UCRL-JC-119787

PREPRINT

CONF-950226-57

Nonlinear Image Filtering Within IDP++

S. K. Lehman
M. G. Wieting
J. M. Brase

RECEIVED

OCT 06 1995

OSTI

This paper was prepared for submittal to
*SPIE's Electronic Imaging:
Science and Technology Conference*
San Jose, CA
February 5-10, 1995

February 9, 1995

Lawrence
Livermore
National
Laboratory

This is a preprint of a paper intended for publication in a journal or proceedings.
Since changes may be made before publication, this preprint is made available with
the understanding that it will not be cited or reproduced without the permission of
the author.

W-7405-ENG-48

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED *mw*

MASTER

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

NONLINEAR IMAGE FILTERING WITHIN IDP++

Sean K. Lehman*, Melvin G. Wieting, James M. Brase
Lawrence Livermore National Laboratory
7000 East Avenue
Livermore, CA 94566

ABSTRACT

IDP++, image and data processing in C++, is a set of signal processing libraries written in C++. It is a multi-dimension (up to four dimensions), multi-data type (implemented through templates) signal processing extension to C++. IDP++ takes advantage of the object-oriented compiler technology to provide "information hiding." Users need only know C, not C++. Signals or data sets are treated like any other variable with a defined set of operators and functions. We present here some examples of the nonlinear filter library within IDP++. Specifically, the results of MIN, MAX, median, α -trimmed mean, and edge-trimmed mean filters as applied to a real aperture radar (RAR) and synthetic aperture radar (SAR) data set.

Keywords: image processing, C++, object-oriented programming, order statistic filters, nonlinear filters

1. INTRODUCTION

IDP++ (image and data processing in C++) is a library of signal processing functions written in C++ taking advantage of the object-oriented nature of the language. We have recently included a collection of nonlinear filtering tools such as MIN, MAX, α -trimmed mean, median, and edge-trimmed mean filters. These are known as order-statistic filters since they operate on ordered subsets of the data.

We present here the results of the application of these filters to a real aperture radar (RAR) image and a synthetic aperture radar (SAR). The images contain a ship-generated internal wave wake. In the RAR image in Figure 1, the wake is vertical and skewed with the vertex at the top. The wake in the SAR image in Figure 2 is horizontal and contains multiple periods in each arm with the vertex at the right. We wish to emphasize these are real, not simulated data sets with real, not necessarily white additive gaussian, noise.

In Section 2, we present a history of IDP++, its current status, usage requirements, advantages and disadvantages, and functionality. Section 3 presents an overview of non-linear filters with comparisons to linear filters, and a description of the filters used in processing the radar images. Section 4 discusses the application of nonlinear filtering to radar imagery. Conclusions are in Section 5.

2. IDP++: IMAGE AND DATA PROCESSING IN C++

IDP++ was developed as a partial replacement for View¹, an interactive multi-dimensional, multi-data type signal processing package developed within our group. View is principally written in C but takes advantage of existing FORTRAN code. It has a scripting or macro capability and is very good at prototyping a solution but is not ideal in real-time situations or as a final solution. Its interactive, interpreted nature makes it slow when processing large data sets.

Thus we had a need for a compiled multi-dimensional signal processing package supporting multiple data types. Not wishing to develop another interpreter or compiler we decided to use the current object-oriented language technology of C++^{2,3}.

In object-oriented programming, concepts such as a complex number, a vector, a signal, etc., can be defined, attributed to variables, and used in an intuitive manner. Each one of these concepts is called a *class*. For example, FORTRAN users are familiar with the complex type which implicitly has real and imaginary parts. In C++, the real and imaginary parts of a complex class are considered to be properties of the class.

IDP++ consists of four classes: complex, string, array, and signal. The latter class concerns us here. The signal class contains the actual signal data of course, but also properties which we intuitively attribute to data sets. Some of them are:

- The signal name or name of the data set;
- The signal directory or where the data reside within the computer file system;
- The signal data type, for example 8-bit, floating-point, complex, etc., (since IDP++ supports multiple data types, this property is used to reflect the type of a particular signal variable);
- The signal dimensionality. IDP++ supports up to four dimensional signals with the axes nominally being X, Y, Z, and t;
- The size of each dimension, that is the number of elements along each axis;
- The data axes origins;
- The data axes sample intervals;
- The time the signal was created or collected;

When a signal is created (*instantiated* in object-oriented language terms) and these properties assigned, they will "travel" with the signal throughout its processing without the need for the user to keep track of them. Thus, instead of manipulating several variables during the processing of a signal, the IDP++ developer uses only one: the signal.

Another advantage is that, being compiled, it will run faster than interpreted signal processing packages and can be used in real-time environments. It can be easily used to prototype a solution but the user must be prepared for the extra time required for compilation. Functions can be nested to more closely reflect mathematical equations making the code easier to read.

Naturally, a good C++ compiler is required. Currently IDP++ builds on the latest Sun and Silicon Graphics compilers. Once built, IDP++ consists of a core library, `idp.a`, which includes all the principal signal processing functions from autocorrelation to zero-padding. The list of functions includes fast Fourier transforms; de-trend; extract; uniform, gaussian, and gamma random noise generators; various window functions; linear filtering functions; etc. There are currently two additional support libraries. One for linear filters (`idplf.a`) and one for order statistic filters (`idposf.a`). The results of the latter are presented here.

The multi-data type support is provided by the use of C++ templates which are used to parameterize class and function definitions. Through the use of templates, a function can be written to operate on a generic data type. The compiler creates or instantiates the function for a specific type, say a float. The current data types supported by IDP++ are 8-bit character (both signed and unsigned), two byte short, long, integer, floating point, floating point complex, double, and 8-bit complex.

3. NONLINEAR FILTERS

Our interest in nonlinear filters^{4,5} stems from our need to enhance features in our data sets corrupted by noise which is not necessarily white additive gaussian. We have used an experimental approach; designing a filter, selecting parameters, and applying it to a data set to observe the results. How a filter type and design affect given distribution with a particular set of parameters is not fully understood at this time. Our approach shows promise although at this point the theory is not completely understood. The non-stationary, non-isotropic, and correlated nature of our data sets make the theory more difficult.

Generally, the nonlinear filters affect different areas of the signal's probability density function (pdf) differently, and if chosen judiciously can emphasize the desired features within the signal. In particular, nonlinear filters based on robust statistics can provide improved estimation performance in non-Gaussian noise environments⁴. Probability density functions with long tails can severely degrade linear estimates of location and size of the distribution. Robust estimators such as the median are much less sensitive to these outliers in finite samples of data.

To date, the filters we have studied fall into a family called *order statistic* filters. They are:

- MIN and MAX;
- Median;
- α -trimmed mean;
- Edge-trimmed mean.

Order statistic filters operate on ordered sub-sets of the data. Given a set of data points, $x_i, i = 0, \dots, n-1$, the i -th order statistic, $x_{(i)}$, is the i -th element after the set of data points have been ordered from lowest to highest in value.

Thus MIN and MAX filters select the lowest ($x_{(0)}$) and highest ($x_{(n-1)}$) order statistic respectively.

The median of the n sample points is given by

$$\text{med}(x_i) = \begin{cases} x_{(v)} & n = 2v + 1 \\ \frac{x_{(v-1)} + x_{(v)}}{2} & n = 2v \end{cases} \quad (1)$$

Thus, in contrast to the MIN and MAX filters which select the lowest and highest valued data point, the median selects the middle value or average of the two middle values if the number of points is even.

The trimming filters perform arithmetic averages of sub-sets of the ordered statistics. They require a parameter specifying the percentage of points to trim out. The α -trimmed mean filter selects points from the middle of the ordered set. It is defined as:

$$\bar{x}_\alpha = \frac{1}{n(1-2\alpha)} \sum_{j=\alpha n}^{n(1-\alpha)-1} x_{(j)} \quad 0 \leq \alpha \leq 0.5 \quad (2)$$

α selects the percentage of order statistics at each end of the set to eliminate from the average. Note, for $\alpha = 0$, $\bar{x}_\alpha$ reduces to the arithmetic mean, and as $\alpha \rightarrow 0.5$ (50%), $\bar{x}_\alpha \rightarrow \text{med}(x_i)$.

The edge-trimmed mean filter selects points from either the low or high end. It is defined as:

$$\bar{x}_\alpha = \begin{cases} \frac{-1}{\alpha n} \sum_{j=-\alpha n}^{-1} x_{(j)} & -1 \leq \alpha < 0 \\ \frac{1}{\alpha n} \sum_{j=0}^{\alpha n-1} x_{(j)} & 0 < \alpha \leq 1 \end{cases} \quad (3)$$

This variation on the α -trimmed mean filter eliminates a percentage of data points from one or the other edge of the distribution. α selects the percentage. As a matter of bookkeeping, negative α trims the low end of the distribution, and positive α trims the high end. Note, for $\alpha = 1$ (i.e. 100%), $\bar{x}_\alpha$ becomes the arithmetic mean.

4. APPLICATION TO RADAR IMAGERY

Enhancement of radar imagery is an important application area for nonlinear filtering. In this section we will illustrate the use of IDP++ nonlinear filtering operations with the two radar data sets. The filter performance will be measured by estimating the signal-to-noise ratio of linear image features before and after filtering.

The images that we use were acquired in ocean remote sensing experiments⁶. The first is from a hillside 10 GHz real-aperture radar (RAR) looking at the ocean surface at low-grazing angle (approximately 6 degrees). It shows a section of an extended ship wake and is shown in Figure 1. The radar stares at a fixed point on the ocean surface; the image axes are range horizontally and time vertically. The strong linear features moving toward the radar (at the left) are thought to be surface wave groups⁷. The ship wake is the faint nearly vertical dark stripe which runs the entire vertical length of the image. The second test image, shown in Figure 2, is a 1.2 GHz SAR image acquired with the NASA/JPL system⁸ at Loch Linnhe, Scotland in 1989.

The statistical properties of radar images make them particularly interesting cases for noise reduction by nonlinear filters. As we discussed above, filters based on robust statistics are effective for smoothing images which have long-tailed intensity distributions. Both SAR and RAR images of the ocean have non-Gaussian intensity distributions – particularly at low-grazing angles.

Coherent radar images are usually modeled as a multiplicative random process⁹ of the form $I(x,y) = N(x,y)\sigma_0(x,y)$ where $N(x,y)$ is an exponentially distributed speckle noise component and $\sigma_0(x,y)$ is a slowly varying radar cross section. For ambient ocean clutter images, $\sigma_0(x,y)$ is often modeled as a gamma distributed random process¹⁰ which produces a long-tailed distribution for $I(x,y)$. Figure 3 shows a measured probability density function (pdf) for the RAR image of Figure 1. The plot compares it with an exponential distribution based on the data mean. The pdf for the SAR image shown in Figure 4 agrees with the modeled exponential distribution.

To test the effectiveness of the nonlinear filters described above for smoothing these ship wake images, we ran the filters for our two test cases with varying region sizes and filter parameters. We then measured the SNR of the wake in the resulting images. The region sizes are listed in Table 1.

Region
3x1
1x3
3x3
10x1
1x10
10x3
3x10
20x1

Table 1. Region sizes over which the filters were applied. The region $m \times n$ represent m -pixels in the X-direction and n -pixels in the Y-direction.

The signal level is measured by calculating the image modulation by averaging along the wake – a series of 1D cuts are made perpendicular to the wake which are then averaged. The peak to peak difference is used as the signal level. The noise level is estimated by calculating the standard deviation of a region near the location on the wake where the signal was measured. The ratio of signal level to noise standard deviation will be used as the image quality metric. We call this the signal-to-noise ratio (SNR).

The regions chosen include both square and rectangular regions at different orientations to test the directional dependence of between the filter region and the signal. We have included a linear mean filter (lowpass) in the comparison. Figures 5 through 10 show the performance for the six filters tested. They show SNR histograms as a function of the regions listed in Table 1.

As a bench-mark to determine improvement, we list the unfiltered SNR values for the two data sets in Table 2. These bench-marks are shown as a horizontal line in the performance histograms. We consider an image to have been improved when an SNR histogram exceeds the bench-mark.

Data Set	SNR
RAR	0.20
SAR	1.06

Table 2. Bench-mark metrics for unfiltered images.

The MIN, MAX, median, and mean filters require only two parameters, those describing the region size. The α - and edge-trimmed means require three, two for the region and the percentage of values to be trimmed.

General observations

The RAR and SAR data sets have different statistics. The low-grazing angle RAR image is believed to follow a gamma distribution with differing correlation lengths in both the X- and Y-directions. The SAR image, on the other hand, follows an exponential distribution coming from a normally distributed complex data set.

RAR MIN	The best SNR improvement occurs with filters whose regions favor the X-direction, i.e. across the vertical ship track. Specifically the 10x1 and 10x3 regions. We believe this is due to the wake correlation length which is shorter in the X-direction than in the Y-direction.
RAR MAX	No significant improvement in SNR above the 0.20 bench-mark leading us to conclude the wake information is in the lower end of the pdf.
RAR median	The best SNR improvement occurs with filters whose regions favor the X-direction. The results are better than those of the MAX filter but not as good as the MIN filter suggesting the wake information is in the lower end of the pdf.
RAR mean	No significant improvement in SNR. This is not surprising given the statistics of the data set.
SAR MIN	Minor SNR improvement with filters whose regions favor the X-direction.
SAR MAX	Best SNR improvement with filters favoring the X-direction which is along the horizontal ship track.
SAR median	The best SNR improvement occurs with filters whose regions favor the X-direction.
SAR mean	SNR improvement for all regions.
RAR α -trimmed mean	Except for the smallest regions, the best SNR improvements are when α is 50%, i.e. the median.
RAR edge-trimmed mean	Best SNR gain in X-favored regions using 30 to 50% of the lowest order data. This reaffirms the wake statistics lie within the low end of the pdf.
SAR α -trimmed mean	Except for the smallest regions, the best SNR improvements are when α is 5 to 10%. This is consistent with the observation of the mean being better than the median.
SAR edge-trimmed mean	Best SNR gain in X-favored regions using 70% of the data. This approaches an arithmetic mean filter.

Conventional, linear low-pass, filter is best for SAR. Nonlinear shows better performance for RAR. The nonlinear filters discussed here perform probability density function (pdf) domain filtering. They need to be tailored to the shape of the distribution with regions selected to be compatible with the correlation lengths. The differing results due to region sizes are accountable by the fact the data statistics are non-stationary and non-isotropic (they differ in the X- and Y-directions).

Figure 1. An example of nonlinear filtering of real aperture radar data. On the left is the original image; on the right is the result of the application of a 10x1 MIN filter.

Figure 2. An example of nonlinear filtering of a SAR image. On the top is the original image; on the bottom is the result of the application of a 10x3 edge-trimmed mean filter which removes the top 30%.

Figure 3. Probability density function (pdf) for the RAR data set. The long tail of the distribution is shown in contrast to an exponential pdf modeled using the parameter (the data mean) from the data set.

Figure 4. Probability density function (pdf) for the SAR data set. It is shown to agree with an exponential distribution.

Figure 5. SNR results for the SAR data set. It has an exponential probability density function (pdf). The benchmark SNR is 1.06 as indicated by the horizontal line. Best improvement is with the linear mean filter.

Figure 6. SNR results for the RAR data set with the long-tailed pdf. The benchmark SNR is 0.20 as indicated by the horizontal line. Best improvement is with the MIN filter.

Figure 7. α -trimmed mean results for the SAR data set. Best improvement for $\alpha = 5\%$ which is close to the mean.

Figure 8. α -trimmed mean results for the RAR data set. Best improvement as α approaches 50%, that is the median.

Figure 9. SNR results of the upper edge-trimmed mean filter with the SAR data set. Best results using the largest percentage of the data, that is the mean.

Figure 10. SNR results of the upper edge-trimmed mean filter with the RAR data set. Best results using the lowest end of the data, that is the min.

5. CONCLUSIONS

We have used IDP++ to test order statistic filters on two example radar images: a SAR image with Gaussian statistics (exponential in intensity) and a RAR image with a very long tailed noise distribution. The nonlinear filters, particularly MIN and edge-trimmed mean, provided improved noise reduction compared to linear filters for the non-Gaussian noise case. The linear smoothing filter performed best in the Gaussian noise environment. We intend to continue this work by generalizing the trimmed mean filter to a pdf domain filter. Such a filter would only pass values that lie within a specified region of the pdf.

We have explored only a small subset of the possible nonlinear filtering approaches for non-Gaussian noise smoothing. IDP++ provides a tool for easily and efficiently implementing, testing, and evaluating filters as we expands the scope of these computational experiments.

6. ACKNOWLEDGMENTS

This work was performed under the auspices of the Department of Energy by the Lawrence Livermore National Laboratory under contract W-7405-Eng-48. We wish to thank Brian Barber of DRA, UK, for making the field experiments possible, and Hughes Aircraft Corp., and the Jet Propulsion Laboratory.

7. REFERENCES

1. H. Szoke, J.M. Brase, V.J. Miller, and M.G. Wieting, "The VIEW Signal and Imaging Processing System," Lawrence Livermore National Laboratory, *UCID-21368*, March 1988.
2. M.A. Ellis and B. Stroustrup, *The Annotated C++ Reference Manual*, Addison-Wesley Publishing Company, Reading, MA, 1990.
3. S.B. Lippman, *C++ Primer 2ed*, Addison-Wesley Publishing Company, Reading, MA, 1991.
4. I. Pitas and A.N. Venetsanopoulos, *Nonlinear Digital Filters, Principles and Applications*, Kluwer Academic Publishers, Boston, MA, 1990.
5. I. Pitas and A.N. Venetsanopoulos, "Order Statistics in Digital Image Processing," *Proceedings of the IEEE*, Vol. 80, No. 12, pp. 1893-1921, December 1992.
6. G. Watson, R.D. Chapman, and J.R. Apel, "Measurements of the internal wave wake of a ship in a highly stratified sea loch", *J. Geophys. Res.*, 97(C6), 9689, 1992.
7. M.P. Tulin, University of California, Santa Barbara, personal communication, 1994.
8. D. Held and W. Brown, "The NASA/JPL multifrequency, multipolarisation airborne SAR system", *Proc. IGARSS*, 1988.
9. L.M. Novak, M.B. Sechtin, and M.J. Cardullo, "Studies of target detection algorithms that use polarimetric radar data", *IEEE Tr. Aerospace and Electronic Systems*, 25, 150, 1989.
10. K.D. Ward, C.J. Baker, and S. Watts, "Maritime surveillance radar, Part 1: Radar scattering from the ocean surface", *IEE Proc.*, 137, Part F (2), 51, 1990.