



HAL
open science

Local Jet Based Similarity for NL-Means Filtering

Antoine Manzanera

► **To cite this version:**

Antoine Manzanera. Local Jet Based Similarity for NL-Means Filtering. International Conference on Pattern Recognition (ICPR), Aug 2010, Istamboul, Turkey. 10.1109/ICPR.2010.654 . hal-01130884

HAL Id: hal-01130884

<https://hal.science/hal-01130884>

Submitted on 12 Mar 2015

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

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

Local Jet based similarity for NL-Means filtering

Antoine Manzanera
ENSTA-ParisTech, 32 Bd Victor, Paris
antoine.manzanera@ensta.fr

Abstract

Reducing the dimension of local descriptors in images is useful to perform pixels comparison faster. We show here that, for enhancing and optimising the computation of the NL-means denoising filter, image patches can be favourably replaced by a vector of spatial derivatives (local jet), to compute the similarity between pixels. First, we present the basic, limited range implementation, and compare it with the original NL-means. We use a fast estimation of the noise variance to automatically adjust the main parameter of the filter. Next, we present an unlimited range implementation using nearest neighbours search in the local jet space, based on a binary search tree representation.

1. Introduction and background

Removing the noise from images while preserving their significant features is an old and difficult problem in image processing. The general linear framework can be expressed as follows:

$$\hat{f}(x) = \frac{1}{Z(x)} \sum_{y \in \mathcal{N}(x)} f(y) \omega(x, y) \quad (1)$$

where f is the image to be filtered, x and y are pixels in \mathbb{Z}^2 , $\mathcal{N}(x) \subset \mathbb{Z}^2$ is a neighbourhood of x , $\omega(x, y) \in \mathbb{R}$ is the weight attached to y , and $Z(x) = \sum_{y \in \mathcal{N}(x)} \omega(x, y)$

is the normalising constant.

In traditional convolution, $\omega(x, y) = g(\|x - y\|)$, where g is a decreasing real function. The advantage is that $\mathcal{N}(x)$ (referred to as the searching window) can then be limited to $\{y \in \mathbb{Z}^2; g(\|x - y\|) > \varepsilon\}$, but because such filter tends to remove all high frequencies, it performs poorly in denoising. Many authors have designed other weighting function depending on $f(x)$ and $f(y)$. Maybe one of the most extreme, and most successful version was proposed by Buades *et al* in [3],

where $\omega(x, y)$ no longer depends on $\|x - y\|$, but only on the local resemblance of the image f around x and y :

$$\omega(x, y) = e^{-\frac{d_f(x, y)}{h^2}} \quad (2)$$

where h is a decay parameter, related to the amount of noise to be removed, and $d_f(x, y)$ measures the local similarity of image f at x and y , defined as follows:

$$d_f(x, y) = \sum_{i \in \mathcal{W}(O)} k(\|i\|) (f(x + i) - f(y + i))^2 \quad (3)$$

where $\mathcal{W}(O)$ is a neighbourhood of the origin, defining the patch, and k is a real decreasing function (a Gaussian kernel in [3]).

By making the most of physically meaningful correlations from the periodicity of natural textures, the so called Non Local Means (NL-means) have demonstrated impressive results in image denoising. But because $\mathcal{N}(x)$ is no longer bounded, the critical disadvantage of NL-Means is a great computational complexity.

Different optimisations have been proposed in the literature, using: multi-resolution to reduce the searching window [4], a vector version of equation 1 to restore the blocks as a whole [4, 5], a certain type of integral images to reduce the computation cost of equation 3 [6], a reduction of $\mathcal{N}(x)$ to a subset according to a certain relevance selection [9, 11, 5], a reduction of dimension of the appearance descriptors using SVD or PCA [11], or binary trees for performing efficient nearest neighbours search [1].

The method proposed here is related to dimension reduction, but it uses one fixed set of appearance features: the spatial derivatives. The scale parameter do not depend on the patch size, but on the standard deviation of the Gaussian kernel used to compute the derivatives. The lower dimension of the local descriptors not only reduces the computation cost of similarity in limited range implementation, but also lends itself to efficient nearest neighbours search in unlimited range implementation. It can also be combined with other kind of optimisation: hierarchical, block-wise or selective.

2. Local jet based similarity

The first justification for using the derivatives as a compact representation of image patches naturally comes from the Taylor expansion (here at order 2):

$$f(x+c) = f(x) + \nabla_f(x).c + \frac{1}{2}c^t.H_f(x).c + o(\|c\|^2) \quad (4)$$

Where ∇_f is the gradient vector and H_f the Hessian matrix of f . Figure 1 illustrates on a natural image, split into 7×7 patches, how the local jet of different order can (or cannot) represent an image patch. The reconstruction is performed on patches of the same size, using only the local jet computed at the patch centre. This experiment is shown to justify the fact that we use order 2 local jet from now on, as the minimal order to reasonably discriminate similar pixels in images.

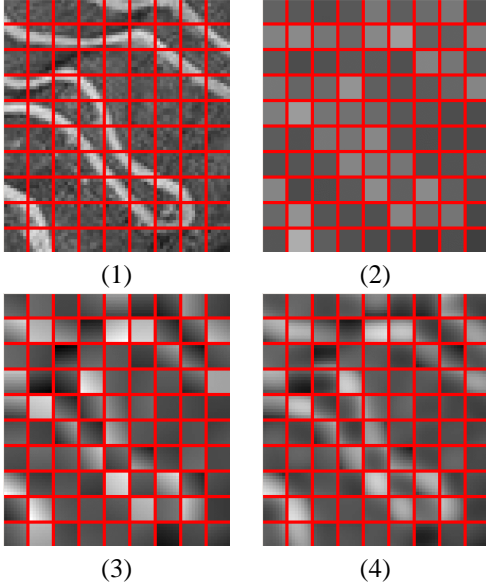


Figure 1. Reconstructing patches by Taylor expansion using the local jet of the patch centre at scale $\sigma = 2.0$. (2) Order 0 (1d feature) (3) Order 1 (3d feature), (4) Order 2 (6d feature)

Now our purpose is not to reconstruct patches but to use the local jet instead of them. By replacing expression of $f(x+c)$ given by equation 4 within equation 3, supposing that $\mathcal{W}(O)$ is symmetrical, the local similarity function $d_f(x,y)$ turns into a linear combination of the sum of squared differences of each derivative:

$$d_f(x,y) = \sum_{i+j \leq 2} c_{(i,j)}(f_{ij}(x) - f_{ij}(y))^2 \quad (5)$$

using the notation $f_{ij} = \frac{\partial f^{i+j}}{\partial x_1^i \partial x_2^j}$, with x_1 and x_2 a basis of \mathbb{R}^2 . $c_{(i,j)}$ the coefficients attached to every derivative should actually be larger for higher degrees derivatives, when the radius of the patch increases. In fact, we do not use patches anymore but compute the derivatives in every pixel at a certain scale σ by estimating every derivative using a convolution of the image by the corresponding derivative of Gaussian:

$$f_{ij}^\sigma = f \star \frac{\partial G_\sigma^{i+j}}{\partial x_1^i \partial x_2^j} \quad (6)$$

where G_σ is the 2d Gaussian function of standard deviation σ . We then propose the following coefficients:

$$c_{(i,j)} = \frac{\sigma^{i+j}}{i+j+1} \quad (7)$$

where σ^{i+j} is the scale normalisation factor from scale space theory [8] (Histogram based normalisation can be used as well), and $i+j+1$ is the number of $(i+j)$ -order derivatives. As in the original NL-means, the decay parameter has to be adapted to the dynamic range of the similarity measure. Figure 2 illustrates the application of the local jet based similarity on a texture image, for 3 different pixels.

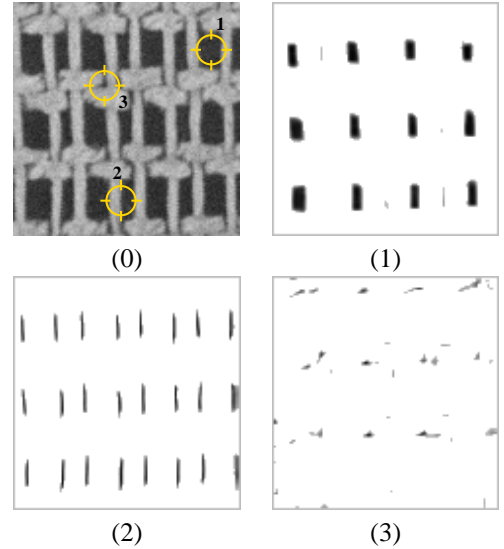


Figure 2. Similarity map (in reverse video) using the local jet of order 2, with $\sigma = 2.0$, and $h = 30$ for 3 different pixels.

Another justification of using local jet as description vector is the fact that the first singular (or eigen) vectors that arise in SVD or PCA based optimisation on natural images look much like the first derivatives of a 2d Gaussian function (see [11] for example).

3. Local jet based NL-means

The main benefit of this method is to reduce the dimension of the vectors used in the similarity metrics. Once computed the local jet, the computation time of $d_f(x, y)$ is then reduced accordingly. Compared to the original NL-means using 7×7 patches, the number of operations is reduced by approximately 8 for local jet of order 2. To keep this acceleration significant however, efficient methods must be used to compute the local jet. We use the recursive implementation of the Gaussian and its derivatives proposed in [13], which allows to compute the 6 components of the local jet in constant time per pixel, for any σ .

So the first results we present correspond to the simple adaptation of the original NL-means, using a limited searching window $\mathcal{N}(x)$. Figure 3 compares the results with the original NL-means on an image detail. $\mathcal{N}(x)$ is of size 13×13 for the two methods. For the original method, the patch $\mathcal{W}(0)$ is a 7×7 square, and the decay parameter is $h = 150$. For the local jet method, $\sigma = 1.5$ and h is automatically adjusted as explained below.

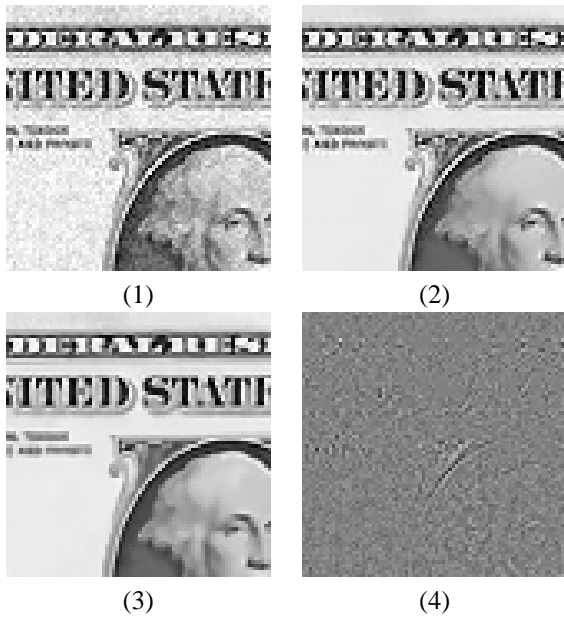


Figure 3. Limited range NL-means: (1) noisy image (2) original method, (3) local jet method. (4) removed noise

The decay parameter h both depends on the range of the metrics similarity and on the amount of noise to be removed. We normalise our distance and estimate the noise variance $\hat{\sigma}$ in the image, then adjust $h = \beta \hat{\sigma}$, with β a constant independent of the image content. Follow-

ing [7, 12, 5], we estimate the noise by summing the local residuals (squared difference between the value and the local average) over the image, excluding areas with high regularised contrast, corresponding to pixels whose gradient modulus is amongst the τ highest ($0 < \tau < 1$). However, unlike [12], we exploit the fact that the gradient modulus of natural images have Rayleigh shaped distribution, and then the τ highest contrast pixels are theoretically identical to those pixels whose gradient modulus is beyond $2\mu\sqrt{-\frac{\log(\tau)}{\pi}}$, where μ is the measured average value of the gradient modulus. We used $\tau = 0.3$ and then $\beta = 0.25$ for all tested images.

Another advantage of using a low dimension feature space is the possibility to explicitly embed the image data so that any pair of pixels that are close in the metrics of the feature space could be rapidly retrieved. Using a multidimensional array whose dimensions correspond to the components of the local jet would result in huge memory requirement, so one better solution is to use a dedicated data structure like a binary search tree (see [2] for details). This representation allows to use a true non-local, i.e. unlimited range, local jet based similarity mean filter. Now, instead of computing the weights of all the neighbours of every pixel within a limited range, we compute the weights of a given number of Nearest Neighbours (NN) of the pixel, in the metrics of the local jet space, whatever the distance in \mathbb{R}^2 space. We have used in our implementation the ANN library developed by Arya and Mount [10]. Example of results are shown on Figure 4.

One remarkable property of the NN method is to be almost insensitive to the choice of the weight function. Indeed, the variation interval of similarity distances is much more narrow. However, comparing the quality of denoising for the limited range vs NN methods is not obvious: the NN results may seem more natural, but the removed noise (difference between filtered and original images) usually show more image structure, while the unlimited search seems to find patterns in large noisy homogeneous surfaces, which tends to exaggerate the texturation of these regions (see Figure 4).

Regarding the computational complexity, the search tree is constructed in $\mathcal{O}(dn \log(n))$, where n is the number of pixels and d the dimension of the local jet vector. For the exact k-NN search, the (worst case) complexity is not much better than the quadratic complexity of the unlimited range traditional approach. However, experiments show that in practice, k-NN exact search are still significantly faster than patch-based limited range method, when the number of neighbours k equals the dimension of the searching window. In the case of approximate k-NN search [2], the query complexity is

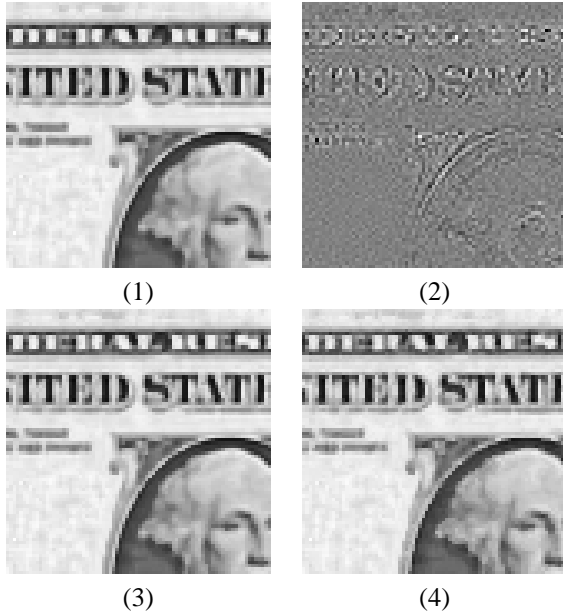


Figure 4. Unlimited range, local jet based nearest neighbours NL-means: (1) 30 NN, exact search (2) removed noise (3) 30 NN, approximate search ($\varepsilon = 1.0$). (4) Idem, ($\varepsilon = 6.0$).

lowered. If x and y are two d -dimensional vectors, y is said to be an ε -approximate k -th nearest neighbour of x if the distance between x and y is less than $1 + \varepsilon$ larger than the distance between x and its actual k -th nearest neighbour. In that case, the computation of the k -NN for all the pixels is made in $\mathcal{O}(n(c_\varepsilon^d + kd) \log(n))$, where c_ε^d is a constant whose order of magnitude is $(\frac{d}{\varepsilon})^d$. In practice, increasing ε may significantly speed up the computation without affecting too much the results (see Figure 4).

4. Conclusion and prospective works

We have shown in this paper that the low dimension local jet vectors could favourably replace the patches to compute pixel similarity in NL-Means filtering. Basically, in limited range implementation, it speeds up the computation by lowering the number of operations while remaining fully regular, with good filtering quality. More experiments must be done to better quantify the difference with the original NL-Means, but more fundamentally we believe that, by varying the orders and number of scales, our method can provide relevant intermediate filters between the bilateral filter and the

original NL-means.

By using a data structure dedicated to NN search, we have also proposed a true unlimited range, NN based NL-Means in the local jet basis. Good speed-up can be obtained, but at the price of an important loss of regularity which makes the true unlimited range still problematic for embedded implementation. We plan to investigate new NN search data structures that have been proposed very recently [1] in the context of local jet descriptors, and develop a sparse, incrementally updated, model of the similarity features, to adapt our algorithm to Real-Time video.

References

- [1] A. Adams, N. Gelfand, J. Dolson, and L. M. Gaussian kd-trees for fast high-dimensional filtering. In *ACM SIGGRAPH 2009*, 2009.
- [2] S. Arya, D. Mount, N. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *Journal of the ACM*, 45(6):891–923, 1998.
- [3] A. Buades, B. Coll, and J. Morel. A non-local algorithm for image denoising. In *Proc. IEEE CVPR*, volume 2, pages 60–65, 2005.
- [4] A. Buades, B. Coll, and J. Morel. A review of image denoising algorithms, with a new one. *SIAM J. Multiscale Modeling and Sim.*, 4(2):490–530, 2005.
- [5] P. Coupé, P. Yger, S. Prima, P. Hellier, C. Kervrann, and C. Barillot. An optimized blockwise non local means denoising filter for 3d magnetic resonance images. *IEEE Transactions on Medical Imaging*, 27(4):425–441, 2008.
- [6] J. Darbon, A. Cunha, T. Chan, S. Osher, and G. Jensen. Fast nonlocal filtering applied to electron cryomicroscopy. In *ISBI'08*, pages 1331–1334, 2008.
- [7] J. Immerkær. Fast noise variance estimation. *Computer Vision and Image Understanding*, 64(2):300–302, 1996.
- [8] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116, 1998.
- [9] M. Mahmoudi and G. Sapiro. Fast image and video denoising via non-local means of similar neighborhoods. *IEEE Signal Processing Letters*, 12(12):839–842, Dec. 2005.
- [10] D. Mount and S. Arya. ANN: A library for approximate nearest neighbor searching. In *CGC Workshop on Computational Geometry*, 1997. <http://www.cs.umd.edu/~mount/ANN/>.
- [11] J. Orchard, M. Ebrahimi, and A. Wong. Efficient non-local means denoising using the SVD. In *Proc. IEEE ICIP*, pages 1732–1735, 2008.
- [12] S. Tai and S. Yang. A fast method for image noise estimation using laplacian operator and adaptive edge detection. In *ISCCSP'08*, pages 1077–1081, 2008.
- [13] L. van Vliet, I. Young, and V. PW. Recursive Gaussian derivative filters. In *Proc. IEEE ICPR*, volume 1, pages 509–514, 1998.