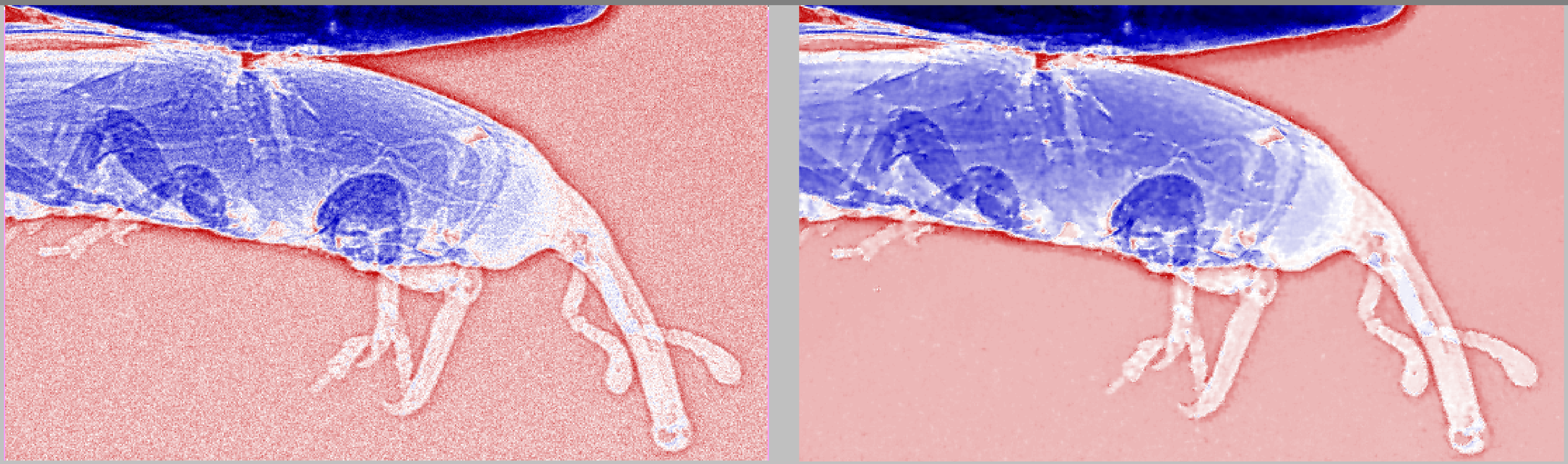
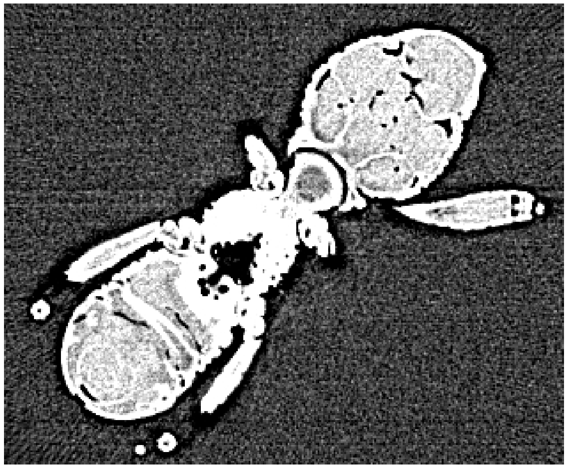
Image denoising with non-local means algorithm

Elena Reznichenko

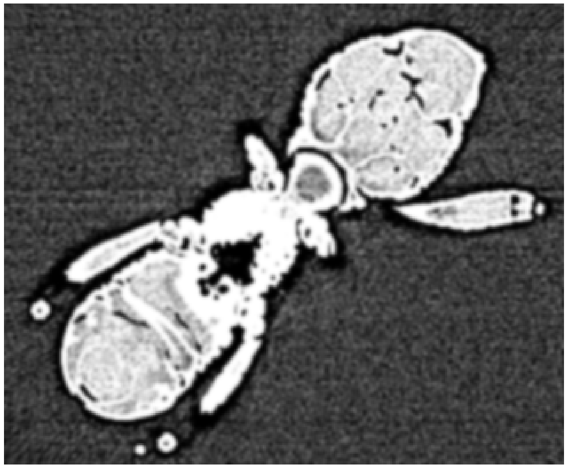


Denoising filters

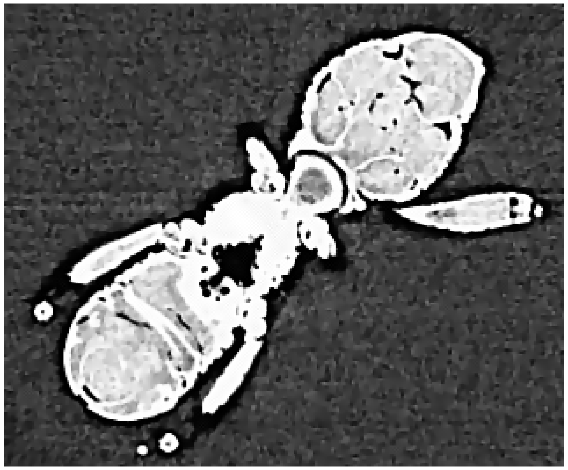
Noisy image



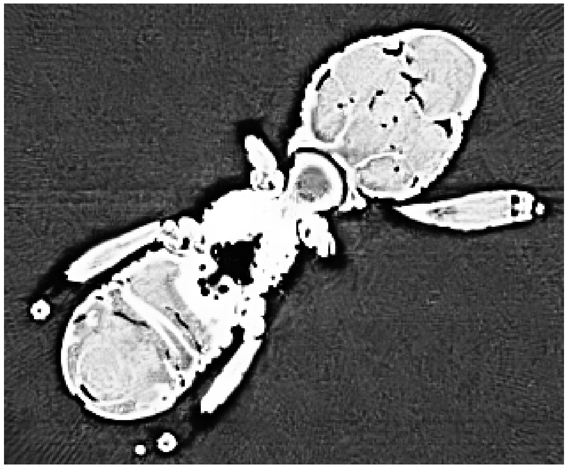
Gaussian filter



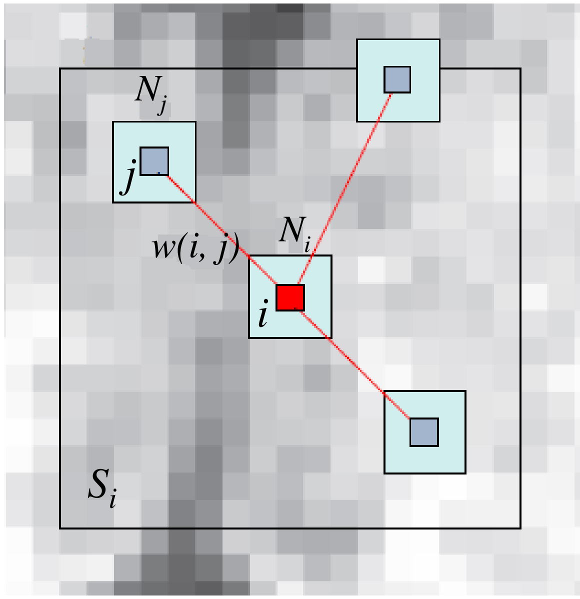
Anisotropic diffusion



Non-local means filter

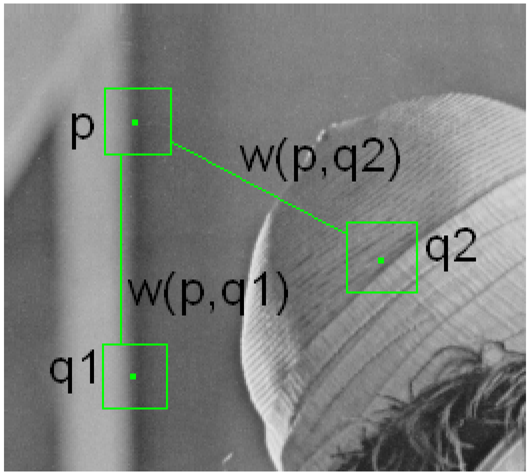


Non-local means algorithm



The restored value of pixel *i* (in red) is the weighted average of all intensities of pixels *j* in the search window S_i, based on the similarity of their neighborhoods N_i and N_j

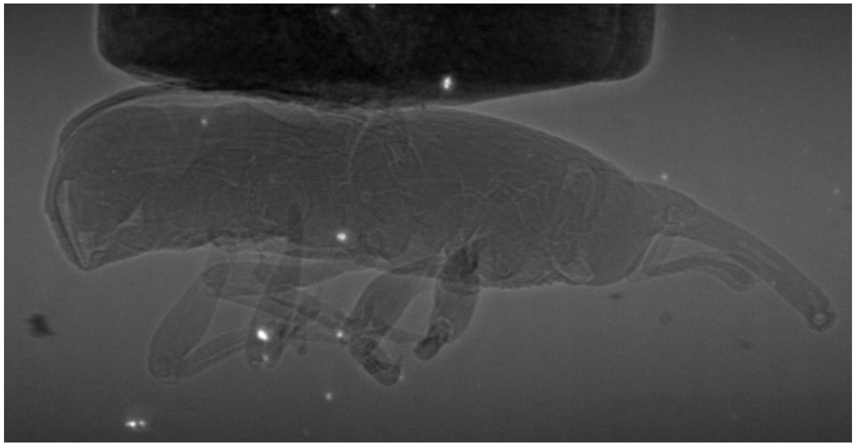
$$NL[v](i) = \sum_{j \in I} w(i, j) v(j)$$

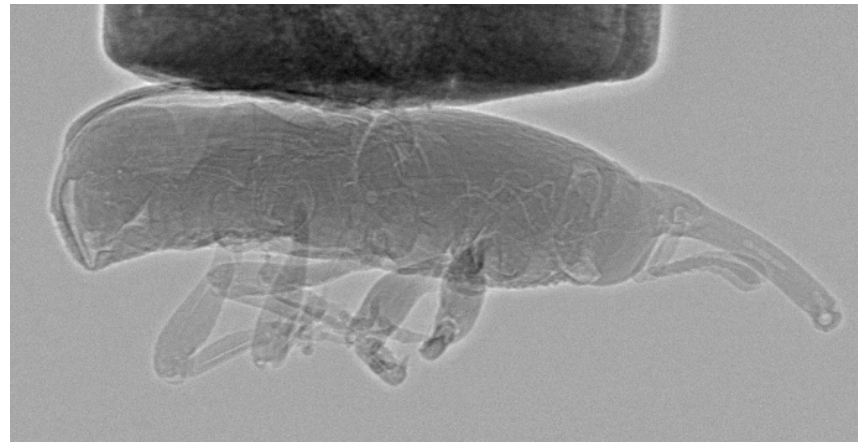


Pixel *q1* will have a stronger influence on the denoised value of *p* than *q2.*

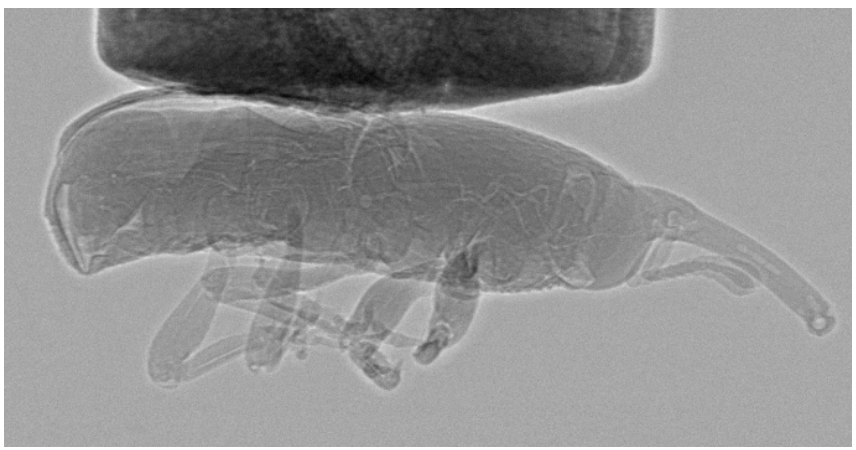
Noise in CT images

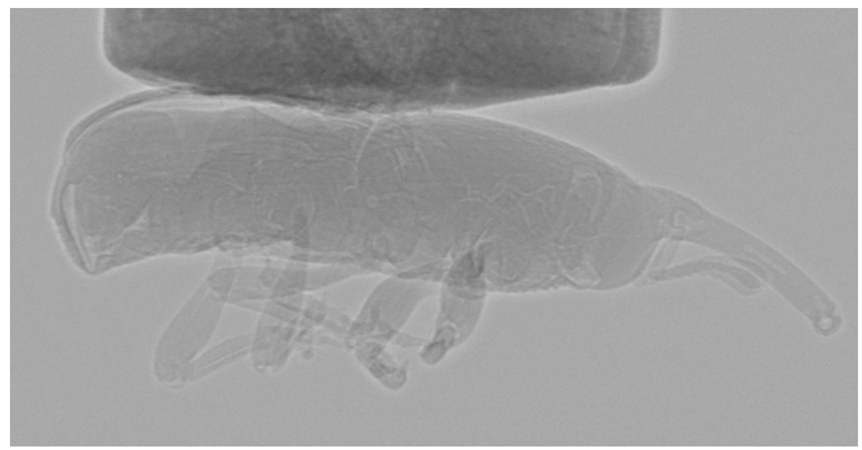
Fixed-pattern noise → Flat-field correction





Poisson noise → Square root transformation





Proposed NLM adaptations

- Using a piece-wise linear function for computing weights

$$w(i,j) = \frac{1}{Z(i)} e^{-\frac{d^2}{h^2}} \longrightarrow w(i,j) = \begin{cases} 1, & d^2 < 2\sigma^2 \\ \frac{d^2 - 2\sigma^2}{2\gamma}, & 2\sigma^2 \le d^2 \le 2\sigma^2 + 2\gamma \\ 0, & d^2 > 2\sigma^2 + 2\gamma \end{cases}$$

- Eliminating unrelated neighborhoods

  It is based on the similarity of the intensities mean value and on the similarity of the average of the gradient orientation.

- Considering edge or gradient pattern similarity

$$d^2 = (1-c) \cdot \left\| v(N_i) - v(N_j) \right\|_{2,a}^2 + c \cdot \left\| grad(N_i) - grad(N_j) \right\|_{2,b}^2$$

Steps of the algorithm

- Initialization of variables
- Loop for each pixel of the image
  - Loop for each pixel in the search window
    - Calculating distance for each pixel in the patch:

$$d = \frac{\sum_{i=1}^{n} (q_i - p_i)^2}{patch\_size}$$

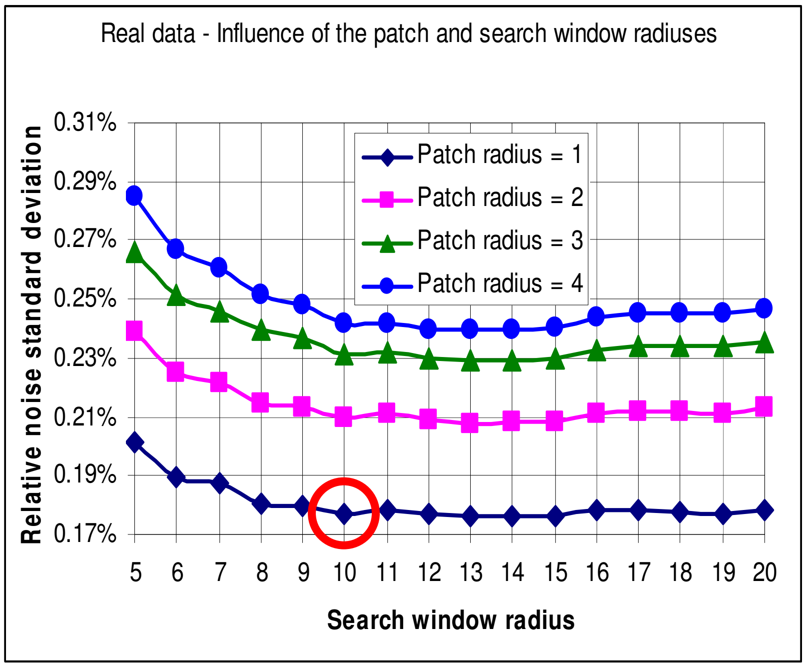
    - Calculating weights:

$$w(i,j) = \begin{cases} 1, & d^2 < 2\sigma^2 \\ \frac{d^2 - 2\sigma^2}{2\gamma}, & 2\sigma^2 \leq d^2 \leq 2\sigma^2 + 2\gamma \\ 0, & d^2 > 2\sigma^2 + 2\gamma \end{cases}$$

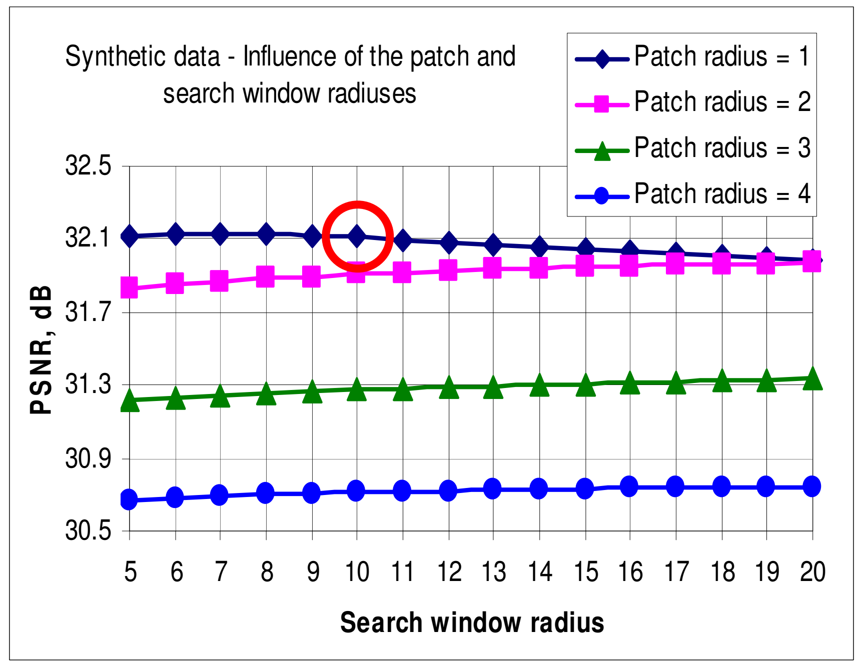
    - Accumulating total weight and weighted pixel values
  - Calculating the restored pixel value by dividing accumulated weighted pixel values by accumulated total weight

Experimental results

Defining the optimal values for the search window and patch radiuses



The best patch radius is 1, the search window radius can be restricted to 10.



The best PSNR is for the patch radius = 1 and the search window radius restricted to 10.

Experimental results

Karlsruhe Institute of Technology

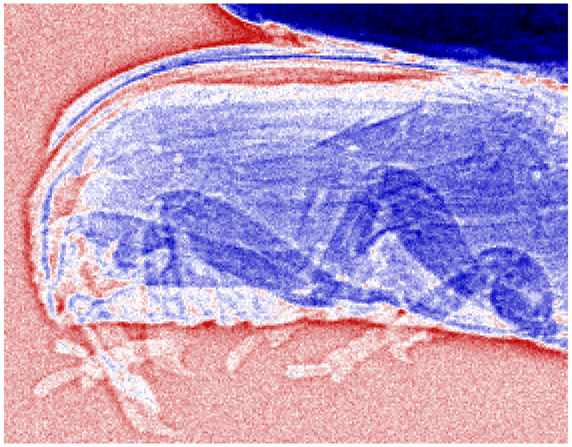
Comparison of the classical and proposed implementations of the NLM filter in terms of computational time and denoising quality

	σ_{Rn}, %	σ_{RNPS}, %	PSNR, dB	Time, s
Real data				
Classical NLM, non-restricted search	0.26	15.08	–	5537.40
Classical NLM	**0.18**	4.61	**–**	10.81
Proposed NLM	0.23	**4.21**	**–**	**3.94**
Synthetic data				
Classical NLM, non-restricted search	2.85	3.27	31.42	3601.91
Classical NLM	2.63	1.39	32.11	6.61
Proposed NLM	**2.49**	**1.06**	**32.58**	**3.08**

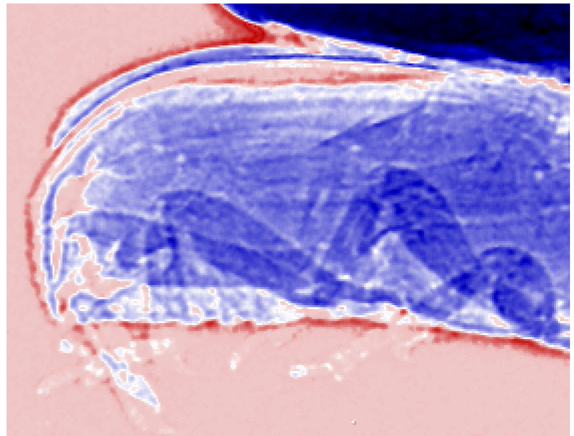
The time is obtained on CPU Intel® Core™ i5-650 3.20 GHz.
The real data is a 832×400 32-bit float grayscale image.
Synthetic data with added Poisson noise is the 719×458 32-bit float grayscale image.

Experimental results

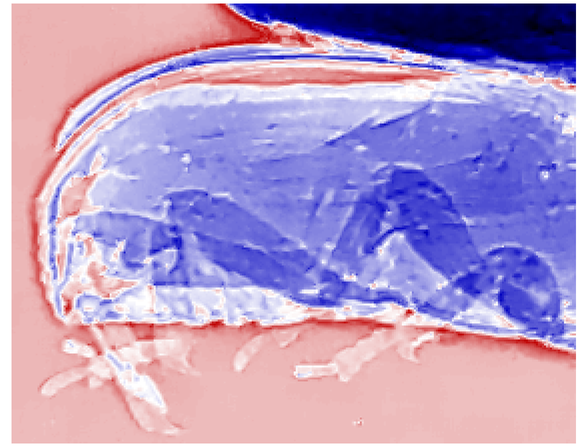
Comparison with the classical NLM filter – real data



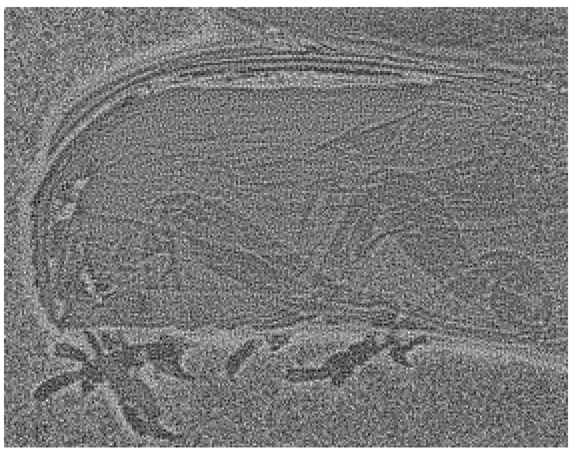
Noisy image



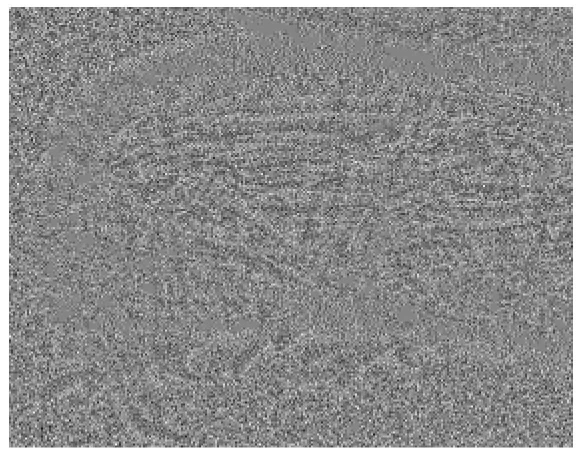
Classical NLM



Proposed NLM with gradient similarity



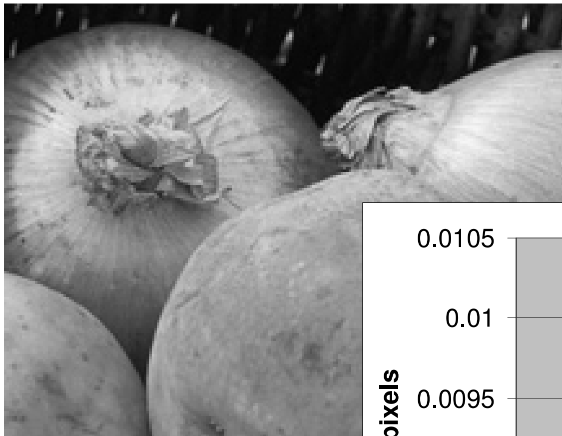
Removed noise – classical NLM



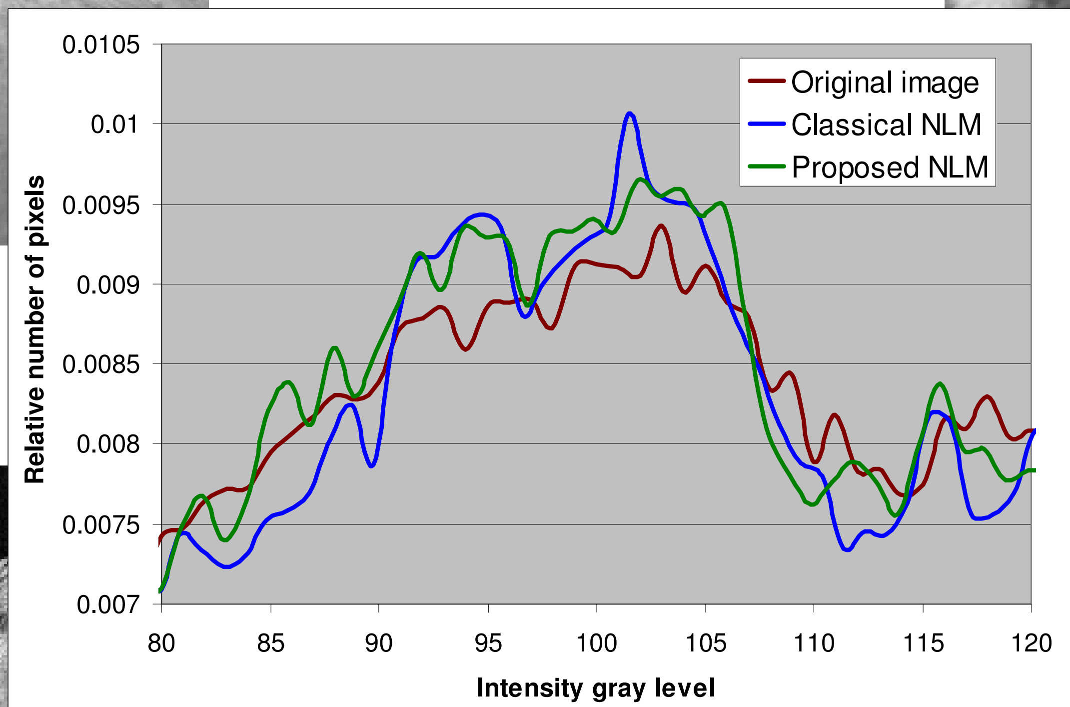
Removed noise – proposed NLM

Experimental results

Comparison with the classical NLM filter – synthetic data

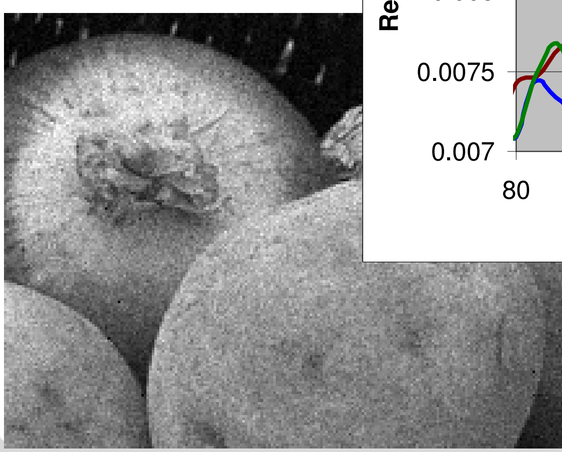


Original image

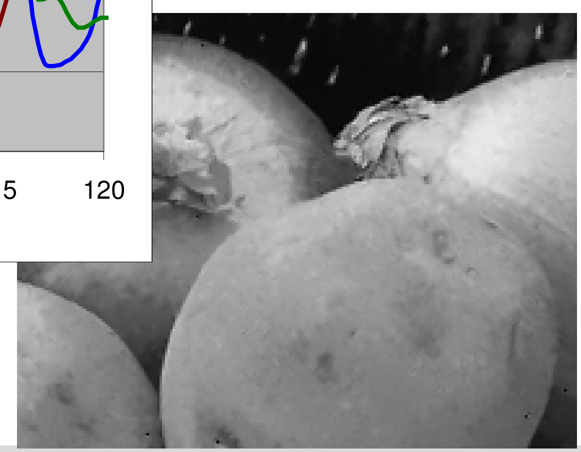


Classical NLM

Noisy image

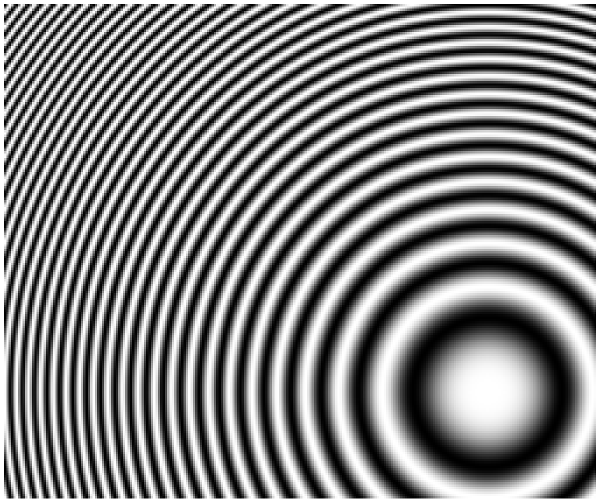


Proposed NLM

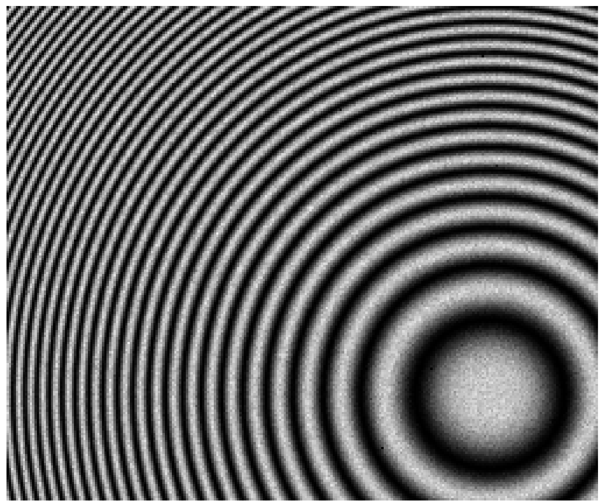


Experimental results

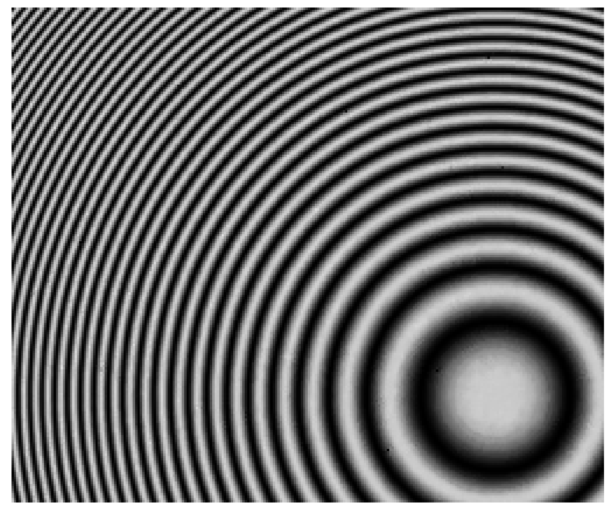
Fresnel zone plate



Original image

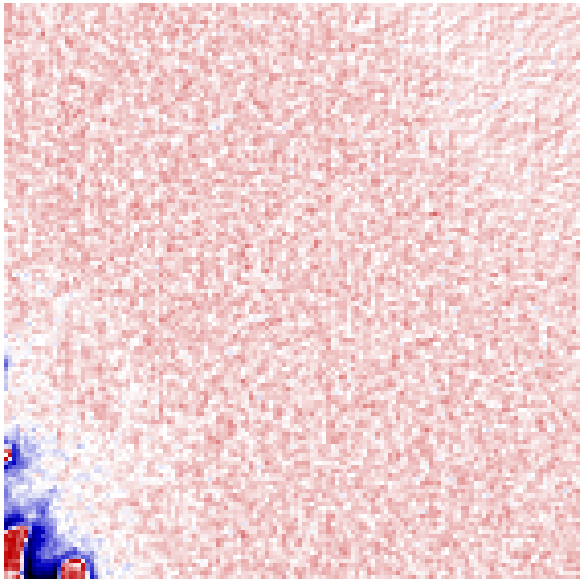


Noisy image

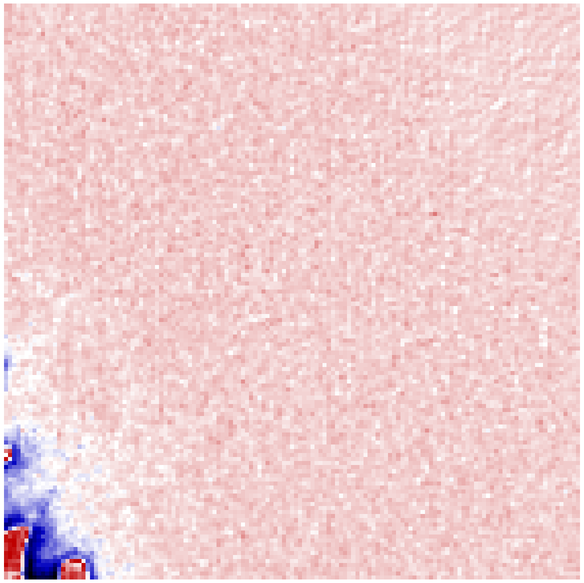


Proposed NLM

3D NLM filtering



Noisy volume



Denoised volume