Universität des Saarlandes

Fachrichtung 6.1 – Mathematik

Preprint Nr. 265

Rotationally Invariant Similarity Measures for Nonlocal Image Denoising

Sven Grewenig, Sebastian Zimmer and
Joachim Weickert

Saarbrücken 2010

Fachrichtung 6.1 – Mathematik
Universität des Saarlandes

Preprint No. 265
submitted: July 15, 2010

Rotationally Invariant Similarity Measures for Nonlocal Image Denoising

Sven Grewenig
Mathematical Image Analysis Group
Dept. of Mathematics and Computer Science
Saarland University, Campus E1.1, 66041 Saarbrücken, Germany
grewenig@mia.uni-saarland.de

Sebastian Zimmer
Katholieke Universiteit Leuven
ESAT-PSI/VISICS
Kasteelpark Arenberg 10, Bus 2441, 3001 Leuven-Heverlee, Belgium
sebastian.zimmer@esat.kuleuven.be

Joachim Weickert
Mathematical Image Analysis Group
Dept. of Mathematics and Computer Science
Saarland University, Campus E1.1, 66041 Saarbrücken, Germany
weickert@mia.uni-saarland.de

Edited by
FR 6.1 – Mathematik
Universität des Saarlandes
Postfach 15 11 50
66041 Saarbrücken
Germany

Fax:	+ 49 681 302 4443
e-Mail:	preprint@math.uni-sb.de
WWW:	http://www.math.uni-sb.de/

Abstract

Many natural or texture images contain structures that appear several times in the image. One of the denoising filters that successfully take advantage of such repetitive regions is the nonlocal means filter. It is simple and yields very good denoising results. Unfortunately, the block matching within the standard nonlocal means filter is not able to handle rotation or mirroring. Rotated or mirrored instances are not detected as variations of the corresponding original structures. In this paper, we analyse two natural approaches for a rotationally invariant similarity measure that will be used as an alternative to, respectively a modification of the well-known block matching algorithm in nonlocal means denoising. The first approach is based on similarity distances computed with the help of moment invariants whereas the second one estimates the rotation angle, rotates the block via interpolation and then uses a standard block matching. In contrast to the standard method, the presented algorithms can find similar regions or patches in an image even if they appear in several rotated or mirrored instances. With this modification, the nonlocal means filter is able to find more suitable regions for its weighted average.

1 Introduction and Motivation

One of the most often applied ideas in correspondence problems such as motion analysis [2, 4, 9] are block matching strategies. They are simple and thus still play an important role in many algorithms, for instance in the MPEG video compression standard. In addition to their application to image sequences, they can also be used to detect repeating structures or regions inside an image. Hence, the basic idea of block matching algorithms has also been applied for image processing methods such as inpainting. Exemplar-based algorithms, e.g. [13], fill in missing image data or remove certain image objects by searching for similar regions or structures in the image and completing the missing data or overwriting the data to be removed according to the information found there. A very popular denoising approach that is motivated by the inpainting method of Efros and Leung [15] is the so-called nonlocal means (NL means) algorithm for image denoising that has been proposed by Buades et al. [7, 8]. This denoising filter looks for similar regions and uses them in a weighted average, where the weights depend on the amount of similarity. It belongs to the class of adaptive averaging filters, like the Yaroslavsky [41] or the bilateral filter [3, 35, 38]. In this context, Mrázek et al. [32] and Pizarro et al. [33] have presented unified frameworks that include many denoising methods like bilateral filtering or M-smoothers

[10, 39] as special cases. The difference between NL means and previous approaches is the way of calculating the weights for the averaging process with the consideration of neighbourhood information. Although it is simple, NL means is able to yield high-quality denoising results. Thus, there is an extensive scientific activity in this field: For example, the methodology has been formalised with the help of variational approaches by Kindermann et al. [26] as well as Gilboa and Osher [19, 20, 21]. Tschumperlé and Brun [40] have shown that the patch-based Tikhonov regularisation [37], where they have used an energy formulation in the so-called patch space, is similar to the NL means filter. An interpretation and analysis in the statistical context has been given by Kervrann and Boulanger [25]. Katkovnik et al. [24] have analysed the evolution of the nonparametric regression modelling from local kernel estimate to non-local means and BM3D by Dabov et al. [14]. In order to further improve the quality, Brox and Cremers have proposed an iterative nonlocal means filter and they have suggested not to average over all pixels but only over a certain number of best matches [5]. A similar idea was used by, for example, both Mahmoudi et al. [31] and Coupé et al. [11, 12]. They have proposed some simple and fast preselection criteria in order to exclude pixels which are not necessary for the filtering process. These methods can decrease the run time of the filtering process as well as yield improvement of the denoising quality at the same time. In the context of speed-up techniques, Brox et al. have presented a fast method that uses cluster trees [6]. Another speed-up approach has been introduced by Liu et al. [28]. They have presented a strategy that integrates both NL means and the Laplacian pyramid, where they show that their method is about fifty times faster than the classical NL means algorithm and can outperform it with respect to numerical results.

However, all the mentioned models do not take into account the invariance under rotations and mirroring. In the context of nonlocal means, it makes sense also to involve pixels in the averaging process which belong to a neighbourhood that differs from the reference patch only by rotation or mirroring. Some models that consider invariances have been presented for example by Alexander et al. [1], Kleinschmidt et al. [27] and Lou et al. [29]. Alexander et al. have proposed a general model for the affine self-similarity of images, whereas the classical NL means algorithm only considers self-similarity in the translational sense. With the help of this general model one could incorporate invariance under affine greyscale transformations within the patch-similarity measure. Kleinschmidt et al. have analysed different types of invariances such as brightness, rotation and scale invariance. In order to incorporate rotational invariance, they have proposed to consider not only the original neighbourhoods but also some rotated versions (e.g. by 90°, 180°

and 270°) in the patch comparison. Lou et al. have suggested to use a similarity-invariant descriptor that is similar to SIFT (scale-invariant feature transform) [30] for the matching procedure. Furthermore, Ji et al. [23] have introduced a moment-based approach that only uses low-order Zernike invariants [36].

The goal of this paper is to present and analyse two different similarity measures that are invariant under arbitrary rotations and mirroring. Our first approach uses a nonlocal means filter whose similarity measure is based on moment invariants. The second rotationally invariant method relies on the rotationally invariant block matching (RIBM), which has already been presented in our conference paper [42]. In addition to this, we present a novel modification of RIBM, which uses the structure tensor for the orientation estimation and can further improve the denoising results.

Our paper is organised as follows: In the next section, we will give a short introduction to NL means and some possible improvements concerning the implementation. Section 3 contains a short overview of moment invariants and explains the corresponding modified NL means filter using these invariants. After this, we will present the RIBM approach and show how the structure tensor can be used in order to further improve this approach. Section 5 examines the performance of NL means using the rotationally invariant similarity measures and the influence of the parameters. In Section 6, we conclude the paper with a summary and discuss some proposals for future work.

2 Nonlocal Means

In this section, we give a short introduction to the nonlocal means filter [7, 8]. Let $f : \Omega \to \mathbb{R}$ be a (noisy) greyscale image defined on a bounded domain $\Omega \subset \mathbb{R}^2$ and let $x \in \Omega$. The filtered image $u : \Omega \to \mathbb{R}$ at the point x is then computed by

$$u(x) = \frac{1}{C(x)} \int\limits_{\Omega} e^{-\frac{(G_a * |f(x+\cdot)-f(y+\cdot)|^2)(0)}{\lambda^2}} f(y)\, dy\ , \tag{1}$$

where $G_a(x) = \exp\left(\frac{-|x|^2}{2a^2}\right)$ is a Gaussian with standard deviation a, λ a smoothing parameter and

$$C(x) = \int\limits_{\Omega} e^{-\frac{(G_a * |f(x+\cdot)-f(y+\cdot)|^2)(0)}{\lambda^2}}\, dy \tag{2}$$

the normalisation factor. The filter parameter λ steers the decay of the weights. If it is too large, almost every pixel of the input image f will get a weight close too $(C(x))^{-1}$. Hence, the filter works approximately as a mean filter and the filtered image will be too smooth. For a very small parameter λ almost all weights are close too zero and there is basically no smoothing. Buades et al. have suggested that λ should be dependent on the amount of the noise and should be selected from the interval $\left[\sqrt{10\sigma_n}, \sqrt{15\sigma_n}\,\right]$, where σ_n is the (estimated) standard deviation of the Gaussian noise contained in the image f.

One can easily see that the similarity or distance measure of the filter is given by

$$\begin{aligned} dist_f(x,y) &= \left(G_a * |f(x+\cdot) - f(y+\cdot)|^2\right)(0) \\ &= \int\limits_{\mathbb{R}^2} G_a(t) \cdot |f(x+t) - f(y+t)|^2 \, dt \ , \end{aligned} \tag{3}$$

with the corresponding weights

$$w(x,y) := \exp\left(-\frac{dist_f(x,y)}{\lambda^2}\right) \ . \tag{4}$$

In practical applications, G_a is in fact a truncated Gaussian

$$\tilde{G}_a(t) := \begin{cases} L^{-1} \cdot G_a(t) & , \ |t| \leq r \\ 0 & , \ |t| > r \end{cases} \tag{5}$$

where $r > 0$ is the radius of the circular patch and $L := \int_{|t| \leq r} G_a(t)\, dt$ the normalisation constant.

For $a \to 0$ we get $dist_f(x,y) = |f(x) - f(y)|^2$. Thus, nonlocal means can be seen as a generalisation of the Yaroslavsky neighbourhood filter [41]. Instead of comparing only two grey values, it uses a weighted distance measure in order to find similar regions in the image. The idea behind this approach is that if two patches are similar, then their central pixel should have a similar meaning for the image and thus similar grey values. Therefore it makes sense to average them. The Gaussian weighting in the distance measuring excludes high influence of noisy pixels that are farther away from the centre pixel of the patch. For example, an approach by Coupé et al. [12] does not use this weighting and there is just a division by the number of pixels within the current processed patch. We will later use this idea for our moment distance measure.

If we consider the distance measure in (3), we see that it has a severe drawback: It is not invariant under any transformation such as rotations or mirroring. There is no reason why patches that are rotated around their centre or mirrored should have a larger distance than the ones that are not rotated or mirrored. A distance measure taking into account some invariances can find more pixels in the image with a similar neighbourhood structure. Instead of a similarity measure based on Euclidean distances we could use, for instance, a comparison of moment invariants. In this case, we do not need to access each pixel of a patch everytime we compute a distance of two blocks. Hence, the use of moment invariants can save a lot of computational effort.

2.1 Search Window

Since the complexity of the algorithm is $\mathcal{O}(|P| \cdot N^2)$, where $|P|$ is the number of pixels within a patch P and N is the total number of pixels, one replaces Ω by a so-called search window around the pixel x, $SW(x)$, in Eq. (1). Then the algorithm should search for similar regions only within this search window. This lowers the complexity to $\mathcal{O}(|P| \cdot N \cdot |SW|)$. Besides the advantage of a reduced running time, the use of a search window can actually improve the denoising results: It excludes some pixels which could have a negative influence on the averaging process and thus degrade the image quality. We call them bad pixels. If one considers an edge for example, then by doubling the radius, the new search window is four times larger (thus the computation lasts longer) and hence the number of bad pixels is quadrupled, whereas the number of good matches according to the edge is only twice as large. While a search window is usually a square around the pixel to be denoised, we use a disc to make the whole algorithm rotationally invariant.

2.2 Image Boundary Condition

At the image boundaries we use reflecting boundary conditions, in order to have full circular patches around those pixels near the boundaries. This approach avoids implementation problems regarding the comparison of patches around pixels close to the image boundary. Pixels within a boundary patch that are located outside the image are not used in the weighted average, i.e. search windows which are crossing the image boundaries will be reduced, so that they only contain pixels within the image.

2.3 Weighting of the Centre Patch

Considering Equation (1), it is obvious that the grey value $f(x)$ always gets the highest weight $w(x, x) = \exp(0) = 1$. If the search window or the whole image only contains patches $P(y)$ that are not similar to $P(x)$, the corresponding weights of the values $f(y)$ will be near zero. Thus, there is basically no filtering. In this case, one has to increase the filter parameter λ in order to see some smoothing effect. However, for $\lambda \gg 0$ all grey values get almost the same weight, which means that the similarity measure is more or less useless. It has been suggested to set the centre weight $w(x,x) := \max_{y \neq x} w(x,y)$ to overcome the problem. We will also use this solution in our implementation of the moment-based approach. For the block matching, we introduce a different solution: As the total distance between two blocks can be decomposed into the distance of the (noise free) structure plus the noise itself, it is obvious that even blocks with the exact same structure will have an expected distance of $2 \cdot \sigma_n^2$, where σ_n is the standard deviation of the additive white Gaussian noise (for a more detailed analysis of the effect of noise on patch-based similarities, we refer to [1]). A block that is compared to itself will not just match the structure, but also the noise, and thus, it will have a distance of zero. In order to compensate for the resulting high weight of the central patch, we replace its distance for the weighting with the expected error of the noise, i.e. $2 \cdot \sigma_n^2$. As there might be patches with a distance less than this value, i.e. with a weight that is larger than the weight of the central patch, we will also replace their distances by the same value. With this modification, we are not only able to achieve better PSNR results than with the previously mentioned modification, but we even achieve these results at significantly smaller block sizes and search windows, resulting in a reduced running time (see Sec. 5).

3 Nonlocal Means with Moment Invariants

As we mentioned in the last section, we use moment invariants to find similar patches in the image f. Moment invariants are a classical statistical tool for object or pattern recognition. They have been introduced by Hu using the theory of algebraic invariants [22]. Hu has derived his famous seven moment invariants that are invariant under translation, rotation and scaling. Six of these moments are also invariant under mirroring, while the seventh moment switches its sign under mirroring, i.e. its absolute value is also invariant. About twenty years later, Teague used moments based on sets of orthogonal basis functions such as Zernike polynomials [36]. They are invariant under translation, rotation, mirroring and scaling, too. Because of their orthogonality, these invariants do not contain any redundant information, which is better for image reconstruction. Furthermore, Teague has shown that Zernike moments of second and third orders are equivalent to the ones of Hu, but can be constructed to arbitrarily high order. Here, equivalence means that there is an algebraical coherence between them, but they are not equal, since Zernike moments are computed in a different way (see [36] for more details). In this paper, we use Zernike moments up to order six, which means a total number of 24. Another type of moments, which are invariant under affine transformations, has been introduced by Flusser and Suk [17]. They have computed four moments explicitly. Besides, Flusser has analysed the independence of rotation moment invariants in another publication [16]. He has shown that the invariants derived by Hu as well as Teague are dependent and has presented two bases for both third- and fourth-order invariants (total number: 11) that we use in this paper. There are also rotation moment invariants that switch their sign under mirroring and thus we have to take the absolute values in the corresponding cases. A lot more publications concerning this important area are of course available, but they are not be used in this paper.

All the above mentioned moments have in common that they can be expressed in terms of usual central moments

$$\nu_{p,q} = \int\limits_{\mathbb{R}} \int\limits_{\mathbb{R}} (x - x_c)^p (y - y_c)^q f(x,y)\, dx\, dy \ , \tag{6}$$

where (x_c, y_c) are the coordinates of the centroid of f and $p + q \geq 0$ the order with $p, q \geq 0$. The centroid can be computed via

$$\begin{pmatrix} x_c \\ y_c \end{pmatrix} = \frac{1}{m_{0,0}} \begin{pmatrix} m_{1,0} \\ m_{0,1} \end{pmatrix} \tag{7}$$

with

$$m_{k,l} = \int\limits_{\mathbb{R}} \int\limits_{\mathbb{R}} x^k y^l f(x,y)\, dx\, dy \ . \tag{8}$$

Setting

$$\mu_{p,q} := \frac{\nu_{p,q}}{\nu_{0,0}^{1+\frac{p+q}{2}}} \tag{9}$$

yields scaling invariance. In this paper, we use the following notation: Φ_p describes a Hu, S_p a Zernike, I_p an affine and Ψ_j a rotation moment invariant. Whenever we talk about the maximum number of moments or moment orders, this refers to the numbers mentioned above. For further details we recommend the respective cited references.

3.1 Modification of the Filter

Now we want to modify the NL-Means filter using a rotationally invariant moment distance measure instead of the standard block matching. Let $\mathcal{M} =$

$\{M_i \mid i = 1, ..., K\}$ be a set of moments that are invariant under mirroring as well as rotation, and $x, y \in \Omega$. We denote the patches inside the image f with the centres x and y by $P(x)$ and $P(y)$ respectively. The moment distance measure is then defined via

$$dist_{\mathcal{M},f}(x,y) := K^{-1} \cdot \sum_{i=1}^{K} \big[M_i(P(x)) - M_i(P(y))\big]^2 \,, \tag{10}$$

where only the grey values inside the corresponding patch $P(\cdot)$ are used in order to compute M_i. It is also necessary to compute the coordinates of the patch's centroid, since we use central moments. For this computation, we only have to change the domain of integration and replace it by the patch $P(\cdot)$ in (8). The pixels within a patch are addressed with a coordinate system that has its origin at the patch's centre. Note that we do not use any weighting inside the moment comparison but a division by the number of moment invariants. The corresponding filter is then given by

$$u(x) = \frac{1}{D(x)} \int\limits_{\Omega} e^{-\frac{dist_{\mathcal{M},f}(x,y)}{\lambda^2}} f(y)\, dy \,, \tag{11}$$

with

$$D(x) = \int\limits_{\Omega} e^{-\frac{dist_{\mathcal{M},f}(x,y)}{\lambda^2}}\, dy \,. \tag{12}$$

Because of the computational effort, Ω will be replaced by a search window $SW(x)$ around the point x, as before in the usual framework.

3.2 Normalisation of the Moment Invariants

The first problem that arises with this definition are the different magnitudes of the invariants. Because of that, it is very difficult to find a good filter parameter λ, and the moments with larger magnitudes are more important within the similarity measure. One solution to this is to perform a normalisation of the values. For example, Ji et al. [23] have suggested a normalisation technique for Zernike invariants. Unfortunately, this technique is not applicable for other invariants. Hence, we propose a normalisation technique that is suitable for all types of invariants. If we consider each moment of the set $\mathcal{M}$ as a random variable, we can use a normalisation such that the mean of the corresponding normalised random variable is 0 and the standard deviation is equal to 1. Thus, we have to compute the mean values

$$\delta_i = \frac{1}{|\Omega|} \int\limits_{\Omega} M_i(P(x))\, dx \tag{13}$$

and variances

$$\sigma_i^2 = \frac{1}{|\Omega|} \int\limits_{\Omega} \left(M_i(P(x)) - \delta_i\right)^2 \, dx \tag{14}$$

for $i = 1, ..., K$.
Setting

$$\tilde{M}_i(P(x)) := \frac{M_i(P(x)) - \delta_i}{\sigma_i} \tag{15}$$

yields the required normalisation. However, even in this case it is not so easy to find a suitable global filter parameter λ as we will see in the experimental section.

4 Rotationally Invariant Block Matching

In this section, we want to discuss another modification of the block matching algorithm that is invariant under rotation and mirroring. First, we present the basic idea of the algorithm in a generic way, followed by a specific description of our implementation. Then we discuss stability issues for some special cases. Finally, we present a novel second implementation, addressing these issues using the so-called structure tensor.

4.1 Basic Idea

The main problem is the estimation of the rotation angle between two corresponding patches. Once this angle is determined, the rest is easy: If we identify all pixels of a block as vectors from the block's centre, all these vectors are rotated by the same angle, thus the pixel coordinates of one block can easily be transformed into pixel coordinates of another block. If one block is a mirrored version of the other one (we have seen a feature capable of detecting this in Section 3, namely Φ_7), we simply mirror one of the blocks at an arbitrary axis before the estimation of the rotation angle. The choice of the mirroring axis introduces another rotation, but as a series of rotations can be described as one single rotation and we estimate the angle of that rotation after we have mirrored the block, the axis of rotation can be chosen arbitrarily.
Adding these steps to the traditional block matching algorithm, we can describe the *rotationally invariant block matching (RIBM)* algorithm as follows:

1. If the second patch is a mirrored version of the first patch, mirror the second patch at an arbitrary axis.

2. Estimate the angle of rotation between the patches.
3. For each pixel in the first patch: Find the position of the corresponding pixel in the second patch by rotating by this estimated angle.
4. Compute the grey value distance between these two pixels.

The summed distances represent the total distance of the two patches. Note that for rotations that are not multiples of 90°, the rotated pixel coordinates might not be integers, thus one would have to apply interpolation.

4.2 Implementation Details

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 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:

$$c_B \quad := \quad \begin{pmatrix} \frac{\int_B x \cdot f(x,y) \quad dx \quad dy}{\int_B f(x,y) \quad dx \quad dy} \\ \\ \frac{\int_B y \cdot f(x,y) \quad dx \quad dy}{\int_B f(x,y) \quad dx \quad dy} \end{pmatrix} . \tag{16}$$

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 (Φ_7, see Section 3) to compute m. Φ_7 is known to be invariant under many transformations such as rotation, but changes its sign under mirroring. While the numerical value of Φ_7 suffers a lot from discretisation and noise, its sign remains quite stable. Our strategy to compensate for mirroring is then given as:

$$m_{B,B'}(v) \quad := \quad \begin{cases} (-v_1, v_2)^\top, & \Phi_7(B') \cdot \Phi_7(B) < 0 \\ \\ (v_1, v_2)^\top, & \text{else} . \end{cases} \tag{17}$$

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

$$\begin{aligned} R_{B,B'} &:= R^{-1}_{\vec{c}_B} \cdot R_{m_{B,B'}(\vec{c}_{B'})} \quad \text{with} && (18) \\ R_v &:= \begin{pmatrix} v_1 & -v_2 \\ v_2 & v_1 \end{pmatrix} . && (19) \end{aligned}$$

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 (20).
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:

$$p_{B'} \;:=\; m_{B,B'}(R_{B,B'} \cdot p_B) \; . \tag{20}$$

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

$$d(B,B') \;:=\; \int\limits_B \left(f_B(p_B) - f_{B'}(p_{B'})\right)^2 \, dp_B \; . \tag{21}$$

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:

$$d(B,B') \;:=\; \sum_{p_B \in B} \left(f_B(p_B) - I(f_{B'}, p_{B'})\right)^2 \; , \tag{22}$$

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 (3).

4.3 RIBM with Structure Tensor

In our previous implementation of the RIBM algorithm, we used the centroid of the blocks to estimate their orientation. While this approach works well for edges, where the centroid is far away from the centre of the block, and homogeneous areas, where a correct estimation of the orientation is not needed, fine structures in an image can cause problems for the algorithm: If the centroid is close to the centre of the block, noise has a strong influence on the estimated orientation of the block. This can lead to incorrect estimations of the rotation angle between two blocks. Especially for parallel lines, this RIBM implementation can lead to higher distances than traditional block matching and thus causing problems when denoising images like the Barbara test image. This motivates a modification of the RIBM implementation, using the so-called structure tensor for a more stable estimation of the orientation of the blocks.

The Structure Tensor

Let $u_\sigma := G_\sigma * u$ be a Gaussian filtered image with the standard deviation $\sigma > 0$. The matrix J_0 resulting from the dyadic product

$$J_0\left(\nabla u_\sigma\right) := \nabla u_\sigma \otimes \nabla u_\sigma := \nabla u_\sigma \nabla {u_\sigma}^T \tag{23}$$

is symmetric and hence has an orthonormal basis of eigenvectors v_1, v_2 with $v_1 \parallel \nabla u_\sigma$ and $v_2 \perp \nabla u_\sigma$. The corresponding eigenvalues $|\nabla u_\sigma|^2$ and 0 then indicate the contrast in the eigendirections. If we convolve $J_0\left(\nabla u_\sigma\right)$ component-wise with a Gaussian G_ρ ($\rho \geq 0$), we achieve the so-called structure tensor [18]

$$J_\rho\left(\nabla u_\sigma\right) := G_\rho * \left(\nabla u_\sigma \otimes \nabla u_\sigma\right)\ . \tag{24}$$

With the help of the structure tensor one can identify image features such as edges and corners or measure the local coherence of structures. This tensor notation enables an averaging for robustness while avoiding the cancellation effect that can be observed on directive derivatives on thin lines. As a drawback, structure tensors only reveal the orientation of these averaged derivatives, not their direction. This means that all orientation information extracted solely from structure tensors will always be limited to the interval $[0°, 180°)$. While this is fine for analysing lines, for a rotationally invariant block matching algorithm one needs the complete 360° spectrum. To achieve this, we combine the structure tensor with the centroid from (16).

Estimating Orientation and Direction

Let $v_{\sigma,\rho,B}$ be the dominant eigenvector (i.e. the one that corresponds to the larger eigenvalue) of the structure tensor $J_\rho\left(\nabla u_\sigma\right)$ at the centre of block B. The modified centroid $c_{\sigma,\rho,B}$ is then defined as

$$c_{\sigma,\rho,B} := \begin{cases} v_{\sigma,\rho,B}, & v_{\sigma,\rho,B} \cdot c_B \geq 0 \\ -v_{\sigma,\rho,B}, & \text{else} \end{cases}, \tag{25}$$

where c_B is the centroid from (16). Our new centroid will point into the same direction as the dominant eigenvector of the structure tensor, unless the old centroid points into an opposing direction, in which case our new centroid will switch its direction to the opposite direction. This is our only modification of the previously presented implementation, all other implementation details remain the same. As this modification only affects the estimation of the blocks' orientation, only the precomputation step needs to be modified. The actual computation of the distances remains unaffected. We will refer to this implementation as RIBM+ST.

5 Experiments and Discussion

Now we want to analyse both modified nonlocal means filters. First, we will study the properties of the two different similarity measures, in particular the robustness with respect to noise and quantisation. After this, we take a look at the dependency of the denoising quality using moment invariants on the different filter parameters like for instance the patch radius or the smoothing parameter λ. Finally we present some results with different well-known test images and compare them to the performance of the classical filter.

5.1 Analysis of the Similarity Measures

Our test image is synthetic and consists of several concentric circles (see Fig. 1). We have picked four points that are on the same line and computed their corresponding invariant values. It is obvious that the circular block around point 3 corresponds to point 1, which is simply rotated by 90°. Points 1 and 2 are rotated by 55° respectively 305°, and the rotation angle between 2 and 4 is exactly 180°. These four pixels have the same distance to the centre of the ring and the same grey value, but the grey values of the other pixels in the corresponding blocks are partially affected by discretisation (see histograms).

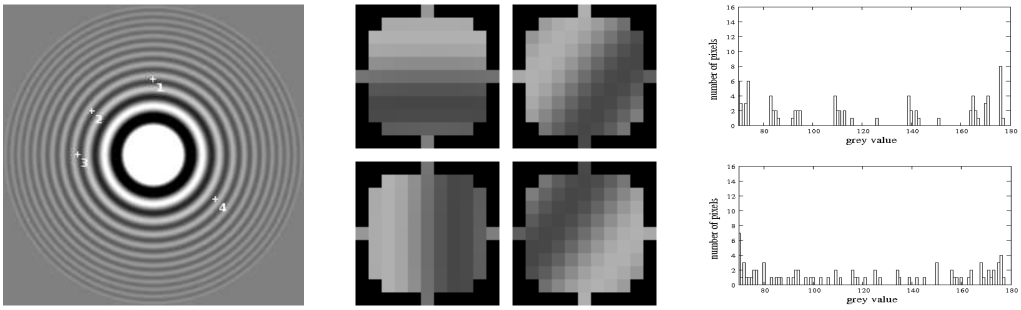

Figure 1: *Ring* image (256×256) with chosen patches of radius 5 (top: 1 and 2, bottom: 3 and 4) and corresponding histograms (top: 1 and 3, bottom: 2 and 4).

Table 1: Values of some invariants for the chosen four patches of the *Ring* image (see Fig. 1).

	without noise				with Gaussian Noise (σ_n = 20)			
moment	P1	P2	P3	P4	P1	P2	P3	P4
Φ_1 (10^{-3})	1.312	1.317	1.312	1.317	1.328	1.321	1.291	1.299
$\|\Phi_7\|$ (10^{-22})	0.102	3.806	0.102	3.806	15.069	7.942	4.085	9.659
S_7 (10^{-3})	0.986	9.661	0.986	9.661	0.184	9.484	2.173	8.243
S_{14} (10^{-1})	1.840	1.824	1.840	1.824	2.608	1.815	1.607	2.127
I_1 (10^{-7})	4.306	4.333	4.306	4.333	4.408	4.360	4.157	4.219
I_2 (10^{-23})	0.369	2.123	0.369	2.123	4.557	2.261	2.781	3.943
$\|\Psi_4\|$ (10^{-16})	0.112	2.254	0.112	2.254	3.717	1.688	14.600	4.172
Ψ_7 (10^{-6})	2.430	2.442	2.430	2.442	2.554	2.472	2.363	2.386

Rotational Invariance

For our first experiment, we compute some moment values of the four blocks, to examine their behaviour under rotation and their robustness against noise. The corresponding results are given in Table 1. At first one can state that the magnitude of the invariant values varies strongly. For example, the values concerning the seven Hu moments vary from 10^{-23} to 10^{-3}. This shows that a normalisation of the values is necessary. Otherwise the Hu invariant with the largest magnitude, i.e. Φ_1, would affect the similarity measure too much. Considering the rotations in the *Ring* image, we notice that the rotations with an angle which is a multiple of 90° do not change the values of the moment invariants in contrast to the rotation from point 1 to point 2, which is not a multiple of 90°. The different results can be explained because of the quantisation: The moment invariants suffer a lot from discretisation when used on small blocks and therefore fail to correctly determine similar blocks that are not rotated by a multiple of 90°. One can realise this by taking a look at the corresponding histograms in Fig. 1. There are clear differences between the histograms of the patches around

point 1 and 2 respectively, whereas the ones of the 90° or 180° rotated patches (see point 3 as well as 4) are the same like the original ones. The different distributions of the grey values cause varying invariant values. Moments with large differences regarding the four points are probably not suitable for replacing the traditional block matching algorithm as a similarity measure, because they are too sensitive with respect to grey value changes.
Another property that the moment invariants should have is the robustness against noise. As one can see in the table, there are some moments that seem to be robust against noise, for instance Φ_1, I_1 and Ψ_7. Furthermore, these moments are more robust w.r.t. discretisation like some other ones. But then the question arises whether this kind of moments can distinguish between different structures or these positive properties only appear due to a small range of values. This is a general problem, since we map a patch with its grey values to a certain small set of moment invariants. Thus, we want to conduct another experiment with the *Ring* image, where we analyse the question about the distinction quality of the invariants.

Distinction Quality

For a certain pixel x_i, we compute its distance to the centre of the *Ring* image x_{ct}, $\|x_i - x_{ct}\|_2$, determine all other pixels x_j with the same distance $\|x_j - x_{ct}\|_2 = \|x_i - x_{ct}\|_2$ and put them into the set D_i, which can be described formally as

$$D_i = D(x_i) := \{x_j \in \Omega \ : \|x_j - x_{ct}\|_2 = \|x_i - x_{ct}\|_2\} \ , \tag{26}$$

where Ω is now the discrete image domain. Thus, all pixels that are contained in D_i lie on the same circular line. In the *Ring* image, this means that D_i contains only pixels whose patches have the same (rotated) structure. Therefore, the invariant values of these patches should be close to each other, whereas the values of the patches with centre pixels in $\Omega \backslash D_i$ should differ from the ones corresponding to D_i. Moment invariants which fulfil this criterion are supposed to be more suitable for block matching than those not fulfilling it. In order to measure whether this criterion is satisfied or not, we compute the errors corresponding to a certain moment invariant M, $err_{D_i}(M)$ and $err_{\Omega\backslash D_i}(M)$, which are defined as follows:

$$err_{D_i}(M) := \sum_{x_j \in D_i} \frac{|M(P(x_i)) - M(P(x_j))|^2}{|D_i| \cdot \|M\|_2^2} \ , \tag{27}$$

$$err_{\Omega\backslash D_i}(M) := \sum_{x_j \in \Omega\backslash D_i} \frac{|M(P(x_i)) - M(P(x_j))|^2}{|\Omega \setminus D_i| \cdot \|M\|_2^2} \ . \tag{28}$$

Table 2: Errors for the moment invariants without tolerance ($\epsilon = 0$)

	without noise			with noise ($\sigma_n = 20$)		
moment	err_D	$err_{\Omega\setminus D}$	ratio	err_D	$err_{\Omega\setminus D}$	ratio
Φ_1	0.000767	1.071771	1397.1	0.004455	0.623925	140.1
Φ_2	0.001595	1.993607	1249.7	0.132561	1.990833	15.0
Φ_3	0.006581	1.992047	302.7	0.402197	1.993152	5.0
Φ_4	0.008366	1.992066	238.1	0.252834	1.988379	7.9
Φ_5	0.026097	1.995035	76.4	0.699715	1.997657	2.9
Φ_6	0.013739	1.994172	145.1	0.427508	1.994175	4.7
$\lvert\Phi_7\rvert$	0.005239	1.996823	381.2	0.845450	1.997832	2.4
S_1	0.001577	1.956041	1240.5	0.032198	1.869204	58.1
S_2	0.006543	1.946672	297.5	0.075690	1.823775	24.1
S_3	0.022882	1.975499	86.3	0.235632	1.838179	7.8
S_4	0.018737	1.970727	105.2	0.082933	1.881217	22.7
S_5	0.050776	1.993239	39.3	0.321234	1.986130	6.2
S_6	0.030365	1.992604	65.6	0.202782	1.989394	9.8
S_7	0.037918	1.949748	51.4	0.262653	1.424564	5.4
S_8	0.011277	1.988116	176.3	0.189972	1.953793	10.3
S_9	0.003688	1.617448	438.6	0.036999	1.182605	32.0
S_{10}	0.041024	1.992125	48.6	0.420063	1.985427	4.7
S_{11}	0.032620	1.993632	61.1	0.148535	1.995452	13.4
S_{12}	0.046101	1.990330	43.2	0.375246	1.965343	5.2
S_{13}	0.023402	1.990611	85.1	0.289389	1.981010	6.8
S_{14}	0.000234	1.791076	7643.0	0.003250	1.774528	545.9
S_{15}	0.001596	1.692187	1060.6	0.015294	1.680024	109.8
S_{16}	0.025135	1.989600	79.2	0.270997	1.974686	7.3
S_{17}	0.111277	1.995003	17.9	0.527136	1.995643	3.8
S_{18}	0.059881	1.992490	33.3	0.368404	1.979815	5.4
S_{19}	0.024127	1.991486	82.5	0.346992	1.981832	5.7
S_{20}	0.010333	1.991126	192.7	0.260280	1.978523	7.6
S_{21}	0.002863	1.920790	670.9	0.073390	1.666896	22.7
S_{22}	0.087258	1.993547	22.8	0.426938	1.995113	4.7
S_{23}	0.029268	1.988618	67.9	0.347652	1.935488	5.6
S_{24}	0.047659	1.991790	41.8	0.189565	1.990915	10.5
I_1	0.014401	1.948644	135.3	0.040183	1.876575	46.7
I_2	0.003120	2.000065	641.1	0.900546	1.996928	2.2
I_3	0.006243	1.998401	320.1	0.440133	1.996002	4.5
I_4	0.014013	1.996543	142.5	0.346886	1.998093	5.8
$\lvert\Psi_4\rvert$	0.011161	1.994103	178.7	0.648539	1.994031	3.1
Ψ_7	0.003351	1.973931	589.0	0.041258	1.906581	46.2
Ψ_8	0.018651	1.993722	106.9	0.525741	1.994946	3.8
$\lvert\Psi_9\rvert$	0.025580	1.994893	78.0	0.862040	1.995477	2.3
Ψ_{10}	0.035747	1.995526	55.8	0.881787	1.998914	2.3
$\lvert\Psi_{11}\rvert$	0.141864	1.996207	14.1	0.867091	1.998207	2.3

In (27) and (28) respectively, $|S|$ denotes the number of elements of an arbitrary set S. $\|M\|_2^2$ is the averaged squared L_2-norm of the invariant M and is computed by

$$\|M\|_2^2 = |\Omega|^{-1} \cdot \sum_{x_k \in \Omega} |M(P(x_k))|^2 \ . \tag{29}$$

It acts like a normalisation factor for the errors. We evaluate both (27) and (28) for all $x_i \in \Omega$ and compute the average errors err_D as well as $err_{\Omega\setminus D}$,

$$err_D(M) := |\Omega|^{-1} \sum_{x_i \in \Omega} err_{D_i}(M) \ , \tag{30}$$

$$err_{\Omega\setminus D}(M) := |\Omega|^{-1} \sum_{x_i \in \Omega} err_{\Omega\setminus D_i}(M) \ . \tag{31}$$

Since we work in a discrete setting, the set D_i does not contain many elements (8 on average). An alternative would be to incorporate a tolerance $\epsilon \geq 0$ and define

$$D_i^\epsilon := \{x_j \in \Omega \ : \|x_i - x_{ct}\|_2 + \epsilon \geq \|x_j - x_{ct}\|_2 \geq \|x_i - x_{ct}\|_2 - \epsilon\} \ . \tag{32}$$

Table 3: Errors for the moment invariants with tolerance $\epsilon = 0.5$

	without noise			with noise ($\sigma_n = 20$)		
moment	err_D	$err_{\Omega \setminus D}$	ratio	err_D	$err_{\Omega \setminus D}$	ratio
Φ_1	0.082402	1.077236	13.1	0.026619	0.627183	23.6
Φ_2	0.376665	2.003054	5.3	0.472403	2.000015	4.2
Φ_3	0.711698	2.000514	2.8	1.150339	2.000319	1.7
Φ_4	0.712459	2.000525	2.8	0.731308	1.996787	2.7
Φ_5	1.153363	2.002163	1.7	1.648489	2.003359	1.2
Φ_6	0.877332	2.002140	2.3	1.153767	2.001349	1.7
$\lvert\Phi_7\rvert$	1.394602	2.003184	1.4	1.688530	2.003423	1.2
S_1	0.087500	1.966765	22.5	0.104921	1.879651	17.9
S_2	0.330235	1.956114	5.9	0.280004	1.833045	6.5
S_3	0.601691	1.984219	3.3	0.687189	1.846133	2.7
S_4	0.573631	1.979541	3.5	0.293239	1.890712	6.4
S_5	0.928122	2.001026	2.2	0.934425	1.993916	2.1
S_6	0.773505	2.000878	2.6	0.706833	1.997904	2.8
S_7	0.529373	1.958518	3.7	0.760855	1.429623	1.9
S_8	0.553450	1.997011	3.6	0.494497	1.962813	4.0
S_9	0.230662	1.625486	7.0	0.106059	1.188821	11.2
S_{10}	0.775116	2.000350	2.6	1.103330	1.992710	1.8
S_{11}	0.482551	2.002795	4.2	0.439852	2.004811	4.6
S_{12}	0.733169	1.998694	2.7	1.081507	1.972594	1.8
S_{13}	0.646837	1.999228	3.1	0.852967	1.989028	2.3
S_{14}	0.015304	1.801236	117.7	0.022166	1.784637	80.5
S_{15}	0.059598	1.702540	28.6	0.055950	1.690010	30.2
S_{16}	0.617065	1.998313	3.2	0.802092	1.982849	2.5
S_{17}	1.157300	2.002095	1.7	1.421186	2.001984	1.4
S_{18}	0.880753	2.000414	2.3	1.121688	1.987016	1.8
S_{19}	0.731901	1.999844	2.7	1.034297	1.989300	1.9
S_{20}	0.676662	1.999643	3.0	0.731378	1.986904	2.7
S_{21}	0.378349	1.929922	5.1	0.219844	1.675382	7.6
S_{22}	0.949811	2.001256	2.1	1.429254	2.001422	1.4
S_{23}	0.675596	1.997138	3.0	1.028239	1.942719	1.9
S_{24}	0.732722	2.000167	2.7	0.654839	1.999577	3.1
I_1	0.278169	1.958195	7.0	0.189803	1.886024	9.9
I_2	1.964911	2.004696	1.0	1.588055	2.002818	1.3
I_3	1.439082	2.004661	1.4	1.428229	2.002323	1.4
I_4	1.264295	2.003337	1.6	1.628869	2.003814	1.2
$\lvert\Psi_4\rvert$	1.016542	2.001624	2.0	1.228530	2.001006	1.6
Ψ_7	0.157738	1.983988	12.6	0.153560	1.916310	12.5
Ψ_8	0.897197	2.001628	2.2	1.291680	2.001709	1.5
$\lvert\Psi_9\rvert$	0.993382	2.002470	2.0	1.361751	2.002061	1.5
Ψ_{10}	1.239383	2.002397	1.6	1.826050	2.004101	1.1
$\lvert\Psi_{11}\rvert$	1.340237	2.002763	1.5	1.714281	2.003714	1.2

The results with $\epsilon \in \{0, 0.5\}$ are listed in both Table 2 and 3. We have computed the errors for the original image and with additive Gaussian noise ($\sigma_n = 20$) in order to see how the noise influences the errors. A comparison between the results with $\epsilon = 0$ and $\epsilon = 0.5$ gives us some information about the effect of the discretisation. The average number of elements contained in $D_i^{0.5}$ is about 504 which is 63 times the number without any tolerance. The most interesting case is then of course the error with noise and $\epsilon = 0.5$.
At first we analyse the seven moments of Hu. Regarding the error err_D, the first Hu moment Φ_1 seems to work very well. On the other hand the error $err_{\Omega \setminus D}$ is very low compared to the other invariants, which means that Φ_1 is stable but can not well distinguish between different structures. However, it has also the largest ratios, which are computed by $err_{\Omega \setminus D}/err_D$. Introducing a tolerance can increase err_D with a factor up to about 100. Only the increase concerning Φ_2 and Φ_7 is significantly larger (about 250). Adding Gaussian noise creates an interesting observation in the case of $\epsilon = 0.5$: err_D is decreasing with a factor about three, i.e. the distribution of invariant

values of the pixels with the same distance to the image centre seems to be more dense. Also the ratio becomes better.
Concerning the Zernike moments, the best ratios appear for the fifth order moment S_{14} and they are also the best among all moments within Table 2 and 3. S_{14} is for instance better than the first Hu moment Φ_1 regarding both err_D and $err_{\Omega\setminus D}$. Besides this, it is more robust under noise and less sensitive with respect to the quantisation. The second best regarding all depicted invariants is S_{15}. It also has good values for the ratio and seems to be less sensitive with respect to noise and discretisation compared to S_{14}. Regarding the case with tolerance $\epsilon = 0.5$ and additive Gaussian noise, there are two further moments with a ratio larger than 10, namely the first order moment S_1 and the fourth order moment S_9.
If we take a look at the errors of the affine moment invariants, we can see that I_1 has the best overall performance among them. Without any noise and tolerance, the other three moments yield better results, but are very sensitive with respect to noise and quantisation. In the case of tolerance $\epsilon = 0.5$, I_1 has a similar behaviour as Φ_1: err_D also decreases with a factor about two and the ratio becomes better, when noise is added.
Finally we consider the independent rotation moment invariants, except for the ones that are equal to the Hu moments. Ψ_7 is the best moment concerning the error ratios and seems to be very robust under noise and discretisation compared to its related invariants that do not yield convincing results, since the ratios decrease enormously.

We also conduct this experiment with RIBM. In this case, the errors are computed by

$$err_{D_i} := \sum_{x_j \in D_i} \frac{d\big(P(x_i), P(x_j)\big)}{|D_i|} \,, \tag{33}$$

$$err_{\Omega\setminus D_i} := \sum_{x_j \in \Omega\setminus D_i} \frac{d\big(P(x_i), P(x_j)\big)}{|\Omega \setminus D_i|} \,, \tag{34}$$

where $d(\cdot\,,\,\cdot)$ is the corresponding rotationally invariant block distance measure. The average errors err_D and $err_{\Omega\setminus D}$ as well as the ratio are similarly defined as above. Our parameter setting contains a patch radius of size five, the standard deviation $a = 3/\sqrt{2}$ of the inner Gaussian and for the structure tensor we use $\sigma = 0.5$ as well as $\rho = 2$. The results are illustrated in Table 4. We can see that the RIBM strategy outperforms the weighted L_2-distance with respect to this experiment. If we employ RIBM together with the structure tensor, the results are even better, which is due to the appearance of fine structures in the *Ring* image. The corresponding similar-

Table 4: Errors for RIBM.

(a) without tolerance

	without noise			with noise ($\sigma_n = 20$)		
method	err_D	$err_{\Omega\setminus D}$	ratio	err_D	$err_{\Omega\setminus D}$	ratio
NL means	561.17	3364.78	6.0	904.04	3908.13	4.3
RIBM	49.71	3125.22	62.9	376.08	3526.44	9.4
RIBM with ST	35.54	3131.61	88.1	315.12	3467.62	11.0

(b) with tolerance $\epsilon = 0.5$

	without noise			with noise ($\sigma_n = 20$)		
method	err_D	$err_{\Omega\setminus D}$	ratio	err_D	$err_{\Omega\setminus D}$	ratio
NL means	1128.94	3378.18	3.0	1858.92	3920.42	2.1
RIBM	121.92	3141.94	25.8	800.37	3541.54	4.4
RIBM with ST	80.63	3148.70	39.1	663.55	3483.37	5.2

ity measure is more stable with respect to the quantisation, since the values only get approximately halved by introducing a tolerance $\epsilon = 0.5$. For the moment invariants the decrease is much larger. If we add white Gaussian noise with $\sigma_n = 20$, the ratio of the two average errors decreases for both tolerances with a factor around seven in the case of RIBM. With $\epsilon = 0.5$, the moment-based similarity measure seems to have less problems concerning the introduction of noise. Hence, there are some moments with better error ratios.

Distribution of the Weights

At this point we want to discuss the distribution of the weights which is influenced by the smoothing parameter λ. The larger the parameter the larger the weights will be, i.e. the filtered image will be smoother. If it is too large, important image structures could be destroyed. On the other hand, if one chooses a value that is too small, only a few pixels are a significant part of the weighted average. Hence, the noise is not removed in some regions of the image. For this reason it is very important to choose a filter parameter that allows the removal of noise and at the same time the preservation of important image structures like edges, corners etc. In the case of original NL means, Buades et al. have already stated that the parameter should depend on the amount of noise and they have made a suggestion about the value. The use of moments within the distance measure makes it more difficult to find a good global λ, which is illustrated in Fig. 2. We take the *Ring* image and pick three patches of radius five with different contrasts. When we add some noise, it is observable that the structure concerning the patch around point 1 is almost destroyed, whereas the other patches are less affected. The patch whose centre pixel is point 3 is almost in its original state. In order to show the dependence between the filter parameter and the

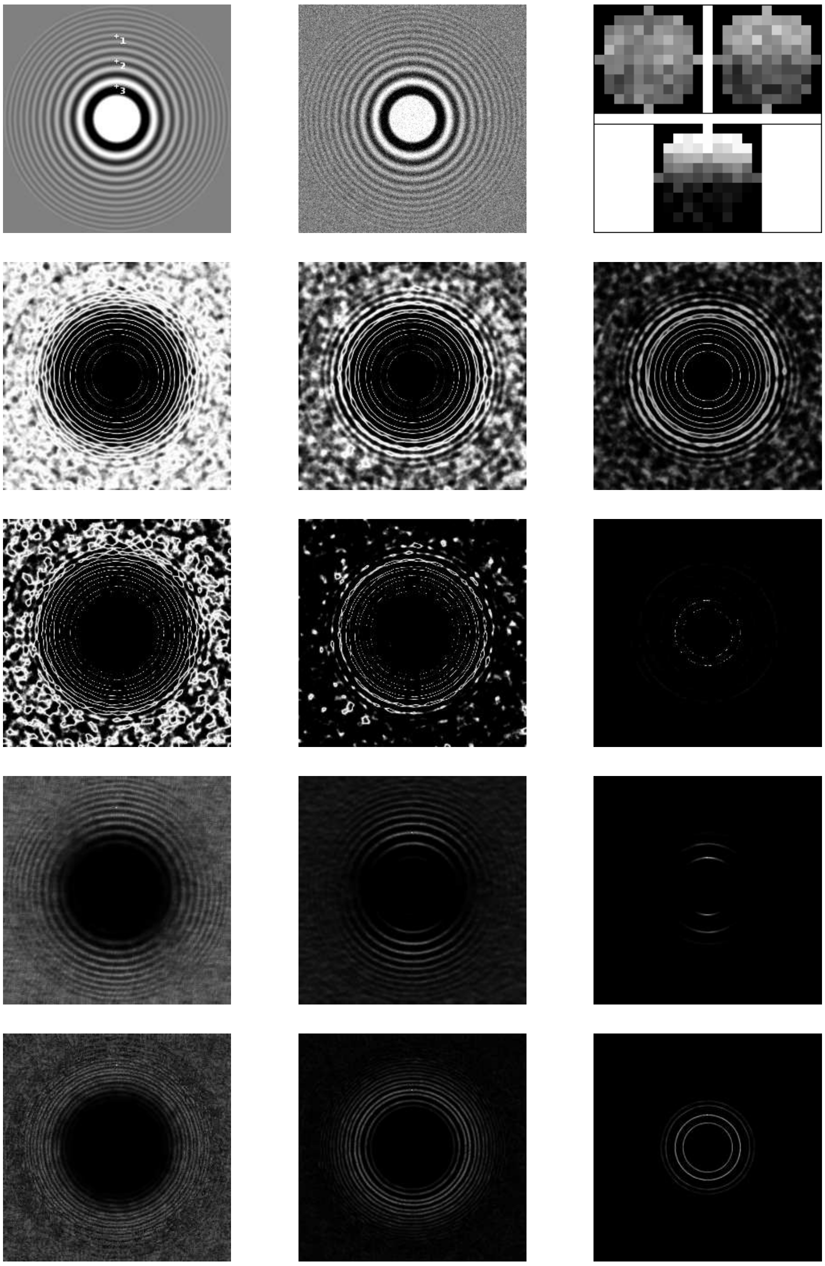

Figure 2: Visualisation of the weights for different methods. **First row**: *Ring* with three chosen points, image with Gaussian noise ($\sigma_n = 20$) and the three noisy patches. **Second row**: Weights using a moment-based similarity measure (Hu moments) for the noisy image with $\lambda^2 = 10^{-3}$ concerning the points $1, 2$ and 3. **Third row**: Weights with $\lambda^2 = 10^{-4}$. **Fourth row**: Weights using standard block matching with $\lambda^2 = 1000$. **Fifth row**: Weights using RIBM with $\lambda^2 = 600$.

distribution of the weights, we work with two different parameters, $\lambda^2 = 10^{-3}$ and $\lambda^2 = 10^{-4}$. Instead of a small search window, we use the whole image for the illustration. The intensity of a pixel's grey value stands for the magnitude of its corresponding weight. Bright grey values correspond to large weights. We have used a distance measure using the seven Hu moments (normalised) for this experiment.

The second row of Fig. 2 shows the distribution of the weights using $\lambda^2 = 10^{-3}$. It is obvious that this value is too large, since both pixel 1 and pixel 2 get too many pixels having large weights despite the different structures within their neighbourhoods. Even pixel 3 gets a lot of bad pixels. One possible solution for getting rid of this problem might be a smaller filter parameter. The third row shows the results for $\lambda^2 = 10^{-4}$. There are still a lot of bad pixels that will be used for denoising the first point. The result for the second point looks much better, since a bigger part of bad pixels has less influence. For the third pixel we get a quite good result where bad pixels are hardly visible and only relevant pixels can be seen. Even if the whole *Ring* image is less smooth with $\lambda^2 = 10^{-4}$, it is still blurred near the boundary of the *Ring* (see point 1) because of the multitude of bad pixels. These are the first indications for the difficulty of finding a suitable value for λ.

The last two rows show the result for the standard block matching and RIBM (without tensor) with relatively large filter parameters $\lambda^2 = 1000$ and $\lambda^2 = 600$ respectively. These values might be too large for the corresponding filtering process, but are used here for visualisation purposes. One can see that even such large values allow only a limited influence of bad pixels for all three points. Thus, the methods using the weighted block matching distance seem to be more stable than the moment-based similarity measure. Furthermore one can notice the tolerance of standard block matching against slight rotations, whereas the RIBM yields circular lines despite the noise.

The simplest solution for avoiding the problem with bad pixels is the use of a small search window instead of the whole image, which is used in practice in order to save CPU time. This restricts the number of bad pixels and might yield a higher quality of the denoised image. However, the proportion of bad pixels could still be too large in some cases.

5.2 Filter Parameters

This subsection discusses the influence and significance of the parameters for the moment-based approach. We would like to see which settings are appropriate for this modified filter and thus will discuss all the filter parameters (patch radius, search window radius and λ) that are used. Since RIBM is related to the standard block matching used in NL means and Buades et al.

have already discussed the parameters, for example in [8], we do not discuss them here again. In order to measure the quality of the filtered images, we will use the peak signal-to-noise ratio (PSNR). It is computed by

$$PSNR(f,g) = 10 \cdot log_{10}\left(\frac{255^2 \cdot |\Omega|}{\sum\limits_{x_i \in \Omega} |f(x_i) - g(x_i)|^2}\right) , \tag{35}$$

with the noisy image g and its denoised version f. The higher the PSNR, the better is the quality of the denoised image. In our experiments, we work with normalised moments (see 3.2) and use a non-iterated filter.

Patch Radius

The first parameter we want to analyse is the patch radius. This parameter steers the size of the circular neighbourhood around a pixel, in which the moments are computed. Thus, it has an important influence on the moment distance measure. In our experiments, the patch radius varies from 2 to 10. The search window radius is always 10 and the filter parameter λ is adapted to the used type of moment invariants, but it is fixed for each curve. Furthermore, we always use the maximum number of moments or moment orders for each of the four types. Figure 3 shows the relation between the patch radius and the PSNR for the *Ring* image. Whereas Hu's, the affine and the rotation moment invariants seem to prefer a small patch radius like 2 or 3, the Zernike moments need a larger radius equal to 5 or greater. If the radius is too large, the result gets worse, because smaller structures are more blurred. Because of the large radius, the smaller structures within the corresponding large patch have less influence on the similarity measure. Thus, there are a lot of patches that do not contain these structures, but have a higher weight in the averaging process.

Search Window Radius

The search window radius is another important parameter. It defines those pixels that are used in the weighted average. Using a search window instead of the whole image has two main advantages: The algorithm is much faster and further bad pixels outside of the search window that might have a negative influence on the weighted average are excluded. In our experiments, we keep the other parameters fixed for each curve (adopted to the type of moments and the test image) and change the search window radius from 5 to 30 with step size 5. As before, the maximum number of moments or moment orders is used within each type. The results are depicted in Figure 4. In all cases,

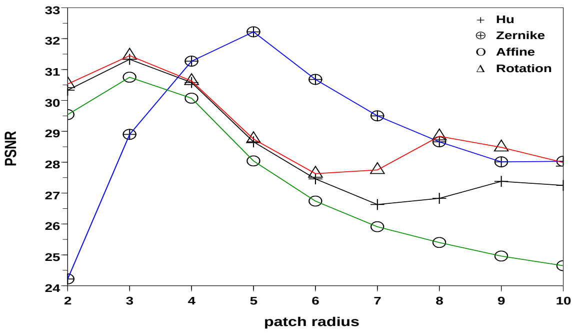

Figure 3: Coherence between the patch radius and the PSNR. The search window radius is 10 in all cases and λ is optimised according to the particular type of invariants.

a larger search window yields worse results, in particular for the Zernike moments. Best results can be obtained with a radius between 5 and 10. The absolute loss concerning the PSNR for the other three invariant types, going from 10 to 30, is around 1. Overall, the recommendation is to use small search windows.

Smoothing Parameter λ

At last, we analyse the λ-parameter. It is responsible for the distribution of the weights as we have mentioned above. In Fig. 5, one can see the behaviour of the PSNR regarding the value of λ^2, where the x-axis is logarithmically scaled. The patch radius is fixed for each curve, but is changed according to the invariants, because for example Zernike moments need a larger radius than the other types. Concerning the number of moment invariants or orders, we always choose the maximum for each type. Although the moment invariants are normalised, they have different optimal λ^2-values, as we can observe in Fig. 5. The course of the curves in both images looks similar. Affine moment invariants require the smallest value in order to reach the best PSNR within the corresponding parameter setting. Both the Hu invariants and the independent rotation moments need almost the same value. This is not really surprising, because five Hu moments belong to this independent set. The largest parameter belongs to the set of Zernike moments.
We will later see that there is another problem with this parameter: In contrast to block-based NL means, it heavily depends on the image. If, however, one considers the original NL means filter, good results for different images can be achieved with a fixed λ.

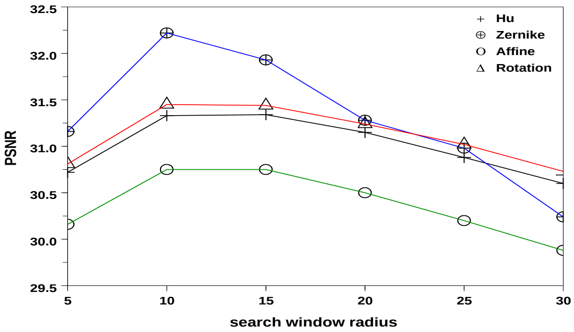

Figure 4: Coherence between the search window radius and the PSNR. The patch radius and λ vary adapted to the particular type of invariants.

5.3 Running Time

One of the main reasons for introducing a moment-based similarity measure besides the rotational invariance is the running time. If we consider for instance a patch with radius 5 that contains 81 pixels, then the standard block matching needs 81 comparisons in order to compute the distance. The moment-based approach replaces the set of grey values by a smaller set of invariant values. Thus, the computation of the distance is much faster and the similarity measure is rotationally invariant at the same time. The only two parameters with a significant influence on the running time are both the patch and the search window radius. If we choose for example a parameter setting including a circular patch with radius 3 and a search window radius of size 5, then the running times for the filters using block matching methods for a 512 × 512 test image are 23 seconds with classical block matching (NL means), 31 seconds with RIBM and 33 seconds for the combination with the structure tensor ($\sigma = 0.5, \rho = 2$) on a standard desktop PC (Pentium 4, 3.2 GHz). This means that the running time for a filter using RIBM is only about 30% larger and is for this reason very fast in contrast to a method which uses fixed rotation angles within the patch comparison (see for example [27]). Depending on the machine and the choice of parameters, the increase of running time varies from 10% to 40%, but never exceeds this upper limit in our experiments. Even the integration of the structure tensor does not affect the running time too much, i.e. increases it additionally by less than 10%. We emphasise that the computation of the structure tensor is only part of the precomputation step and not optimised. Considering the moment-based approach using all Hu invariants and the above mentioned radii, the running time is only about 7 seconds and thus three times faster than standard nonlocal means. Even the method with the use of 24 Zernike

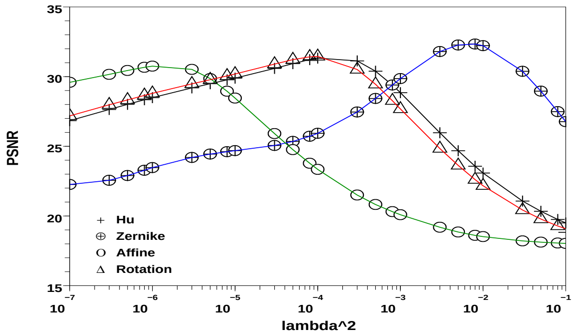

Figure 5: Coherence between λ^2 and the PSNR. The patch radius varies adapted to the particular type of invariants and the search window radius is always 10.

moment invariants only needs about 9 seconds. For a parameter setting with a patch radius of size 5 and a search window with radius 10, this method is finished in 36 seconds, whereas the running times of the other filters are about 194, 288 and 316 seconds, i.e. up to nine times slower. Overall we can state that the moment-based approach is able to save a lot of CPU time and the RIBM methods are much faster than any method using several fixed rotation angles.

5.4 Denoising Quality and Comparison

Now, we want to analyse the denoising quality of our modified NL means filters and compare them to the classical filter. To this end, we use well-known test images (see Fig. 8 and Fig. 10) in addition to the *Ring* image. As we have mentioned in Sec. 2.3, we use some special modifications for the weighting of the centre patch. The evaluation of the denoising results is presented in Table 5. Some of the results are illustrated in the figures 8, 9, 10 and 11. We have tried different parameter settings for each image and each type of moment invariants and only show the best result concerning these settings.

The filters based on block matching strategies have only one parameter setting that is described in the table.

Moment-Based Method

At first glance one can see that the classical NL means filter with only one parameter setting yields better results for all images except for *Ring* and *Trui* with optimised parameter settings. This already shows one problem: It

Table 5: Denoising results (PSNR) of the moment-based approach and comparison to classical NL means (NLM) as well as RIBM, where Gaussian noise with $\sigma_n = 20$ is used. The parameters for the filter using the moment-based distance measure are optimised, whereas in the case of the other three filters only one parameter setting is used for all images: $a = 3/\sqrt{2}$, $\lambda^2 = 225$, patch radius 3, search window radius 5 and for the structure tensor $\sigma = 0.5$, $\rho = 2$.

Image	**Barbara**	**Boats**	**Lena**	**House**	**Peppers**	**Ring**	**Trui**
NLM with Hu	25.14	28.52	30.51	30.63	28.86	31.49	32.00
patch radius	2	2	3	2	2	3	3
sw radius	7	7	5	8	4	9	6
λ^2	$5 \cdot 10^{-3}$	$5 \cdot 10^{-3}$	$2.5 \cdot 10^{-2}$	$5 \cdot 10^{-2}$	$1.5 \cdot 10^{-2}$	10^{-3}	$3.5 \cdot 10^{-2}$
moments	Φ_1	Φ_1	Φ_1	Φ_1	Φ_1	Φ_1	Φ_1
NLM with Zernike	26.62	28.24	30.43	30.86	28.40	**32.48**	30.05
patch radius	4	4	5	7	4	5	7
sw radius	5	4	4	7	4	13	6
λ^2	$5 \cdot 10^{-2}$	$7 \cdot 10^{-2}$	$2.5 \cdot 10^{-1}$	$6 \cdot 10^{-2}$	$4 \cdot 10^{-2}$	10^{-2}	$1.5 \cdot 10^{-1}$
moments	$S_1 - S_{24}$	$S_1 - S_{17}$	$S_1 - S_{11}$	$S_1 - S_{11}$	$S_1 - S_{24}$	$S_1, S_2,$ $S_7 - S_{17}$	$S_1 - S_{24}$
NLM with Affine	24.90	28.13	30.02	30.19	28.21	30.84	31.46
patch radius	2	2	3	3	2	3	3
sw radius	12	10	7	7	4	10	12
λ^2	$5 \cdot 10^{-4}$	$7 \cdot 10^{-5}$	$5 \cdot 10^{-3}$	$5 \cdot 10^{-3}$	$6 \cdot 10^{-4}$	$6 \cdot 10^{-6}$	$5 \cdot 10^{-3}$
moments	I_1	I_1, I_2	I_1	I_1	I_1	I_1	I_1
NLM with Ind. Rot.	24.93	28.48	29.92	30.53	28.82	31.48	30.20
patch radius	2	2	2	2	2	3	2
sw radius	7	7	6	9	4	9	3
λ^2	10^{-3}	$9 \cdot 10^{-4}$	10^{-2}	$8 \cdot 10^{-3}$	$2 \cdot 10^{-3}$	$2 \cdot 10^{-4}$	$1.5 \cdot 10^{-1}$
moments	$\Psi_1 - \Psi_6$	$\Psi_1 - \Psi_6$	$\Psi_1 - \Psi_6$	$\Psi_1 - \Psi_6$	$\Psi_1 - \Psi_6$	$\Psi_1 - \Psi_6$	$\Psi_1 - \Psi_6$
NLM	29.67	29.58	31.56	31.89	30.21	30.54	31.96
RIBM	28.93	29.53	31.66	32.06	30.43	29.65	**32.53**
RIBM+ST	**29.73**	**29.71**	**31.84**	**32.17**	**30.51**	30.29	32.47

is difficult to find a single parameter setting that gives reasonable results in all cases in contrast to the standard filter, and even optimised results can be worse, in particular the performance on *Barbara* decreases significantly. The necessity of very different parameter settings shows that this simple moment-based method is difficult to use in practice. If we compute the average PSNR for each type of moment invariants, then the invariants of Hu and the Zernike moments have the best values, 29.59 and 29.58. The optimised results are depicted in Fig. 9 and Fig. 11. One can see, in particular for *Trui*, that the method using Zernike moments can have problems at image edges and the one with Hu moment invariants might have problems in homogenous areas (e.g. black hair in *Trui*).

The other two types, rotation moments with 29.19 and affine invariants with 29.11, are inferior. For NL means, the average value is better, i.e. 30.77. The parameter setting for nonlocal means is of course not optimal, but it yields even with a small patch and search window radius reasonable results in addition to a better running time. These results can be seen in Fig. 8 as well as Fig. 10.

In the case of the Hu, the affine as well as the independent rotation moment invariants one can also see that the best results are obtained with the use of only one low-order moment (Φ_1, I_1) and low-order basis $(\Psi_1, ..., \Psi_6)$ respectively. For affine moments, only *Boats* needs the integration of two moments, i.e. I_1 and I_2. Concerning the Zernike moments, it is visible that incorporating higher orders is necessary in order to get a better performance. The result for the *Ring* image shows that a moment-based approach can be useful for denoising synthetic images with many rotationally invariant structures.

Another conclusion that has been already treated in Fig. 3 concerns the size of the patch radius. It should be small, i.e. around 2 or 3, in all cases except for the Zernike ones that allow larger radii up to 7. As we have observed in Fig. 4, smaller search window radii are better, and we can also see this in Table 5. The values for the filter parameter λ do not only depend on the different types of moments, but also on the different images. This makes it very difficult to find a good global value for λ, which is very important with respect to practical applications.

Finally we would like to combine all treated types of moment invariants in order to denoise the *Ring* image. For this experiment, we consider Table 3 that shows the results with tolerance $\epsilon = 0.5$ and additive white Gaussian noise. We take those moments that have a ratio larger than or equal to 9.9 (minimum value that allows the integration of all types) in this case, i.e. Φ_1, S_1, S_9, S_{14}, S_{15}, I_1 and Ψ_7.

The main problem with the combination of different types are the varying optimal values of λ. Our first experiment considers only one λ-parameter. In this case, our best result has got a PSNR value of 32.37, where $\lambda^2 = 10^{-2}$, the patch radius is 5 and the search window radius 14. We can see that the parameter setting is similar to the one of Table 5 with Zernike moments. This might be explained by the fact that four of the seven chosen moment invariants are Zernike ones. The second experiment covers four different λ-values, i.e. one for each invariant type. Here, our best result yields a PSNR of 32.48 with the same patch radius as before, search window radius 15 and $\lambda_\Phi^2 = 10^{-2}$, $\lambda_I^2 = 1.25 \cdot 10^{-4}$, $\lambda_\Psi^2 = 10^{-3}$, $\lambda_S^2 = 1.25 \cdot 10^{-2}$, but it is still not better than the best result in Table 5.

Compared to the approach of Ji et al. [23], our PSNR values with respect to Zernike moments are up to 3.5 dB (*Barbara*) smaller. Therefore, we used their invariant setting and normalisation technique, but we were not able to reproduce the reported results of Ji et al.

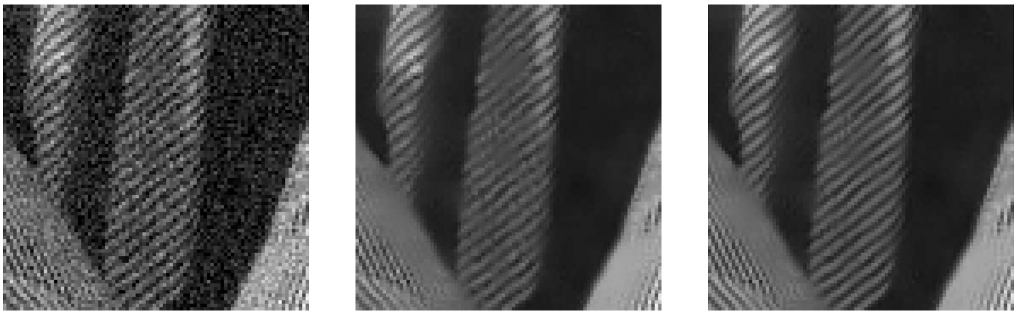

Figure 6: Segment of *Barbara* (90 × 90). **Left**: Noisy segment ($\sigma_n = 20$). **Middle**: Filtered using RIBM. **Right**: Filtered using RIBM with structure tensor.

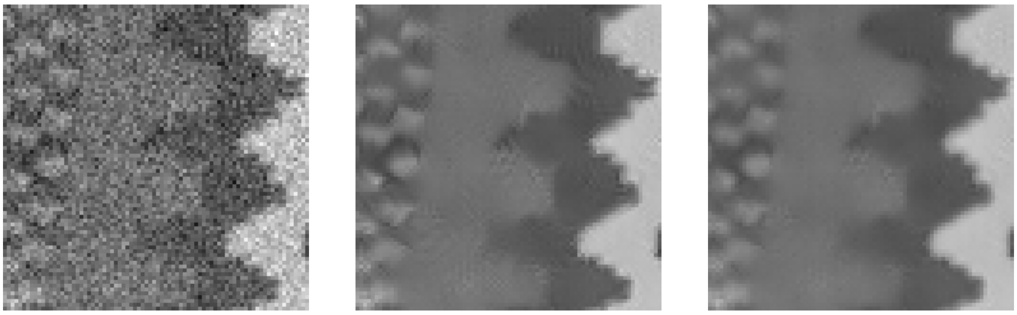

Figure 7: Segment of *Trui* (80 × 80). **Left**: Noisy segment ($\sigma_n = 20$). **Middle**: Filtered using NL means. **Right**: Filtered using RIBM.

RIBM

In contrast to the moment-based method, RIBM is able to achieve good results with only one parameter setting for all images (see Table 5) and is hence more comparable to NL means. This setting contains only small radii in order to have a better running time. The overall performance of the RIBM filter is clearly better than the moment-based one. It can outperform NL means in many cases, especially on images with lots of edges and corners, like in the *Trui* test image (see Fig. 7). The additional use of the structure tensor improves all results except for the *Trui*, where the PSNR slightly degrades. This is due to the fact, that the smoothing of the structure tensor hinders the correct estimation of the orientation at sharp corners. But for Barbara and *Ring*, the incorporation of the structure tensor yields a significant improvement compared to the RIBM algorithm. In both images, the original RIBM has problems with the parallel lines as one can see in Fig. 6, where a small segment of the *Barbara* image containing parallel lines is depicted. It is not really surprising that RIBM achieves worse results than original NL means: Standard block matching is tolerant against small rotations (see Fig. 2). As we use small search windows, we have only small rotations in the *Ring* image and basically no rotations at all in the *Barbara* image. Thus, RIBM cannot benefit from its rotational invariance, while at the same time it suffers from its instability on blocks with fine lines. As both test images contain a lot of such patches, RIBM+ST can achieve more stable results on these images. The average PSNR values for the RIBM methods are 30.68 without and 30.96 with the structure tensor. The denoising results with RIBM+ST can be found in Fig. 9 and 11. As we have mentioned above, the average value for NL means is 30.77. The worse value of RIBM without the structure tensor is mainly due to the result for the *Barbara* image.

6 Conclusion and Future Work

In this paper, we have presented two different ideas for a rotationally invariant similarity measure that is used for image denoising with the NL means algorithm. Our first approach replaces the set of grey values within a patch, which is usually used in block matching algorithms, by a set of moment invariants for the patch comparison. The advantage of such a procedure is that the number of the set of moment invariants can be much smaller than the number of pixels within a patch and hence it is able to greatly reduce the CPU time, though it allows for some invariances. Furthermore, a moment-based approach avoids costly methods which do not only consider the original patch but also some rotated versions with fixed angles, where maybe some additional interpolation is necessary.

The main disadvantage of a rotationally invariant similarity measure using moment invariants is that the invariants have different magnitudes. Without any normalisation, the moments with a relative large magnitude will dominate the similarity measure and significantly attenuate the influence of invariants with relative small magnitudes. In order to suppress this problem, we have used a simple normalisation technique that can be used for arbitrary types of invariants. Due to its simplicity this normalisation might not be optimal for all moment invariants.

Another problem that goes along with the different magnitudes is the choice of the smoothing parameter λ. Despite the normalisation it is difficult to find an appropriate value, since λ seems to depend not only on the type of invariants but also on the image. Hence, the filter is less attractive for practical applications.

Our second approach, RIBM, reverts to classical block matching ideas and even serves as a generalisation of them. A second implementation, i.e. RIBM+ST, was presented to improve the stability of the algorithm that causes problems in some test images. The implementations of the presented algorithm only add a small overhead to the running time, as opposed to methods that simply try different rotations. We have shown that this modification can improve the results of the NL means algorithm even on small search windows, where natural image usually only contain small rotations. A natural question for future work would be, how this effect develops on larger search windows. Unfortunately, larger search windows in general decrease the PSNR performance of NL means, as the exponential weighting function is not capable of perfectly separating good matches form bad matches. Thus, better results are usually achieved on small search windows. A modified weighting function might be able to better separate good pixels from bad pixels. With such a modified weighting, the influence of rotationally invariant similarity measures might be even stronger. Another modification to RIBM+ST could be the incorporation of Higher-Order Structure Tensors [34]. This might avoid the drawback of the use of a smoothed structure tensor at sharp edges, like in the *Trui* image. Another very interesting open question is whether one can also benefit from RIBM in the context of other denoising methods such as collaborative filtering [14] or other image processing applications like for example inpainting [15]. Concerning the moment-based approach, our future work will concentrate on better normalisation techniques and adaptive strategies for the smoothing parameter in order to improve its practical use and attenuate the sensitivity with respect to λ. With the help of such strategies it is likely that one can improve the quality of the denoising results.

Acknowledgements

The first author gratefully acknowledges funding by the Deutsche Forschungsgemeinschaft (DFG), project WE 2602/7-1.

References

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

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

[3] V. Aurich and J. Weule. Non-linear Gaussian filters performing edge preservating diffusion. In G. Sagerer, S. Posch, and F. Kummert, editors, *Mustererkennung*, Informatik Aktuell, pages 538–545. Springer, 1995.

[4] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.

[5] T. Brox and D. Cremers. Iterated nonlocal means for texture restoration. In F. Sgallari, A. Murli, and N. Paragios, editors, *Proc. of the 1st International Conference on Scale Space and Variational Methods in Computer Vision*, volume 4485 of *Lecture Notes in Computer Science*, pages 13–24. Springer, Berlin, 2007.

[6] T. Brox, O. Kleinschmidt, and D. Cremers. Efficient nonlocal means for denoising of textural patterns. *IEEE Transactions on Image Processing*, 17(7):1083–1092, July 2008.

[7] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Proc. of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 60–65, 2005.

[8] A. Buades, B. Coll, and J.-M. Morel. A review of image denoising algorithms, with a new one. *SIAM Multiscale Modeling and Simulation*, 4(2):490–530, 2005.

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

[10] C. K. Chu, I. K. Glad, F. Godtliebsen, and J. S. Marron. Edge-preserving smoothers for image processing. *Journal of the American Statistical Association*, 93(442):526–541, 1998.

[11] P. Coupé, P. Yger, and C. Barillot. Fast non local means denoising for 3D MR images. In R. Larsen, M. Nielsen, and J. Sporring, editors, *MICCAI 2006*, volume 4191 of *Lecture Notes in Computer Science*, pages 33–40. Springer, Berlin Heidelberg, 2006.

[12] P. Coupé, P. Yger, S. Prima, P. Hellier, C. Kervrann, and C. Barillot. An optimized blockwise nonlocal means denoising filter for 3-D magnetic resonance images. *IEEE Transactions on Medical Imaging*, 27(4):425–441, April 2008.

[13] A. Criminisi, P. Pérez, and K. Toyama. Object removal by exemplar-based inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 721–728, 2003.

[14] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, August 2007.

[15] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the International Conference on Computer Vision (ICCV 1999)*, volume 2, pages 1033–1038, 1999.

[16] J. Flusser. On the independence of rotation moment invariants. *Pattern Recognition*, 33:1405–1410, 2000.

[17] J. Flusser and T. Suk. Pattern recognition by affine moment invariants. *Pattern Recognition*, 26(1):167–174, 1993.

[18] W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. of the ISPRS Intercommission Workshop on Fast Processing of Photogrammetric Data*, pages 281–305, 1987.

[19] G. Gilboa, J. Darbon, S. Osher, and T. F. Chan. Nonlocal convex functionals for image regularization. Technical Report CAM-06-57, Department of Mathematics, University of California at Los Angeles, CA, U.S.A., 2006.

[20] G. Gilboa and S. Osher. Nonlocal linear image regularization and supervised segmentation. *SIAM Multiscale Modeling and Simulation*, 6(2):595–630, 2007.

[21] G. Gilboa and S. Osher. Nonlocal operators with applications to image processing. *SIAM Multiscale Modeling and Simulation*, 7(3):1005–1028, 2008.

[22] M. K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8:179–187, February 1962.

[23] Z. Ji, Q. Chen, Q.-S. Sun, and D.-S. Xia. A moment-based nonlocal-means algorithm for image denoising. *Information Processing Letters*, 109(23-24):1238–1244, 2009.

[24] V. Katkovnik, A. Foi, K. Egiazarian, and J. Astola. From local kernel to nonlocal multiple-model image denoising. *International Journal of Computer Vision*, 86(1):1–32, 2010.

[25] C. Kervrann and J. Boulanger. Local adaptivity to variable smoothness for exemplar-based image regularization and representation. *International Journal of Computer Vision*, 79(1):45–69, 2008.

[26] S. Kindermann, S. Osher, and P. W. Jones. Deblurring and denoising of images by nonlocal functionals. *SIAM Multiscale Modeling and Simulation*, 4(4):1091–1115, 2005.

[27] O. Kleinschmidt, T. Brox, and D. Cremers. Nonlocal texture filtering with efficient tree structures and invariant patch similarity measures. In A. Foi and A. Gotchev, editors, *Proc. International Workshop on Local and Non-Local Approximation in Image Processing*, pages 103–113, 2008.

[28] Y.-L. Liu, J. Wang, X. Chen, and Y.-W. Guo. A robust and fast nonlocal means algorithm for image denoising. *Journal of Computer Science and Technology*, 23(2):270–279, 2008.

[29] Y. Lou, P. Favaro, S. Soatto, and A. Bertozzi. Nonlocal similarity image filtering. Technical Report CAM-08-26, Department of Mathematics, University of California at Los Angeles, CA, U.S.A., July 2009.

[30] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[31] M. Mahmoudi and G. Sapiro. Fast image and video denoising via nonlocal means of similar neighborhoods. *IEEE Signal Processing Letters*, 12(12):839–842, December 2005.

[32] P. Mrázek, J. Weickert, and A. Bruhn. On robust estimation and smoothing with spatial and tonal kernels. In R. Klette, R. Kozera, L. Noakes, and J. Weickert, editors, *Geometric Properties for Incomplete Data*, volume 31 of *Computational Imaging and Vision*, pages 335–352. Springer, Dordrecht, 2006.

[33] L. Pizarro, P. Mrázek, S. Didas, S. Grewenig, and J.Weickert. Generalised nonlocal image smoothing. Technical Report No. 248, Department of Mathematics, Saarland University, Germany, 2009. To appear in International Journal of Computer Vision.

[34] T. Schultz, J. Weickert, and H.-P. Seidel. A higher-order structure tensor. In D. H. Laidlaw and J. Weickert, editors, *Visualization and Processing of Tensor Fields – Advances and Perspectives*, pages 263–280. Springer, 2009.

[35] S. M. Smith and J. M. Brady. SUSAN – A new approach to low level image processing. *International Journal of Computer Vision*, 23(1):43–78, 1997.

[36] M. R. Teague. Image analysis via the general theory of moments. *Journal of the Optical Society of America*, 70(8):920–930, 1980.

[37] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Mathematics Doklady*, 4(2):1035–1038, 1963.

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

[39] P. L. Torroba, N. L. Cap, H. J. Rabal, and W. D. Furlan. Fractional order mean in image processing. *Optical Engineering*, 33(2):528–534, 1994.

[40] D. Tschumperlé and L. Brun. Image denoising and registration by PDE's on the space of patches. In A. Foi and A. Gotchev, editors, *Proc. International Workshop on Local and Non-Local Approximation in Image Processing*, pages 32–40, 2008.

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

[42] S. Zimmer, S. Didas, and J. Weickert. A rotationally invariant block matching strategy improving image denoising with non-local means. In A. Foi and A. Gotchev, editors, *Proc. International Workshop on Local and Non-Local Approximation in Image Processing*, pages 135–142, 2008.

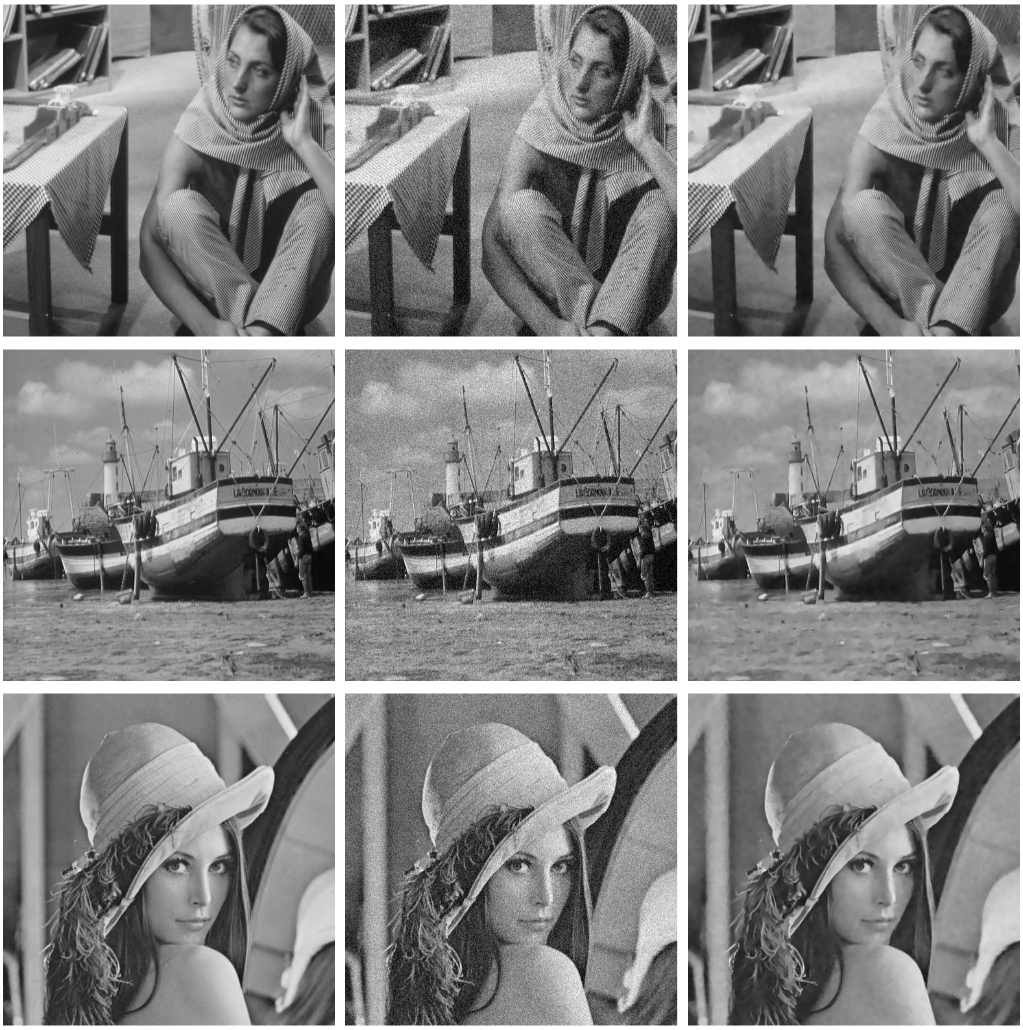

Figure 8: Images with size 512×512. **Left**: Original image. **Middle**: Noisy image with additive Gaussian noise ($\sigma_n = 20$). **Right**: Denoised image with classical NL means. **1st row**: *Barbara*. **2nd row**: *Boats*. **3rd row**: *Lena*.

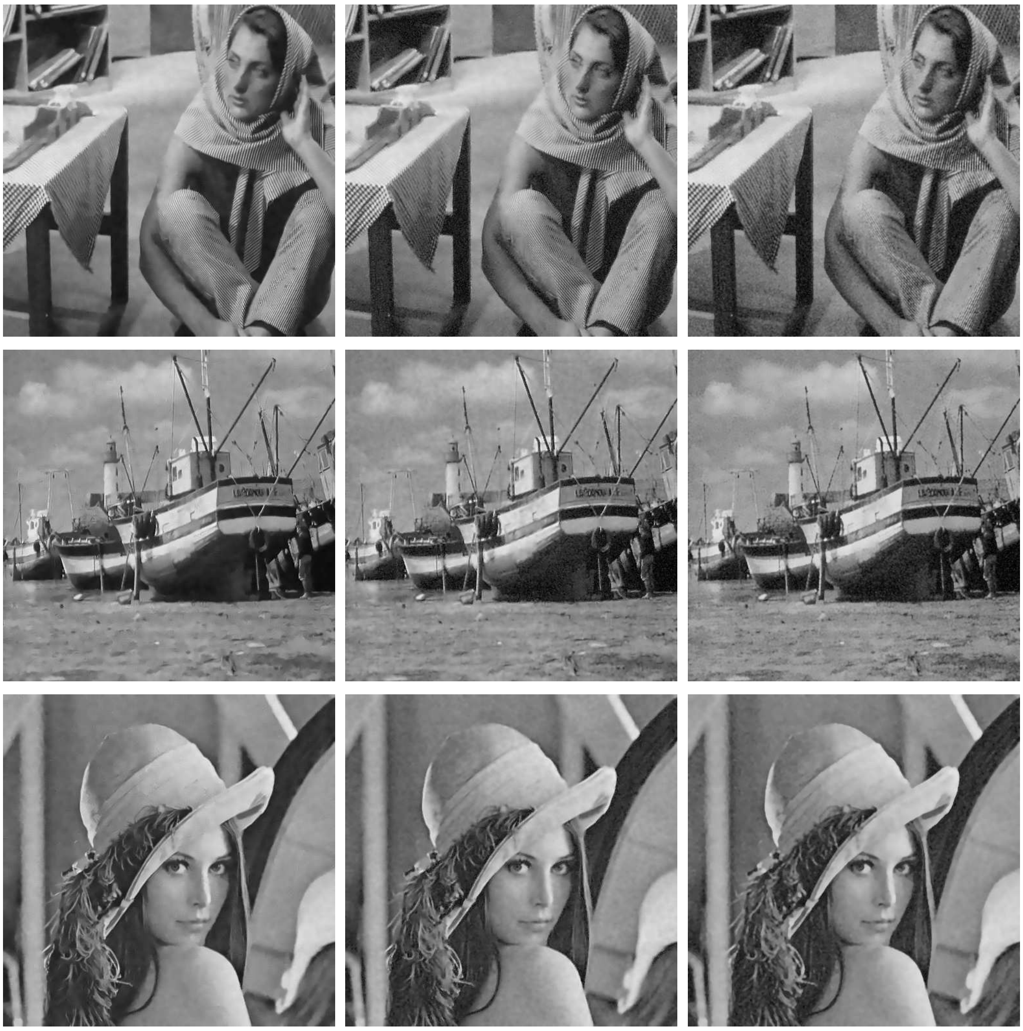

Figure 9: Denoised images (512 × 512) with rotationally invariant methods. **Left**: RIBM+ST. **Middle**: Zernike moments. **Right**: Hu moments.

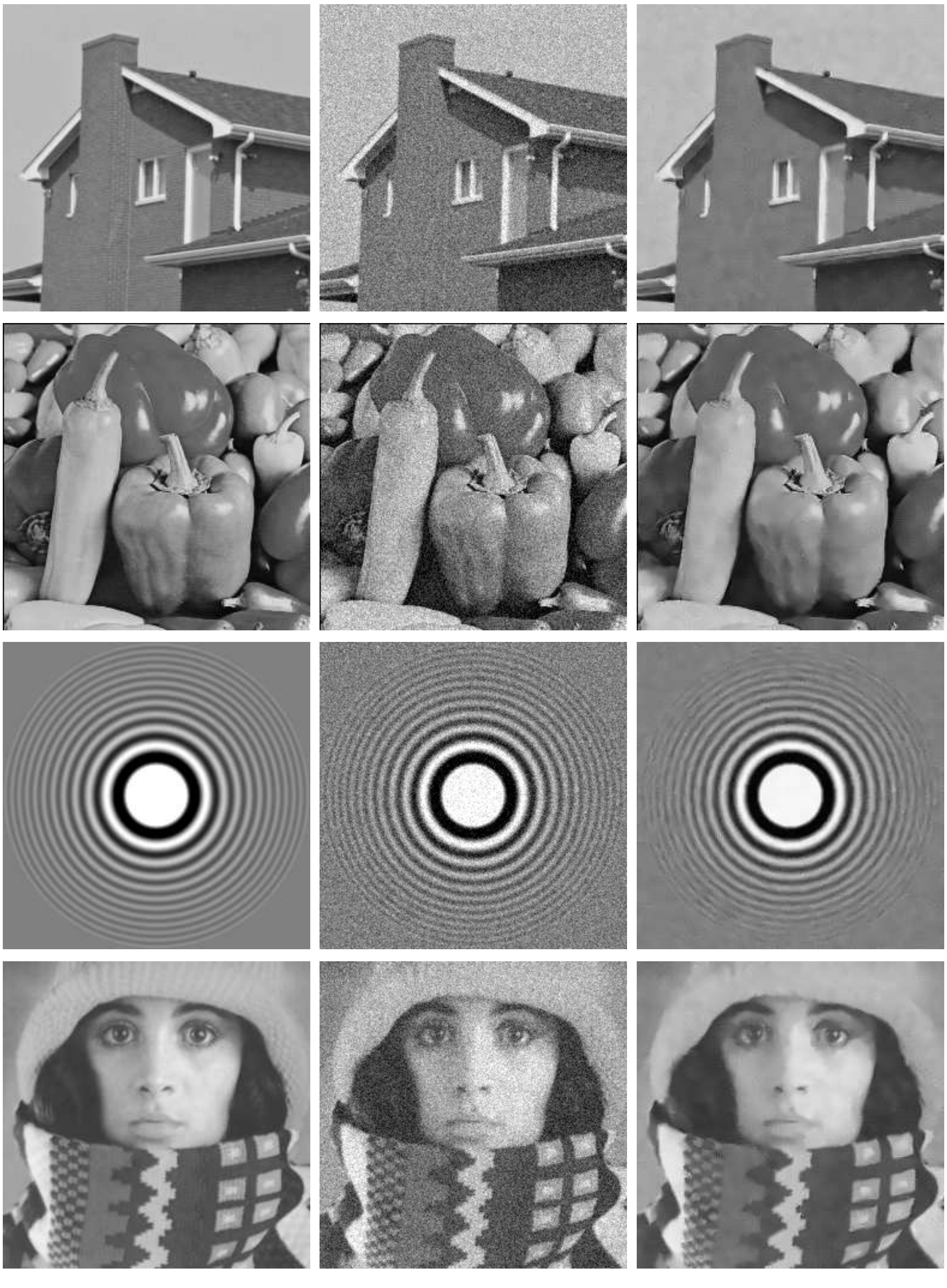

Figure 10: Images with size 256×256. **Left**: Original image. **Middle**: Noisy image with additive Gaussian noise ($\sigma_n = 20$). **Right**: Denoised image with classical NL means. **1st row**: *House*. **2nd row**: *Peppers*. **3rd row**: *Ring*. **4th row**: *Trui*.

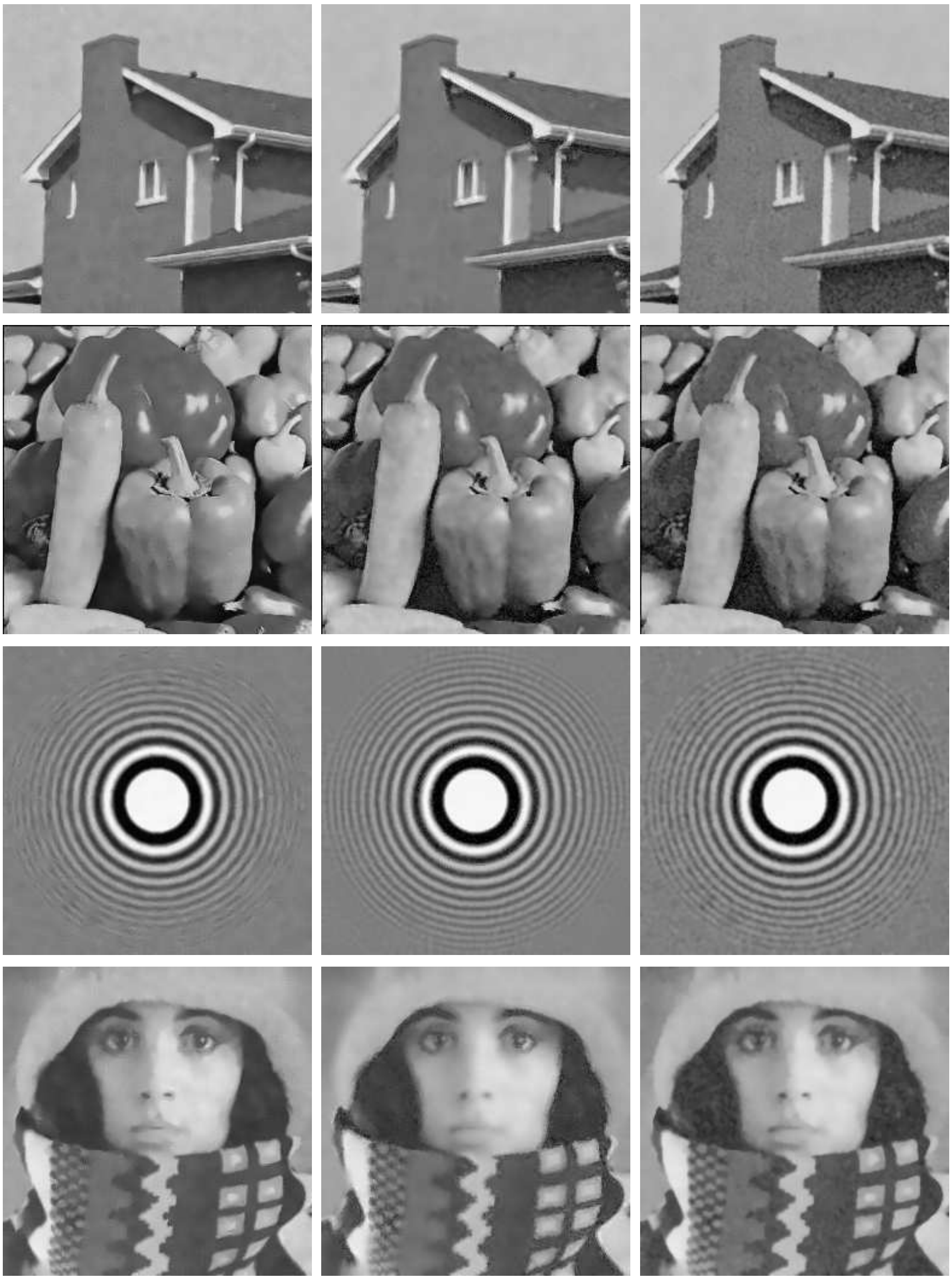

Figure 11: Denoised images (256×256) with rotationally invariant methods. **Left**: RIBM+ST. **Middle**: Zernike moments. **Right**: Hu moments.