

# Foveated self-similarity in nonlocal image filtering

Alessandro Foi<sup>a</sup> and Giacomo Boracchi<sup>b</sup>

<sup>a</sup>Department of Signal Processing, Tampere University of Technology,  
P.O. Box 553, 33101 Tampere, Finland

<sup>b</sup>Dipartimento di Elettronica e Informazione, Politecnico di Milano,  
via G. Ponzio, 34/5, 20133 Milano, Italy

## ABSTRACT

Nonlocal image filters suppress noise and other distortions by searching for similar patches at different locations within the image, thus exploiting the self-similarity present in natural images. This similarity is typically assessed by a windowed distance of the patches pixels. Inspired by the human visual system, we introduce a patch foveation operator and measure patch similarity through a foveated distance, where each patch is blurred with spatially variant point-spread functions having standard deviation increasing with the spatial distance from the patch center. In this way, we install a different form of self-similarity in images: the foveated self-similarity.

We consider the Nonlocal Means algorithm (NL-means) for the removal of additive white Gaussian noise as a simple prototype of nonlocal image filtering and derive an explicit construction of its corresponding foveation operator, thus yielding the Foveated NL-means algorithm.

Our analysis and experimental study show that, to the purpose of image denoising, the foveated self-similarity can be a far more effective regularity assumption than the conventional windowed self-similarity. In the comparison with NL-means, the proposed foveated algorithm achieves a substantial improvement in denoising quality, according to both objective criteria and visual appearance, particularly due to better contrast and sharpness. Moreover, foveation is introduced at a negligible cost in terms of computational complexity.

**Keywords:** nonlocal image modeling, foveation, image denoising, human visual system, self-similarity.

## 1. INTRODUCTION

We investigate the connections between the human visual system (HVS) and nonlocal algorithms in imaging. Specifically, we focus on the role of foveation in nonlocal image filtering. As a simple prototype of nonlocal filters for images, we consider the Nonlocal Means (NL-means) algorithm<sup>1</sup> for removal of additive white Gaussian noise. Its basic idea is to build a pointwise estimate of the image where each pixel is obtained as a weighted average of pixels that lie at the center of patches (i.e. neighborhoods, regions) that are similar to the patch centered at the estimated pixel. The estimates are nonlocal since similar patches can be found at different locations within the image and, in principle, the averages can be calculated over all pixels of the image. The pairwise similarity between patches (often referred to as nonlocal self-similarity) determines the weights used in the averaging, so that the higher is the similarity the larger are the weights. In NL-means, the similarity between different patches is measured through the windowed quadratic distance of the gray-level intensities (i.e. windowed photometric distance), where the window function decays smoothly to zero as we move away from center of the corresponding patch.

In this work, we replace such windowed distance with a foveated distance: instead of multiplying the photometric differences against a window function, we compute the distance between foveated patches, i.e. patches blurred by point-spread functions having increasing standard deviation (thus increasing spread) as the spatial distance from the center of the patch grows. If we treat the point to be estimated as a fixation point, this foveated distance mimics the inability of the HVS to perceive details at the periphery of the center of attention.

Our analysis and experimental study show that utilizing a foveated self-similarity instead of the conventional windowed self-similarity leads to an improvement in the resulting image estimate, according both to objective criteria and visual appearance, particularly due to better contrast and sharpness.

---

This work was supported by the Academy of Finland (project no. 213462, Finnish Programme for Centres of Excellence in Research 2006-2011, project no. 129118, Postdoctoral Researcher's Project 2009-2011, and project no. 252547, Academy Research Fellow 2011-2016). E-mail contacts: [alessandro.foi@tut.fi](mailto:alessandro.foi@tut.fi), [giacomo.boracchi@polimi.it](mailto:giacomo.boracchi@polimi.it).

## 2. OBSERVATION MODEL

We consider observations as noisy grayscale images  $z : X \rightarrow \mathbb{R}$  that can be modeled as

$$z(x) = y(x) + \eta(x), \quad x \in X \subset \mathbb{Z}^2, \quad (1)$$

where  $X \subset \mathbb{Z}^2$  is the image domain, which is assumed a regular pixel grid,  $y : X \rightarrow \mathbb{R}$  is the unknown original image, and  $\eta : X \rightarrow \mathbb{R}$  is i.i.d. Gaussian white noise,  $\eta(\cdot) \sim \mathcal{N}(0, \sigma^2)$ . The purpose of any denoising algorithm is to provide an estimate  $\hat{y}$  of the original image  $y$ .

Image patches play a central role and, for the sake of clarity, we introduce a specific notation for them. Let  $U \subset \mathbb{Z}^2$  be a neighborhood centered at the origin; we define a patch centered at a pixel  $x \in X$  in the observation  $z$  as

$$\mathbf{z}_x(u) = z(u + x), \quad u \in U.$$

Similarly, we define the noise-free patches as  $\mathbf{y}_x(u) = y(u + x)$ ,  $u \in U$ .

## 3. NONLOCAL MEANS

The use of nonlocal self-similarity in image processing gained significance with the fractal models of natural images (e.g.,<sup>2</sup>). For the specific task of noise attenuation, nonlocality has been initially considered in,<sup>3</sup> and later developed in the NL-means algorithm,<sup>1,4</sup> representing a breakthrough in image denoising and inspiring several powerful algorithms in the following years (e.g., BM3D<sup>5</sup> or SAFIR<sup>6</sup>). Nonlocal filtering and, more generally, patch-based algorithms lately became established paradigms, which are successfully applied to a broad range of imaging applications. For a comprehensive overview on modern denoising approaches and algorithms we refer the reader to,<sup>7</sup> while in the present paper, for the sake of simplicity and to elucidate our contribution, we restrict our attention to the NL-means algorithm.

In its basic implementation, the NL-means admits the following simple and elegant formulation. The denoised image  $\hat{y}$  consists in a weighted average of potentially\* all the image pixels, i.e.

$$\hat{y}(x_1) = \sum_{x_2 \in X} w(x_1, x_2) z(x_2), \quad \forall x_1 \in X, \quad (2)$$

where  $\{w(x_1, x_2)\}_{x_2 \in X}$  is the set of adaptive weights that characterize the pixel  $x_1$ , which are positive and sum to one. Each weight  $w(x_1, x_2)$  is determined by the similarity between the two patches  $\mathbf{z}_{x_1}$  and  $\mathbf{z}_{x_2}$ , as

$$w(x_1, x_2) = e^{-\frac{d(x_1, x_2)}{h^2}} / \sum_{x \in X} e^{-\frac{d(x_1, x)}{h^2}}, \quad (3)$$

where  $d(x_1, x_2)$  is a distance measure between image patches centered at  $x_1$  and  $x_2$ , and  $h > 0$  is a filtering parameter controlling the decay of the exponential function. Equation (3) assigns larger weights to the terms  $z(\cdot)$  in (2) that correspond to pixels belonging to similar patches (i.e. where the distance between patches  $d(x_1, x_2)$  is small), regardless of their location within the image. This motivates the nonlocal denomination of the algorithm. Thanks to such a weighting mechanism, the NL-means enforces the self-similarity of natural images, which turns out to be an effective regularity prior for suppressing the noise.

The distance  $d$  is defined as a windowed quadratic distance between patches, i.e.

$$d(x_1, x_2) = \left\| \mathbf{z}_{x_1} \sqrt{\mathbf{k}} - \mathbf{z}_{x_2} \sqrt{\mathbf{k}} \right\|_2^2 = \left\| (\mathbf{z}_{x_1} - \mathbf{z}_{x_2})^2 \mathbf{k} \right\|_1 = \sum_{u \in U} (z(u + x_1) - z(u + x_2))^2 \mathbf{k}(u), \quad (4)$$

being  $\mathbf{k}$  a non-negative windowing kernel defined over  $U$ , which allows to adjust the contribution of each term depending on the position of  $u$  with respect to the patch center. Typically,  $\mathbf{k}$  is rotational symmetric and the weights  $\mathbf{k}(u)$  are determined by the spatial distance from the center. In the original formulation of NL-means,<sup>1</sup> the authors suggest using a Gaussian function (with fixed standard deviation) as windowing kernel  $\mathbf{k}$ , while in practice various windowing kernels having maximum at the patch center are also adopted.<sup>8</sup>

---

\*In practice, due to computational aspects, for each  $x_1$ , only pixels  $x_2$  belonging to a limited search window around  $x_1$  are considered for the sum (2).

The motivating idea of using (4) is very practical: the similarity between  $y(x_1)$  and  $y(x_2)$  (which are not available) can be assessed by computing the similarity of the corresponding noisy patches, i.e.,  $\mathbf{z}_{x_1}$  and  $\mathbf{z}_{x_2}$ . The noise influence is thus reduced by evaluating the similarity over patches, and the decay of  $\mathbf{k}$  reflects to which extent the similarity between  $y(u+x_1)$  and  $y(u+x_2)$  at each  $u \neq 0$  implicates the similarity between  $y(x_1)$  and  $y(x_2)$ .

Given a pair of arbitrary noise-free patches  $\mathbf{y}_{x_1}, \mathbf{y}_{x_2}$ , the term  $(\mathbf{z}_{x_1} - \mathbf{z}_{x_2})^2(u)$  follows a non-central chi-squared distribution having mean  $(\mathbf{y}_{x_1} - \mathbf{y}_{x_2})^2(u) + 2\sigma^2$ , thus, we can easily compute the expectation of the distance operator (4)  $d(x_1, x_2)$  as

$$\begin{aligned} E\{d(x_1, x_2)\} &= E\left\{\left\|\left(\mathbf{z}_{x_1} - \mathbf{z}_{x_2}\right)^2 \mathbf{k}\right\|_1\right\} = \left\|E\left\{\left(\mathbf{z}_{x_1} - \mathbf{z}_{x_2}\right)^2\right\} \mathbf{k}\right\|_1 = \\ &= \left\|\left(\left(\mathbf{y}_{x_1} - \mathbf{y}_{x_2}\right)^2 + 2\sigma^2\right) \mathbf{k}\right\|_1 = \left\|\left(\mathbf{y}_{x_1} - \mathbf{y}_{x_2}\right)^2 \mathbf{k}\right\|_1 + 2\sigma^2 \sum_{u \in U} \mathbf{k}(u), \end{aligned} \quad (5)$$

where  $E\{\cdot\}$  denotes the mathematical expectation.

The above equations point out that the content of the patches is compared in a pixelwise manner, and that all the information from the full high-resolution image is conveyed into the similarity test (4), which indeed determines the filtering at the central pixel only. The likely variations in the high-frequency content at pixels close to the patch center may therefore prevent the joint nonlocal filtering of otherwise mutually similar patches.

### 3.1 Extensions on the basic NL-means algorithm

The literature presents several modification to the basic NL-means scheme, typically aiming at speeding up the computations, or at improving the algorithm by introducing adaptive mechanisms to set the parameters, similarity measures enjoying invariant properties, or modifications at the filtering stage.

Already in<sup>1</sup> a multiscale solution that speeds up the basic implementation of NL-means is presented: to reduce the computational burden, similar patches are searched within a sub-sampled image, while the similarity measure is then refined on the (full-size) noisy image, where the filtering is also performed.

An extension of NL-means where the filtering parameters as well as the patch size vary spatially within the image, depending on the image content, is proposed in,<sup>6</sup> while a numerical study concerning the selection of the search window size, as well as a new criteria to set  $w(x_1, x_1)$ , i.e. the weight used the central pixel, are presented in.<sup>9</sup> The same issues have been addressed in.<sup>10</sup>

Several extensions of the patch-similarity measure have been proposed to exploit image redundancy up to a class of geometric transformation: in the basic NL-means, the patch similarity is defined up to translations and therefore similar patches can be at different locations of the image but must share the same scale and orientation. Modified similarity measures that handle rotated patches have been presented in,<sup>11</sup> by exploiting patch clustering, and in,<sup>10</sup> where patch centroids or structure tensors are used to align rotated patches. In<sup>12</sup> patch similarity is extended up to scale and rigid transformations, by enforcing SIFT descriptors<sup>13</sup> to map patches in a canonical form, which does not depend on scale and orientation.

## 4. FOVEATION

Because of the uneven size and organization of photo-receptive cells and ganglions in the human eye, the retinal image features various spatially variant properties.<sup>14-17</sup> In particular, the visual acuity (and hence the sharpness of the resulting retinal image) is highest at the middle of the retina, in a depression termed *fovea*, where the concentration of cones has its peak. At the periphery of the retina, vision is instead blurry. Thus, the retinal image is sharp at the center of the gaze (point of fixation) and becomes progressively blurred as the distance from the center increases. This phenomenon, termed *foveated vision* or *foveated imaging*, is illustrated in Figure 1. There are many anatomical and physiological explanations for foveation, including improved low-light vision and the limited capacity of the optic nerve.

### 4.1 Foveation operator, foveated patches, and foveated distance

Inspired by the HVS, we replace the windowed distance  $d(x_1, x_2)$  (4), by the *foveated distance*

$$d^{\text{FOV}}(x_1, x_2) = \|\mathcal{F}[z, x_1] - \mathcal{F}[z, x_2]\|_2^2 = \|\mathbf{z}_{x_1}^{\text{FOV}} - \mathbf{z}_{x_2}^{\text{FOV}}\|_2^2, \quad (6)$$

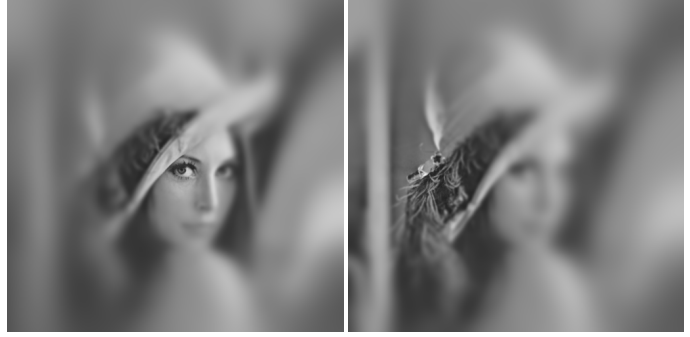
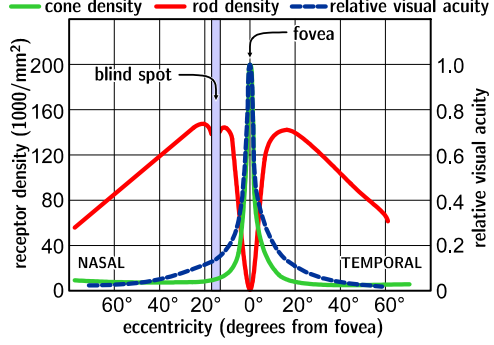


Figure 1. Left: Photo-receptor density and relative visual acuity vs. eccentricity. The visual acuity and the cone density peak at the fovea and decrease rapidly as we move toward the periphery of the retina. Plot based on data from<sup>14</sup> and<sup>15</sup>. Right: Examples of the *Lena* image foveated at two different fixation points.

where  $\mathcal{F}$  is a *foveation operator* that, given an image  $z$  and a fixation point  $x$ , outputs a *foveated patch*  $\mathbf{z}^{\text{FOV}} : U \rightarrow \mathbb{R}$ ,

$$\mathcal{F}[z, x](u) = \mathbf{z}_x^{\text{FOV}}(u), \quad u \in U. \quad (7)$$

$\mathcal{F}$  can be regarded as a spatially variant blurring operator with increasing blur (i.e. decreasing bandwidth) as we leave the center. Strictly speaking,  $\mathbf{z}_x^{\text{FOV}}$  is, compared to  $\mathbf{z}_x$ , progressively blurrier as its argument  $|u|$  grows.

## 4.2 Constrained design of the foveation operator

There are many ways in which  $\mathcal{F}$  can be formalized. To design a foveation operator more suitable for being introduced in NL-means, we impose the following four principal requirements.

### Linearity

$\mathcal{F}$  is a linear operator with respect to the input image.

### Flat-field preservation

$\mathcal{F}$  maps a flat image into flat patches, i.e.

$$\exists \alpha > 0 : \forall c > 0 \text{ if } z(x) = c \quad \forall x \in X, \quad \text{then } \mathcal{F}[z, x](u) = \alpha c \quad \forall u \in U, \quad \forall x \in X. \quad (8)$$

While this property might appear as rather natural, it is perhaps striking that seldom it is actually verified in the inner computations of image processing algorithms. For instance, the multiplication against the windowing kernel  $\mathbf{k}$  in NL-means prevents this property.

### Central acuity

$\mathcal{F}$  is fully sharp at the center of the patch, i.e.

$$\exists \alpha > 0 : \mathcal{F}[z, x](0) = \alpha z(x) \quad \forall x \in X.$$

This property aims at mimicking the peak of the visual acuity at the fovea, illustrated in Figure 1.

### Compatibility

The foveated distance  $d^{\text{FOV}}$  can be used as a straightforward replacement of the usual windowed distance  $d$  in (3), without need of further adjustments of parameters or other elements of the NL-means algorithm. We characterize this desired feature in the ideal case where perfectly identical patches are found, for which we want  $d^{\text{FOV}}$  to yield the same adaptive weights  $w$  given by  $d$ . More precisely, we require that

$$\text{if } \mathbf{y}_{x_1}^{\text{FOV}} = \mathbf{y}_{x_2}^{\text{FOV}}, \text{ then } E \{ d^{\text{FOV}}(x_1, x_2) \} = E \left\{ \left\| \mathbf{z}_{x_1}^{\text{FOV}} - \mathbf{z}_{x_2}^{\text{FOV}} \right\|_2^2 \right\} = 2\sigma^2 \sum_{u \in U} \mathbf{k}(u), \quad (9)$$

where  $\mathbf{y}_x^{\text{FOV}}$  denotes the noise-free foveated patches, defined analogous to (7) as  $\mathcal{F}[y, x](u) = \mathbf{y}_x^{\text{FOV}}(u)$ ,  $u \in U$ . In other words, (9) means that for any pair of identical foveated noise-free patches  $\mathbf{y}_{x_1}^{\text{FOV}} = \mathbf{y}_{x_2}^{\text{FOV}}$ , the expectation of the corresponding foveated distance  $d^{\text{FOV}}(x_1, x_2)$  should be equal to the expectation (5) of the windowed distance (4) for the case when the noise-free patches coincide,  $\mathbf{y}_{x_1} = \mathbf{y}_{x_2}$ .

## 5. CONSTRUCTION OF THE FOVEATION OPERATOR

We construct an operator  $\mathcal{F}$  fulfilling the above four requirements as a linear space-variant Gaussian blur. In particular, the corresponding Gaussian blurring kernels (i.e. the point-spread functions) have standard deviation that increases with the decay of  $\mathbf{k}$ . The resulting blur thus mimics the foveation effects illustrated in Figure 1.

For the sake of generality and mathematical simplicity, we present the design of the kernels using a continuous-domain variable  $x \in \mathbb{R}^2$ ; natural discretization to the integers  $\mathbb{Z}^2$  is to be employed in the implementation.

### 5.1 Design of the Gaussian blurring kernels

Let  $K = \{\mathbf{k}(u), u \in U\} \subseteq \mathbb{R}^+$  be the set of values attained by the windowing kernel  $\mathbf{k}$ . The idea is to design, through suitable dilation and scaling, a family of kernels parametrized by  $\kappa \in K$ , where the kernels have same  $\ell^1$  norm but varying  $\ell^2$  norm determined by  $\kappa$ . For this purpose, let  $p \in (0, 1)$  be a fixed parameter (the meaning of  $p$  and the actual choice of its value are discussed further below) and set

$$\varsigma(\kappa) = \sqrt{\frac{1}{-2 \ln(1-p)} \frac{\kappa_0}{\kappa}}, \quad (10)$$

where  $\kappa_0 = \mathbf{k}(0)$  denotes the value of the windowing kernel  $\mathbf{k}$  at the center of the patch. We define  $\tilde{v}_{\varsigma(\kappa)}$  as the (circular symmetric) bivariate Gaussian kernel (probability density function) with mean zero and diagonal covariance matrix  $\mathbb{I}_{\varsigma^2(\kappa)}$ :

$$\tilde{v}_{\varsigma(\kappa)}(x) = \frac{1}{2\pi\varsigma^2(\kappa)} e^{-\frac{|x|^2}{2\varsigma^2(\kappa)}} = \frac{-\ln(1-p)}{\pi} \frac{\kappa}{\kappa_0} e^{\frac{\ln(1-p)\kappa|x|^2}{\kappa_0}}, \quad x \in \mathbb{R}^2. \quad (11)$$

The parameter  $\varsigma(\kappa)$  determines the bandwidth of the Gaussian kernel and hence the amount of blurring produced. For the special case  $\kappa = \kappa_0$ , the probability inside of a circle of radius 1 around the origin is equal to  $p$  (e.g.,<sup>18</sup> p. 254, or<sup>19</sup> p. 940):

$$\int_{|x| \leq 1} \tilde{v}_{\varsigma(\kappa_0)}(x) dx = p. \quad (12)$$

Basic calculus shows that  $\|\tilde{v}_{\varsigma(\kappa)}\|_2^2 = \frac{1}{4\pi\varsigma^2(\kappa)}$ . Therefore, the desired kernel  $v_{\varsigma(\kappa)}$  is obtained by scaling  $\tilde{v}_{\varsigma(\kappa)}$  as

$$v_{\varsigma(\kappa)} = \sqrt{\kappa} \|\tilde{v}_{\varsigma(\kappa)}\|_2^{-1} \times \tilde{v}_{\varsigma(\kappa)} = \sqrt{\frac{-2 \ln(1-p)}{\pi\kappa_0}} \kappa e^{\frac{\ln(1-p)\kappa|x|^2}{\kappa_0}}.$$

The  $\ell^1$  norm and the squared  $\ell^2$  norm of the kernel  $v_{\varsigma(\kappa)}$  are, respectively,

$$\|v_{\varsigma(\kappa)}\|_1 = \sqrt{\kappa} \|\tilde{v}_{\varsigma(\kappa)}\|_2^{-1} = \sqrt{\frac{4\pi\kappa_0}{-2 \ln(1-p)}}, \quad (13)$$

$$\|v_{\varsigma(\kappa)}\|_2^2 = \kappa. \quad (14)$$

### 5.2 The foveation operator

Finally, we define the foveation operator  $\mathcal{F}$  as

$$\mathcal{F}[z, x_1](u) = \sum_{x \in X} z(x_1 + u + x) v_{\varsigma(\mathbf{k}(u))}(x), \quad \forall u \in U. \quad (15)$$

The above definition implicitly assumes that  $z$  has been preliminary padded (or otherwise extended) outside of its native domain  $X$ ; in our implementation of the algorithm we resort to symmetric padding.

### 5.3 Proof of the four requirements

Let us show that the operator (15) satisfies the four principal requirements set forth in Section 4.2.

**Linearity** is trivially guaranteed by the use of linear Gaussian blurring as defined by (8).

**Flat-field preservation** follows easily by observing that if  $z(x) = c \forall x$ , then

$$\mathcal{F}[z, x_1](u) = c \sum_{x \in X} v_{\zeta(\mathbf{k}(u))}(x) = c \|v_{\zeta(\mathbf{k}(u))}\|_1 \quad \forall u \in U.$$

Since  $\|v_{\zeta(\mathbf{k}(u))}\|_1$  (13) is constant and independent of  $\mathbf{k}(u)$ , equation (8) holds with  $\alpha = \|v_{\zeta(\mathbf{k}(u))}\|_1$ .

**Central acuity** requires that the Gaussian kernel approaches (modulo a fixed scaling factor  $\alpha > 0$ ) a Dirac impulse at the center of the patch. This can be achieved by proper selection of the parameter  $p$  in the definition (10) of  $\zeta(\kappa)$ . In particular, (12) implies that the blurring kernel  $v_{\zeta(\mathbf{k}(0))}$  is (modulo a factor  $\alpha$ ) a point-spread function focused on the central pixel of the patch, provided  $p$  is close to 1. To obtain that  $v_{\zeta(\mathbf{k}(0))}$  is virtually a discrete Dirac impulse, we choose  $p = 1 - e^{-2\pi} = 0.998$ . For this particular value we also have that (13) elegantly simplifies to

$$\|v_{\zeta(\kappa)}\|_1 = \alpha = \sqrt{\kappa_0} = \sqrt{\mathbf{k}(0)} \quad \forall \kappa \in K. \quad (16)$$

**Compatibility** is a consequence of (14) and is proved as follows. If  $\mathcal{F}[y, x_1] = \mathbf{y}_{x_1}^{\text{FOV}} = \mathbf{y}_{x_2}^{\text{FOV}} = \mathcal{F}[y, x_2]$ , then, because of linearity,

$$E \{d^{\text{FOV}}(x_1, x_2)\} = E \|\mathcal{F}[z, x_1] - \mathcal{F}[z, x_2]\|_2^2 = E \|\mathbf{y}_{x_1}^{\text{FOV}} - \mathbf{y}_{x_2}^{\text{FOV}} + \mathcal{F}[\eta, x_1] - \mathcal{F}[\eta, x_2]\|_2^2 = E \|\mathcal{F}[\tilde{\eta}, x_1]\|_2^2,$$

where  $\tilde{\eta}(x) \sim \mathcal{N}(0, 2\sigma^2)$ . We therefore have

$$\begin{aligned} E \{d^{\text{FOV}}(x_1, x_2)\} &= E \left\{ \sum_{u \in U} \mathcal{F}^2[\tilde{\eta}, x_1](u) \right\} = \sum_{u \in U} E \{ \mathcal{F}^2[\tilde{\eta}, x_1](u) \} = \sum_{u \in U} \text{var} \{ \mathcal{F}[\tilde{\eta}, x_1](u) \} = \\ &= \sum_{u \in U} \text{var} \left\{ \sum_{x \in X} \tilde{\eta}(x_1 + u + x) v_{\zeta(\mathbf{k}(u))}(x) \right\} = \sum_{u \in U} \sum_{x \in X} \text{var} \{ \tilde{\eta}(x_1 + u + x) v_{\zeta(\mathbf{k}(u))}(x) \} = \\ &= \sum_{u \in U} \sum_{x \in X} 2\sigma^2 v_{\zeta(\mathbf{k}(u))}^2(x) = \sum_{u \in U} 2\sigma^2 \|v_{\zeta(\mathbf{k}(u))}\|_2^2 = 2\sigma^2 \sum_{u \in U} \mathbf{k}(u), \end{aligned}$$

which thus proves (9).

### 5.4 Illustrations

Let us give some concrete illustration of the construction of the foveation operator. Figure 2 shows the windowing kernel  $\mathbf{k}$  used in NL-means.<sup>8</sup> To this window corresponds a codomain  $K = \{\mathbf{k}(u), u \in U\} = \{\kappa_0, \kappa_1, \kappa_2, \kappa_3, \kappa_4\}$ , whose five unique values are reported in the figure. Equation (10) yields  $\zeta(\kappa_0)=0.282$ ,  $\zeta(\kappa_1)=0.434$ ,  $\zeta(\kappa_2)=0.611$ ,  $\zeta(\kappa_3)=0.861$ , and  $\zeta(\kappa_4)=1.360$ . The five blurring kernels  $v_{\zeta(\kappa_i)}$ ,  $i = 0, \dots, 4$ , obtained from these values are shown in Figure 3 along with their frequency response. Each of these kernels is based on a discrete Gaussian kernel  $\tilde{v}_{\zeta(\kappa)}$  of size  $(2 \lceil 3\zeta(\kappa) \rceil + 1) \times (2 \lceil 3\zeta(\kappa) \rceil + 1)$ ,  $\lceil \cdot \rceil$  being the ceiling function. Observe that  $v_{\zeta(\kappa_0)}$  is practically a scaled Dirac impulse, confirming central acuity. The varying bandwidth, decreasing with  $\kappa$ , is clearly visualized. Also note that the frequency responses of the five blurring kernels all attain their maximum at the origin and that the value of this maximum is equal to  $\sqrt{\kappa_0} = 0.196$ , as it is implied by (16). The squared  $\ell^2$ -norm values  $\|v_{\zeta(\kappa)}\|_2^2$  for these five kernels are 0.0379, 0.0231, 0.0090, 0.0041, 0.0017, respectively, which are nearly equal to the corresponding values of  $\kappa \in K$ , as by (14). The minor differences are due to the discretization of the Gaussian kernel  $\tilde{v}_{\zeta(\kappa)}$  used in our implementation, while (11) assumes continuous domain variables.

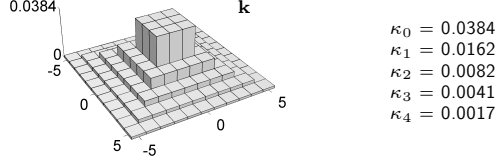


Figure 2. The windowing kernel  $\mathbf{k}$  of size  $11 \times 11$  used for computing the similarity weights in the NL-means algorithm<sup>4,8</sup> and its five unique values.

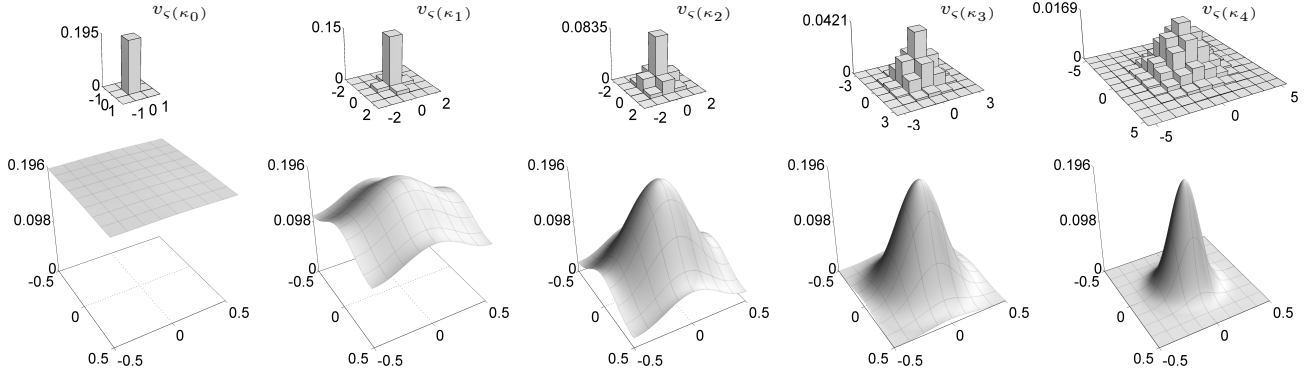


Figure 3. The five blurring kernels  $v_\zeta(\kappa_i)$ ,  $i = 0, \dots, 4$ , corresponding to the five unique values  $\kappa \in K$  of the window  $\mathbf{k}$  (shown in Figure 2) (top row) and their respective frequency response (bottom row).



Figure 4. Two illustrative examples of how patches are transformed when computing the windowed distance  $d$  (4) and the foveated distance  $d^{\text{FOV}}$  (6):  $\mathbf{z}_x$  denotes the patch in the noisy image, while  $\mathbf{z}_x \sqrt{\mathbf{k}}$  and  $\mathbf{z}_x^{\text{FOV}}$  denote, respectively, the patches used for computing the quadratic windowed distance and foveated distance, for the same image and fixation point. The division by  $\sqrt{\kappa_0}$  ensures that these patches are displayed on the same intensity range of  $\mathbf{z}_x$ . In both examples the noise standard deviation in  $\mathbf{z}_x$  is  $\sigma = 20$  and  $\mathbf{y}_x$ , the noise-free counterpart of  $\mathbf{z}_x$ , is also shown.

Figure 4 compares patches used to compute the foveated distance  $d^{\text{FOV}}$  (6) against those used for the windowed distance  $d$  (4), which are obtained by multiplication against the square root of the windowing kernel  $\mathbf{k}$ , as shown in (4). Observe how the foveated patches exhibit effective noise attenuation (due to the design of  $\mathcal{F}$ ), while at the same time do not alter the main structures in the patch. The superior denoising performance, presented in the next section, suggests that foveated patches allow for a more precise assessment of nonlocal self-similarity in noisy images.

## 6. DENOISING EXPERIMENTS

We carried out denoising experiments for five standard  $512 \times 512$  grayscale test images, namely *Lena*, *Barbara*, *Boats*, *Hill*, and *Man*. The noise-free images are shown in Figure 5. In the experiments, the images are corrupted by additive white Gaussian noise according to the observation model (1), and denoised by both the proposed Foveated NL-means<sup>†</sup> and the standard NL-means algorithm.<sup>8</sup> The implementations of the two algorithms share the same parameters (patch size  $11 \times 11$ , search window  $21 \times 21$ ,  $h = \sigma$ ), obtained by tuning NL-means on the test images, and differ only by the employed patch distance: windowed distance  $d$  (4) for NL-means, and foveated distance  $d^{\text{FOV}}$  (6) for Foveated NL-means. Both  $d$  and  $d^{\text{FOV}}$  are defined by the same kernel  $\mathbf{k}$ , shown in Figure 2, and the foveation operator  $\mathcal{F}$  is constructed setting  $p = 1 - e^{-2\pi} = 0.998$ , as explained in Section 5.3. In this way, we enable a fair and direct comparison of the two algorithms.

<sup>†</sup>A Matlab implementation of the proposed Foveated NL-means is provided online at <http://www.cs.tut.fi/~foi>



Figure 5. The five  $512 \times 512$  grayscale test images  $y$  used in the denoising experiments.

$\sigma$	<i>Lena</i>		<i>Barbara</i>		<i>Boats</i>		<i>Hill</i>		<i>Man</i>		Average		
	NLM	FNLM	NLM	FNLM	NLM	FNLM	NLM	FNLM	NLM	FNLM	WIN	FNLM	$\Delta_{\text{FNLM-NLM}}$
10	34.78	35.03	33.71	33.40	32.41	32.72	32.11	32.70	32.24	32.84	33.05	33.34	0.287
20	31.28	32.42	29.78	30.42	28.92	29.98	28.82	29.96	28.84	29.95	29.53	30.55	1.021
30	29.23	30.53	27.17	28.11	26.81	28.13	26.97	28.23	26.99	28.17	27.43	28.63	1.199
40	27.79	29.08	25.42	26.39	25.40	26.76	25.83	26.96	25.79	26.90	26.04	27.22	1.174
50	26.67	27.94	24.18	25.12	24.40	25.68	25.04	26.03	24.91	25.93	25.04	26.14	1.101
60	25.76	27.01	23.29	24.20	23.65	24.80	24.45	25.30	24.21	25.16	24.27	25.29	1.022
70	25.01	26.21	22.62	23.49	23.06	24.09	23.96	24.72	23.65	24.52	23.66	24.60	0.945
80	24.37	25.52	22.09	22.91	22.58	23.49	23.56	24.21	23.18	23.98	23.16	24.02	0.868
90	23.84	24.90	21.66	22.43	22.18	22.99	23.20	23.77	22.77	23.50	22.73	23.52	0.791
100	23.37	24.35	21.31	22.01	21.83	22.55	22.88	23.38	22.40	23.08	22.36	23.07	0.714

Table 1. PSNR (db) results of the standard NL-means (which uses the windowed distance  $d$ ), denoted by “NLM” and by the proposed Foveated NL-means (which uses the foveated patch distance  $d^{\text{FOV}}$ ), denoted by “FNLM”.

$\sigma$	<i>Lena</i>		<i>Barbara</i>		<i>Boats</i>		<i>Hill</i>		<i>Man</i>		Average		
	NLM	FNLM	NLM	FNLM	NLM	FNLM	NLM	FNLM	NLM	FNLM	NLM	FNLM	$\Delta_{\text{FNLM-NLM}}$
10	0.899	0.907	0.928	0.927	0.853	0.871	0.837	0.863	0.867	0.890	0.877	0.891	0.0147
20	0.838	0.859	0.855	0.871	0.762	0.794	0.732	0.769	0.770	0.803	0.791	0.819	0.0281
30	0.786	0.811	0.781	0.806	0.693	0.731	0.659	0.700	0.701	0.737	0.724	0.757	0.0328
40	0.735	0.762	0.712	0.742	0.637	0.676	0.604	0.643	0.646	0.680	0.667	0.700	0.0337
50	0.686	0.713	0.649	0.682	0.588	0.626	0.559	0.594	0.597	0.630	0.616	0.649	0.0332
60	0.637	0.666	0.593	0.627	0.545	0.580	0.521	0.552	0.553	0.584	0.570	0.602	0.0323
70	0.591	0.621	0.544	0.579	0.505	0.538	0.486	0.514	0.513	0.543	0.528	0.559	0.0310
80	0.548	0.577	0.502	0.536	0.470	0.500	0.454	0.479	0.476	0.505	0.490	0.519	0.0294
90	0.508	0.536	0.464	0.497	0.437	0.465	0.425	0.448	0.443	0.470	0.455	0.483	0.0276
100	0.471	0.498	0.431	0.462	0.407	0.433	0.398	0.418	0.413	0.438	0.424	0.450	0.0257

Table 2. SSIM<sup>20</sup> results of the standard NL-means (which uses the windowed distance  $d$ ), denoted by “NLM” and by the proposed Foveated NL-means (which uses the foveated patch distance  $d^{\text{FOV}}$ ), denoted by “FNLM”.

PSNR and SSIM<sup>20</sup> results for  $\sigma = [10, 20, \dots, 90, 100]$  are collected in Table 1 and Table 2, where we also report values averaged over the five images and the gain due to foveation. The proposed Foveated NL-means substantially outperforms the standard NL-means in all cases, with the only exception of *Barbara*  $\sigma = 10$ , where we note a minor loss in PSNR but virtually identical SSIM score. The trend of these numerical results is illustrated in Figure 6, where we plot the PSNR and SSIM results of the Foveated NL-means as function of the corresponding results of the standard NL-means. The tables and the plots attest a significant numerical improvement due to foveation; in particular, for  $10 < \sigma < 90$ , the typical gain is about 1 dB PSNR and 0.03 SSIM units. This also corresponds to a considerable visual difference between the outputs by the NL-means algorithm and by the proposed Foveated NL-means, as can be seen from the various fragments shown in Figure 7 and Figure 8. The estimates produced by the algorithm based on the foveated distance are characterized by a better contrast, sharper appearance, and increased detail preservation. At the same time, the amount of noise suppression in the smooth regions is the same as in the standard NL-means, due to the compatibility constraint.



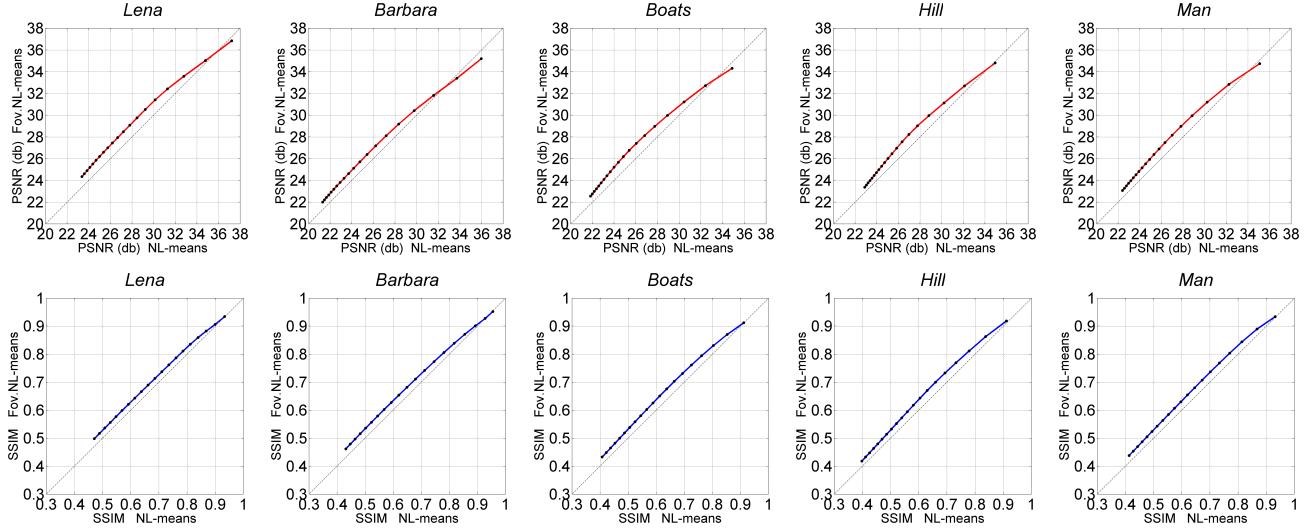


Figure 6. PSNR (db) and SSIM<sup>20</sup> results of the standard NL-means algorithm<sup>4,8</sup> (horizontal axes) vs. the proposed Foveated NL-means (vertical axes). The plots show results obtained for  $\sigma = [5, 10, \dots, 95, 100]$ , where lower PSNR or SSIM values correspond to higher  $\sigma$  values.

## 7. DISCUSSION AND CONCLUSIONS

The superior sharpness and contrast achieved by the Foveated NL-means come perhaps unexpected, because the foveated self-similarity is assessed on blurred data. This paradox is however only apparent, because in the weighted averaging (2) what matters is that  $w(x_1, x_2)$  is small when  $|y(x_1) - y(x_2)|$  is large, and vice versa. The quality improvement achieved using foveation suggests that the similarity between the noise-free central pixels correlates better with the foveated patch similarity than with the windowed similarity. This should not surprise, since foveation suppresses high-frequency components at the periphery of the patch, and these components are typically less correlated with those at center of the patch than the low-frequency components are. While a direct parallel may not be obvious, it is nevertheless interesting to observe that also in the HVS foveation plays indeed a role in increasing the dynamic range and the signal-to-noise ratio at low-light conditions, at the expense of blurring the peripheral perception. In particular, we conjecture that some form of foveated self-similarity may be relevant in trans-saccadic integration (e.g.,<sup>21</sup>), a process that vaguely reminds of the Foveated NL-means.

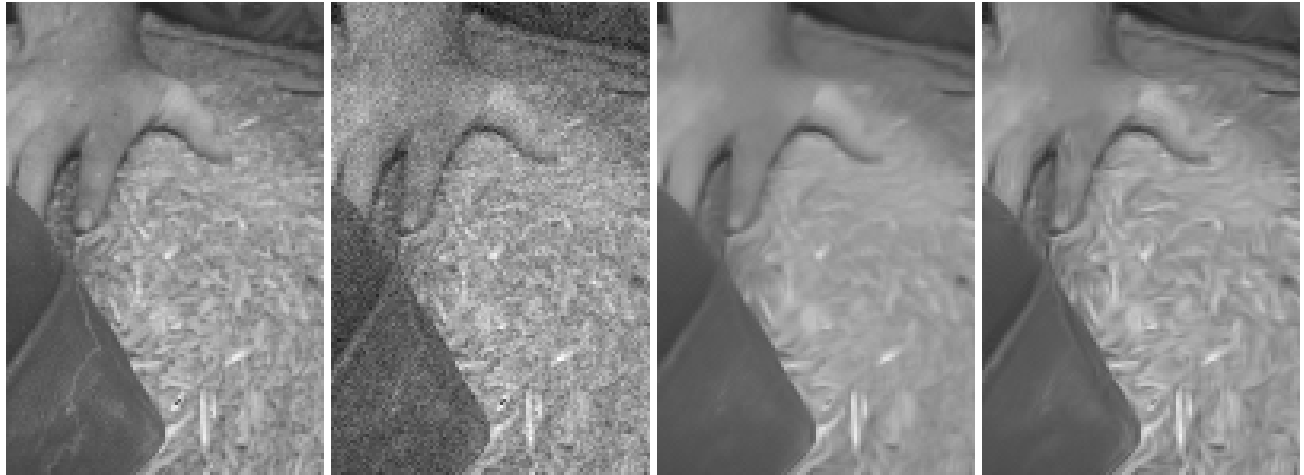
The compatibility requirement ensures that the Foveated NL-means algorithm performs comparably to the standard NL-means whenever there are plenty of mutually similar patches, such as in areas with smooth or periodic content. This allowed us to avoid any customized tuning of the foveated algorithm, and just utilize the parameters that work well for the NL-means.

In terms of computational complexity, the foveated distance can be introduced at negligible computational cost, because the foveation operator (15) requires only the preliminary computation of the  $|K|$  convolutions of  $z$  against the blurring kernels  $v_{\zeta(\kappa_i)}$ ,  $\kappa_i \in K$ . Note that the cardinality of  $K$  is often much smaller than the number  $|U|$  of pixels in the patch (e.g., in our experiments  $|K|=5$  and  $|U|=121$ ).

Finally, we wish to remark that, in spite of the substantial improvement achieved by introducing foveation in NL-means, the performance of these algorithms is still inferior to that of more sophisticated nonlocal filters, such as, e.g., BM3D<sup>5</sup> or SAFIR.<sup>6</sup> As a figure of merit, BM3D achieves a PSNR of 31.26 dB when denoising *Lena* corrupted by noise with  $\sigma = 30$ , versus 29.23 dB of NL-means and 30.53 dB of Foveated NL-means. However, our contribution is not to be intended as the development of yet another denoising algorithm, but rather as the exploration of a new form of nonlocal self-similarity.

### Summary and perspective

Exploiting an essential principle from human vision, namely foveation, we proposed a direct modification of the way self-similarity is assessed within nonlocal imaging filters, by replacing the conventional windowed self-similarity with a foveated self-similarity. For the specific case of the well known NL-means algorithm, we provided

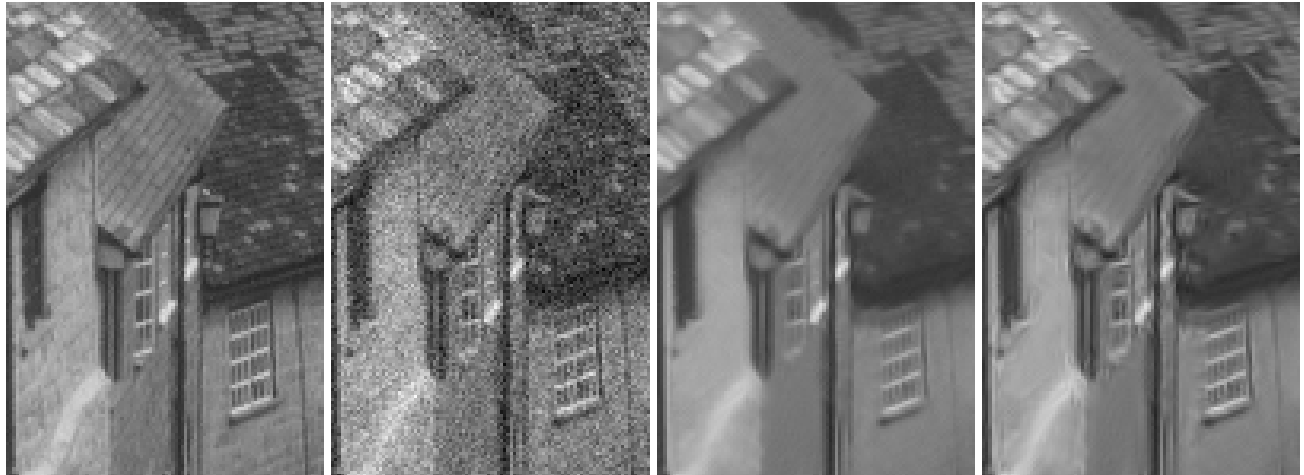


*Man*, original

noisy  $\sigma = 10$  (28.14 0.680)

NL-means (32.24 0.867)

Fov. NL-means (32.84 0.890)

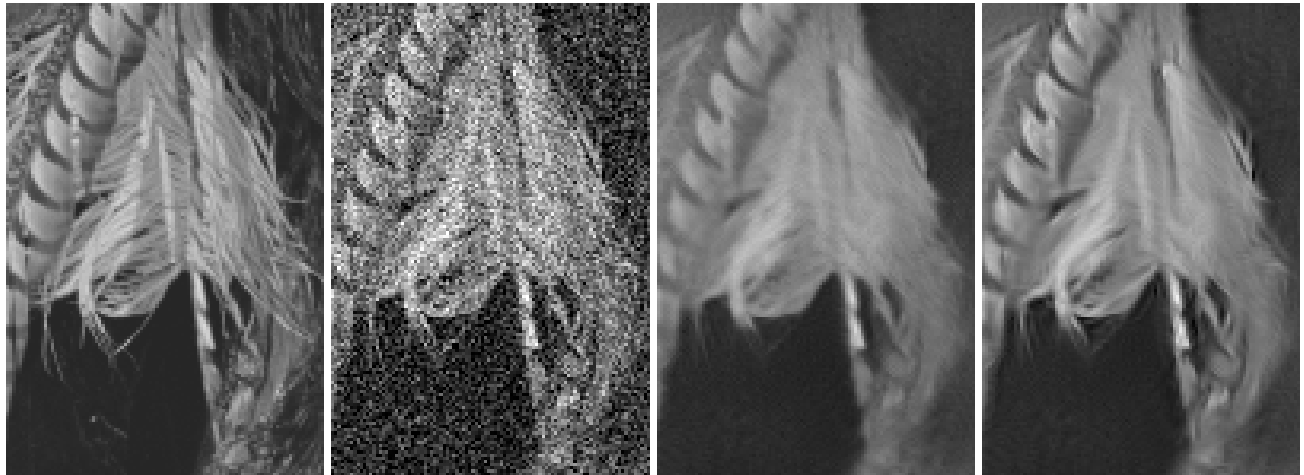


*Hill*, original

noisy  $\sigma = 15$  (24.61 0.531)

NL-means (30.22 0.779)

Fov. NL-means (31.13 0.812)



*Man*, original

noisy  $\sigma = 30$  (18.59 0.271)

NL-means (26.99 0.701)

Fov. NL-means (28.17 0.737)

Figure 7. Comparison between outputs by the NL-means algorithm and by the proposed Foveated NL-means. The shown fragments have size  $100 \times 150$  pixel. The numbers between parentheses are the PSNR (db) and SSIM scores computed for the entire image.

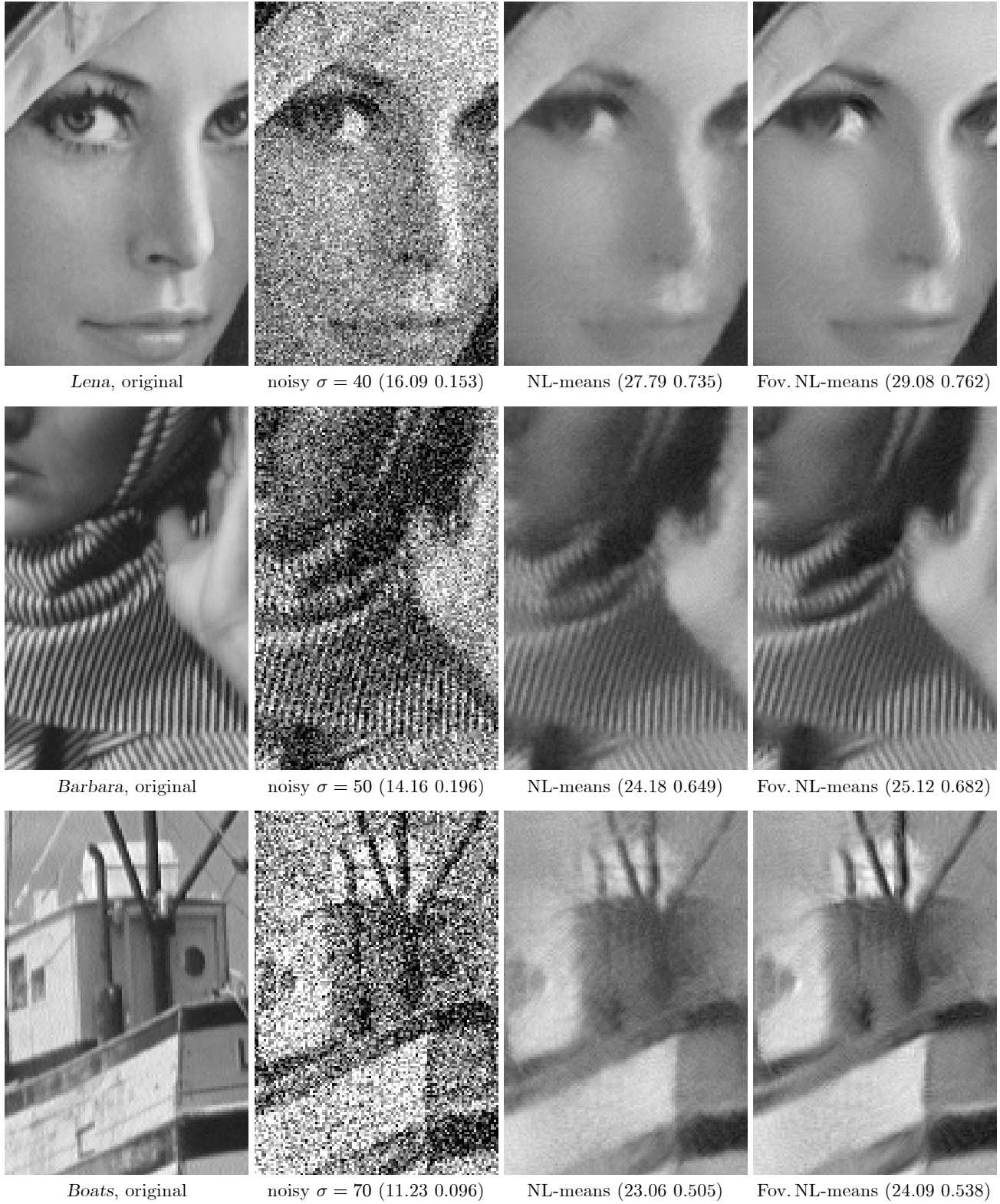


Figure 8. Comparison between outputs by the NL-means algorithm and by the proposed Foveated NL-means. The shown fragments have size  $100 \times 150$  pixel. The numbers between parentheses are the PSNR (db) and SSIM scores computed for the entire image.

an explicit construction of a foveation operator based on the given windowing kernel. At the cost of a negligible computational overhead, the resulting Foveated NL-means algorithm outperforms the conventional NL-means with PSNR gains often in excess of 1 dB, particularly at moderate and high noise levels. Most importantly, there is substantial improvement in sharpness, contrast, and overall visual quality, which is also confirmed by a consistently higher SSIM score.

Our ongoing research aims at analyzing the extent of foveated self-similarity in the context of natural image statistics and at exploiting this particular form of self-similarity within multiscale nonlocal transform-domain filters.

## REFERENCES

1. A. Buades, B. Coll, and J. Morel, “A review of image denoising algorithms, with a new one,” *Multiscale Modeling Simulation* **4**(2), p. 490, 2005.
2. M. F. Barnsley, *Fractals everywhere*, Academic Press, 2nd ed., 1993.
3. J. S. De Bonet, “Noise reduction through detection of signal redundancy.” In Rethinking artificial intelligence. MIT AI Lab, 1997.
4. A. Buades, B. Coll, and J. Morel, “A non-local algorithm for image denoising,” in *Proc. 2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR’05)*, pp. 60–65, 2005.
5. K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3D transform-domain collaborative filtering,” *IEEE Trans. Image Process.* **16**, August 2007.
6. C. Kervrann and J. Boulanger, “Optimal spatial adaptation for patch-based image denoising,” *IEEE Trans. Image Process.* **15**, pp. 2866–2878, October 2006.
7. V. Katkovnik, A. Foi, K. Egiazarian, and J. Astola, “From local kernel to nonlocal multiple-model image denoising,” *Int. J. Comput. Vision* **86**(1), pp. 1–32, 2010.
8. J. V. M. Herrera and A. Buades, “Non-local means filter, Matlab code.” Matlab Central File Exchange, 2008. <http://www.mathworks.com/matlabcentral/fileexchange/13176-non-local-means-filter>.
9. J. Salmon, “On two parameters for denoising with non-local means,” *IEEE Signal Process. Letters* **17**, pp. 269–272, March 2010.
10. S. Grewnig, S. Zimmer, and J. Weickert, “Rotationally invariant similarity measures for nonlocal image denoising,” *Journal of Visual Communication and Image Representation* **22**, pp. 117–130, February 2011.
11. T. Thaipanich, B. Oh, P.-H. Wu, D. Xu, and C.-C. Kuo, “Improved image denoising with adaptive nonlocal means (ANL-means) algorithm,” *IEEE Trans. Consum. Electr.* **56**, pp. 2623–2630, November 2010.
12. Y. Lou, P. Favaro, S. Soatto, and A. Bertozzi, “Nonlocal similarity image filtering,” in *Proc. 15th Int. Conf. Image Analysis and Processing, ICIAP ’09*, pp. 62–71, (Berlin, Heidelberg), 2009.
13. D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vision* **60**, pp. 91–110, November 2004.
14. C. Curcio, K. Sloan, R. Kalina, and A. Hendrickson, “Human photoreceptor topography,” *J. Comparative Neurology* **292**, pp. 497–523, 1990.
15. T. Wertheim, “Über die indirekte sehschärfe,” *Zeitschrift für Psychologie und Physiologie der Sinnesorgane* **7**, pp. 172–187, 1894.
16. B. A. Wandell, *Foundations of Vision*, Sinauer Associates, Inc., 1995.
17. M. P. Eckstein, “Visual search: A retrospective,” *Journal of Vision* **11**(5), 2011.
18. S. Kotz, N. Balakrishnan, and N. Johnson, *Continuous Multivariate Distributions*, vol. 1 (Models and Applications), Wiley, 2nd ed., 2000.
19. M. Abramowitz and I. Stegun, eds., *Handbook of Mathematical Functions*, National Bureau of Standards, 10th ed., 1972.
20. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error measurement to structural similarity,” *IEEE Trans. Image Process.* **13**, April 2004.
21. M. Demeyer, P. De Graef, J. Wagemans, and K. Verfaillie, “Transsaccadic identification of highly similar artificial shapes,” *Journal of Vision* **9**(4), 2009.