A ROTATIONALLY INVARIANT BLOCK MATCHING STRATEGY IMPROVING IMAGE DENOISING WITH NON-LOCAL MEANS

Sebastian Zimmer, Stephan Didas and Joachim Weickert

Mathematical Image Analysis Group
Faculty of Mathematics and Computer Science
Saarland University, Building E1.1
66041 Saarbrücken, Germany
{szimmer,didas,weickert}@mia.uni-saarland.de

ABSTRACT

We propose a rotationally invariant similarity measure as a modification of the well-known block matching algorithm for finding similar regions in an image or an image sequence. This algorithm can find similar patches even if they appear in several rotated or even mirrored instances. We demonstrate the application of this approach to enhance the quality of the non-local means algorithm for image denoising. For this filtering method, we also introduce a locally adaptive way of choosing the parameters. Numerical examples show that both modifications lead to a visible and measurable qualitative improvement of the denoising results.

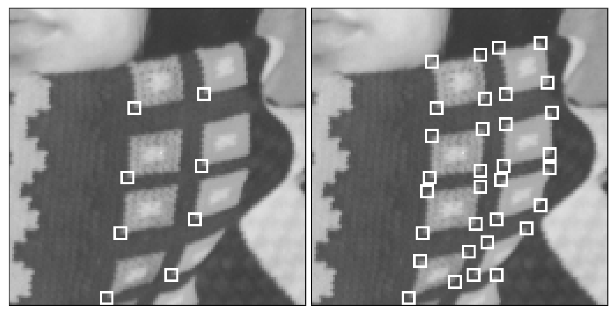

Figure 1. Similar blocks are manually marked with white squares. **Left:** Traditional block matching. **Right:** Rotationally invariant similarity measure.

1. MOTIVATION

Block matching strategies belong to the earliest and most often applied ideas in motion analysis [1, 2, 3]. Due to their simplicity, they still play an important role in modern algorithms, for example in the MPEG video compression standard. Besides the application to image sequences, it is clear that they can also be used to detect repeating structures or regions inside a single image. Thus, the basic idea of block matching has also been applied for image processing methods: For example, the inpainting algorithm by Efros and Leung [4] fills in missing information by searching for similar regions in the image and completing the missing details according to the information found there.

A very popular denoising approach motivated by this inpainting method is the so-called *nonlocal means (NL-means)* algorithm for image denoising that has been proposed by Buades et al. [5]. To calculate the denoised value of one pixel, NL-means searches for similar neighbourhoods in the whole image and averages the corresponding pixels of these neighbourhoods. NL-means belongs to the class of adaptive averaging filters, like the sigma filter [6], the Yaroslavsky filter [7], or the bilateral filter [8, 9, 10]. The difference to previous approaches is the way of calculating the weights for the averaging process with the consideration of neighbourhood information.

In spite of its simplicity, this approach is able to yield high-quality denoising results. This has motivated scientific activity in this field: For example, the methodology has been formalised with the help of variational approaches by Kindermann et al. [11] and Gilboa and Osher [12]. An interpretation and analysis in the statistical context has been given by Kervrann and Boulanger [13]. On the application side, Mahmoudi and Sapiro have proposed a fast implementation method which allows for the application in the context of video denoising [14]. To improve the quality, Brox and Cremers have proposed not to average over all pixels, but only over a certain number of best matches [15]. They also emphasise that it is often not appropriate to choose the parameters of the method globally. The idea to use self-similarity in the image for denoising has also been refined by the so-called *collaborative filtering* approach by Davob et al. [16].

The goal of this paper is to present a similarity measure that is invariant under rotations and mirroring. This extends very recent ideas by Alexander et al. [17] to arbitrary rotations. In the context of nonlocal means, it makes sense also to involve pixels in the averaging which belong to a neighbourhood that differs from the reference patch only by rotation and mirroring. Figure 1 visualises the aim of finding more useful pixels for averaging by considering rotated and mirrored image patches. It turns out that this in fact can help to find a better choice of pixels and more suitable weights for the average and thus improve the quality of the results of NL-means. Further, we propose a method for a locally adaptive parameter choice.

This paper is organised as follows: The next section reviews the mathematical formulation of block matching and the nonlocal means algorithm. The new approach for rotationally invariant block matching and its application within the nonlocal means algorithm is described in detail in Section 3. Section 4 presents further improvements of NL-means, namely an adaptive choice of parameters. Numerical experiments comparing the qualitative results are shown in Section 5. Section 6 concludes the paper with a summary and an outlook.

2. BLOCK MATCHING AND NONLOCAL MEANS

In this section, we give a mathematical formulation for block matching and NL-means as we will use it later on. Let $f : \Omega \longrightarrow \mathbb{R}$ denote a greyscale image, typically with domain $\Omega \subseteq \mathbb{R}^2$. The distances of two blocks in classical block matching is calculated as *sum of squared distances*:

$$d_{SSD}(p,q) \;=\; \int_B (f(p+b) - f(q+b))^2 \; db \;\;. \qquad (1)$$

In this continuous formulation $B \subset \mathbb{R}^2$ is the set defining the neighbourhood and $b \in B$ is the displacement inside this neighbourhood. Usually, this will be a square; for symmetry reasons especially in our context of rotational invariance, we will use a disc here. Instead of the quadratic penalisation of the distance, it is also common to use more robust subquadratic functions in (1).

For denoising an image, the NL-means algorithm uses similarity measures to search for similar areas in the whole image. For every pixel to be denoised, the algorithm looks for areas that are similar to the window around that pixel. The idea is that if two areas are similar, then their central pixels should have a similar meaning for the image and thus similar grey values. If two areas in an image are very similar, they can be understood as two noisy measurements of the same noise-free patch, and thus it makes sense to average their central pixel to estimate the original value. Consequently, the denoising result is obtained by an average as follows:

$$D_f(p) \;=\; \int\limits_{A(p)} w(p,q) \cdot f(q) \;\; dq \qquad (2)$$

$$w(p,q) \;=\; \frac{\exp(\frac{-d(p,q)^2}{\lambda^2})}{\int\limits_{A(p)} \exp(\frac{-d(p,q')^2}{\lambda^2}) \;\; dq'} \qquad (3)$$

where $A(p)$ defines a search window, in which the algorithm should search for similar pixels. As described above, this search window is the whole image in theory. Nevertheless, for complexity reasons most implementations restrict $A(p)$ to a window surrounding p; typical sizes are 21×21 pixels. Instead of (1) a weighted distance measure is used here:

$$d(p,q) \;=\; \int\limits_B G_\alpha(b) \cdot (f(p+b) - f(q+b))^2 \;\; db \qquad (4)$$

where G_α is a Gaussian kernel:

$$G_\alpha(b) \;=\; \exp\left(\frac{-|b|^2}{\alpha^2}\right) \;. \qquad (5)$$

However, while block matching is known to be robust under noise, it has the severe drawback that it is not invariant under any transformation such as rotations or mirroring. As for every pixel the algorithm searches for blocks that are similar to the window with the original pixel in its centre, there is no reason why blocks that are rotated around their centre should have a higher distance than blocks that are not rotated. We have already seen a visualisation of the idea in Figure 1: A search strategy which is invariant under simple transformations can find many more pixels in the image with a similar meaning.

3. ROTATIONALLY INVARIANT BLOCK MATCHING

In this section, we describe our generalisation of a similarity measure which is invariant under rotations and mirroring. First we sketch the approach in a generic way. In the second step, we focus in detail on a specific practical implementation.

3.1. Basic idea

The central problem is to estimate the angle of rotation between two corresponding blocks. This is done with the help of one point correspondence: We estimate the angle by which a certain pixel in the block, the so-called *centroid*, is rotated around the block's centre. Natural requirements at such a centroid are that it is robust under noise and easy to calculate. Furthermore, we identify all points within a block by vectors pointing from its centre to the points' coordinates (see Figure 2). We can then describe the basic idea of *rotationally invariant block matching (RIBM)* in a simple generic algorithm:

1. Estimate the angle of rotation between the blocks.
2. To each pixel in the first block, find the position of the corresponding pixel in the second block by rotating its vector by this estimated angle.

The summed distances represent the total distance of the two blocks. It obviously makes sense to use circles as blocks here. This algorithm can be extended to detect not only rotated, but also mirrored versions of the reference block. If we have found a mirrored version, we can again mirror it at an arbitrary axis and then apply the algorithm as described above.

In the following subsection, we will present an example for an implementation of this approach. An evaluation for the application of denoising with NL-means is shown in Section 5.

3.2. An implementation of RIBM

Let two blocks B and B' in the image be given such that B' is a noisy, rotated around the centre (and possibly mirrored) version of B. In our sample implementation, we

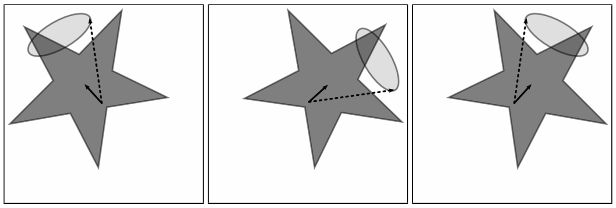

Figure 2. Three identical blocks. The middle block is rotated by 90° to the right, while the right block is mirrored at the y-axis. The continuous arrow points from the centre to the centroid of the block (in this example, dark grey values are given high numerical values). The dotted arrow in the left block points to some arbitrary point, the dotted vector in the middle and the right block are its corresponding points.

use centroids, which are commonly used for the computation of shift-invariant moments, to estimate the angle of rotation. To define the centroid, we assume that pixels within a block are addressed with a coordinate system that has its origin at the block's centre (see Figure 2):

$$c_B := \begin{pmatrix} \frac{\int_B x \cdot f(x,y) \; dx \; dy}{\int_B f(x,y) \; dx \; dy} \\ \\ \frac{\int_B y \cdot f(x,y) \; dx \; dy}{\int_B f(x,y) \; dx \; dy} \end{pmatrix} \quad (6)$$

The calculations of the angles can be done without expensive trigonometric functions by using rotation matrices. Let $\vec{c_B}$ denote the normalised vector corresponding to the centroid of B and let $m_{B,B'}(v)$ be a function that flips the sign of the first component of the vector v (i.e. mirrors the vector at the y-axis) if block B' is a mirrored version of block B. In our implementation we use the seventh moment of Hu (Hu_7, see [18] for the definition and some properties) to compute m. Hu_7 is known to be invariant under many transformations such as rotation, but changes its sign under mirroring. While the numerical value of Hu_7 suffers a lot from discretisation and noise, its sign remains quite stable. Our strategy to compensate for mirroring is then given as:

$$m_{B,B'}(v) := \begin{cases} (-v_1, v_2)^\top, & \text{Hu}_7(B') \cdot \text{Hu}_7(B) < 0 \\ (v_1, v_2)^\top, & \text{else} \end{cases} \quad (7)$$

We can write the rotation matrix that describes the estimated rotation between the blocks as:

$$R_{B,B'} := R_{\vec{c_B}}^{-1} \cdot R_{m_{B,B'}(\vec{c_{B'}})} \quad \text{with} \quad (8)$$

$$R_v := \begin{pmatrix} v_1 & -v_2 \\ v_2 & v_1 \end{pmatrix} . \quad (9)$$

The normalisation in the Euclidean norm guarantees that $R_{B,B'}$ is a rotation matrix. If the block's centre and centroid coincide, this approach can not be used to estimate a rotation matrix since $c_B = 0$: We then simply use classical block matching by setting $p_{B'}$ to p_B in (10).

If, however, we can compute a rotation matrix, finding the corresponding coordinates of a point p_B in another block B' is a simple matter of matrix-vector multiplication:

$$p_{B'} := m_{B,B'}(R_{B,B'} \cdot p_B) \; . \quad (10)$$

Again, we compensate for mirroring using our function m. Now $p_{B'}$ represents the corresponding coordinates of point p in block B' relative to the centre of B'

To simplify the notation of the final formulation we denote the grey value of f at the coordinates that are given by adding the relative coordinates p_B to the centre of block B with $f_B(p_B)$. Now we can finally define our new similarity measure as

$$d(B, B') := \int_B (f_B(p_B) - f_{B'}(p_{B'}))^2 \, dp_B . \quad (11)$$

To transfer this to the discrete case we replace this integral by a sum. Since we work on a rectangular pixel grid, the rotation of a patch will only map pixels on the grid if the angle is a multiple of 90°. In all other cases, we need some kind of interpolation. For a discrete image and rotations that are not multiples of 90° one will of course not achieve perfect invariance, but even with simple interpolation methods one can get good results. The similarity measure then looks as follows:

$$d(B, B') := \sum_{p_B \in B} (f_B(p_B) - I(f_{B'}, p_{B'}))^2 \, , \quad (12)$$

where I denotes an interpolation function. For our implementation we used bilinear interpolation. Both formulations can of course be combined with an inner Gaussian weighting, as in (4).

Figure 3 shows the results of classical block matching and the rotationally invariant approach presented in this section with numerical examples. It is clearly visible that our method is able to find many more possible candidates of pixels with similar meaning in the image than the standard approach.

4. FURTHER IMPROVEMENTS OF NL-MEANS

After presenting the similarity measure in the previous section, we now turn our attention again to the NL-means method. We are going to present two modifications of the method that can enhance the quality of the denoising results in practical examples.

Firstly, we discuss the weighting of the pixels distances in the similarity calculation, and secondly, we focus on an adaptive choice of parameters.

4.1. Weighting of the central pixel

As already seen in (4), a weighted sum of squared distances is used as similarity measure in the classical NL-means algorithm. The grey value distances of the single

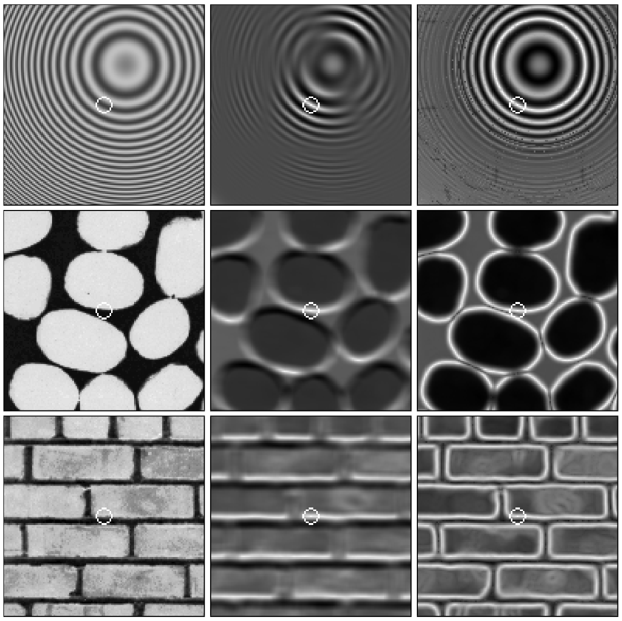

Figure 3. Comparison of classical and rotationally invariant block matching. **Left:** Texture test images, size 129×129 pixels. **Middle:** Similarity to the centre pixel of the section computed on a block of circular shape with radius 4 (framed white) using traditional block matching and unweighted Euclidean distances (linearly scaled, bright pixels have small distances). **Right:** Rotationally invariant block matching using bilinear interpolation.

pixels are weighted with by a Gaussian factor respecting the spatial distance in the image domain. The idea behind such a weighting is that pixels close to the centre are more relevant than pixels at the block's boundary. This idea is quite common also in averaging filters like the bilateral filter [10]. The problem with this approach is that for the denoising we actually want to know the value of the pixel at centre of the block, but we give it the highest weight and thus punish deviation from that value more than everything else. This leads to the problem that e. g. in a noisy but otherwise homogeneous area blocks with similar values at the centre are considered more similar than other blocks. To sufficiently denoise those areas one would have to give even blocks with high distances a high weight (i.e. increasing λ in equation (3)). Such an action would lead to problems in other areas of the image, e.g. edges or fine structures. To overcome this problem we propose a modified Gaussian kernel for the inner weighting: The weight of the centre pixel is set to a new value $\gamma < 1$. In a discrete setting this can done easily by using

$$G_{\alpha,\gamma}(b) \;:=\; \begin{cases} \gamma, & b = (0,0) \\ \exp\left(\frac{-|b|^2}{\alpha^2}\right), & \text{else}\,. \end{cases} \qquad (13)$$

The continuous case requires a slightly different solution. One way to achieve the same effect would be to subtract a

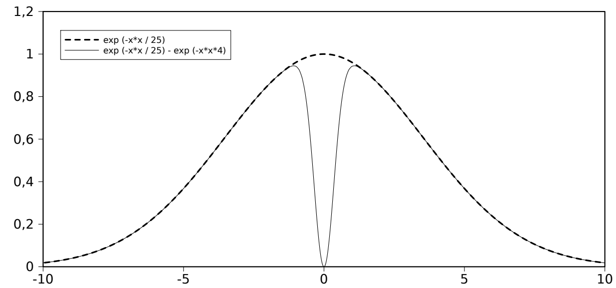

Figure 4. The middle pixel is weighted down by a second Gaussian

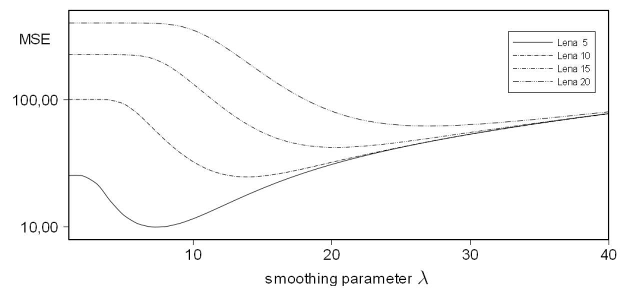

Figure 5. Behaviour of the MSE depending on the smoothness parameter λ for Lena with additive Gaussian noise with $\sigma = 5, 10, 15, 20$. The y-axis is logarithmically scaled.

second Gaussian from the first one:

$$G_{\alpha,\beta,\gamma}(b) := \exp\left(\frac{-|b|^2}{\alpha^2}\right) - (1-\gamma)\cdot\exp\left(\frac{-|b|^2}{\beta^2}\right) \qquad (14)$$

with $\beta < \alpha$. The shape of this function can be seen in Figure 4.

4.2. Locally adaptive weighting

Our second improvement of NL-means concerns the choice of the parameters: The parameter λ in (3) controlling the weighting of the distances is the most crucial parameter of the NL-means filter. Its optimal value of course depends on the amount of noise in the image: Choosing λ too low leads to a noisy result while setting it too high blurs fine structures and edges. Figure 5 displays the behaviour of the *mean square error (MSE)* depending on λ for some denoising examples.

It has been shown by Brox and Cremers [15] that there is in general not even a global optimal λ for one image such that all areas of that image are sufficiently denoised without blurring other areas in the same image too much. This directly motivates a replacement of the global parameter λ by a locally adaptive function $\Lambda(\lambda, B)$.

One possibility to achieve this is to take into account the empirical variance resp. the standard deviation s_B of the block B. The standard deviation depends on both the structure of the block and the amount of noise. This qualifies it for this task, as a sharp structure (which leads to a

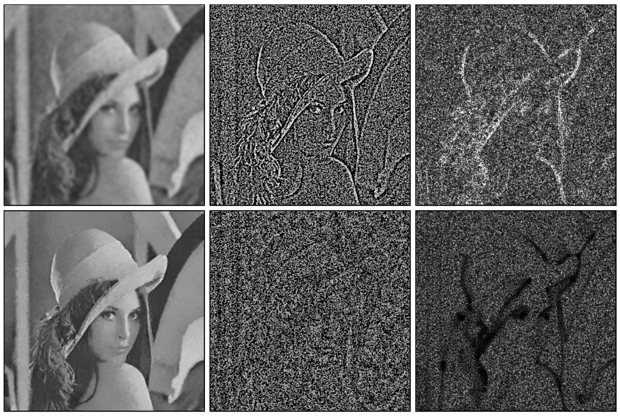

Figure 6. Comparison of method noise (MN) and absolute method noise (AMN). **Top:** Lena with Gaussian noise ($\sigma = 20$), convolved with a Gaussian kernel. **Bottom:** NL-means denoising. **From Left to Right:** Filtered image of size 256×256 pixels, MN, AMN.

high variance) usually means that there are only few similar blocks (at least significantly less than in homogeneous areas), which requires more smoothing. The same holds true for a high amount of noise, enabling us to even handle locally different noise within an image. Scaling down s_B by a sub-linear function Ψ avoids a too high influence on the smoothing. A locally adaptive function could then be given by

$$\Lambda(\lambda, B) := \lambda \cdot \Psi(s_B). \tag{15}$$

We achieved good results for $\Psi(s_B) = \sqrt{s_B}$.

Another possibility is to simply look at the best match within the search window A apart from the pixel itself, i. e. the smallest distance larger than a threshold which should not be chosen too small to avoid divisions by small numbers in the exponent of the weighting function. For our experiments, we set this threshold to 1. The corresponding function could then look like:

$$\Lambda(\lambda, A) := \lambda \cdot \Psi\left(\min_{q \in A(p)} \{dist(p,q) | dist(p,q) > 1\}\right). \tag{16}$$

Taking the identity function for Ψ and setting λ to 1 we can completely avoid the formerly most important parameter, resulting in a fully automatic filter. This yields very good visual results for different amounts of noise. The numerical results, however, are worse than the results of the original NL-means filter. Details and reasons for this are given in the next section.

5. NUMERICAL RESULTS

In this section, we demonstrate the influence of our proposed modifications with all three quality measures that are commonly used for benchmarking denoising filter: the visual impression of the filtered image, the *mean squared error (MSE)* and the so-called *method noise (MN)*. MN is simply the difference between the noisy image f and its denoised version $D(f)$:

$$MN \quad := \quad f - D(f). \tag{17}$$

The idea behind this measure is that this difference should always look like random noise. The problem is that the values can have either sign, and thus one has to shift those values to visualise the results. Blurred edges become visible as edges with a dark and a bright side. Areas where most of the noise survives, however, are not visible that easy. That is why we slightly modify the definition of MN and introduce the *absolute method noise*

$$AMN \quad := \quad |f - D(f)|\,. \tag{18}$$

While blurred structures still remain visible with this approach as bright structures, areas where the noise is still present are now visible as dark structures (see an example in Figure 6).

Figure 7. NL-means with block matching vs. RIBM. **Left:** Segment (48×48 pixels) of Flintstones with Gaussian noise ($\sigma = 20$). **Middle:** Classical NL-means. **Right:** NL-means with RIBM: round structures are denoised better.

5.1. RIBM

The usage of our RIBM implementation improved the denoising quality for most of our test images. The influence is very well visible on round edges; see Figure 7. The influence on the MSE is rather moderate. The benefit of RIBM is most prominent in structured areas that appear repeatedly in several rotated instances, for example in textures. The running time is currently about 3-4 times longer than a comparable NL-means implementation. Since the speed-up with prefiltering by Mahmoudi and Sapiro [14] relies on the comparison of the average gradient, the two ideas can unfortunately not be directly used together.

5.2. Weighting of the central pixel

Weighting down the inner Gaussian increases the amount of noise removed in homogeneous areas without increasing the blurring artifacts at edges (see Figure 8). The MSE of all of our test images was improved by this modification. The improvement becomes also visible in the AMN: the images visualising the differences become brighter without the arisal of bright areas. This improvement is done at no computational costs: As shown in (13), one single line of code is sufficient to adapt existing NL-means implementations.

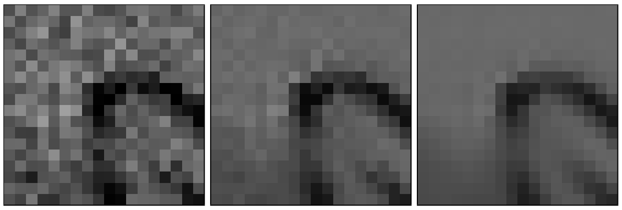

Figure 8. Weighting of the central pixel. **Left:** Section (18×18 pixels) of Flintstones with Gaussian noise ($\sigma = 20$). **Middle:** Classical NL-means. **Right:** NL-means with weight of middle pixel of inner Gaussian set to 0: More noise is removed without edges being blurred.

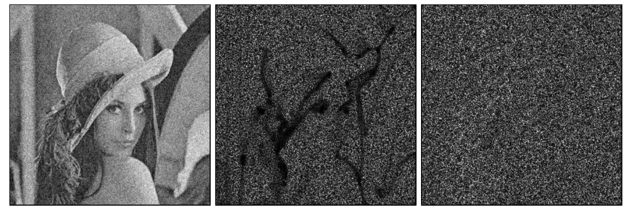

Figure 9. Locally adaptive weighting. **Left:** Lena with Gaussian noise added ($\sigma = 20$), size 256×256 pixels. **Middle:** AMN of filtered image using traditional NL-means. **Right:** AMN of filtered image using NL-means with a local adaptive smoothing parameter $\Lambda(\lambda, B) = \lambda \cdot \sqrt{S_B}$.

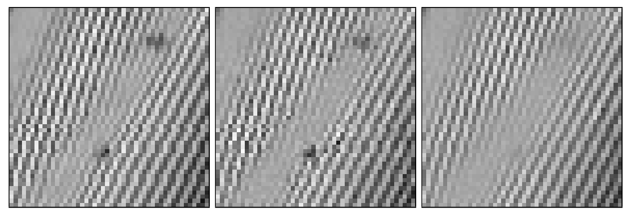

Figure 10. Locally adaptive weighting with minimal distances. **Left:** Section of Barbara. Two dark spots are visible. **Middle:** Gaussian noise added ($\sigma = 20$) and filtered using standard NL-means. **Right:** NL-means with a local adaptive smoothing parameter using minimal distances.

5.3. Locally adaptive weighting

The locally adaptive approach finally improves the denoising results at structures and edges. When using the empirical variance of the blocks for computing a locally adaptive weighting, the AMN looks even more random than the AMN of images filtered with traditional NL-means (see Figure 9). The MSE was also improved for all of our test images except for Barbara. For the local adaptive approach using minimal distances as weighting parameter lambda, the visual impression of all test images was very good, but the MSE increased on all of our test images. This is due to the fact that this approach removes some non-repetitive structures in the images without adding additional blur (see Figure 10). Another nice effect of using minimal distances is that the parameters are significantly less dependent on the actual noise. Figure 11 shows an example of images with different amounts of noise that have been filtered with the same parameters. In all images the noise is removed without blurring the edges.

A comparison in terms of the absolute method noise for all variants of NL-means presented in this paper can be found Figure 12. It is clearly visible that there are more black regions for NL-means than for the modified versions, and so more noise remains for the standard algorithm. In Table 1, the MSE of the denoising results for several test images is displayed. For each test image, most of the modifications lead to an improved quality compared to the standard NL-means algorithm. On the other hand, the combination of all modifications is not always better than the best one of the single variants.

It is also visible that rotationally invariant block matching performs best for images with round edges. A counterexample is the Barbara image: It contains large regions with parallel line-like structure (see Figure 10). In these regions, the centroid and the centre are very close to each other and an orientation estimation is highly deteriorated by noise.

6. SUMMARY

We have presented a similarity measure for image patches which is invariant under rotations and mirroring. It serves as a generalisation of classical block matching strategies. We have called this approach *rotationally invariant block matching (RIBM)*. The direction of the central pixel to a feature point is used to determine the local rotation of the blocks. Mirroring of patches is detected with the seventh Hu moment. Examples show that this method is able to find many more semantically corresponding pixels in test images than the standard approach.

We have applied RIBM to image denoising with the NL-means algorithm. It can be seen that this can increase the image quality, especially for images with many round edges. This is achieved at a moderate price: The running time of the algorithm is about 3-4 times the running time of the classical NL-means algorithm.

We have proposed two additional modifications of the NL-means algorithm which can enhance the quality even further. The first modification reduces the weighting of the central pixel in the similarity calculation which is motivated by the fact that the central pixel is still to be determined in the averaging. Thus, the confidence in this pixel should be low. The second improvement replaces the global weighting parameter λ by a locally adaptive one. This can be motivated by previous results in the literature [15]. These two modification can also be applied directly to classical NL-means. They improve the quality and leave the computational costs almost unchanged.

To conclude the paper, we sketch some questions of ongoing and further research: In the similarity measure, robust functions instead of the square could be used in order to allow for an adaptation to different kinds of noise.

Figure 11. Locally adaptive weighting using minimal distances. **Top:** Section (212 $\times$ 212 pixels) of Barbara with Gaussian noise ($\sigma = 5, 10, 20$). **Bottom:** Filtered with NL-means using minimal distances for local adaptive weighting. All images have been filtered with the same fixed parameter set.

A very interesting question is whether one can also benefit from RIBM in the context of other denoising methods like collaborative filtering [16] or other image processing applications like inpainting [4], for example.

7. REFERENCES

[1] P. J. Burt, C. Yen, and X. Xu, "Multiresolution flow-through motion analysis," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Washington, 1983, pp. 246–252.

[2] P. Anandan, "A computational framework and an algorithm for the measurement of visual motion," *International Journal of Computer Vision*, vol. 2, pp. 283–310, 1989.

[3] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, "Performance of optical flow techniques," *International Journal of Computer Vision*, vol. 12, no. 1, pp. 43–77, 1994.

[4] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proceedings of the International Conference on Computer Vision (ICCV 1999)*, Sept. 1999, vol. 2, pp. 1033–1038.

[5] A. Buades, B. Coll, and J.-M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[6] J.-S. Lee, "Digital image smoothing and the sigma filter," *Computer Vision, Graphics, and Image Processing*, vol. 24, pp. 255–269, 1983.

[7] L. P. Yaroslavsky, *Digital Picture Processing*, Springer, New York, 1985.

[8] V. Aurich and J. Weule, "Non-linear Gaussian filters performing edge preservating diffusion," in *Mustererkennung*, G. Sagerer, S. Posch, and F. Kummert, Eds. 1995, Informatik Aktuell, pp. 538–545, Springer.

[9] S. M. Smith and J. M. Brady, "SUSAN – A new approach to low lewel image processing," *International Journal of Computer Vision*, vol. 23, no. 1, pp. 43–78, 1997.

[10] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and colour images," in *Proc. of the 1998 IEEE International Conference on Computer Vision*, Bombay, India, Jan. 1998, pp. 839–846, Narosa Publishing House.

[11] S. Kindermann, S. Osher, and P. W. Jones, "Deblurring and denoising of images by nonlocal functionals," *Multiscale Modeling and Simulation*, vol. 4, no. 4, pp. 1091–1115, 2005.

[12] G. Gilboa and S. Osher, "Nonlocal operators with applications to image processing," Tech. Rep. CAM-07-23, Department of Mathematics, University of California at Los Angeles, CA, U.S.A., 2007.

[13] C. Kervrann and J. Boulanger, "Local adaptivity to variable smoothness for exemplar-based image regularization and representation," *International Journal of Computer Vision*, vol. 79, no. 1, pp. 45–69, Aug. 2008.

[14] M. Mahmoudi and G. Sapiro, "Fast image and video denoising via nonlocal means of similar neighborhoods," *IEEE Signal Processing Letters*, vol. 12, no. 12, pp. 839–842, Dec. 2005.

[15] T. Brox and D. Cremers, "Iterated nonlocal means for texture restoration," in *Scale Space and Variational Methods in Computer Vision*, F. Sgallari, A. Murli, and N. Paragios, Eds. 2007, vol. 4485 of *Lecture Notes in Computer Science*, pp. 13–24, Springer, Berlin.

[16] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[17] S. K. Alexander, E. R. Vrscay, and S. Tsurumi, "A simple, general model for the affine self-similarity of images," in *Image Analysis and Recognition*, A. Campilho and M. Kamel, Eds. 2008, vol. 5112 of *Lecture Notes in Computer Science*, pp. 192–203, Springer, Berlin.

[18] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, vol. 8, pp. 179–187, Feb. 1962.

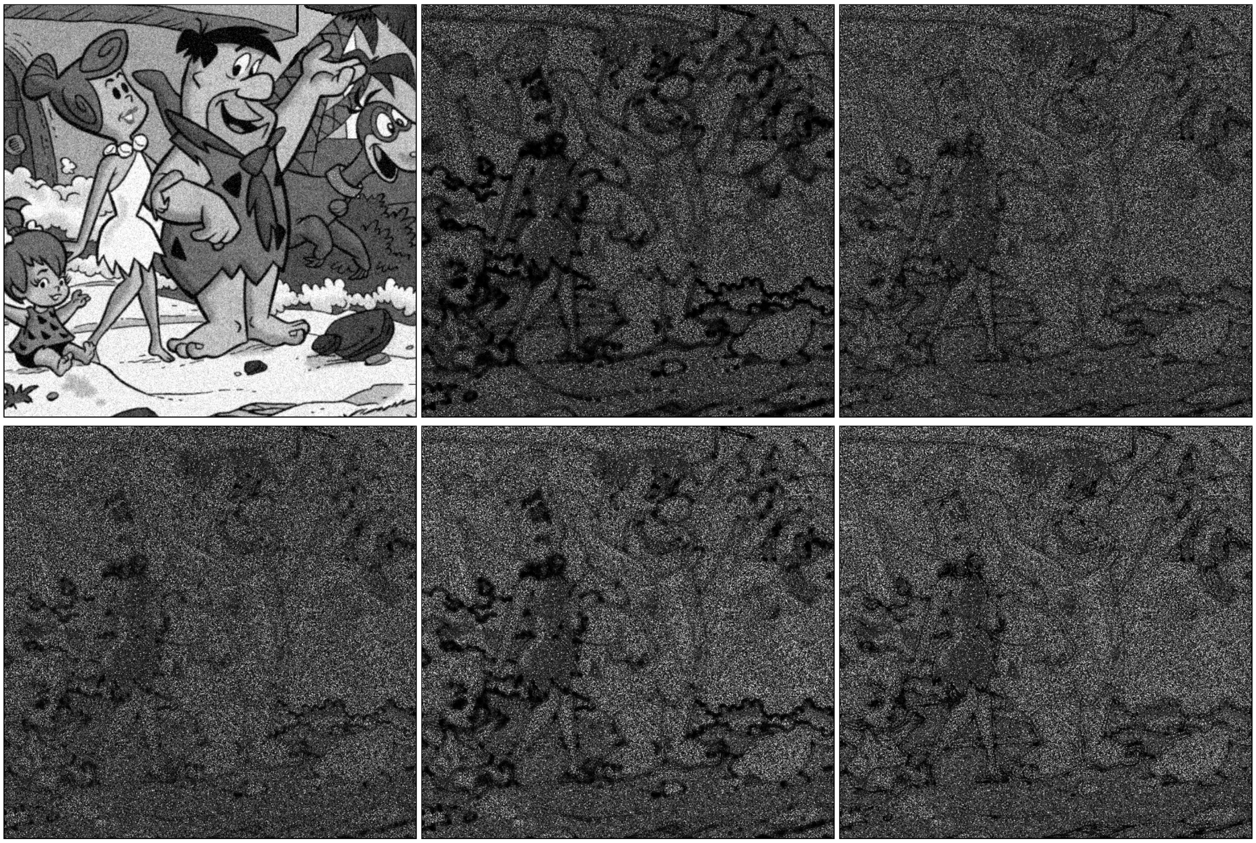

Figure 12. Comparison of absolute method noise for all method variants. **Top Left:** Flintstones (sometimes also referred to as 'Flinstones') with additive Gaussian noise ($\sigma = 20$), size 512×512 pixels. **Top Middle:** AMN for classical NL-means. **Top Right:** NL-means with RIBM. **Bottom Left:** NL-means with local adaptive smoothing parameter $\Lambda(\lambda, B) = \lambda \cdot \sqrt{S_B}$. **Bottom Middle:** NL-means with $G_{\alpha,0}$ as Gaussian kernel. **Bottom Right:** NL-means with all three modifications combined. For MSE see Table 1.

Image Size Noise	**Barbara** 512x512 $\sigma = 20$	**Boats** 512x512 $\sigma = 20$	**Flintstones** 512x512 $\sigma = 20$	**Lena** 512x512 $\sigma = 20$	**House** 256x256 $\sigma = 20$	**Peppers** 256x256 $\sigma = 20$	**Trui** 256x256 $\sigma = 5$	**Trui** 256x256 $\sigma = 20$
Traditional NL-means	72.66	78.68	108.92	48.27	44.83	72.19	11.32	46.47
NL-means with RIBM	93.11	77.21	93.17	47.67	43.66	67.43	9.33	41.53
NL-means with $G_{\alpha,0}$	**67.02**	78.33	103.24	43.74	**38.64**	67.02	10.94	39.55
NL-means with $\Lambda(\lambda, B)$	76.06	**74.77**	**89.20**	46.44	40.96	**66.80**	**8.54**	43.29
NL-means with RIBM, $G_{\alpha,0}$ and $\Lambda(\lambda, B)$	91.21	77.12	97.61	**43.50**	39.91	67.12	9.13	**39.43**
NL-means with $\Lambda(\lambda, A)$ and $G_{\alpha,0}$	91.50	105.49	137.28	54.83	47.56	95.82	9.13	45.85

Table 1. Benchmark results given as MSE. $\Lambda(\lambda, B)$ denotes the local adaptive approach that takes into account the variance of the block, $\Lambda(\lambda, A)$ denotes the approach that uses the minimal distance. As explained in Section 6 the MSE increases for the latter one in most images.