Examining the Role of Scale in the Context of the Non-Local-Means Filter

Mehran Ebrahimi and Edward R. Vrscay

Department of Applied Mathematics
Faculty of Mathematics
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
m2ebrahi@uwaterloo.ca, ervrscay@uwaterloo.ca

Abstract. We consider the role of *scale* in the context of the recently-developed non-local-means (NL-means) filter. A new example-based variant of the NL-means is introduced and results based on same-scale and cross-scale counterparts will be compared for a set of images. We consider the cases in which neighborhoods are taken from the observed image itself as well as from other irrelevant images, varying the smoothness parameter as well. Our experiments indicate that using cross-scale (i.e., downsampled) neighborhoods in the NL-means filter yields results that are quite comparable to those obtained by using neighborhoods at the same-scale.

1 Introduction

Natural images exhibit various *regularity properties* such as the degree of smoothness, total variation, decay as well as sparsity of the transform domain coefficients, which have been exploited in various image processing tasks, including compression [13,14]. Many of these regularity properties are *local*, in the sense that the greyscale value at a pixel is correlated with values in its neighborhood.

Self-similarity is an example of a *nonlocal* regularity property, in the sense that local neighborhoods of an image can be highly correlated (i.e., affinely similar) to other neighborhoods throughout the image. Self-similarity is the basis of fractal image coding [3,12,8] which, historically, concentrated on image compression. Fractal coding has also been shown to be effective in other image processing tasks, such as denoising [10,1]. More recently, the translational self-similarity of images has been exploited for the purpose of denoising [4,5].

The essence of the fractal transform is to approximate smaller *range* subblocks of an image with modified copies of larger *domain* subblocks [3,12,8]. The underlying basis for the success of these methods is that of *approximability*: "How well can an image be approximated using self-similarity constraints?" Equivalently, this may be viewed in terms of *regularity*: "How regular or invariant is an image under the imposition of self-similarity constraints?" Some recent investigations [1,2] have shown that images generally possess a great deal of local (affine) self-similarity.

A. Campilho and M. Kamel (Eds.): ICIAR 2008, LNCS 5112, pp. 170–181, 2008.

The non-local-means (NL-means) filter [4,5] performs denoising using the *same-scale* translational redundancy of information in an image. This is done by a weighted averaging process, where the weights are determined by the similarities of neighborhoods, or "patches." The efficacy of the NL-means as a denoising method is again an issue of *approximability and regularity*. Indeed, fractal-based methods and the NL-means filter may be considered as special cases of a more general model of affine image self-similarity [2].

In this paper, we examine the self-similarity properties of an image by formulating a simple *cross-scale* variant of the NL-means and comparing its effectiveness to that of the standard *same-scale* NL-means method. The former will use neighborhoods/patches that are taken from the same image, or even from another image, *at a different scale*. Such a role of scale for the NL-means filter has not yet been investigated.

In Section 2, we introduce the background material required for this paper. We also introduce the cross-scale variant of the NL-means which will be used in our experiments. A rather simple statistical examination of patch-similarity at two-scales for various images is given in Section 3. In Sections 4 and 5, we examine the role of scale for a couple of images in the context of the NL-means filter. In Section 4, the patches are taken from the observed image, i.e., "self-examples." whereas in Section 5, they are taken from another image. Some concluding remarks are presented in Section 6.

2 The NL-Means Filter and Some New Variants

2.1 NL-Means Image Denoising Filter

Consider the following image denoising problem [4,5],

$$\mathbf{u} = \mathbf{f} + \mathbf{n},$$

where $\mathbf{u}$ is is a given observation, $\mathbf{n} \in l^2(\Omega)$ is additive white, independent Gaussian noise with zero-mean and variance σ^2, and $\mathbf{f} \in l^2(\Omega)$ is the image to be recovered, where

$$\Omega = [1, \ldots, m] \times [1, \ldots, n].$$

For any $x \in \Omega$ define the approximation of $\mathbf{f}$ denoted by $\widetilde{\mathbf{f}}$ as

$$\widetilde{\mathbf{f}}(x) = \frac{1}{C(x)} \sum_{y \in \Omega} w(x, y)\mathbf{u}(y), \text{ such that} \tag{1}$$

$$w(x, y) = \exp\Big(- \frac{\| \mathbf{u}(\mathcal{N}^d\{x\}) - \mathbf{u}(\mathcal{N}^d\{y\}) \|^2_{2,a}}{h^2} \Big), \text{ and} \tag{2}$$

$$C(x) = \sum_{y \in \Omega} w(x, y), \tag{3}$$

where the expressions $\mathcal{N}^d\{\ldots\}$ and $\| \, . \, \|^2_{2,a}$ are defined in the following way.

Neighborhoods: For any point in the domain of observation $(i,j) \in \Omega$, define

$$\mathcal{N}^d\big\{(i,j)\big\} = \Big\{(i+i_1, j+j_1) \mid (i_1,j_1) \in \mathbb{Z}^2, \max\{|i_1|,|j_1|\} \leq d\Big\}. \tag{4}$$

Gaussian-weighted-semi-norm: For a discrete image $\mathbf{I} \in l^2\big([1,\ldots,2d+1] \times [1,\ldots,2d+1]\big)$ define $\| \, . \, \|_{2,a}^2$ as

$$\| \, \mathbf{I} \, \|_{2,a}^2 = \sum_{-d \leq s_1 \leq d, -d \leq s_2 \leq d} G_a(s_1,s_2) |\mathbf{I}(s_1+d+1, s_2+d+1)|^2$$

in which G_a is a two-dimensional Gaussian kernel of standard deviation a, centered at $(0,0)$, and of the same dimension as $\mathbf{I}$.

The idea of the NL-means algorithm is that given a discrete noisy image $\mathbf{u}$, the estimated denoised value $\widetilde{\mathbf{f}}(x)$ is computed as a weighted average of all pixels in the image, $\mathbf{u}(y)$, where the weights $w(x,y)$ depend on the similarity of neighborhoods of the pixels x and y, and w is a decreasing function of the weighted Euclidean distance of the neighborhoods. The parameter h acts as a degree of filtering and controls decay of the exponential function and therefore the decay of the weights as a function of the Euclidean distances.

2.2 Restricting the Search-Window in the NL-Means Filter

It turns out that the denoising algorithm above is computationally intensive. A possibility to overcome the computational complexity introduced in [4,5] is to restrict $y \in \Omega \cap \mathcal{N}^r\{x\}$ i.e., in a neighborhood of x than the whole field of Ω in the corresponding equations above.

2.3 An Example-Based Variant of the NL-Means Filter

In this section, we introduce an extension of the NL-means method which employs neighborhoods taken from an *example image*, $\mathbf{v}$, similar to what we introduced in [6].

Example image: The notion of *example image* is denoted by $\mathbf{v} \in l^2(\Phi)$, where Φ is the $k \times l$ pixel lattice defined by

$$\Phi = [1,\ldots,k] \times [1,\ldots,l].$$

Algorithm 1: An Example-Based Variant of the NL-Means Filter. The approximation of $\mathbf{f}$, denoted by $\widetilde{\mathbf{f}}$, given the observation $\mathbf{u}$, using *example image* $\mathbf{v}$, is computed in the following way. For any $x \in \Omega$,

$$\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{v},h,d,a)}(x) = \frac{1}{C(x)} \sum_{y \in \Phi} w(x,y)\mathbf{v}(y), \tag{5}$$

where

$$w(x,y) = \exp\Big(-\frac{\| \, \mathbf{u}(\mathcal{N}^d\{x\}) - \mathbf{v}(\mathcal{N}^d\{y\}) \, \|_{2,a}^2}{h^2}\Big) \tag{6}$$

and

$$C(x) = \sum_{y \in \Phi} w(x, y). \tag{7}$$

It is clear that NL-means image denoising is equivalent to Algorithm 1 in the case that $\mathbf{v} = \mathbf{u}$, so that the approximation is $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{u},h,d,a)}(x)$.

In a manner similar to the restricted search-window algorithm, we may speed up this algorithm by restricting y. Due to the fact that the size of the example image $\mathbf{v}$ is $k \times l$ and not necessarily identical to the size of u, $m \times n$, we consider a corresponding pixel to x in $\mathbf{v}$ called x_{map} defined as $x_{map} = ([k/mx_1], [l/nx_2])$, in which $x = (x1, x2)$, and $[\ldots]$ is the floor function. Therefore, we define $y \in \Phi \cap \mathcal{N}^r\{x_{map}\}$ i.e., in a neighborhood of x_{map} rather than the whole field of Φ in the corresponding equations above.

2.4 Projection Operator to a Coarser Scale Image

We now consider the approximation of images yielded by the NL-means filter at varying scales. To do so, we need a precise definition of a projection to a coarser scale, to be denoted as $\mathcal{P}_z$. Let us define $\mathcal{P}_z : l^2(\Omega) \to l^2(\Omega)$ to be $\mathcal{P}_z = \mathcal{S} \circ \mathcal{B}$, where $\mathcal{B} : l^2(\Omega) \to l^2(\Omega)$ is the local averaging operator of length z: For any $(p, q) \in \Omega$,

$$(\mathcal{B}\mathbf{f})(p, q) = \frac{1}{z^2} \sum_{0 \le p_1 < z, 0 \le q_1 < z} \mathbf{f}(p + p_1, q + q_1). \tag{8}$$

(Note that boundary conditions on $\mathbf{f}$ may be required so that $\mathcal{B}$ is well-defined). Also, the downsampling operator $\mathcal{S} : l^2(\Omega) \to l^2(\Omega)$ is defined for any image $\mathbf{I} \in l^2(\Omega)$ such that for any $(i, j) \in \Omega$,

$$(\mathcal{S}\mathbf{I})(i, j) = \mathbf{I}\Big((i-1)z + 1, (j-1)z + 1\Big). \tag{9}$$

In the experiments presented throughout the rest of the paper, we shall be using $\mathcal{P}_2$, i.e., the case that $z = 2$.

3 Distribution of Neighborhood Distances at Various Scales

Some recent investigations [1,2] in the context of fractal-image coding have shown that images generally possess a great deal of local (affine) self-similarity: Given a subimage $\mathbf{u}|_{R_i}$ there are often a good number of domain blocks D_j whose subimages $\mathbf{u}|_{D_j}$ approximate it roughly as well as the "best" domain block. This feature, which never seems to have been quantified previously, accounts for the rather small degradations that are experienced when the size of the *domain pools*– the domain blocks D_j to be examined – is decreased.

In a parallel fashion, the NL-means algorithm described in the previous section relies not only on the intensity values of pixels but also on the neighborhood (or

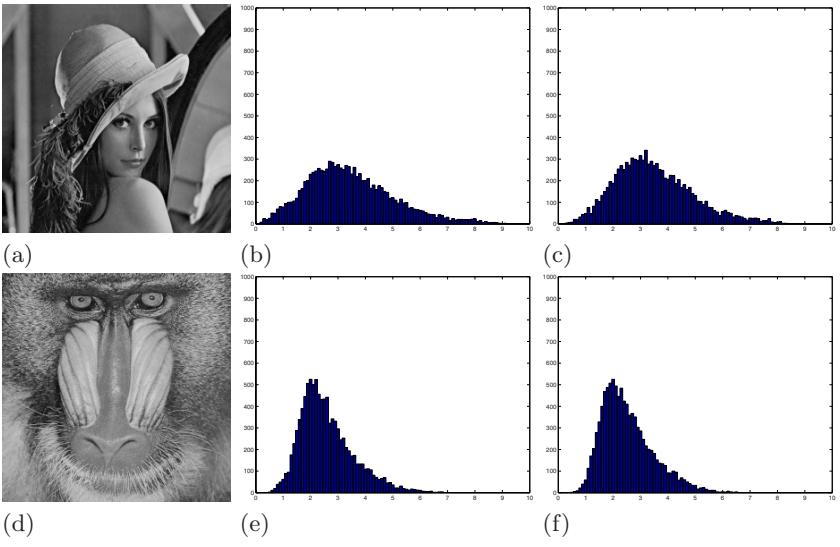

Fig. 1. (a) and (d) are the 256×256 test images. (b) and (e) are plots of the distributions of $\| \mathbf{u}(\mathcal{N}^6\{x\}) - \mathbf{u}(\mathcal{N}^6\{y\}) \|_2$ corresponding to the images in (a) and (d), respectively. The plots in (c) and (f) are the distributions of $\| \mathbf{u}(\mathcal{N}^6\{x\}) - (\mathcal{P}_2\mathbf{u})(\mathcal{N}^6\{y\}) \|_2$ for the images in (a) and (d), respectively.

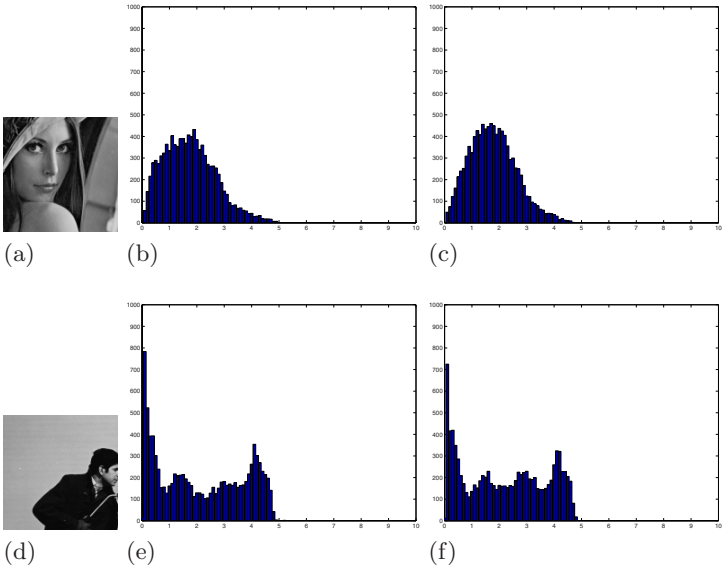

Fig. 2. (a) and (d) are the 128×128 test images. (b) and (e) are plots of the distributions of $\| \mathbf{u}(\mathcal{N}^3\{x\}) - \mathbf{u}(\mathcal{N}^3\{y\}) \|_2$ corresponding to the images in (a) and (d), respectively. The plots in (c) and (f) are the distributions of $\| \mathbf{u}(\mathcal{N}^3\{x\}) - (\mathcal{P}_2\mathbf{u})(\mathcal{N}^3\{y\}) \|_2$ for the images in (a) and (d), respectively.

patch) similarities existing in an image. Below, we present experiments to evaluate same-scale and cross-scale similarities of an image simply under the L^2 metric.

In Figures 1(a) and 1(d) are shown the test images used here: 256×256 pixel, 8 bits per pixel, *Mandrill* and *Lena* images for which the intensity values are rescaled to the interval $[0, 1]$. We have chosen 10,000 randomly selected pairs of 13 by 13 pixel-blocks from each image and have plotted the distribution of the L^2 distances of these blocks on the left hand-side. More precisely, Figs. 1(b) and 1(e) show plots of the distributions of $\| \mathbf{u}(\mathcal{N}^6\{x\}) - \mathbf{u}(\mathcal{N}^6\{y\}) \|_2$ for the images in Figs. 1(a) and 1(d), respectively. Here, we have performed the experiment for 10,000 uniformly-randomly selected pairs of (x,y). The plots in Fig. 1(c) and 1(f) are the distributions of $\| \mathbf{u}(\mathcal{N}^6\{x\}) - (\mathcal{P}_2\mathbf{u})(\mathcal{N}^6\{y\}) \|_2$ for the images in Figs. 1(a) and 1(d), respectively. As a result, the histograms of Figs. 1(b) and 1(e) correspond to *same-scale* patch comparisons, while those in Figs. 1(c) and 1(f) correspond to *cross-scale* patch comparisons.

Similarly, Figs. 2(a) and 2(d) show 8 bit-per-pixel 128×128 pixel test images, *Lena* and *Cameraman* for which the intensity values are rescaled to the interval $[0, 1]$. From these images, we have chosen 10,000 randomly-selected pairs of 7×7 pixel-blocks. Figs. 2(b) and 2(e) show plots of the distributions of $\| \mathbf{u}(\mathcal{N}^6\{x\}) - \mathbf{u}(\mathcal{N}^6\{y\}) \|_2$ corresponding to the images in 2(a) and 2(d), respectively. Similarly, the plots in Figs. 2(c) and 2(f) are the distributions of $\| \mathbf{u}(\mathcal{N}^6\{x\}) - (\mathcal{P}_2\mathbf{u})(\mathcal{N}^6\{y\}) \|_2$ corresponding to the images in 2(a) and 2(d), respectively.

For each test image, the similarity between the histograms yielded by the two experiments, i.e., same-scale and cross-scale, is striking. This indicates that the self-similarity properties of these images at both same-scale and cross-scale are remarkably similar. This is a hint that an appropriately defined cross-scale version of the NL-means filter should behave in a similar fashion to the traditional same-scale NL-means filter. In the next section we compare these two methods.

4 Same-Scale vs. Cross-Scale Approximations

In this section, we compare the results of two approximations: the traditional, same-scale NL-means method which yields the approximation $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{u},h,d,a)}(x)$ and its cross-scale NL-means counterpart, with approximation $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathcal{P}_2(\mathbf{u}),h,d,a)}(x)$. Given an observed image $\mathbf{u}$ and using Algorithm 1, we choose the example image $\mathbf{v} = \mathbf{u}$ to obtain results of the NL-means filter. Choosing the example image $\mathbf{v} = \mathcal{P}_2(\mathbf{u})$, translates to cross-scale neighborhood-comparison and averaging. Furthermore, we examine how these two schemes differ from each as the smoothness parameter h is varied. In the experiments presented below, we have used Algorithm 1 with a search-window of size 21×21, i.e., $r = 10$. Furthermore, the standard deviation of the Gaussian kernel a was set to $d = 3$.

Figs. 3(a) and 4(a) display the two test images $\mathbf{f}$ used, while the observations $\mathbf{u}$ are shown in Figs. 3(b) and 4(b). Here, the standard deviation of noise was $\sigma = 0.01$. Two experiments using NL-means are shown, first by taking the example image $\mathbf{v}_1 = \mathbf{u}$, i.e., the observed image itself, shown in 3(c) and 4(c), and second

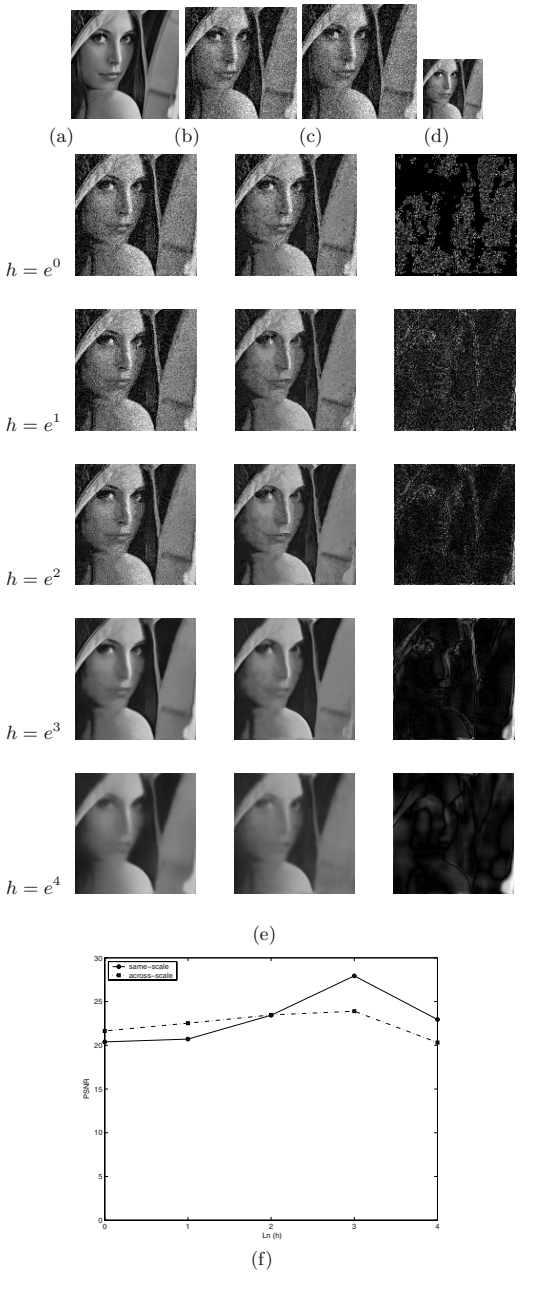

Fig. 3. (a) shows the original image $\mathbf{f}$, (b) the observation $\mathbf{u}$, (c) $\mathbf{v}_1 = \mathbf{u}$, (d) $\mathbf{v}_2 = \mathcal{P}_2\mathbf{u}$. In (e): For each value of h are shown: $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x)$ (left), $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{v}_2,h,d,a)}(x)$ (middle) and $|\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x)$- $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{v}_2,h,d,a)}(x)|$ (right). In (f): PSNR vs ln h for images plotted on left and in middle of (e).

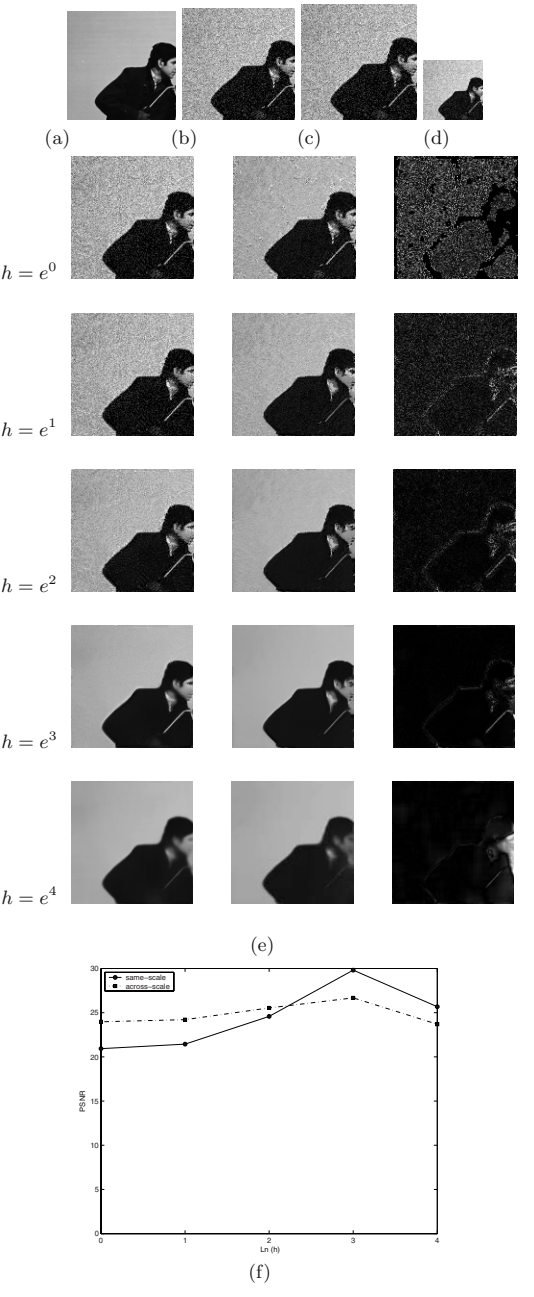

Fig. 4. (a) shows the original image $\mathbf{f}$, (b) the observation $\mathbf{u}$, (c) $\mathbf{v}_1 = \mathbf{u}$, (d) $\mathbf{v}_2 = \mathcal{P}_2\mathbf{u}$. In (e): For each value of h are shown: $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x)$ (left), $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{v}_2,h,d,a)}(x)$ (middle) and $|\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x) - \widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{v}_2,h,d,a)}(x)|$ (right). In (f): PSNR vs ln h for images plotted on left and in middle of (e).

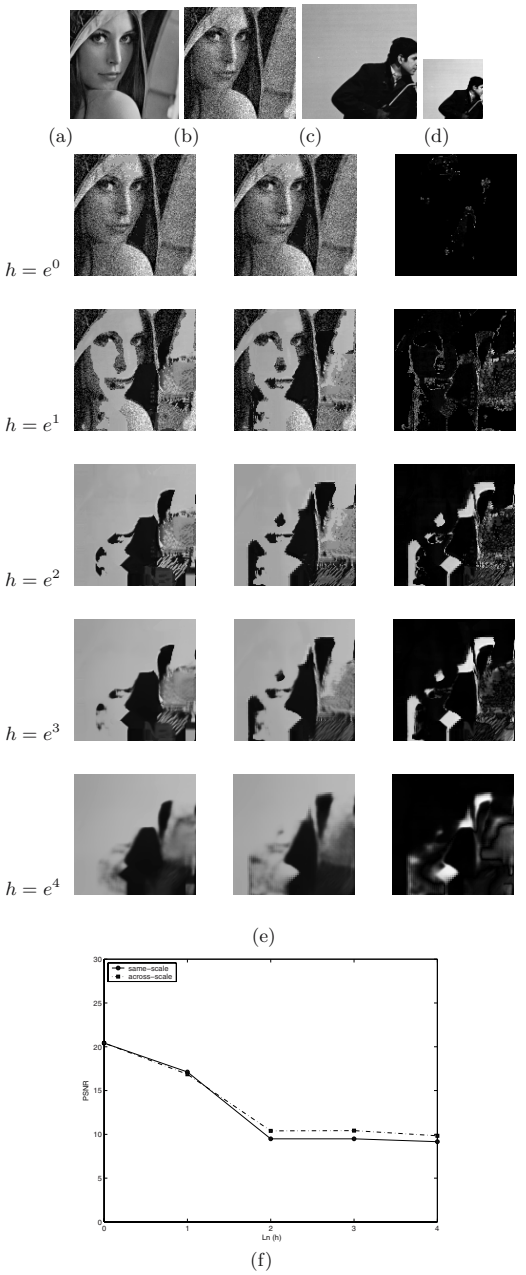

Fig. 5. (a) contains the image $\mathbf{f}$, (b) the observation $\mathbf{u}$, (c) $\mathbf{v}_1$, (d) $\mathbf{v}_2 = \mathcal{P}_2(\mathbf{v}_1)$, (e) For each value of h, Left hand-side: $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x)$, middle: $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{v}_2,h,d,a)}(x)$, right hand-side: $|\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x)$- $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{v}_2,h,d,a)}(x)|$, In (f): PSNR vs ln h for images plotted on left and in middle of (e).

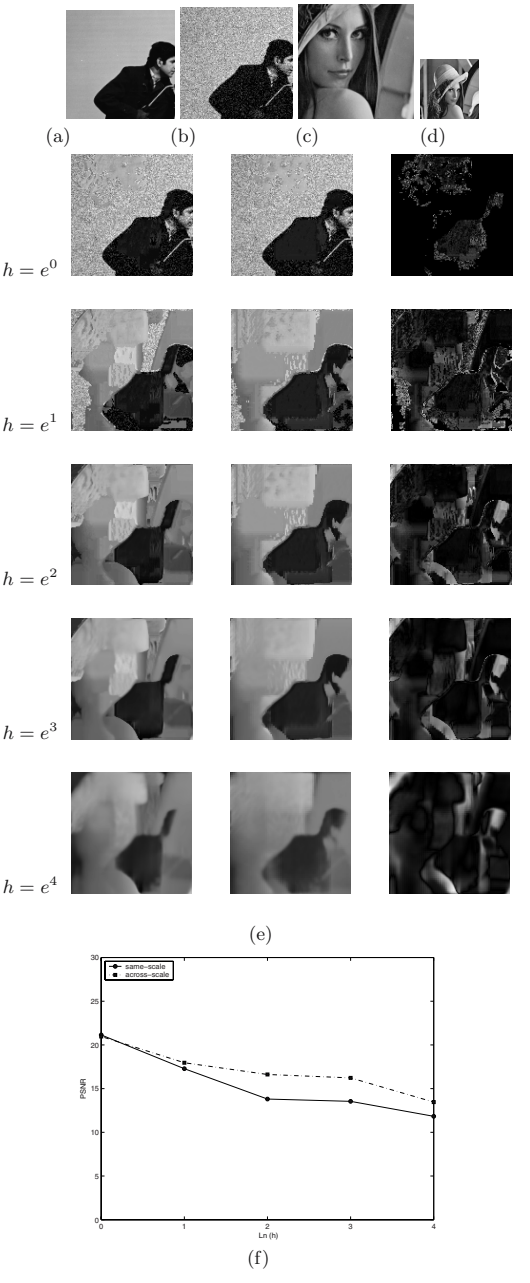

Fig. 6. (a) shows the original image $\mathbf{f}$, (b) the observation $\mathbf{u}$, (c) $\mathbf{v}_1$, (d) $\mathbf{v}_2 = \mathcal{P}_2(\mathbf{v}_1)$, (e) For each value of h, Left hand-side: $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x)$, middle: $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{v}_2,h,d,a)}(x)$, right hand-side: $|\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x)$- $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{v}_2,h,d,a)}(x)|$, (f) Plot of PSNR vs. ln h for images plotted on left and in middle of (e).

by taking the example image $\mathbf{v}_2 = \mathcal{P}_2(\mathbf{u})$, i.e., the coarser scale approximation of the observed image, with factor 2, shown in 3(d) and 4(d). In 3(e) and 4(e), we have chosen various values of the smoothness parameter h. For each value of h, we have plotted $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{u},h,d,a)}(x)$ on the left and $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathcal{P}_2(\mathbf{u}),h,d,a)}(x)$ in the middle. On the right, for each h-value, is plotted the absolute value of the difference of these two images, $|\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{u},h,d,a)}(x)$- $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathcal{P}_2(\mathbf{u}),h,d,a)}(x)|$. Finally in 3(f) and 4(f) plots of PSNR vs. ln h for the images plotted on the left and middle of (e) are presented. It can be observed that varying the smoothness parameter h, the PSNR is increased for both same-scale and cross-scale experiments, and eventually it is decreased as h approaches infinity. We have used a logarithmic scale for h so that the decay of PSNR becomes more visible for high-values of h on the graph.

These experiments suggests that good denoising is accomplished with the use of either $\mathbf{u}$ and $\mathcal{P}_2(\mathbf{u})$ as examples, i.e, same-scale or cross-scale neighborhoods, in the NL-means filter.

5 Filters Based on Irrelevant Examples and Results

We now examine the performance of Algorithm 1 in the case of irrelevant examples, i.e., examples taken from another image, both in the same-scale and cross-scale case. Some results on the use of examples from other images for the same-scale case have been presented in [6]. In what follows, we use the *Cameraman* and *Lena* images. Once again, a search-window of size 21×21, i.e., $r = 10$, was used, and the standard deviation of the Gaussian kernel a was set to $d = 3$.

Figs. 5(a) and 6(a), show the images $\mathbf{f}$ while the respective observations $\mathbf{u}$ are shown in 5(b) 6(b). Here, once again, the standard deviation of the added noise was $\sigma = 0.01$. The results of two experiments using NL-means are shown. In the first experiment, we used the example image $\mathbf{v}_1$ shown in (c). In the second experiment, we used a factor-2 coarser scale version of the example image, $\mathbf{v}_2 = \mathcal{P}_2(\mathbf{v}_1)$, shown in (d). In (e), we present results for various values of h the smoothness parameter. On the left are presented the results $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x)$. In the middle are shown $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathcal{P}_2(\mathbf{v}_1),h,d,a)}(x)$. And on the right are shown the the absolute differences of these two images $|\widetilde{\mathbf{f}}_{(\mathbf{u},\mathbf{v}_1,h,d,a)}(x)$- $\widetilde{\mathbf{f}}_{(\mathbf{u},\mathcal{P}_2(\mathbf{v}_1),h,d,a)}(x)|$. Finally in (f), plots of PSNR vs. ln h are shown.

It can be observed that in both cases, PSNR decreases with h, due to the use of the irrelevant examples in the approximation step.

6 Conclusions

In this paper, we have examined the role of scale in the context of the NL-means filter by experimenting over various images and various values of smoothness parameter h. A new example-based variant of the NL-means is introduced and we have also examined the issue of relevant examples. It turns out that cross-scale patches are also "relevant" in approximating the image, as opposed to some "irelevant" image, as intuitively expected. This work in some way provides

a deeper insight to the question, examined in [6], of why examples from cross-scales of an image can be used as a regularizer to solve the problem of single-frame image using using self-examples. A more analytical explanation of the results in the paper seems to be a natural path to follow.

Acknowledgments

This research has been supported in part by the Natural Sciences and Engineering Research Council of Canada, in the form of a Discovery Grant (ERV). ME is supported by an Ontario Graduate Scholarship.

References

1. Alexander, S.K.: Multiscale methods in image modelling and image processing, Ph.D. Thesis, Dept. of Applied Mathematics, University of Waterloo (2005)
2. Alexander, S.K., Vrscay, E.R., Tsurumi, S.: A simple, general model for the affine self-similarity of images. In: Campilho, A., Kamel, M. (eds.) Proceedings on ICIAR 2008. LNCS, vol. 5112, pp. 192–203. Springer, Heidelberg (2008)
3. Barnsley, M.F.: Fractals Everywhere. Academic Press, New York (1988)
4. Buades, A., Coll, B., Morel, J.M.: A nonlocal algorithm for image denoising. In: IEEE International conference on Computer Vision and Pattern Recognition (CVPR), San-Diego, California, June 20-25, vol. 2, pp. 60–65 (2005)
5. Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. SIAM Journal on Multiscale Modeling and Simulation (MMS) 4(2), 490–530 (2005)
6. Ebrahimi, M., Vrscay, E.R.: Solving the Inverse Problem of Image Zooming Using "Self-Examples". In: Kamel, M., Campilho, A. (eds.) ICIAR 2007. LNCS, vol. 4633, pp. 117–130. Springer, Heidelberg (2007)
7. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: IEEE International Conference on Computer Vision (ICCV), Corfu, Greece, September 20-25, pp. 1033–1038 (1999)
8. Fisher, Y. (ed.): Fractal image compression, theory and application. Springer, New York (1995)
9. Freeman, W.T., Jones, T.R., Pasztor, E.C.: Example-based super-resolution. IEEE Comp. Graphics And Appl. 22(2), 56–65 (2002)
10. Ghazel, M., Freeman, G., Vrscay, E.R.: Fractal image denoising. IEEE Trans. on Image Proc. 12(12), 1560–1578 (2003)
11. Haber, E., Tenorio, L.: Learning regularization functionals. Inverse Problems 19, 611–626 (2003)
12. Lu, N.: Fractal Imaging. Academic Press, NY (1997)
13. Mallat, S.: A wavelet tour of signal processing. Academic, San Diego (1998)
14. Mallat, S., Zhong, S.: Characterization of signals from multiscale edge. IEEE Trans. Patt. and Anal. and Mach. Intell. 14(4), 710–732 (1992)
15. Zhu, S.C., Mumford, D.: Prior learning and Gibbs reaction-diffusion. IEEE Trans. on Patt. Analysis and Machine, Intel. 19(11), 1236–1250 (1997)