Letter to the Editor

Comments on the paper ‘A novel 3D wavelet-based filter for visualizing features in noisy biological data’, by Moss et al.

C. L. LUENGO HENDRIKS & D. W. KNOWLES
Life Science Division, Lawrence Berkeley National Laboratory,
One Cyclotron Road, mailstop 84R171, Berkeley, CA 94720, U.S.A.

Key words.

Summary
Moss et al. (2005) describe, in a recent paper, a filter that they use to detect lines. We noticed that the wavelet on which this filter is based is a difference of uniform filters. This filter is an approximation to the second-derivative operator, which is commonly implemented as the Laplace of Gaussian (or Marr–Hildreth) operator. We have compared Moss’ filter with (1) the Laplace of Gaussian operator, (2) an approximation of the Laplace of Gaussian using uniform filters and (3) a few common noise reduction filters. The Laplace-like operators detect lines by suppressing image features both larger and smaller than the filter size. The noise reduction filters only suppress image features smaller than the filter size. By estimating the signal-to-noise ratio and mean square difference of the filtered results, we found that the filter proposed by Moss et al. does not outperform the Laplace of Gaussian operator. We also found that for images with extreme noise content, line detection filters perform better than the noise reduction filters when trying to enhance line structures. In less extreme cases of noise, the standard noise reduction filters perform significantly better than both the Laplace of Gaussian and Moss’ filter.

Method
The comparisons here were done in two dimensions. Moss’ wavelet-based filter is described for three-dimensional (3D) images, but is dimensionality independent. A two-dimensional (2D) test image was created following the recipe given by Moss: We drew three 1-pixel-thick lines of intensity 10 on a background of intensity 0, and dilated them with decreasing intensity to 5-pixels width. To this test image we then added Gaussian distributed noise with a mean of 0 and a standard deviation of 10/SNR (SNR being the chosen value for the signal-to-noise ratio as defined by Moss et al.), and linearly stretched the result to fit the [0, 255] range. This stretching did not change the content of the image, since we used 32-bit floating point representation for the image data. The parameter \(a\) from Moss’ filter was set to 6 pixels. Each of the filters, described below, was then applied to test images with a SNR of 8, 2 and 0.5.

To estimate the SNR in the filtered results we used the following procedure: (1) Find the multiplication factor that minimizes the mean square difference between the noiseless input image and the filter output, taking into account only the pixels that fall within the test image lines (i.e. those pixels that are nonzero in the noiseless input image). (2) Multiply the filtering output with this value. This normalizes the ‘signal level’ to 10. (3) Compute the standard deviation \(s\) over a set of background pixels in the normalized filter output (we exclude all pixels within a distance of 3\(a\) of the lines and the edge, see the mask image in Fig. 1). (4) The SNR is given by 10/\(s\). The value of the minimized mean square difference gives an indication of how well the filter was able to preserve the line.

The scripts we used for this comparison have been made available online at http://clluengo.lbl.gov/mossfilter.html

Moss’ wavelet-based filter
For 2D, Moss’ filter requires eight 1D filter passes, eight clips (which sets all negative values to 0), three image averagings and two rotations. The rotations are the most expensive components of this filter, and are used to improve the rotation invariance of the filter (applying 1D uniform filters, which is what the wavelet is built on, sequentially in each dimension yields a rectangular filter kernel).

The Laplace operator
The Laplace operator \(\nabla^2\) is constructed by addition of the second-order derivative in each dimension (Jähne, 2002), that
The second-order derivative is best implemented as a Gaussian derivative (Marr & Hildreth, 1980; Jähne, 2002), to avoid enhancing the noise. This is often referred to as the Laplace of Gaussian (LoG) or the Marr–Hildreth operator, and can be implemented as a separable filter (Huertas & Medioni, 1986). The Gaussian smoothing gives the LoG operator a parameter, which has to be tuned to the thickness of the lines expected in the image.

For this comparison we choose the parameter for the Gaussian to be $\sigma = 0.6a$, with $a$ being the parameter for Moss’ filter. This makes the two filters quite comparable, as can be seen in Fig. 2. Also, we clipped the result of the operator, setting all negative values to 0, to mimic the behaviour of Moss’ filter.

The difference of uniform filters

The LoG operator can be approximated by a difference of Gaussians (Jähne, 2002). The image is smoothed at two scales, and the difference taken. In 1D, if the Gaussian smoothing is replaced by a uniform smoothing kernel, we obtain an operator identical to the wavelet used by Moss. We implemented this in 2D (and called it difference of uniform filters) by using a disk of diameter $a$ as the smaller smoothing kernel, and a disk of diameter $3a$ as the larger. As with the LoG, we clip negative values to 0.

There are other filters that improve upon the LoG, such as the nonlinear Laplace operator described by van Vliet et al.
Noise reduction filters

Reducing noise must be one of the topics most studied in signal and image processing. We added several common methods to this comparison: the median filter (Jähne, 2002), the Kuwahara–Nagao filter (Kuwahara et al., 1976; Nagao & Matsuyama, 1979) and an alternating sequential filter (Sternberg, 1986; Serra, 1988). These filters do not set the background value to 0, but rather to the noise mean. Thus, to better compare the result of these filters with the line detectors we subtract the median of the noisy input image, and then clip all negative values to 0. For both the median and the Kuwahara–Nagao operator we used a disk with diameter $a - 1$ as the filter kernel. The alternating sequential filter is implemented as an open-close filter applied recursively with disks of increasing size between 2 and $a - 3$ pixels.

Results and discussion

Figure 1 shows the input images and the result of each of the filters described, for different setting of the input SNR. As can be seen none of the methods work particularly well for very high noise levels. Moss et al. show a better result at SNR = 0.5 on their 3D test image than the result the same filter obtains here in 2D. In 3D a higher noise level is allowable because of the increased number of pixels within a neighbourhood, which leads to better statistics on the noise. This advantage would equally improve the results of the other filters in this comparison. For all but the lowest SNR levels, standard noise reduction filters do a really good job of recovering the original noiseless image.

Table 1 shows the SNR and mean square difference estimated for the images discussed above. High values of SNR are desired, but not at the expense of increased mean square difference, which would indicate that the signal is not being recovered well. As can be seen, Moss’ filter and the difference of uniform filters obtain a similar performance (with this measure), whereas the LoG performs slightly better. We attribute this difference to the optimality of the Gaussian as a low-pass filter (Marr & Hildreth, 1980). The noise reduction filters break down for extremely low SNR, but in less severe cases perform much better than any of the line detecting filters.

We cannot compare the execution time of these filters because our implementation of Moss’ filter is not optimized. However, we can compare the computational complexity of this filter with the LoG. As explained above, Moss’ filter is implemented by adding four filtering results (or 24 in 3D), obtained by a separable filtering kernel; some of these filtering passes must be applied to a rotated version of the input image, the output of which must be rotated back (two rotations in 2D or six in 3D). By contrast, the LoG can be computed by adding the output of two separable filters (or three in 3D). On our AMD Opteron (2.2 GHz) processor, the LoG filter with $\sigma = 0.6a$ ($a = 6$) takes 13 s to process a $300 \times 300 \times 300$ voxel image. In comparison, a single rotation of this image
around the z-axis takes 15 s. Moss’ filter requires 6 rotations, 24 filter passes and some additional arithmetic, which Moss et al. reported taking 3.6 min on their Intel Xeon (2.8 GHz) processor, on an image of the same size.

**Conclusion**

The wavelet-based filter presented by Moss et al. is very similar to a Laplace-type filter. The complex implementation of Moss’ filter makes it computationally expensive and, in our study using 2D synthetic data, does not perform better than other established techniques, such as the Laplace of Gaussian operator.

**References**


