, using a variable n seems unavoidable for good performances. Table 1 summarizes our choices of n depending on the estimated p. These values of n were selected experimentally as the values at which the PSNR reaches its steady state value (see Figures 3 and 4).

We illustrate the influence of these parameters on PSNR values in Table 2 and in Figures 3 and 4. In the figures, $f = 3$ and $t = 5$ are fixed and n varies. The vertical dashed lines represent the values of n according to Table 1. In Table 2, we show the PSNR values when f and t vary. In this experiment, n is chosen as displayed in Table 1. We also show for each pair (f, t), the best PSNR obtained when n is allowed to change. Observe that these last values are not very different from the ones obtained with n fixed.

p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
n	7	9	12	16	22	35	60	90

Table 1: Number n of patches used in the maximum-likelihood estimation for each value of p.

3.3 Acceleration of the Algorithm

Our method suffers of two main bottlenecks:

- the histogram estimation for each pixel.

- the computation of the optimal denoised values $(\widehat{\mu}, \widehat{\sigma})$ which necessitates several convolutions (cf. Equation (4)).

Even if the computation of the convolution can be easily accelerated by a fast FFT convolution method (the library FFTW in our case), it seems more difficult to optimize the code of the histograms computation.

In order to accelerate the whole algorithm, we partition the noisy image into a set of K sub-images (where K is twice the number of threads in the computer) and we apply our algorithm in parallel on each sub-image using OpenMP.

3.4 Improvements Analysis

In Section 3.1, we have introduced two rules to improve the quality of the restored image: mix the restored image with the noisy one and apply two iterations of the method. Here, we propose to analyze these criteria efficiency by studying both PSNR results (as shown in Table 3) and visual results (as shown in Figure 5).

First, when we analyze the PSNR values of Table 3, the best performances are usually obtained by applying the mixture criterion and only one iteration of the algorithm. But, this analysis is contradicted by the analysis of the visual performances (Figure 5). Indeed, by applying two iterations,

Impulse noise intensity $p = 0.1$				
	$f = 2$	$f = 3$	$f = 4$	$f = 5$	$f = 6$
$t = 3$	32.87 (34.80)	34.50 (35.00)	35.23 (35.23)	35.53 (35.53)	35.39 (35.39)
$t = 5$	31.55 (35.12)	35.00 (35.09)	35.57 (35.57)	35.81 (35.81)	35.73 (35.73)
$t = 7$	30.29 (35.21)	35.45 (35.45)	35.67 (35.67)	35.87 (35.87)	35.80 (35.80)
$t = 9$	29.06 (35.38)	35.51 (35.51)	35.78 (35.78)	**35.92 (35.92)**	35.82 (35.82)

Impulse noise intensity $p = 0.2$				
	$f = 2$	$f = 3$	$f = 4$	$f = 5$	$f = 6$
$t = 3$	33.50 (33.50)	34.01 (34.01)	34.15 (34.15)	34.15 (34.15)	34.10 (34.10)
$t = 5$	33.80 (33.80)	34.15 (34.15)	**34.24 (34.24)**	**34.24 (34.24)**	34.18 (34.18)
$t = 7$	33.93 (33.93)	34.17 (34.17)	34.19 (34.19)	34.22 (34.22)	34.17 (34.17)
$t = 9$	34.13 (34.13)	34.18 (34.18)	34.12 (34.12)	34.15 (34.15)	34.13 (34.13)

Impulse noise intensity $p = 0.3$				
	$f = 2$	$f = 3$	$f = 4$	$f = 5$	$f = 6$
$t = 3$	31.24 (31.67)	32.22 (32.22)	32.66 (32.66)	32.77 (32.77)	32.73 (32.73)
$t = 5$	31.43 (32.08)	32.43 (32.48)	32.75 (32.75)	32.93 (32.93)	**32.96 (32.96)**
$t = 7$	31.80 (32.41)	32.63 (32.63)	32.79 (32.79)	32.91 (32.91)	32.93 (32.93)
$t = 9$	32.02 (32.60)	32.66 (32.66)	32.78 (32.78)	32.85 (32.85)	32.83 (32.83)

Impulse noise intensity $p = 0.4$				
	$f = 2$	$f = 3$	$f = 4$	$f = 5$	$f = 6$
$t = 3$	29.85 (30.07)	30.79 (30.79)	31.05 (31.05)	31.17 (31.17)	31.16 (31.16)
$t = 5$	30.09 (30.37)	31.02 (31.02)	31.27 (31.27)	31.42 (31.42)	31.38 (31.38)
$t = 7$	30.23 (30.49)	31.01 (31.01)	31.19 (31.19)	31.37 (31.37)	**31.44 (31.44)**
$t = 9$	30.35 (30.65)	31.00 (31.00)	31.14 (31.14)	31.28 (31.28)	31.34 (31.34)

Impulse noise intensity $p = 0.5$				
	$f = 2$	$f = 3$	$f = 4$	$f = 5$	$f = 6$
$t = 3$	27.95 (28.30)	28.72 (28.73)	29.15 (29.15)	29.32 (29.32)	29.34 (29.34)
$t = 5$	28.10 (28.45)	29.03 (29.05)	29.51 (29.51)	29.82 (29.82)	29.95 (29.95)
$t = 7$	28.12 (28.51)	29.11 (29.12)	29.50 (29.50)	29.82 (29.82)	**30.00 (30.00)**
$t = 9$	28.09 (28.53)	29.12 (29.13)	29.46 (29.46)	29.71 (29.71)	29.86 (29.86)

Impulse noise intensity $p = 0.6$				
	$f = 2$	$f = 3$	$f = 4$	$f = 5$	$f = 6$
$t = 3$	26.68 (26.71)	26.89 (26.89)	27.02 (27.07)	27.14 (27.22)	27.16 (27.28)
$t = 5$	26.10 (26.41)	26.76 (26.80)	27.16 (27.16)	27.47 (27.49)	**27.68 (27.70)**
$t = 7$	25.72 (26.03)	26.67 (26.69)	27.06 (27.06)	27.39 (27.40)	27.61 (27.63)
$t = 9$	25.47 (25.85)	26.52 (26.52)	26.93 (26.93)	27.21 (27.23)	27.42 (27.47)

Table 2: Influence of the half patch size f and the search window size t for different intensities p of the impulse noise. In **bold** the best PSNR for a given p. The PSNR shown are estimated by using the number of patches n defined in Table 1. We also show in brackets the best PSNR obtained when n is allowed to change. All these values have been computed over the standard image *Lena* of size 512×512.
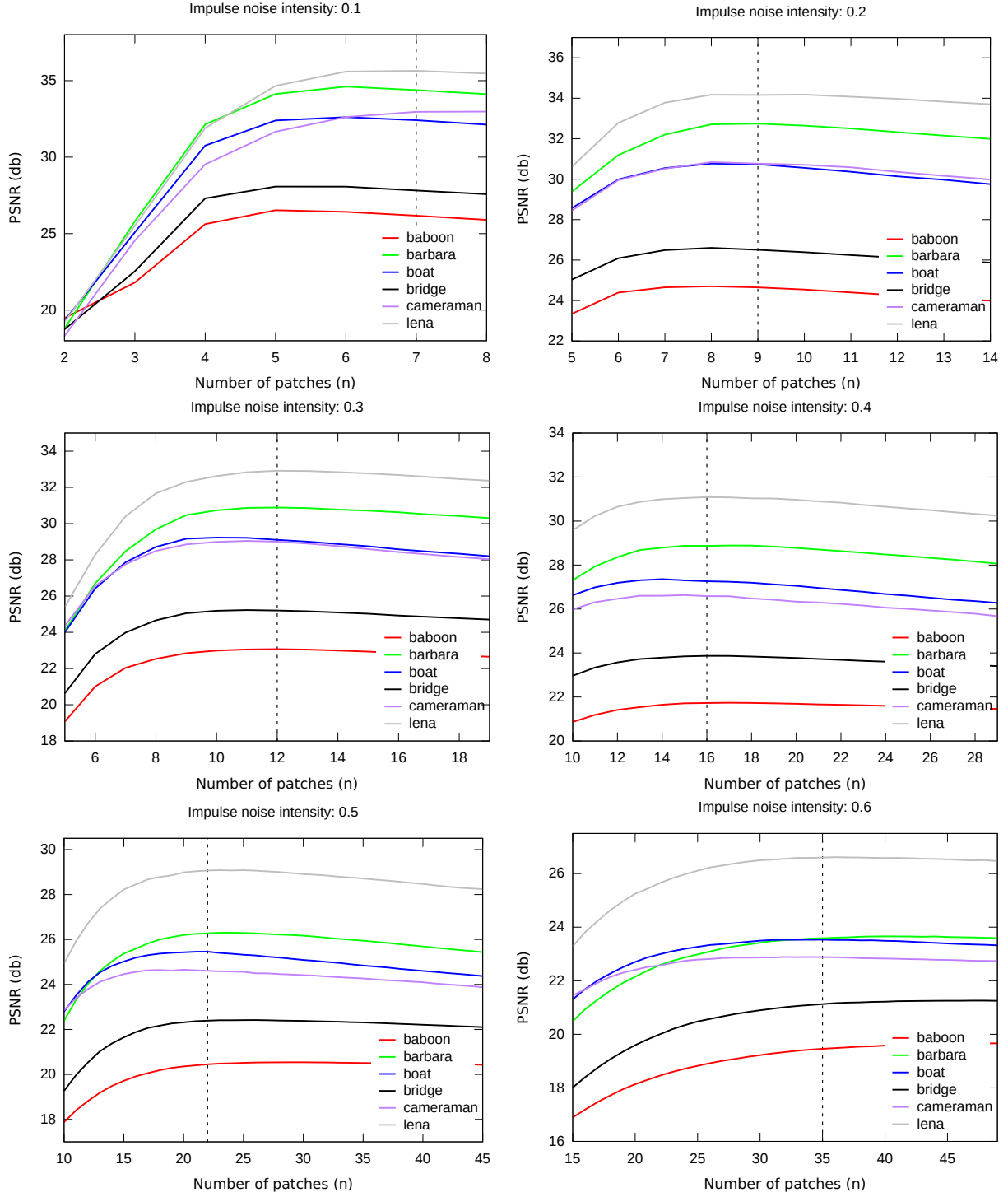
Figure 3: Evolution of the PSNR for different values of the impulse noise intensity p without any Gaussian noise and different numbers of patches n used for the restoration. For the experiments, we have considered a set of six different images. We have fixed the half patch size to $f = 3$ and the search window size to $t = 5$. Vertical dashed lines represent the values of n according to Table 1. Images used for the experiment are all standard images of size 512×512.

the method tends to remove the small details of the image but removes also the defects which are still present after the first iteration. This is particularly the case for high values of noise. For small noise values, the performances are almost similar.
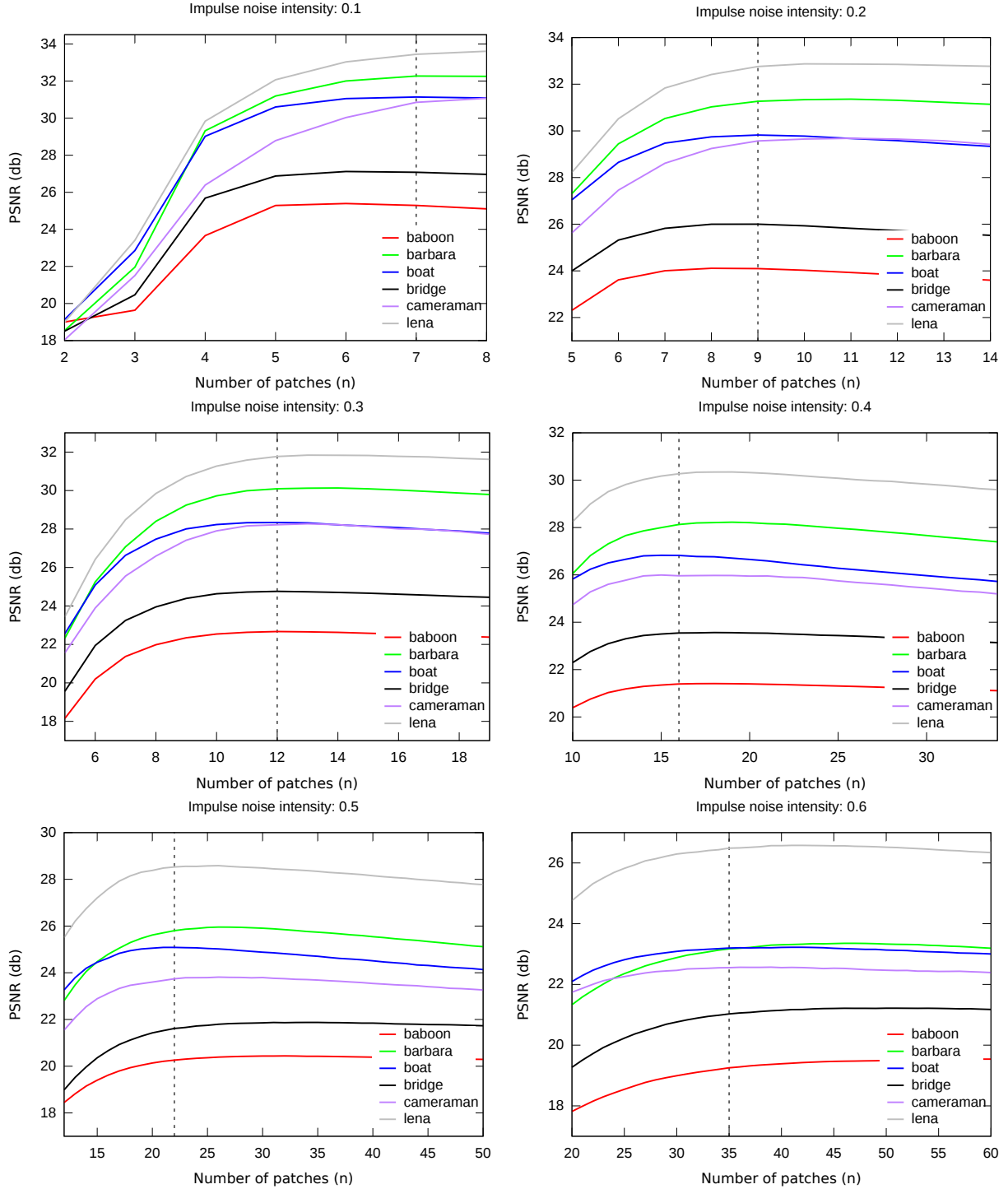
Figure 4: Evolution of the PSNR for different values of the impulse noise intensity p, a Gaussian noise standard deviation $\sigma = 10$ and different numbers of patches n used for the restoration. For the experiments, we have considered a set of six different images. We have fixed the half patch size to $f = 3$ and the search window size to $t = 5$. Vertical dashed lines represent the values of n according to Table 1. Images used for the experiment are all standard images of size 512×512.

4 Results

In this section, we present some results of PARIGI on different images: *Barbara, Simpson, Lena, Bolivie*.

Mixture / 2 Iterations:	no/no	yes/no	no/yes	yes/yes
$\sigma = 0$ $p = 10\%$	34.25	33.74	**37.07**	36.02
$p = 20\%$	32.93	32.86	**34.63**	34.14
$p = 30\%$	31.39	31.74	32.26	**32.31**
$p = 40\%$	29.19	29.94	29.46	**30.04**
$p = 50\%$	26.68	**27.64**	26.63	27.49
$p = 60\%$	23.97	**24.86**	23.80	24.64
$p = 70\%$	21.46	**22.24**	21.33	22.04
$p = 80\%$	18.80	**20.27**	18.72	20.08
$\sigma = 10$ $p = 10\%$	**32.30**	31.92	30.46	31.24
$p = 20\%$	**31.46**	31.39	30.33	31.02
$p = 30\%$	30.12	**30.41**	29.48	30.14
$p = 40\%$	28.42	**29.10**	28.02	28.82
$p = 50\%$	26.24	**27.22**	25.91	26.92
$p = 60\%$	23.65	**24.56**	23.42	24.32
$p = 70\%$	21.32	**22.12**	21.15	21.91
$p = 80\%$	18.58	**20.13**	18.49	19.93
$\sigma = 20$ $p = 10\%$	28.98	**29.07**	25.85	27.40
$p = 20\%$	28.67	**28.91**	26.14	27.60
$p = 30\%$	27.89	**28.35**	26.04	27.37
$p = 40\%$	26.63	**27.38**	25.43	26.69
$p = 50\%$	23.03	**23.66**	22.25	23.24
$p = 60\%$	22.85	**23.67**	22.46	23.38
$p = 70\%$	21.02	**21.81**	20.80	21.58
$p = 80\%$	18.43	**20.11**	18.33	19.91

Table 3: Analysis of the influence of the different options of our method. In **bold** the best PSNR for given σ and p.

The PSNR values for different standard deviations σ of the Gaussian noise and intensities p of the impulse noise are summarized in Table 4. Concerning the visual results, we present them in figures 6 to 14.

For high impulse noise intensities (as in Figure 8), our method can recover textural details by using the image self-similarity (as for example the stripes in the image *Barbara*, first line of Figure 8). When the self similarity is low (for example for the image *Bolivie*, last line of Figure 8), our method has some difficulties to obtain visually good results. Moreover, for these high intensities, small details or edges of the image can disappear. This phenomenon can be observed for the image *Simpson* (second line of Figure 8).

For reasonable intensities of impulse noise (as in figures 6 and 7), these problems are limited. Indeed, as explained in Section 3.2, the number of patches used for the restoration is linked to the intensity of the impulse noise. Consequently, for small intensities, less patches are used, allowing a better restoration.

Our method is also designed to recover an image corrupted by both Gaussian noise and impulse noise. We present the results in figures 9 to 14. For small values of the Gaussian noise standard deviation (as in figures 9, 10 and 11), both Gaussian and impulse noise are removed. Nevertheless for high values of Gaussian noise (as in figures 12, 13 and 14) some Gaussian noise is still present. This phenomenon can be explained by the inability of the method to correctly estimate the Gaussian noise variance.

Julie Delon, Agnès Desolneux, Thierry Guillemot

		Barbara	Simpson	Lena	Bolivie
$\sigma = 0$	$p = 10\%$	36.02	38.35	37.51	21.47
	$p = 20\%$	34.14	36.11	35.65	20.43
	$p = 30\%$	32.31	33.38	33.90	19.45
	$p = 40\%$	30.04	30.52	32.04	18.44
	$p = 50\%$	27.49	28.32	29.91	17.33
	$p = 60\%$	24.64	26.00	27.39	16.36
	$p = 70\%$	22.04	24.21	25.22	15.67
	$p = 80\%$	20.08	22.95	23.36	14.94
$\sigma = 10$	$p = 10\%$	31.29	32.46	32.22	20.94
	$p = 20\%$	31.02	32.24	32.12	20.23
	$p = 30\%$	30.14	31.20	31.70	19.28
	$p = 40\%$	28.82	29.38	30.69	18.25
	$p = 50\%$	26.92	27.72	29.01	17.22
	$p = 60\%$	24.32	25.94	27.28	16.42
	$p = 70\%$	21.91	24.16	25.04	15.73
	$p = 80\%$	19.93	23.03	23.35	14.93
$\sigma = 20$	$p = 10\%$	27.41	27.86	27.82	19.70
	$p = 20\%$	27.60	28.20	28.25	18.52
	$p = 30\%$	27.37	27.62	28.44	18.38
	$p = 40\%$	26.69	27.00	28.09	17.68
	$p = 50\%$	23.24	26.10	27.50	16.98
	$p = 60\%$	23.38	24.96	26.16	16.27
	$p = 70\%$	21.58	23.77	24.70	15.69
	$p = 80\%$	19.91	22.59	22.87	14.89

Table 4: PSNR values of the restored images for different Gaussian noise standard deviations σ and different intensities p of the impulse noise.

5 Conclusion and Discussion

In this work, we have proposed an implementation of the PARIGI method to remove impulse and Gaussian noise from gray scale images. The method is a patch-based algorithm that first finds similar patches (using a special distance that is robust and adapted to the noise), and then recovers the gray level values by a maximum likelihood estimator. Similarly to other patch-based approaches, this method deals particularly well with repetitive patterns and texture. However, it has a tendency to oversmooth singular regions. To improve the results, an interesting direction would be to explore the use of patch prior models.

A possible extension of this work would be to deal with color images. In the multichannel case, impulse noise can be defined in an analogous way. However, the detection of impacted pixels is much easier, since it can rely on the correlation and redundancy between the different channels. As a consequence, removing impulse noise in color images should be treated as an inpainting problem.

Acknowledgments

Figure 5: Zoom on restored images for an impulse noise intensity $p = 60\%$. From top to bottom and from left to right : original image, noisy image, restored image using one iteration without the mixture criterion (PSNR=23.97), restored image using two iterations without the mixture criterion (PSNR=23.80), restored image using one iteration with the mixture criterion (PSNR=24.86), restored image using two iterations with the mixture criterion (PSNR=24.64). The PSNR are estimated by considering the complete image (and not only the zoom presented in this figure). Contrary to what we expect by analyzing Table 3, using two iterations gives better visual performances.

Figure 6: Restored images for an intensity of impulse noise $p = 20\%$. For each column, from left to right: original image, noisy image, our method.
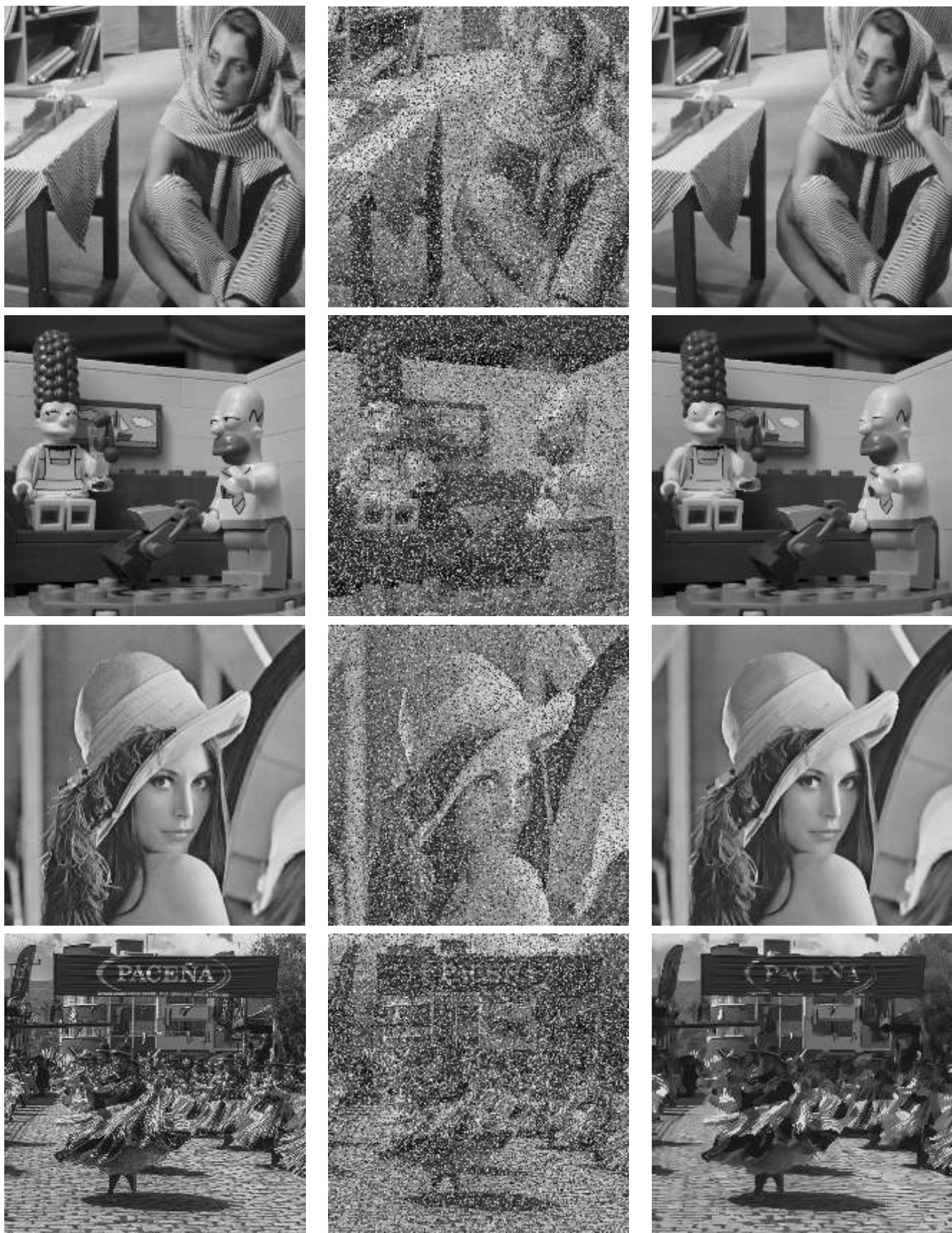
Figure 7: Restored images for an intensity of impulse noise $p = 40\%$. For each column, from left to right: original image, noisy image, our method.
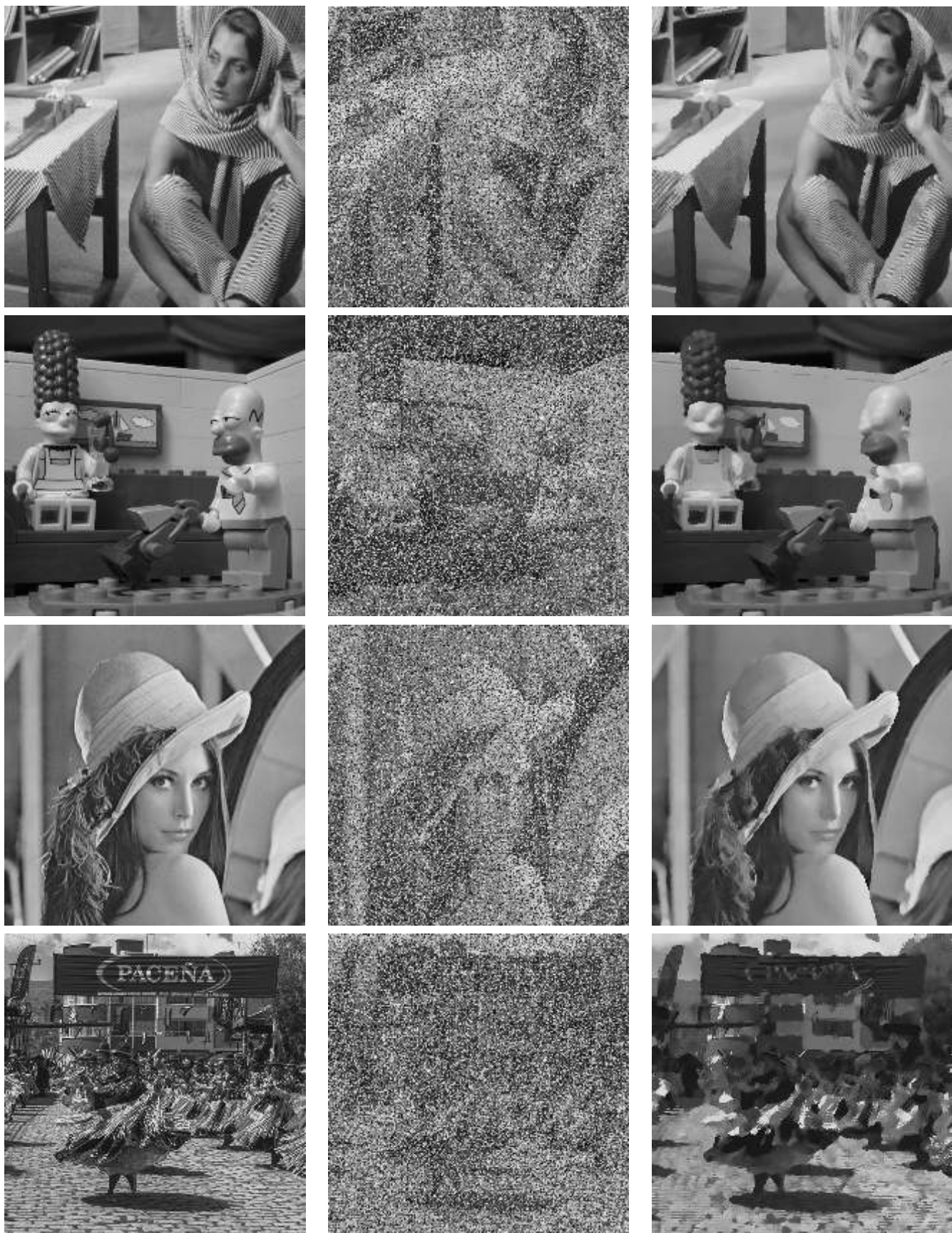
Figure 8: Restored images for an intensity of impulse noise $p = 60\%$. For each column, from left to right: original image, noisy image, our method.

Figure 9: Restored images for a Gaussian noise of standard deviation $\sigma = 10$ and an intensity of impulse noise $p = 20\%$. For each column, from left to right: original image, noisy image, our method.

Figure 10: Restored images for a Gaussian noise of standard deviation $\sigma = 10$ and an intensity of impulse noise $p = 40\%$. For each column, from left to right: original image, noisy image, our method.
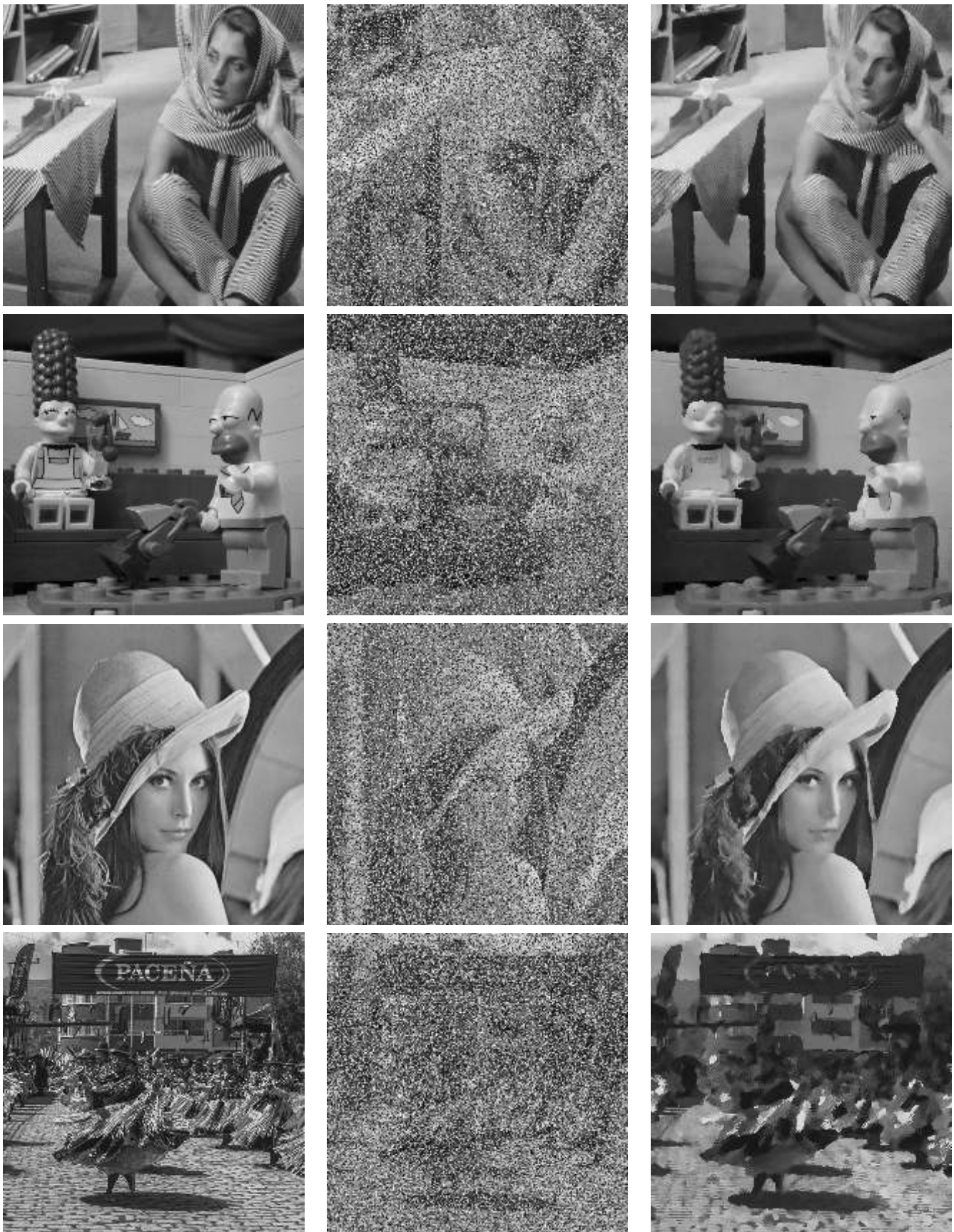
Figure 11: Restored images for a Gaussian noise of standard deviation $\sigma = 10$ and an intensity of impulse noise $p = 60\%$. For each column, from left to right: original image, noisy image, our method.

Figure 12: Restored images for a Gaussian noise of standard deviation $\sigma = 20$ and an intensity of impulse noise $p = 20\%$. For each column, from left to right: original image, noisy image, our method.

Figure 13: Restored images for a Gaussian noise of standard deviation $\sigma = 20$ and an intensity of impulse noise $p = 40\%$. For each column, from left to right: original image, noisy image, our method.
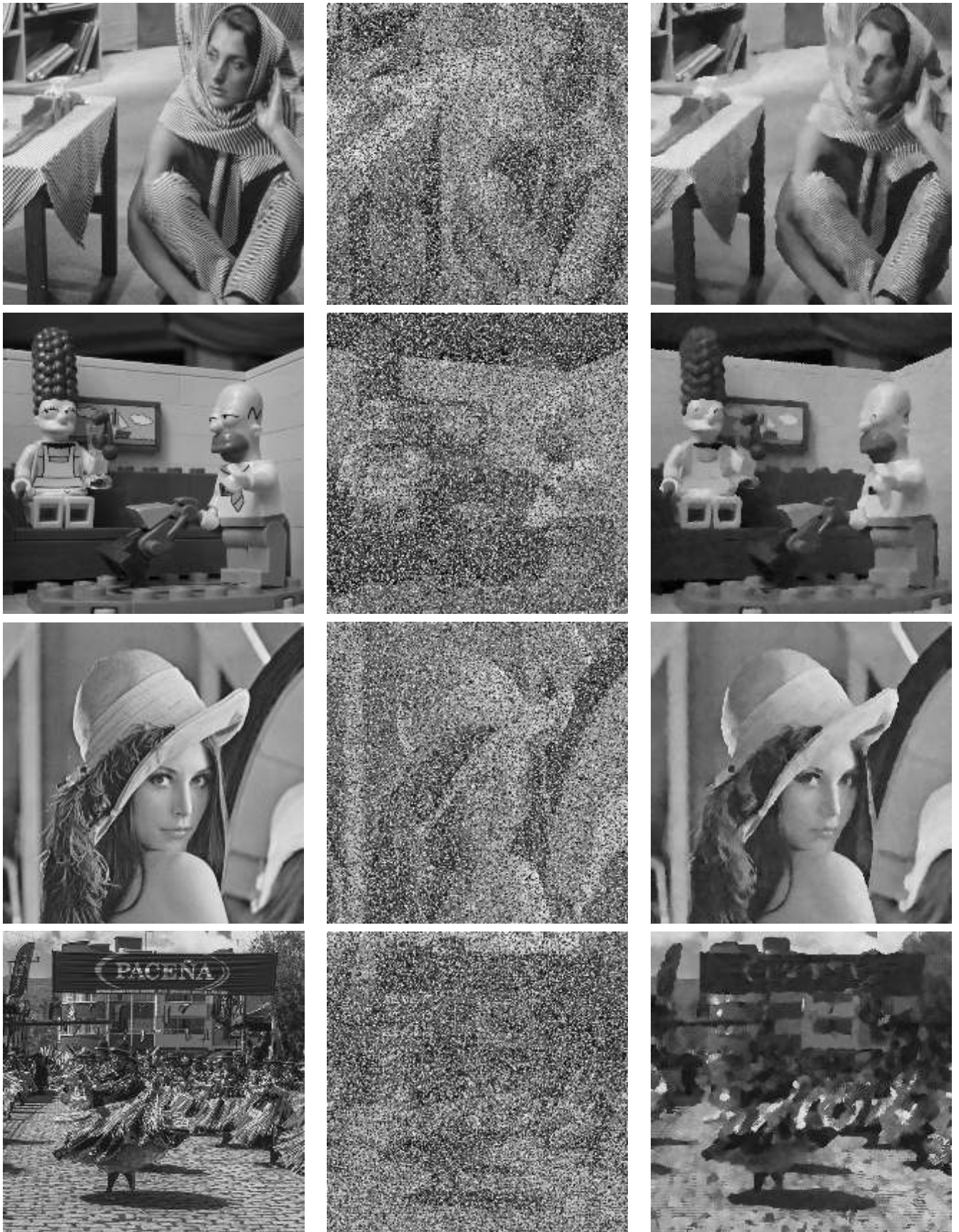
Figure 14: Restored images for a Gaussian noise of standard deviation $\sigma = 20$ and an intensity of impulse noise $p = 60\%$. For each column, from left to right: original image, noisy image, our method.

Image Credits

 Standard test images

 by J. Delon, CC-BY-NC 4.0 license.

 by T. Guillemot, CC-BY-NC 4.0 license.

References

[1] T. Chen and H.R. Wu, *Adaptive impulse detection using center-weighted median filters*, IEEE Signal Processing Letters, 8 (2001), pp. 1–3. http://dx.doi.org/10.1109/97.889633.

[2] V. Crnojevic, V. Senk, and Z. Trpovski, *Advanced impulse detection based on pixel-wise MAD*, IEEE Signal Processing Letters, 11 (2004), pp. 589–592. http://dx.doi.org/10.1109/LSP.2004.830117.

[3] J. Delon and A. Desolneux, *A patch-based approach for removing mixed Gaussian-impulse noise*, SIAM Journal on Imaging Sciences, 6 (2013), pp. 1140–1174. http://dx.doi.org/10.1137/120885000.

[4] Y. Dong, R.H. Chan, and S. Xu, *A detection statistic for random-valued impulse noise*, IEEE Transactions on Image Processing, 16 (2007), pp. 1112–1120. http://dx.doi.org/10.1109/TIP.2006.891348.

[5] Y. Dong and S. Xu, *A New Directional Weighted Median Filter for Removal of Random-Valued Impulse Noise*, IEEE Signal Processing Letters, 14 (2007), pp. 193–196. http://dx.doi.org/10.1109/LSP.2006.884014.

[6] R. Garnett, T. Huegerich, C. Chui, and W. He, *A universal noise removal algorithm with an impulse detector*, IEEE Transactions on Image Processing, 14 (2005), pp. 1747–1754. http://dx.doi.org/10.1109/TIP.2005.857261.

[7] Haijuan Hu, Bing Li, and Quansheng Liu, *Removing mixture of Gaussian and impulse noise by patch-based weighted means*, Journal of Scientific Computing, 67 (2015), pp. 103–129. http://dx.doi.org/10.1007/s10915-015-0073-9.

[8] Y.-M. Huang, M.K. Ng, and Y.-W. Wen, *Fast Image Restoration Methods for Impulse and Gaussian Noises Removal*, IEEE Signal Processing Letters, 16 (2009), pp. 457–460. http://dx.doi.org/10.1109/LSP.2009.2016835.

[9] M. Nikolova, *A variational approach to remove outliers and impulse noise*, Journal of Mathematical Imaging and Vision, 20 (2004), pp. 99–120. http://dx.doi.org/10.1023/B:JMIV.0000011326.88682.e5.

[10] J. Salmon and Y. Strozecki, *Patch reprojections for Non Local methods*, Signal Processing, 92 (2012), pp. 447–489. http://dx.doi.org/10.1016/j.sigpro.2011.08.011.

[11] Yu Xiao, Tieyong Zeng, Jian Yu, and Michael K. Ng, *Restoration of images corrupted by mixed Gaussian-impulse noise via l_1-l_0 minimization*, Pattern Recognition, 44 (2011), pp. 1708–1720. http://dx.doi.org/10.1016/j.patcog.2011.02.002.

[12] Bo Xiong and Zhouping Yin, *A universal denoising framework with a new impulse detector and nonlocal means*, IEEE Transactions on Image Processing, 21 (2012), pp. 1663–1675. http://dx.doi.org/10.1109/TIP.2011.2172804.

[12] Bo Xiong and Zhouping Yin, *A universal denoising framework with a new impulse detector and nonlocal means*, IEEE Transactions on Image Processing, 21 (2012), pp. 1663–1675.