

GAMMA Technical Report: Offset estimation programs update

Christophe Magnard, Charles Werner, and Urs Wegmüller

Gamma Remote Sensing, Worbstrasse 225, CH-3073 Gümligen, Switzerland



Nov 2017

TABLE OF CONTENTS

Summary, usage changes and recommendations	3
1 Introduction	5
2 Modifications.....	5
2.1 Oversampling	5
2.1.1 Lanczos interpolation	6
2.1.2 FFT-based interpolation	6
2.2 Filtering.....	7
2.2.1 Low-pass filter of the complex data	7
2.2.2 Low-pass filter of the intensity data	7
2.2.3 Windowing of the intensity data	7
2.3 Peak detection	8
2.4 Deramp flag	8
2.5 Iterative processing	9
3 Results.....	9
3.1 Results using the synthetic pair.....	9
3.2 Actual data	28
3.2.1 TerraSAR-X – Venezia	28
3.2.2 Devon Ice Cap.....	32
4 Discussion	37
4.1 Sinusoidal biases	37
4.2 Offset underestimation.....	38
4.3 Iterative processing	38
4.4 Interpolation changes - issues using very small patches.....	38
4.5 Peak detection	38
4.6 Processing time	38
4.7 Outlook	39
5 References.....	40
6 Appendix: Programs usage.....	41
6.1 Offset_pwr	41
6.2 Offset_pwr_tracking	41
6.3 Offset_pwr_tracking2	42
6.4 Offset_pwrwm	43
6.5 Offset_pwr_trackingm	43
6.6 Offset_pwr_trackingm2	44
6.7 Offset_pwr_list	45

SUMMARY, USAGE CHANGES AND RECOMMENDATIONS

This update concerns the following programs:

offset_pwr, offset_pwr_tracking, offset_pwr_tracking2, offset_pwr, offset_pwr_trackingm, offset_pwr_trackingm2, and offset_pwr_list

Modifications include revised interpolation, data filtering, iterative processing, and peak estimation. These modifications should improve the reliability of the results and minimize systematic biases introduced by finite patch size and variable overlap of the patches. The programs were tested on various datasets and are expected to perform correctly even for very small patches (tested using 8x8 patches). Note that the accuracy of the offset measurements is dependent on the patch size.

General recommendations and modifications:

- If the **SLC** data are available, it is preferable to perform the offset estimation on these data rather than on **MLI** data. Using SLC data, biases caused by an aliasing of the spectrum can be avoided, resulting in more accurate results.
- The **cross-correlation function oversampling factor option** was removed (`c_ovr` command line parameter), because peak detection now uses a gradient descent algorithm that incorporates high order interpolation.
- For **offset_pwr**, **offset_pwr_tracking**, **offset_pwr**, and **offset_pwr_trackingm**, a new **Lanczos interpolation order** option has been added (Lanczos order 5 -> 9). 5 (default) provides good interpolation quality and is relatively fast, 9 provides outstanding interpolation quality but is somewhat slower.
- While the programs work with 8x8 patches, it is recommended to use at least 64x64 patches. 128x128 patches ensure a very reliable offset detection. The patches can have differing values for width and height.
- In the case where there is a difference in the spatial resolution, the window dimension should in general be adjusted to give approximately square patches.
- In order to estimate offsets with highest accuracy and minimum bias, it is recommended to carry out an iterative offset estimation using **offset_pwr_tracking2** or **offset_pwr_trackingm2**.

Estimating offsets between SLCs:

- For SLC data, a **2x oversampling** should be used to improve the accuracy.
- ScanSAR and TOPS data should be deramped prior to using the offset estimation programs.
- A **bandwidth fraction** option was added. It defines a low-pass filter applied on the complex data and might be useful when no oversampling of the SLC is applied. Its default value is 1.0 (full bandwidth). Reducing the bandwidth is not recommended for ScanSAR and TOPS data that were not previously deramped.
- A **deramp flag** option was added: the default behavior is that no deramping of the phase is applied. For most cases the Doppler centroid is close to 0, and therefore no deramping is required. In case of data with larger Doppler centroids, such as for spotlight SAR as well as for some ERS and RADARSAT data, the phase deramping flag should be set to 1. Deramping of the phase requires extra computation so the option can be specified only when required using the command line parameter. The deramping carried out in the offset estimation programs does not support ScanSAR or TOPS data deramping; for this, a dedicated program must be used.

- A flag was added that allows disabling the low-pass filter on the intensity data (*int_filt* parameter, the filtering is enabled by default). The low-pass filter is very important for avoiding biases when no oversampling is applied. In that case, it should not be disabled. Disabling the low-pass filter may be useful for speeding up the processing when 2x oversampling and large patches are used (128 or larger).

Estimating offsets between MLIs:

- A **bandwidth fraction** option was added, its default value is 0.8 (80% of the bandwidth). This option permits removing part of the aliased spectrum and thus improves the accuracy of the offset estimates (decrease of the offset bias error). As a drawback, the smaller the processing bandwidth, the greater the uncertainty in the estimated offsets (noisier results). Where greater reduction of bias error is required, reducing the bandwidth from the default value may be required.
- **Oversampling** the data provides a minimal improvement in the offset detection because once the SLC data are detected, the spectral aliasing has already occurred and cannot be significantly reduced by oversampling the detected data. If the SLC data were oversampled before detection, then further oversampling of the MLI data will also not be of significant benefit.

1 INTRODUCTION

While the previous versions of the offset estimation programs were already very effective, some shortcomings were identified in the following cases:

- When no oversampling of the SLC was performed, sinusoidal biases varying between plus and minus 0.05 pixels were observed. They were caused by the aliasing of the image spectrum in the image detection process (going from complex to intensity values).
- A general underestimation of the offsets was observed, that increased with the offset magnitude. It also depended on the patch size: the smaller the patch, the larger the underestimation of the offset.
- Iterative processing using `offset_pwr_tracking2` and `offset_pwr_trackingm2` used the nearest integer offset as initialization. It resulted in uncertainties at half pixel offsets and more generally provided sub-optimal use of the results from previous iteration.
- Offset estimation using very small patches ($\leq 16 \times 16$) in combination with oversampling yielded unreliable offsets (errors up to ± 0.25 pixels). This issue was probably caused by a poor FFT-based interpolation quality when applied to very small patches.
- The peak detection to estimate the maximum of the cross-correlation function with sub-pixel accuracy used a polynomial fit that may not reproduce accurately the true shape of the cross-correlation function. The output cross-correlation was that of the integer pixel in the non-oversampled cross-correlation function.

The update aims at minimizing the issues found wherever possible, while avoiding disproportionate increase in computational load.

2 MODIFICATIONS

2.1 Oversampling

Oversampling of the SLCs by a factor of 2 or more avoids aliasing the spectrum when detecting the images (going from complex to intensity values). Oversampling MLI data however provides minimal improvement in the offset detection, it probably only slightly reduces border effects occurring in the FFTs.

Oversampling is a computationally expensive step; when patches used for the offset estimation are overlapping, it would be advantageous to oversample only once the entire images. It is however not practical, since large datasets could overflow the computer memory (RAM). As a compromise, in `offset_pwr`, `offset_pwr_tracking`, `offset_pwr_m`, and `offset_pwr_trackingm`, the oversampling is now carried out for all the image columns (the whole range). This avoids repetitively oversampling the same data. In `offset_pwr_tracking2`, `offset_pwr_trackingm2`, and `offset_pwr_list` programs, each patch is oversampled separately. Oversampling whole blocks was found unpractical (too slow) for the way these programs are functioning.

For intensity data, interpolation is now achieved on the square root of the data. Tests showed that interpolation of intensity data using high quality interpolation methods yields major ringing issues. This ringing artifact is barely noticeable when interpolating on the square root of the intensity.

The oversampling can be achieved either using time-domain interpolation such as the Lanczos interpolation or by via FFT-based interpolation. Both methods have their advantaged and drawbacks.

2.1.1 Lanczos interpolation

Lanczos interpolation uses a truncated SINC interpolant, weighted by the main lobe of another SINC that was stretched to the window size [1].

Using Lanczos interpolation, provided a buffer half the size of the interpolation kernel is added at the edges, the resulting quality is independent on the patch size. As a drawback, it is generally relatively slow, since an interpolation kernel of 10 pixels (Lanczos order = 5) has to be used to reach sufficient interpolation quality, and kernels \geq to 14 (Lanczos order = 7 or more) are required for outstanding interpolation quality.

The computational load of time-domain interpolation can be noticeably reduced in the case of data oversampling: the position of the interpolated points relative to the original points is constant over the whole dataset. This allows separating the 2D interpolation into two 1D interpolations. Moreover, the original values do not need to be recomputed. This reduces the number of computations compared to a conventional 2D interpolation by

$$\frac{m \cdot ovr}{3 \cdot (ovr - 1)} \quad (1)$$

where m is the size of the interpolation kernel and ovr the oversampling factor. For example, with $ovr = 2$ and $m = 12$, the number of computations is divided by 8. In addition, the interpolation weighting factors (the kernel) can be precomputed, thus further accelerating the interpolation process.

Lanczos interpolation is now used in *offset_pwr*, *offset_pwr_tracking*, *offset_pwr_m*, and *offset_pwr_trackingm* programs. There, the whole range length is oversampled at a time, and a buffer half the size of the interpolation kernel is used in the azimuth direction. This method ensures accurate interpolation at the edges of the patches. The Lanczos order can be specified through a new input option.

2.1.2 FFT-based interpolation

FFT-based interpolation is carried out as follows: a “zero-interleaving” process is carried out in time-domain, i.e. zeros are added between the actual data points. A 2D-FFT is applied on the zero-interleaved data. A low-pass filter is then applied on the 2D spectrum to discard the spectrum repetitions introduced by the zero interleaving. The low-pass filter bandwidth and transition band sizes are divided by the oversampling factor. An inverse-FFT is finally applied to the spectrum, yielding the oversampled data.

FFT-based interpolation generally provides high quality oversampling and is very fast for moderate size patches. However, its results are less accurate for small patches and its computing time becomes problematic for very large datasets such as the Lanczos interpolation may become more appropriate in such cases.

FFT-based interpolation is now used in *offset_pwr_tracking2*, *offset_pwr_trackingm2*, and *offset_pwr_list* programs. To improve the interpolation quality at the window edges and in case of very small patches, a buffer of 8 pixels is added at the patch edges. The low-pass filter has also been

redesigned and has now the following characteristics: bandwidth fraction: 0.95, transition band: 0.05, attenuation: 40 dB (to compute the effectively used bandwidths and transition bands, these values have to be divided by the oversampling factor).

2.2 Filtering

2.2.1 Low-pass filter of the complex data

When using SLC inputs, a low-pass filter can be applied to the complex data. This filter can be useful in particular when no oversampling is applied: a bandwidth reduction lowers the spectrum aliasing when detecting the images. When the resulting spectrum bandwidth is smaller than half the sampling frequency, hence no aliasing occurs at all. As a drawback, the smaller the bandwidth, the greater the offset uncertainty (noisier results).

The bandwidth fraction is a new input option in `offset_pwr`, `offset_pwr_tracking`, `offset_pwr_tracking2`, and `offset_pwr_list`.

2.2.2 Low-pass filter of the intensity data

Intensity data may be aliased, in particular if the complex data were not oversampled or low-pass filtered before performing the multi-looking step. The most severe aliasing occurs in the higher frequencies, hence a low-pass filter permits removing the largest part of the aliased spectrum and thus improving the accuracy of the results (decrease of the bias). As a drawback, the smaller the bandwidth, the greater the offset uncertainty (noisier results).

The bandwidth fraction is a new input option in `offset_pwr`, `offset_pwr_tracking`, `offset_pwr_tracking2`, and `offset_pwr_list`. In `offset_pwr`, `offset_pwr_tracking`, `offset_pwr_tracking2`, and `offset_pwr_list`, the bandwidth fraction is set to 0.8 when no oversampling is used and 0.9 when the SLC data were previously oversampled. It can be disabled using the `int_filt` option.

2.2.3 Windowing of the intensity data

Although some uncertainty is remaining, the general offset underestimation is probably caused by data wrapping during the calculation of the cross-correlation. A slight windowing of the intensity data decreases this underestimation.

The last 16 pixels at each edge are weighted down using a cosine function. For small patches (< 128 pixels), we only filter one eighth of the pixels at the edge. The window is scaled up in case of oversampling. The window used for patches with 128x128 pixels is shown in Fig. 1.

Applying this window artificially increases the cross-correlation value, in particular for uncorrelated data. To compensate this effect, a suppression of DC and low frequency components is carried out on the cross-correlation spectrum.

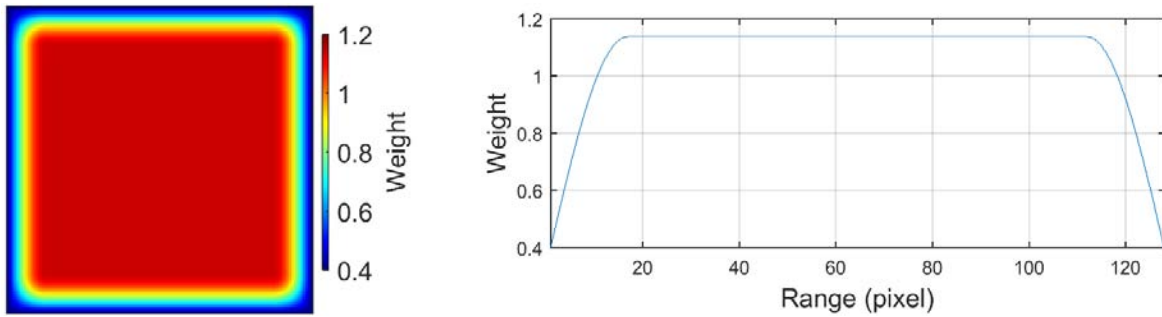


Fig. 1 Left: 128x128 window applied to the intensity data. Right: window values along range for mid-azimuth position. The window was weighted to have an average value equal to 1.

2.3 Peak detection

Peak detection with sub-pixel accuracy is now achieved using a gradient ascent or descent algorithm [2], also sometimes called “steepest ascent”. The output cross-correlation value is now the maximum at sub-pixel accuracy.

In an initialization step, the algorithm scans the whole patch to find the global maximum position with integer pixel accuracy.

The cross-correlation is then interpolated (a 7th or 9th degree B-spline interpolation is used depending on the program) at three positions around the initial maximum estimation. The interpolated values permit estimating a tangent plane defined by its normal vector: $\vec{v}_n = (v_{nx}, v_{ny}, v_{nz})$, with $\|\vec{v}_n\| = 1$. Two normal vectors can actually characterize the tangent plane, pointing either upwards or downwards, depending on the sign of v_{nz} . Let us only consider the downward pointing vector. This vector has the following properties:

- The horizontal vector (v_{nx}, v_{ny}) is pointing into the direction of the maximum of the function.
- The magnitude of the horizontal components varies between 1 for a vertical tangent plane (horizontal normal vector) and 0 for a horizontal tangent plane (vertical normal vector).

The closer to the maximum location, the lower the gradient. Hence adding (v_{nx}, v_{ny}) to the current maximum estimate typically provides an improved maximum estimate. In case of overshooting the maximum, (v_{nx}, v_{ny}) can be weighted downwards.

The maximum is searched through an iterative process. It ends when the distance between the updated position and the previous position is smaller than a specified value.

2.4 Deramp flag

A deramp flag option was added to the programs working with SLC data. For most cases where the Doppler centroid is close to 0, no phase deramping is required. In case of data with larger Doppler centroids, such as for spotlight SAR as well as for some ERS and RADARSAT data, a phase deramping has to be carried out. This flag was added so that the user can specify that phase deramping be applied only when necessary.

2.5 Iterative processing

In *offset_pwr_tracking2* and *offset_pwr_trackingm2*, the offsets estimated in a previous iteration are used in a two-step process:

- In a first step, the nearest integer of the offset estimate is used to read a “slave” patch centered on that integer pixel.
- In a second step, the patch is shifted by the fractional part of the offset in both range and azimuth, such that “master” and “slave” patches should match according to the previous offset estimation. This process is achieved by applying a phase ramp in frequency domain.

3 RESULTS

The results were evaluated using both a synthetic image pair and actual data consisting of two interferometric image pairs.

- The synthetic image pair is generated using a TerraSAR-X scene acquired over the Etna, Italy, and the same acquisition stretched in both range and azimuth direction by a factor of 1.0002. The stretched image was resampled using *resamp_image* program, with a Lanczos-9 interpolation method to ensure the very best quality and avoid introduction of resampling artifacts.
- The actual data are a pair of TerraSAR-X acquisitions in Venezia, Italy, and a pair of Sentinel-1 acquisitions over Devon Ice Cap, Canada.

3.1 Results using the synthetic pair

The input data have 8000 range samples and 10000 azimuth lines. Offsets were estimated on a rectangular grid, 80 patches along range and 100 patches along azimuth.

For the assessment of the updated software, we used *offset_pwr_tracking*, *offset_pwr_trackingm* and *offset_pwr_tracking2*. The most significant results are as follows:

1. In the first experiment using *offset_pwr_tracking*, two different patch sizes were used: 128x128 and 32x32. In each case, the data were processed both with and without oversampling. Using the new updated program version, no low-pass filter was applied to the SLC data (bandwidth fraction = 1.0). The results without oversampling also correspond to those obtained using MLI data (*offset_pwr* and *offset_pwr_trackingm*), with a bandwidth fraction of 0.8 for the new program versions (see Section 2.2.2). Results are shown in Fig. 2-9.
2. In the second experiment using the updated *offset_pwr_trackingm* (for MLI data), the effect of the bandwidth reduction on the bias is shown: three results using the updated program are shown, where the bandwidth fraction parameter was set to 1.0, 0.8, and 0.5. The results using the original programs are also provided. The size of the used patches was 128x128. Results are shown in Fig. 10-13.
3. The benefit of the updated version of *offset_pwr_tracking2* program is illustrated next. Fig. 14 shows results obtained using the nearest integer offset estimated in a previous iteration, while Fig. 15 shows the results obtained using the real-valued offsets estimated in a previous iteration. A 32x32 patch size was used to emphasize the effect of the iteration for improving the estimates. The results should be compared to those in Fig. 6.

4. Oversampling very small patches yielded unreliable results using the original program versions, due to inaccurate oversampling of small patches. The improvement is highlighted in Fig. 16 and Fig. 17.
5. Finally, the effect of the intensity data windowing described in Section 2.2.3 is highlighted by showing in Fig. 18 results where the windowing was turned off. They should be compared to those in Fig. 2, where all other parameters were kept the same.

For each test, we provide three plots:

- The two first plots show the results of the offsets detection along range and azimuth. Blue points show the difference between the estimated offsets and the theoretical offsets. The black line shows the median difference, e.g. in the “Range offsets along range” case, for each range position, the median was computed for the 100 patches along the azimuth direction. The red line shows the median of the estimated offsets.
- The third plot shows the output cross-correlation.

The processing time is provided in the figure captions, **note that the scales are not the same for all plots!**

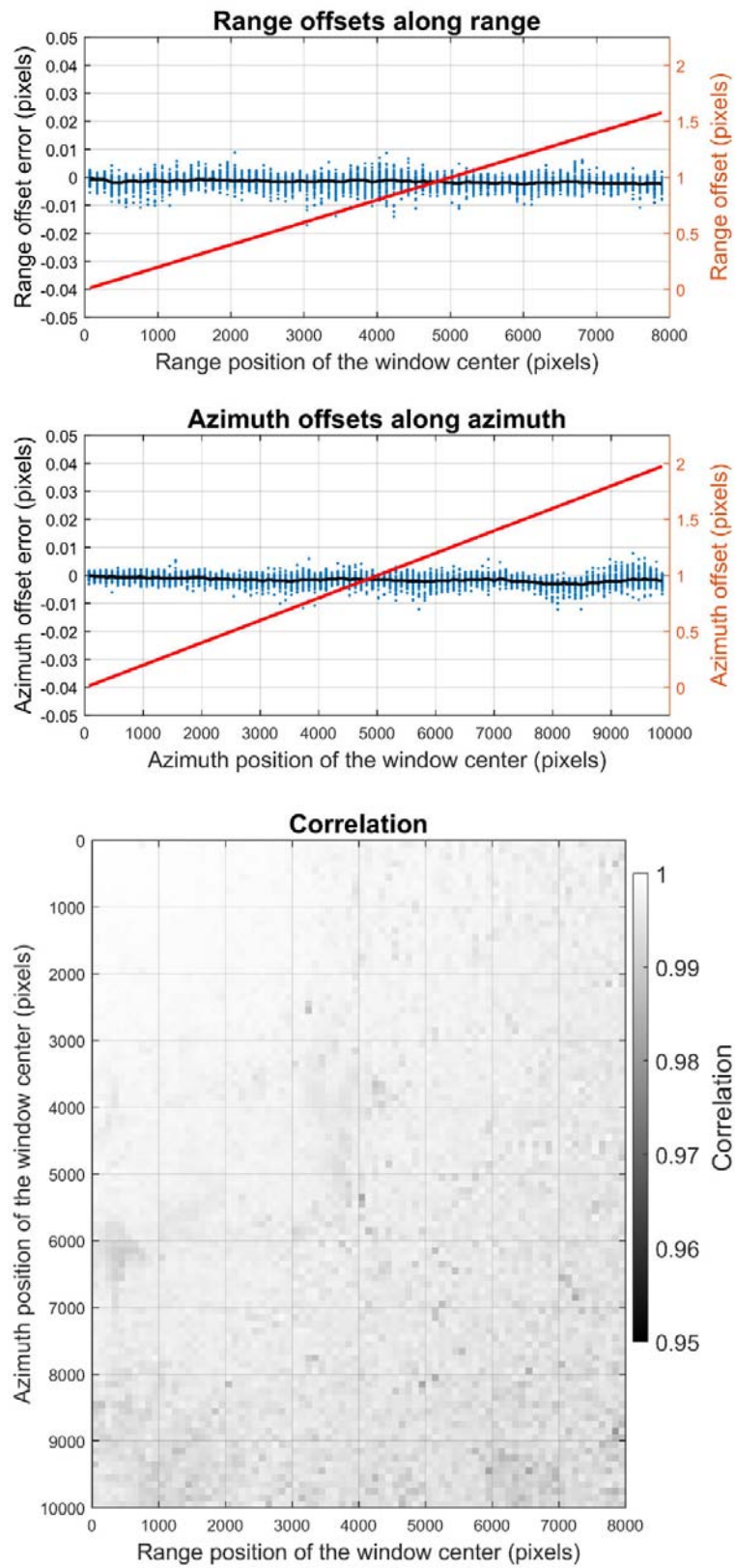


Fig. 2 Updated version of offset_pwr_tracking program: 128x128 patch size, 2x SLC oversampling, Lanczos 9 interpolation, no deramp, 1.0 bandwidth fraction. Processing time: 22.413 (s).

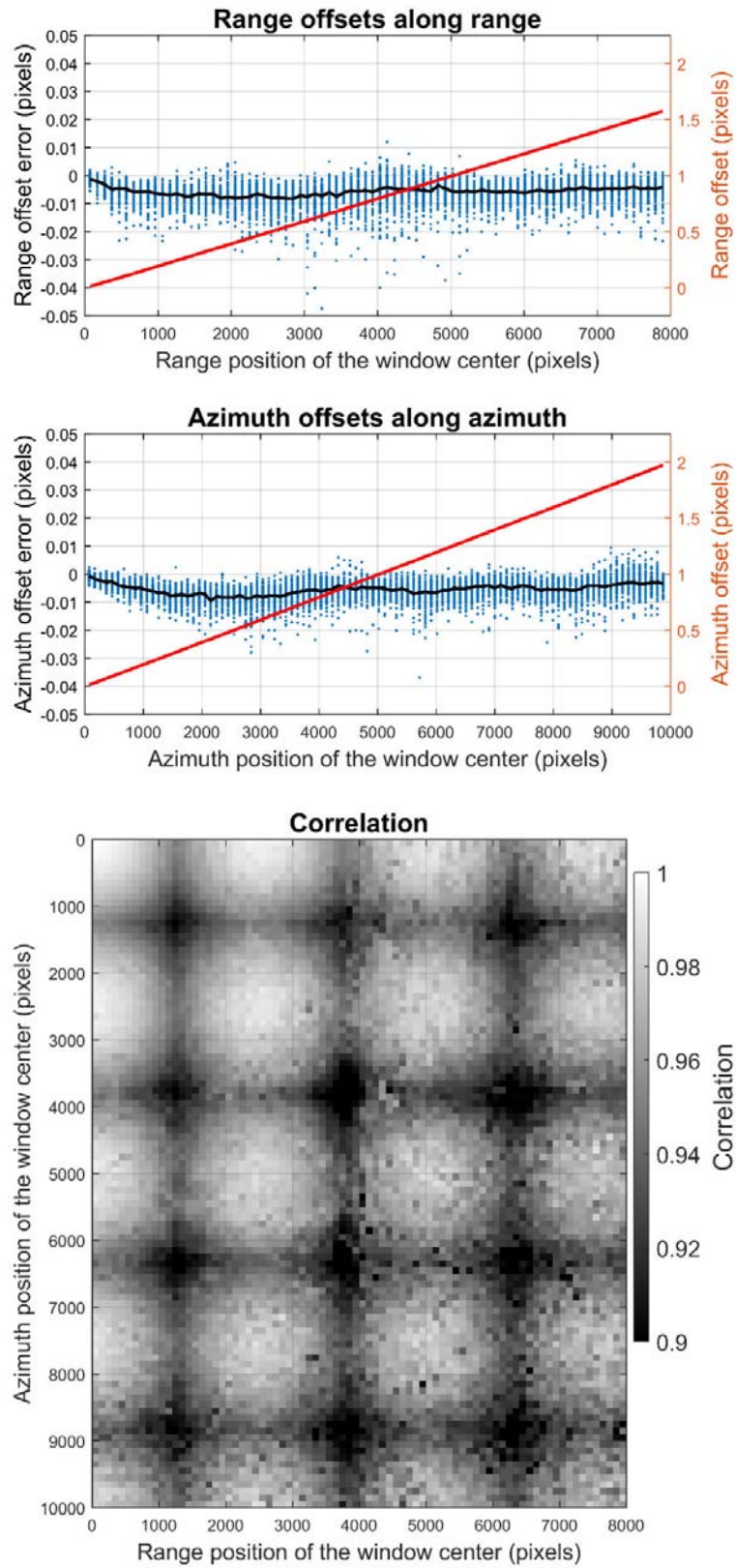


Fig. 3 Original version of offset_pwr_tracking program: 128x128 patch size, 2x SLC oversampling, 4x cross-correlation function oversampling factor. Processing time: 14.731 (s).

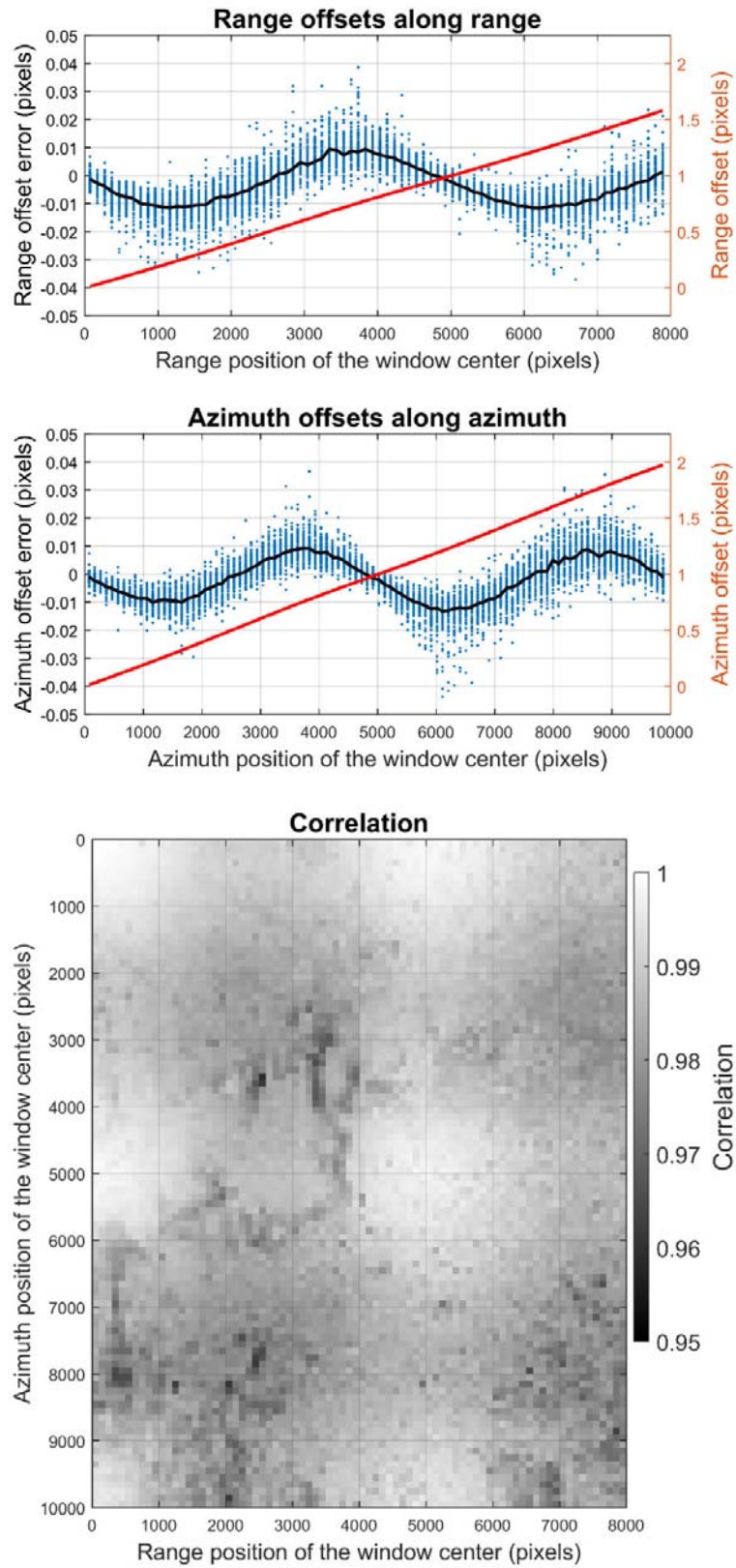


Fig. 4 Updated version of offset_pwr_tracking program: 128x128 patch size, no SLC oversampling, no deramp, 1.0 bandwidth fraction. Processing time: 3.300 (s).

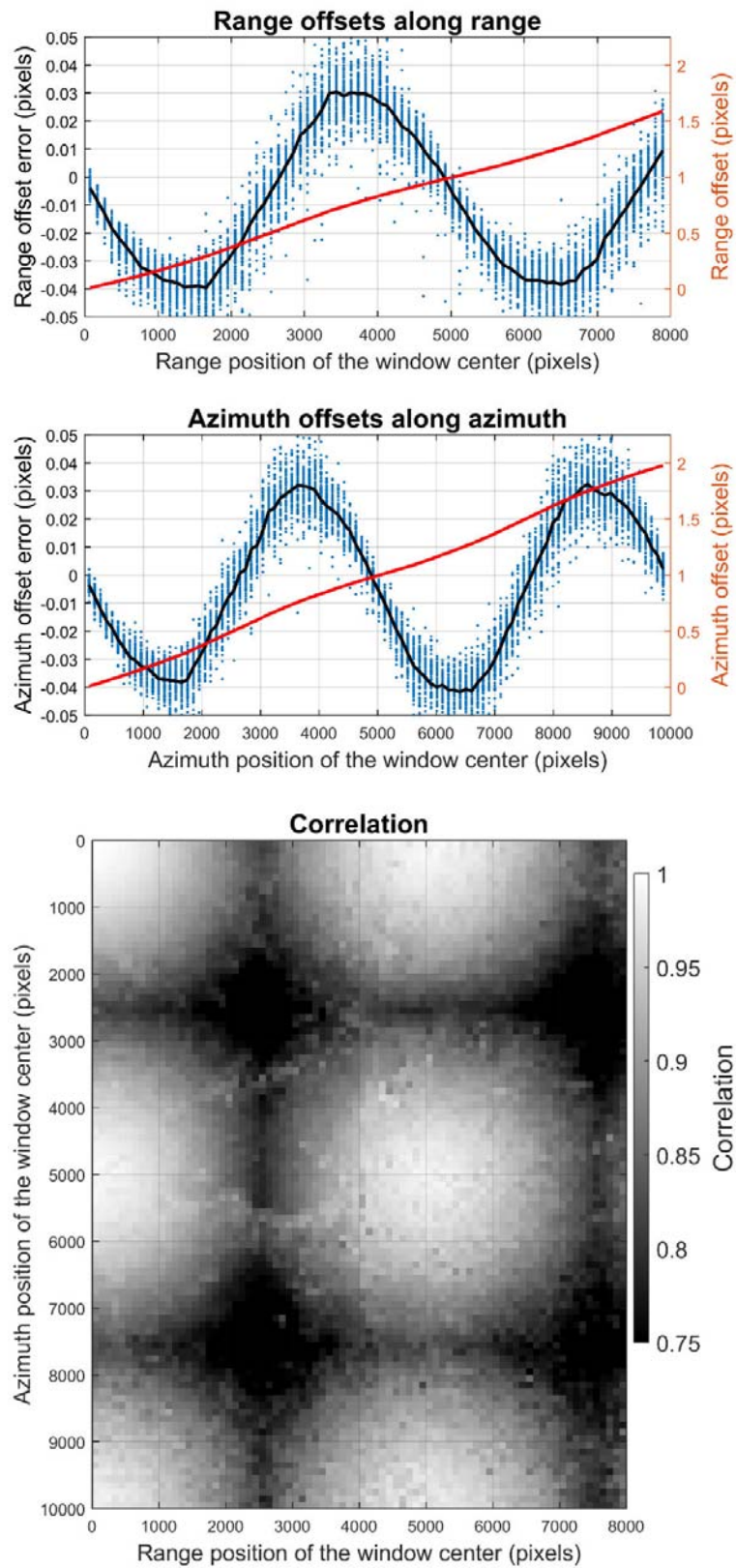


Fig. 5 Original version of offset_pwr_tracking program: 128x128 patch size, no SLC oversampling, 4x cross-correlation function oversampling factor. Processing time: 6.836 (s).

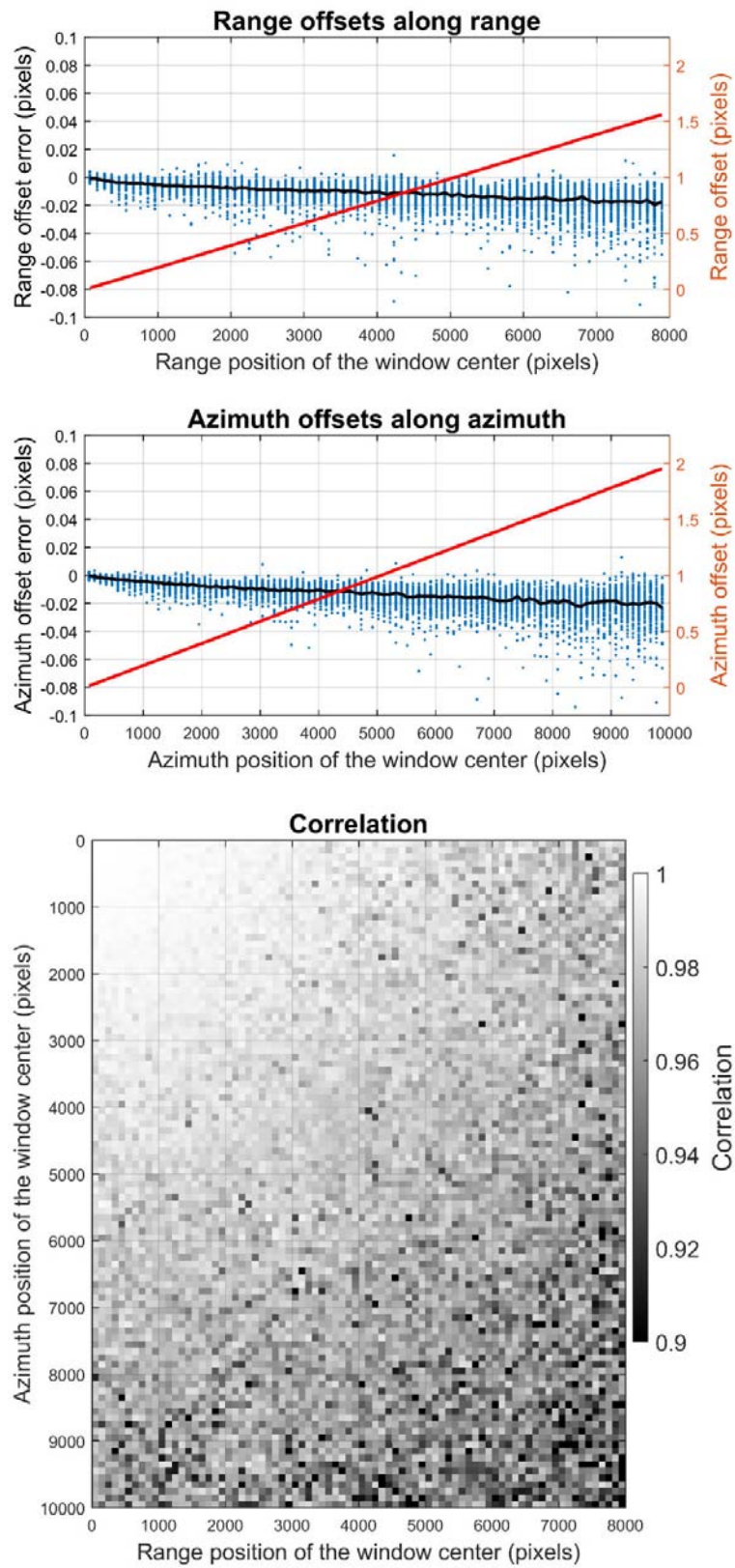


Fig. 6 Updated version of offset_pwr_tracking program: 32x32 patch size, 2x SLC oversampling, Lanczos 9 interpolation, no deramp, 1.0 bandwidth fraction. Processing time: 4.966 (s).

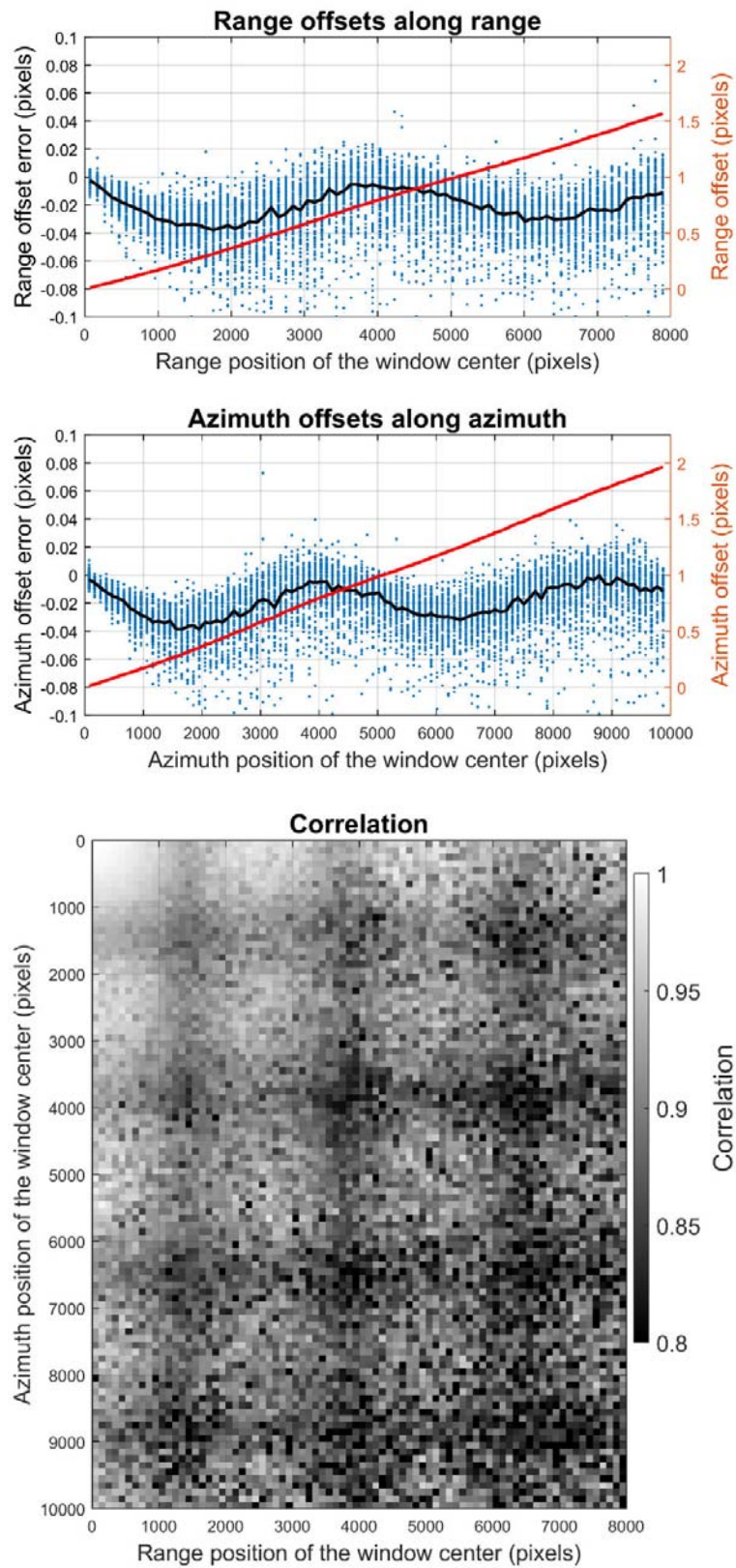


Fig. 7 Original version of offset_pwr_tracking program: 32x32 patch size, 2x SLC oversampling, 4x cross-correlation function oversampling factor. Processing time: 2.067 (s).

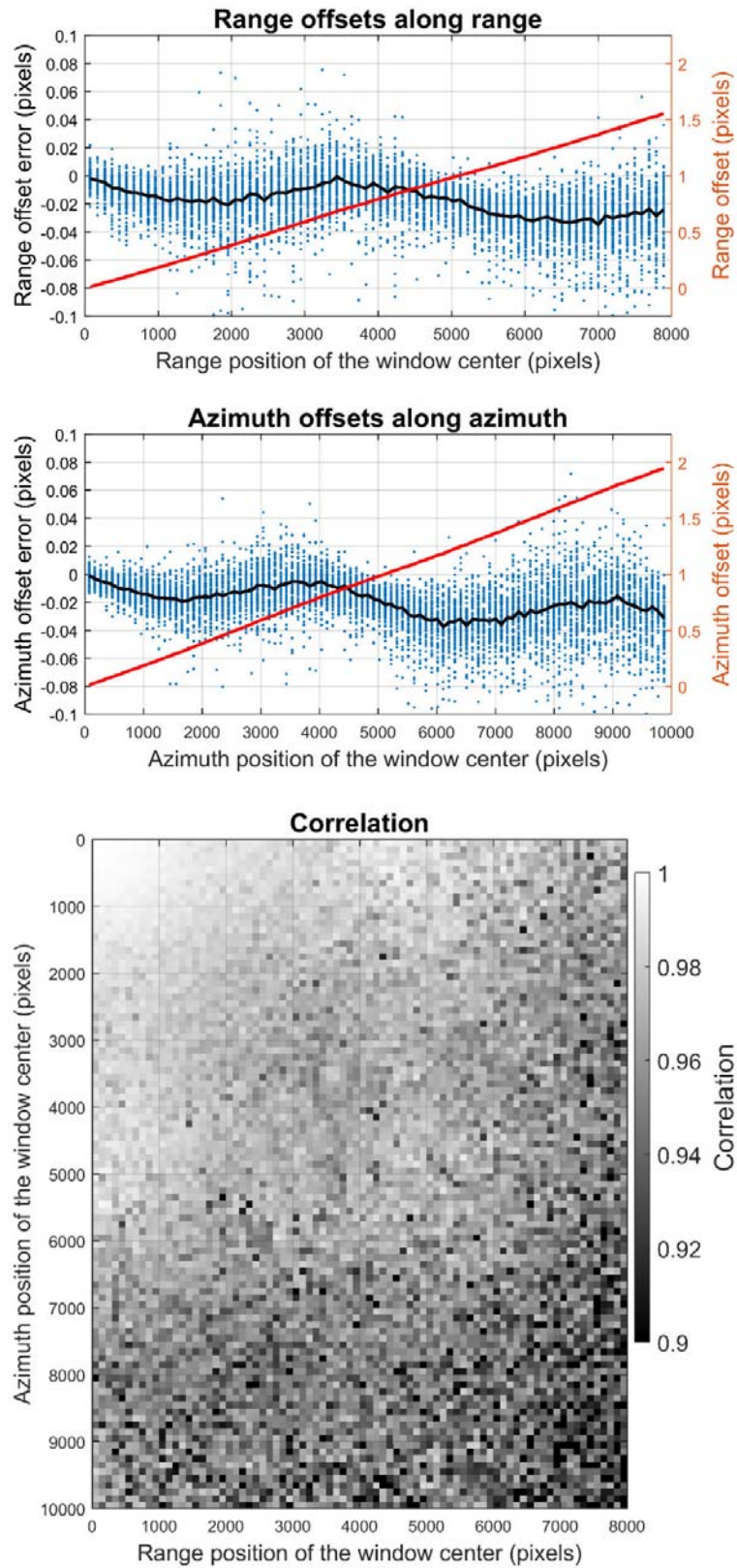


Fig. 8 Updated version of offset_pwr_tracking program: 32x32 patch size, no SLC oversampling, no deramp, 1.0 bandwidth fraction. Processing time: 0.881 (s).

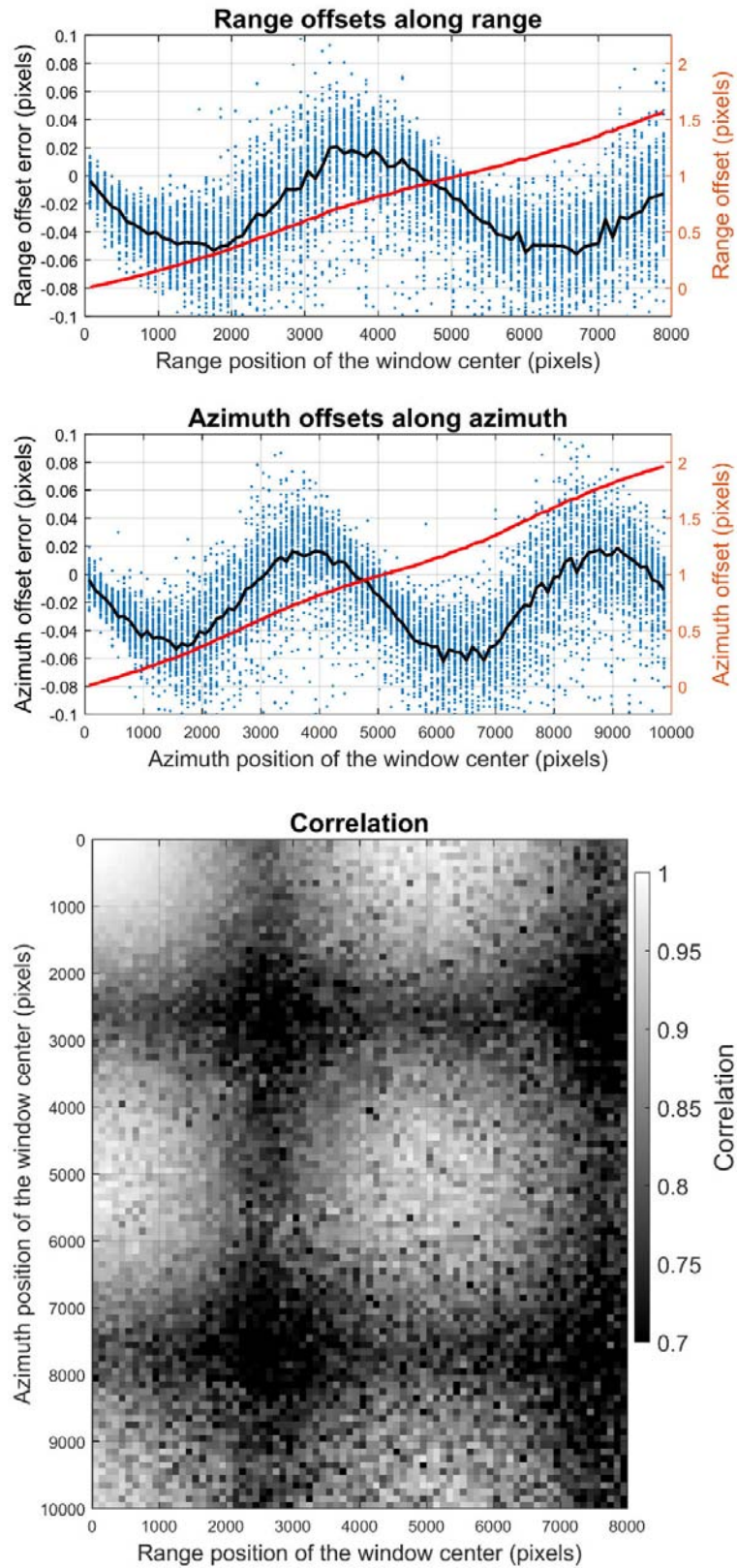


Fig. 9 Original version of offset_pwr_tracking program: 32x32 patch size, no SLC oversampling, 4x cross-correlation function oversampling factor. Processing time: 1.744 (s).

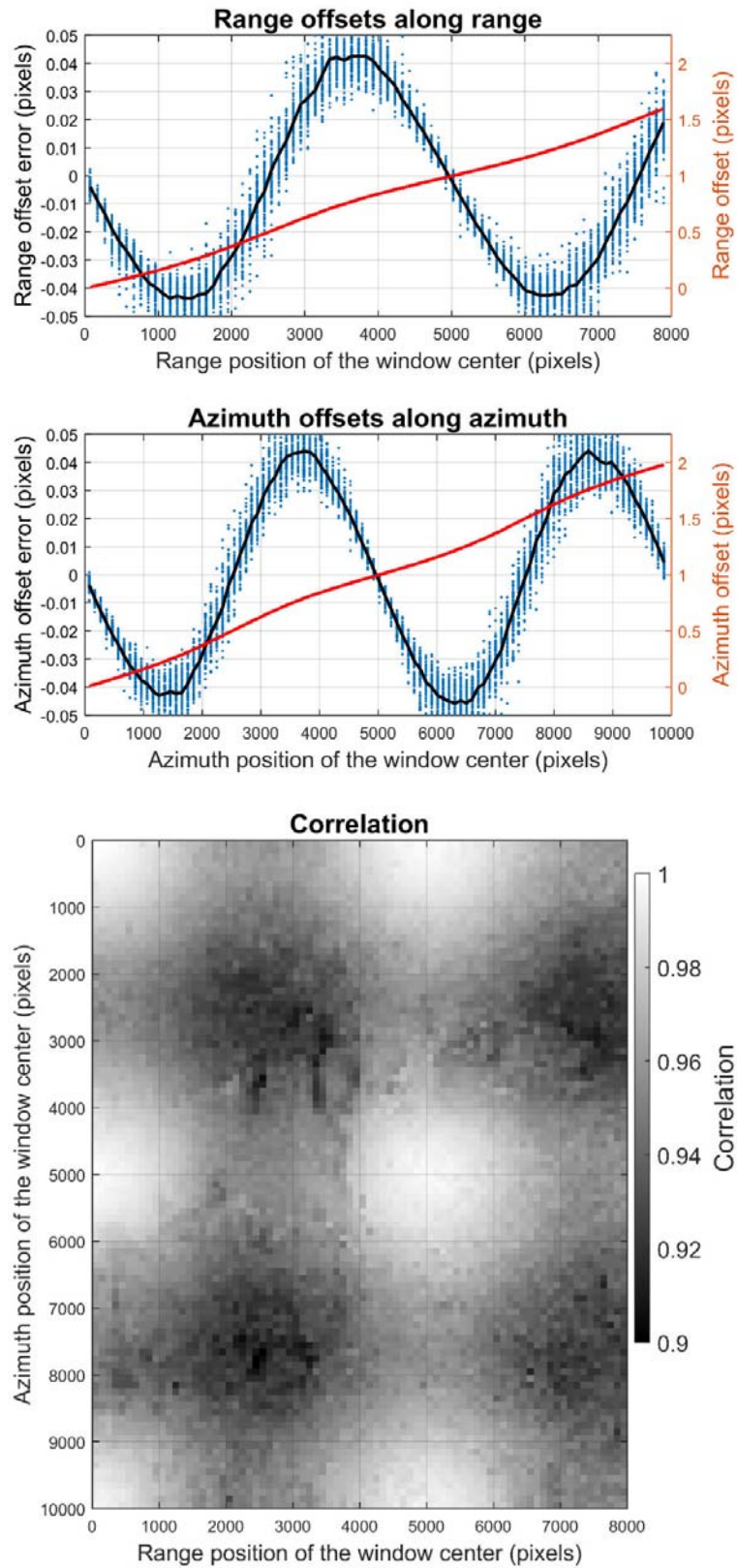


Fig. 10 Updated version of offset_pwr_trackingm program: 128x128 patch size, no MLI oversampling, 1.0 bandwidth fraction. Processing time: 2.163 (s).

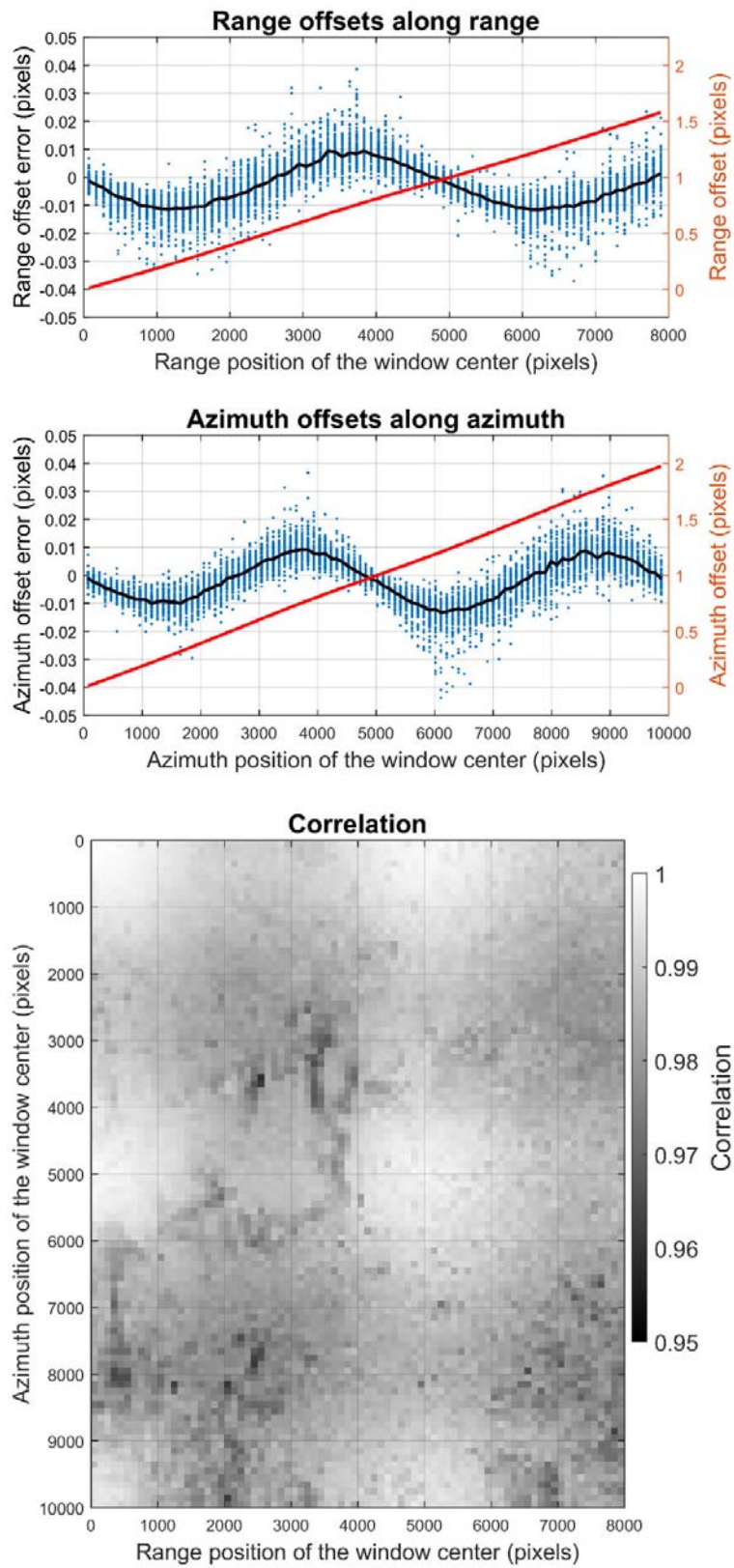


Fig. 11 Updated version of offset_pwr_trackingm program: 128x128 patch size, no MLI oversampling, 0.8 bandwidth fraction (default value). Processing time: 2.766 (s).

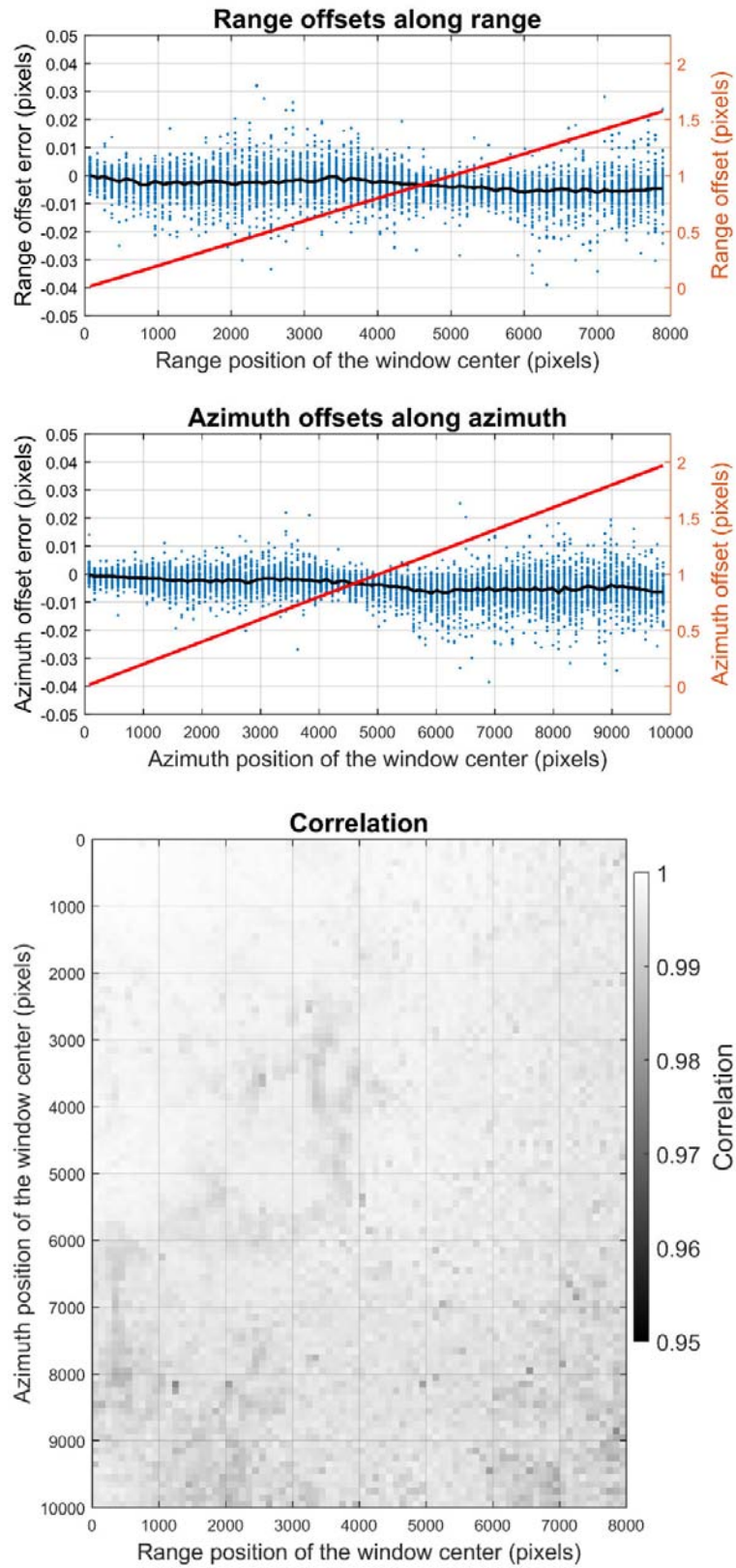


Fig. 12 Updated version of offset_pwr_trackingm program: 128x128 patch size, no MLI oversampling, 0.5 bandwidth fraction. Processing time: 2.863 (s).

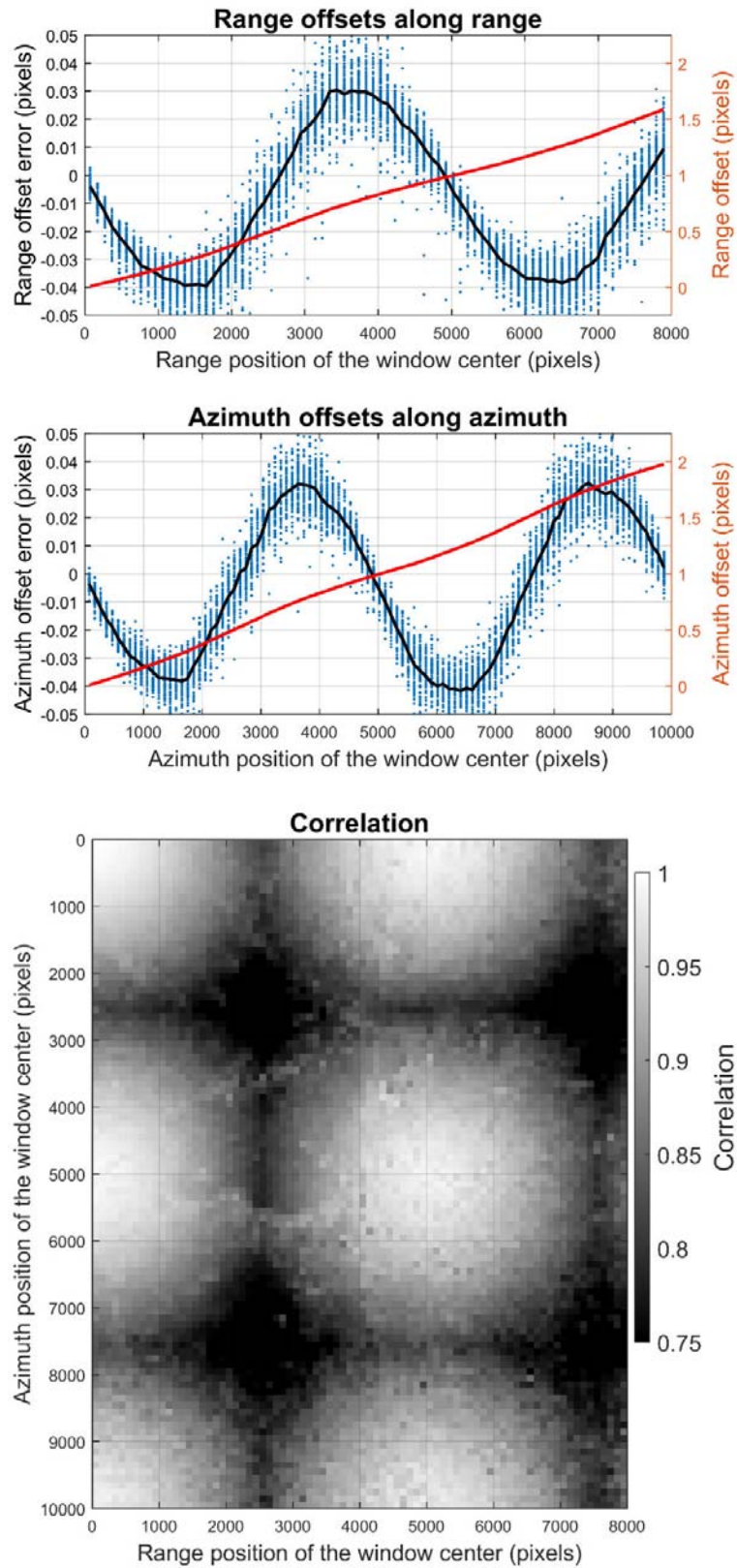


Fig. 13 Original version of offset_pwr_trackingm program: 128x128 patch size, no MLI oversampling, 4x cross-correlation function oversampling factor. Processing time: 1.615 (s).

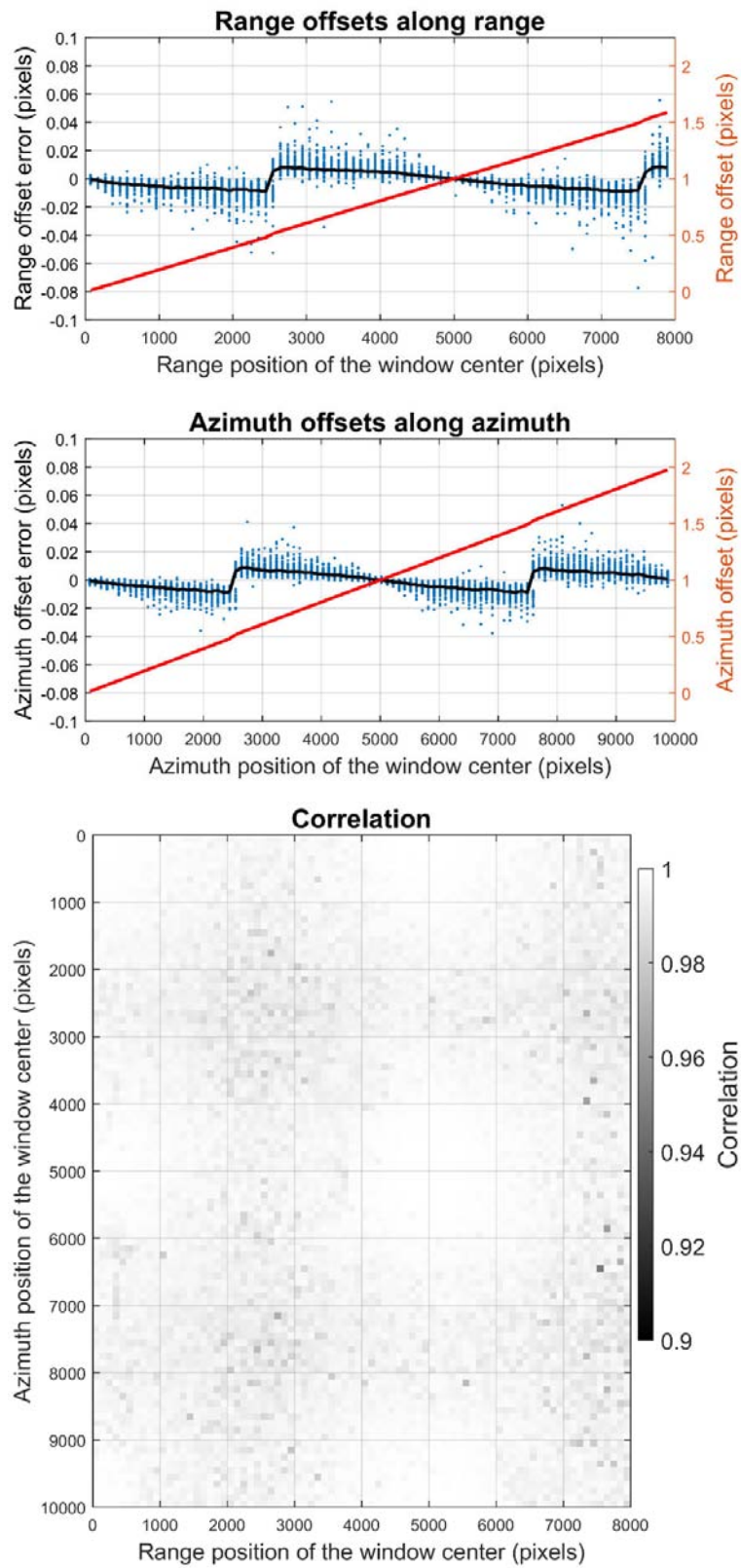


Fig. 14 Updated version of offset_pwr_tracking2 program using integer offsets from previous iteration: 32x32 patch size, 2x SLC oversampling, no deramp, 1.0 bandwidth fraction. Processing time: 1.905 (s).

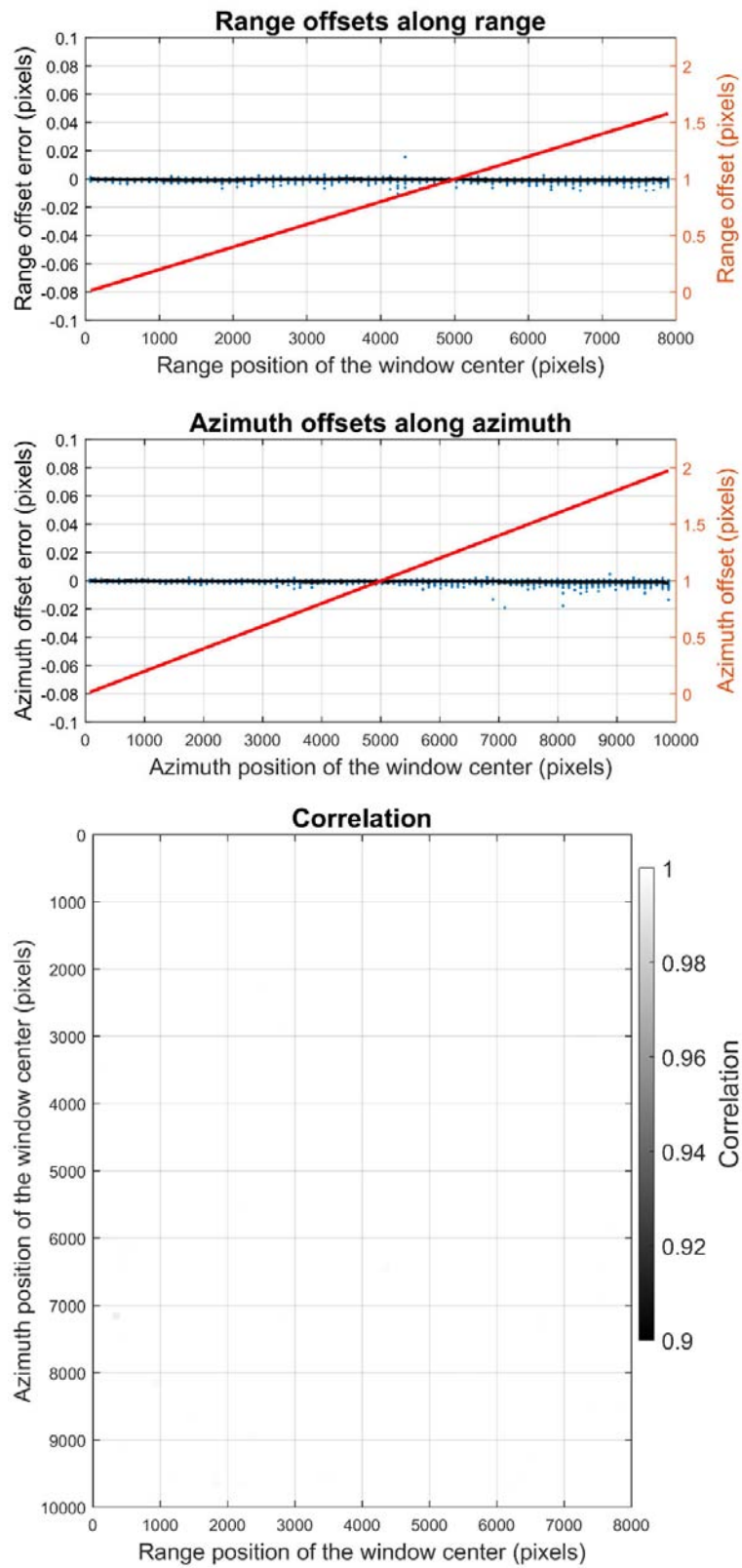


Fig. 15 Updated version of offset_pwr_tracking2 program using real-valued offsets from previous iteration: 32x32 patch size, 2x SLC oversampling, no deramp, 1.0 bandwidth fraction. Processing time: 1.971 (s).

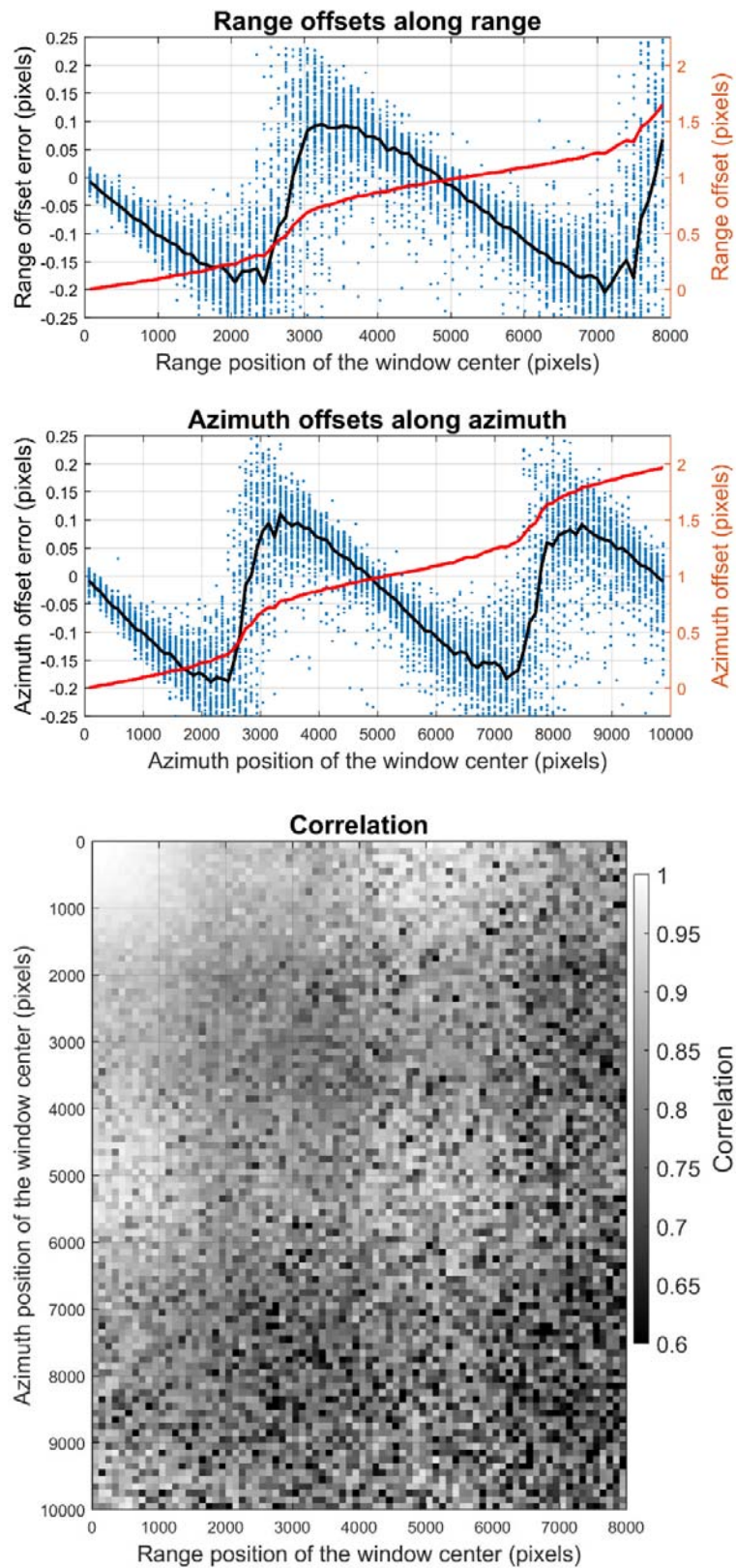


Fig. 16 Original version of offset_pwr_tracking program: 16x16 patch size, 2x SLC oversampling, no deramp, 1.0 bandwidth fraction. Processing time: 1.086 (s).

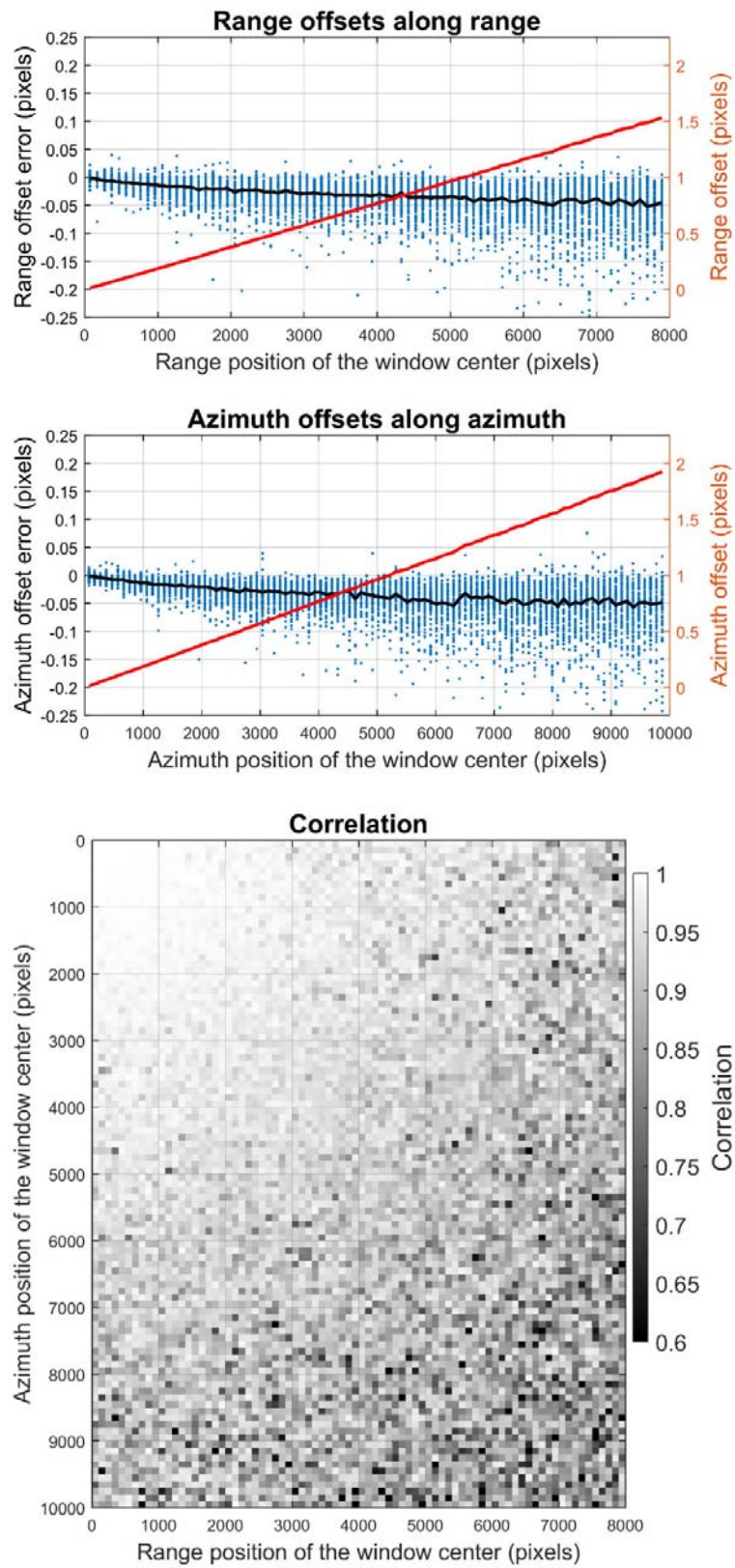


Fig. 17 Updated version of offset_pwr_tracking program: 16x16 patch size, 2x SLC oversampling, no deramp, 1.0 bandwidth fraction. Processing time: 3.304 (s).

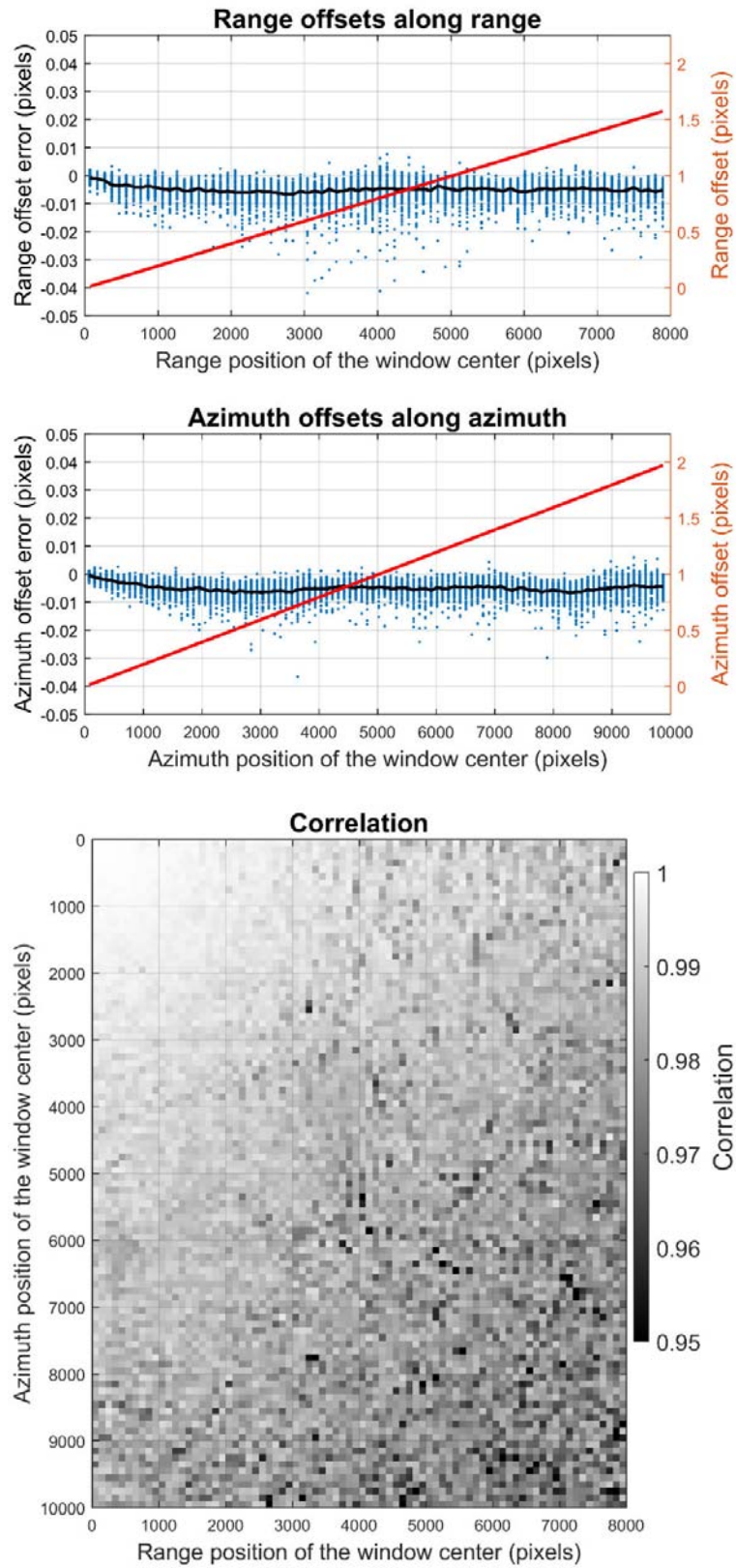


Fig. 18 Updated version of offset_pwr_tracking program without windowing: 128x128 patch size, 2x SLC oversampling, no deramp, 1.0 bandwidth fraction. Processing time: 21.700 (s).

3.2 Actual data

For the analysis using actual data, no absolute reference is available. From the results obtained using the synthetic pair (see Section 3.1), we consider the results obtained iteratively (*offset_pwr_tracking2* program) using the updated version and 2x oversampling as the best available reference.

3.2.1 TerraSAR-X – Venezia

This test consists of a pair of TerraSAR-X datasets acquired the 5th and 16th of February, 2011. Offsets were computed on the first 8000 lines (of 29700), since the second part mostly covers water. We used 64x64 patches separated by 32 pixels from each other. Offsets were estimated using the original and updated versions of *offset_pwr_tracking* program without oversampling and with 2x oversampling, as well as using the original and updated versions of *offset_pwr_tracking2* program with 2x oversampling.

The processing time and median correlation of the investigated scenarios are reported in Tab. 1. Fig. 19 shows the range offsets and Fig. 20 the correlation obtained using the updated version of *offset_pwr_tracking2*, with 2x oversampling (after a first estimation using *offset_pwr_tracking*). Fig. 21-25 compare the range offsets obtained with the various scenarios to those reported in Fig. 19-20. To enhance the readability, the figures show the median along azimuth of the offset difference. Note that it consists of a relative and not an absolute comparison.

The azimuth offsets are not reported since they are relatively constant over the whole image.

	Original 1x	Original 2x	Original iterative 2x	Updated 1x	Updated 2x	Updated iterative 2x
Processing Time [s]	17.971	37.484	31.318	15.237	73.993	65.084
Median correlation	0.2217	0.2343	0.2341	0.2613	0.2489	0.2485

Tab. 1 Processing times and median correlations for TerraSAR-X – Venezia example.

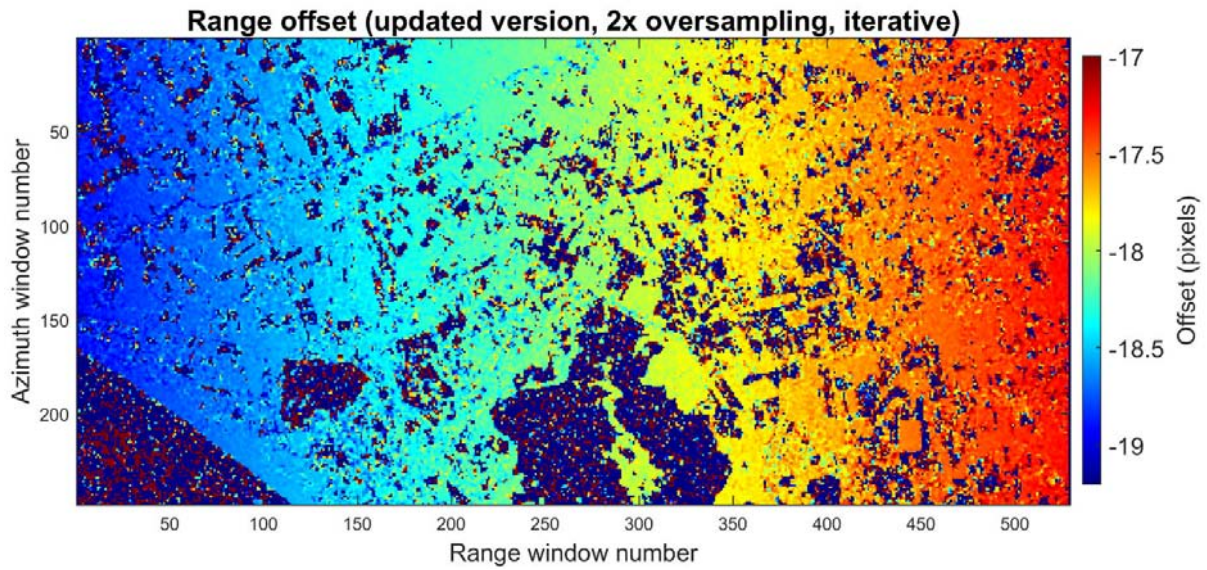


Fig. 19 Range offsets estimated using updated offset_pwr_tracking2 program: 64x64 patch size, 2x SLC oversampling, no deramp, 1.0 bandwidth fraction.

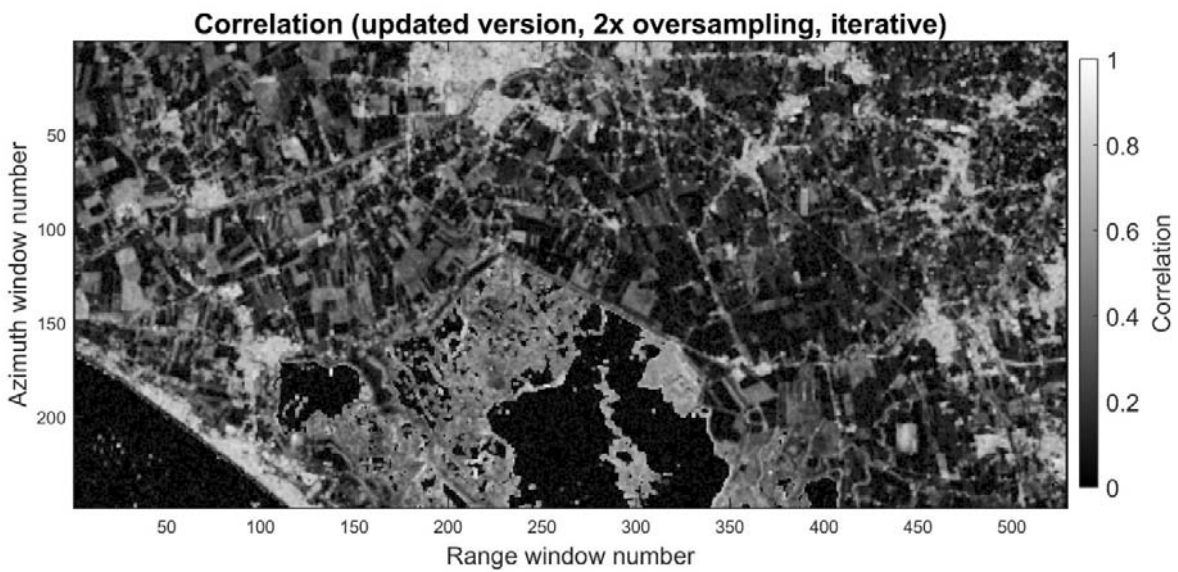


Fig. 20 Correlation obtained using updated offset_pwr_tracking2 program: 64x64 patch size, 2x SLC oversampling, no deramp, 1.0 bandwidth fraction.

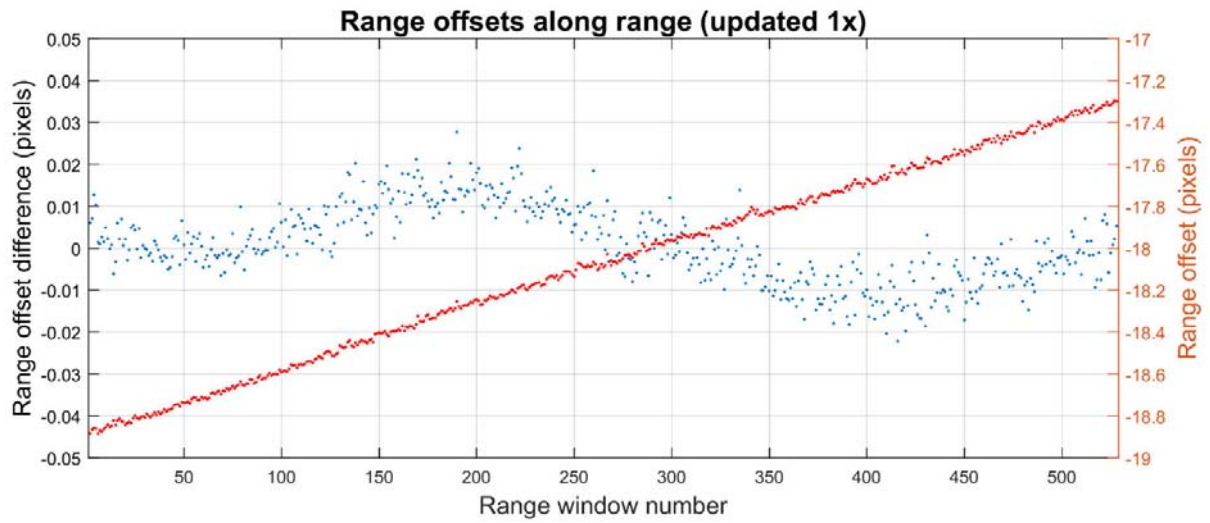


Fig. 21 Comparison between the updated version of `offset_pwr_tracking` program using no oversampling and the updated version of `offset_pwr_tracking2` using 2x oversampling.

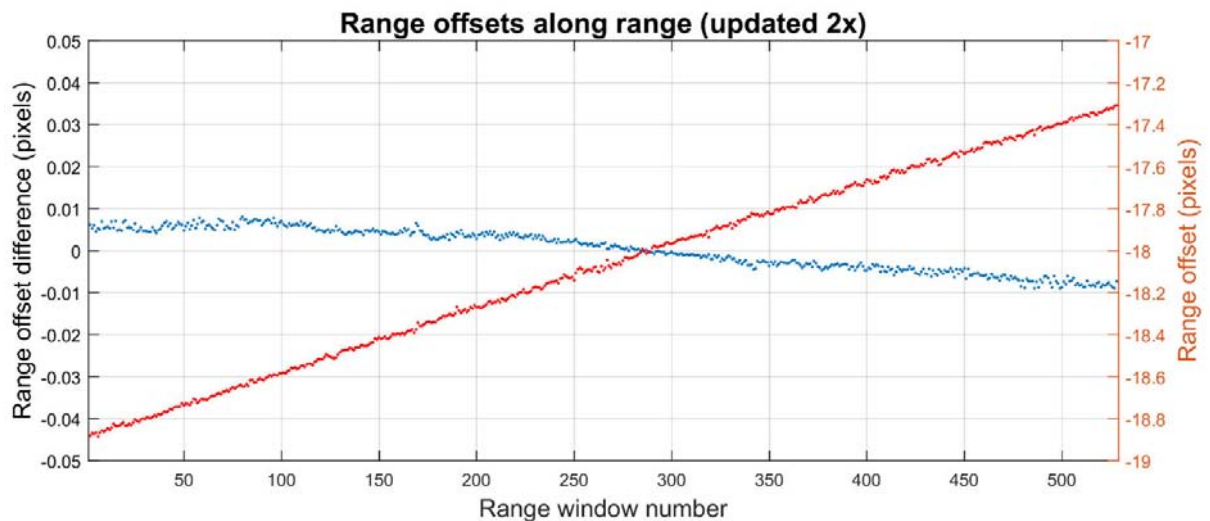


Fig. 22 Comparison between the updated version of `offset_pwr_tracking` program using 2x oversampling and the updated version of `offset_pwr_tracking2` using 2x oversampling.

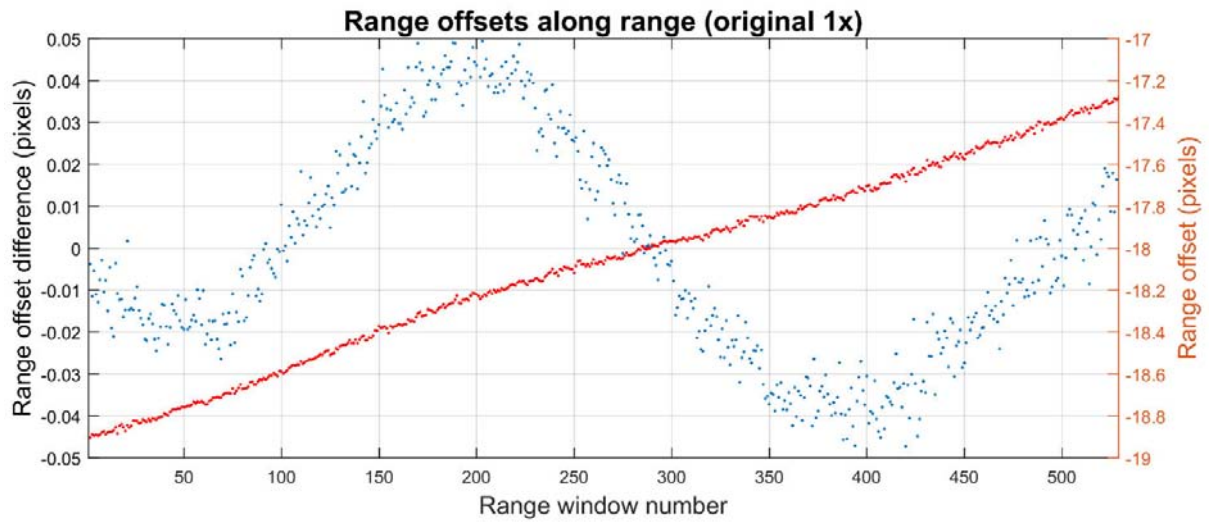


Fig. 23 Comparison between the original version of `offset_pwr_tracking` program using no oversampling and the updated version of `offset_pwr_tracking2` using 2x oversampling.

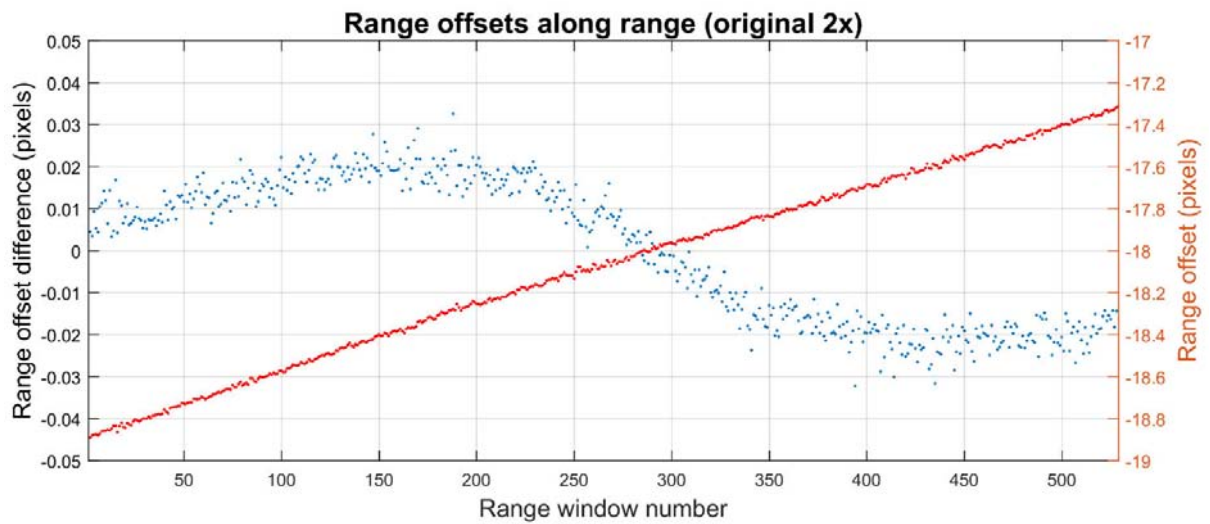


Fig. 24 Comparison between the original version of `offset_pwr_tracking` program using 2x oversampling and the updated version of `offset_pwr_tracking2` using 2x oversampling.

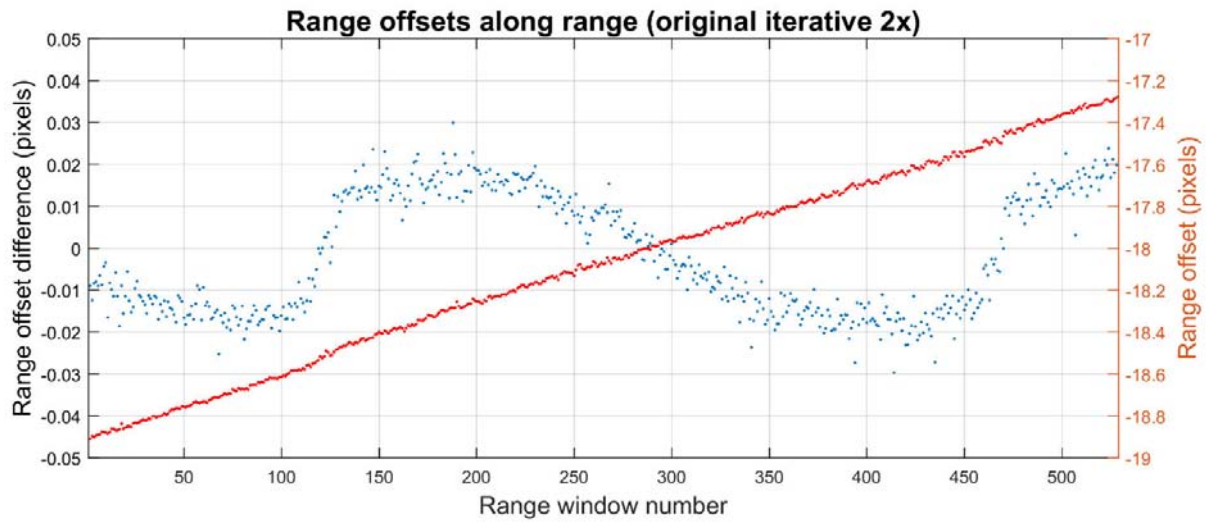


Fig. 25 Comparison between the original version of `offset_pwr_tracking2` program using 2x oversampling and the updated version of `offset_pwr_tracking2` using 2x oversampling.

3.2.2 Devon Ice Cap

The Devon Ice Cap example consists of two Sentinel-1A TOPS acquisitions made the 17th and 29th of January, 2015, over the Devon Ice Cap in Northern Canada. The acquisitions allow estimating glacial flow. The TOPS data were deramped prior to testing the offset estimation programs, to avoid introducing aliasing errors.

Tab. 2 reports the processing times obtained using the original and updated versions without and with 2x oversampling, as well as using the original and updated versions of `offset_pwr_tracking2` program with 2x oversampling. A multi-look image, the range and azimuth offsets estimated using the updated `offset_pwr_tracking2` program using 2x oversampling, and the estimated correlation are shown in Fig. 26-29.

The results obtained using the updated version of `offset_pwr_tracking2`, 2x oversampling, were considered as the reference. We compared the five other results to that reference. The R^2 , RMSE, and slope of the regression line are reported in Tab. 3-5. For the statistical computation, outliers were discarded ($-1.5 \leq \text{offsets} \leq 1.5$). As an example, Fig. 30-32 show scatter plots comparing the results obtained using the original `offset_pwr_tracking` program without oversampling vs. the reference.

	Original 1x	Original 2x	Original iterative 2x	Updated 1x	Updated 2x	Updated iterative 2x
Processing Time [s]	283.969	1447.917	1455.642	272.499	1958.478	2870.877

Tab. 2 Processing times for Sentinel-1 Devon Ice Cap example.

Range offsets	Original 1x	Original 2x	Original iterative 2x	Updated 1x	Updated 2x
Adjusted R2	0.9010	0.9515	0.9513	0.9234	0.9932
RMSE (pixels)	0.0187	0.0139	0.0140	0.0194	0.0054
Slope	0.8575	0.9359	0.9437	1.0130	0.9927

Tab. 3 Statistical comparison of the range offsets relative to those estimated using the updated version, 2x oversampling.

Azimuth offsets	Original 1x	Original 2x	Original iterative 2x	Updated 1x	Updated 2x
Adjusted R ²	0.8366	0.8976	0.9012	0.8910	0.9872
RMSE (pixels)	0.0179	0.0138	0.0137	0.0176	0.0055
Slope	0.8507	0.8588	0.8697	1.0255	0.9811

Tab. 4 Statistical comparison of the azimuth offsets relative to those estimated using the updated version, 2x oversampling.

Correlation	Original 1x	Original 2x	Original iterative 2x	Updated 1x	Updated 2x
Adjusted R ²	0.9859	0.9858	0.9857	0.9987	1.0000
RMSE	0.0201	0.0202	0.0202	0.0061	0.0004
Slope	0.9983	0.9937	0.9936	0.9972	0.9998

Tab. 5 Statistical comparison of the correlation relative to that estimated using the updated version, 2x oversampling.

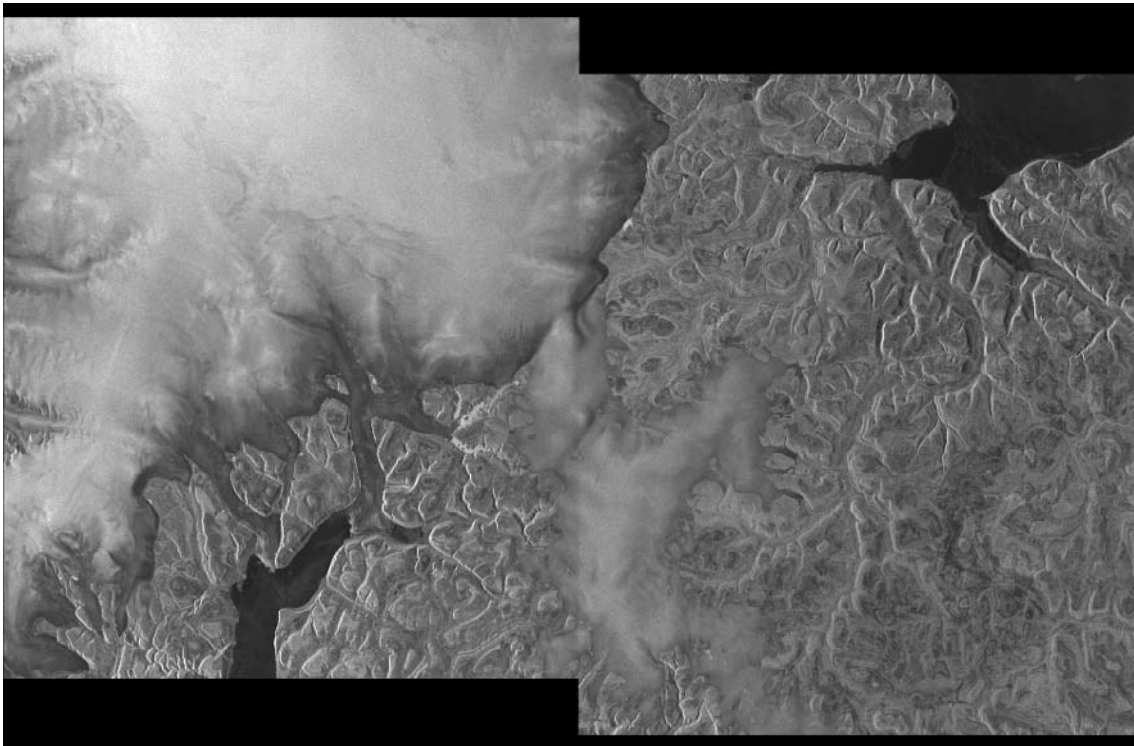


Fig. 26 Multi-look (50 range x 10 azimuth looks) image of Devon Ice Cap first acquisition.

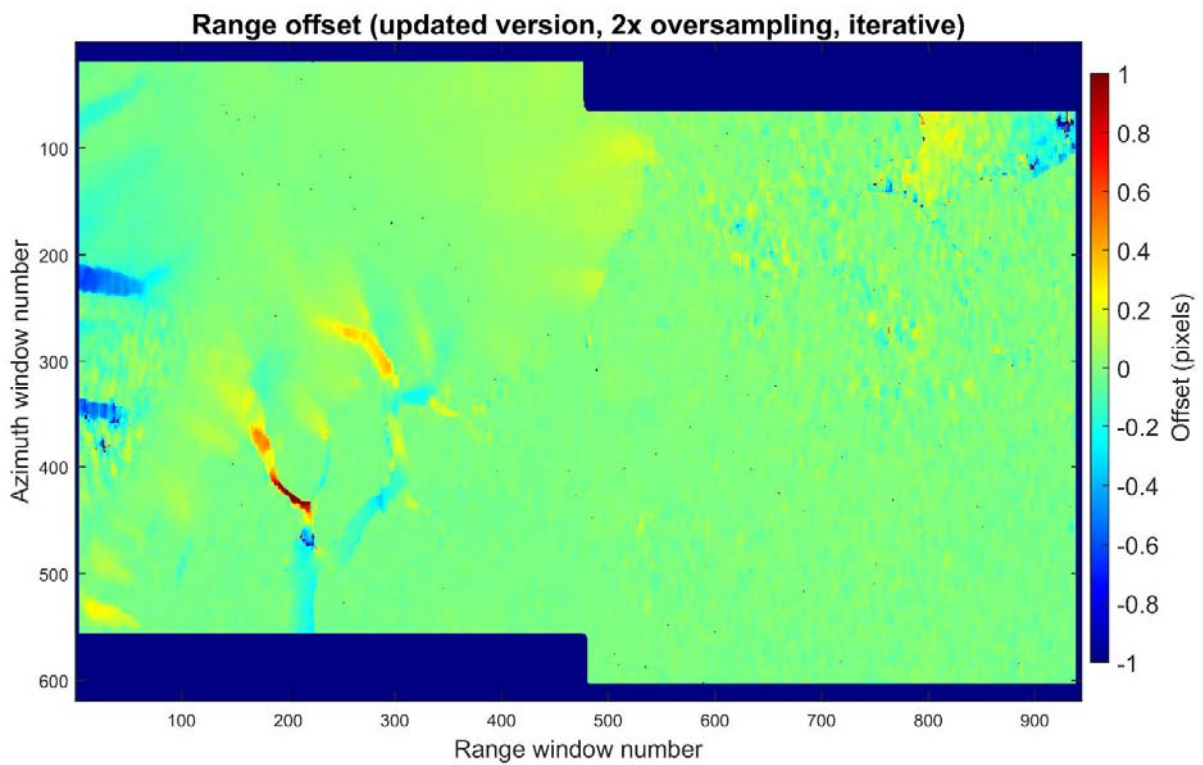


Fig. 27 Range offsets estimated using updated offset_pwr_tracking2 program: 256x128 patch size, 2x SLC oversampling, no deramp, 1.0 bandwidth fraction.

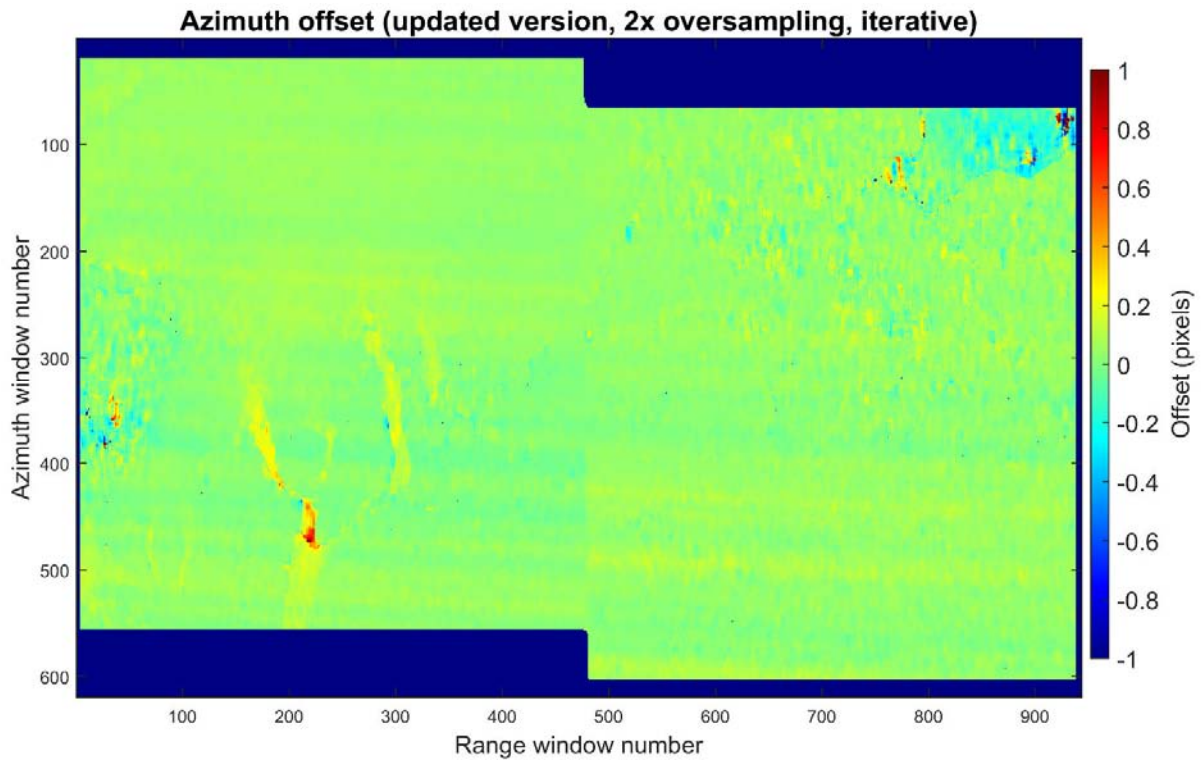


Fig. 28 Azimuth offsets estimated using updated offset_pwr_tracking2 program: 256x128 patch size, 2x SLC oversampling, no deramp, 1.0 bandwidth fraction.

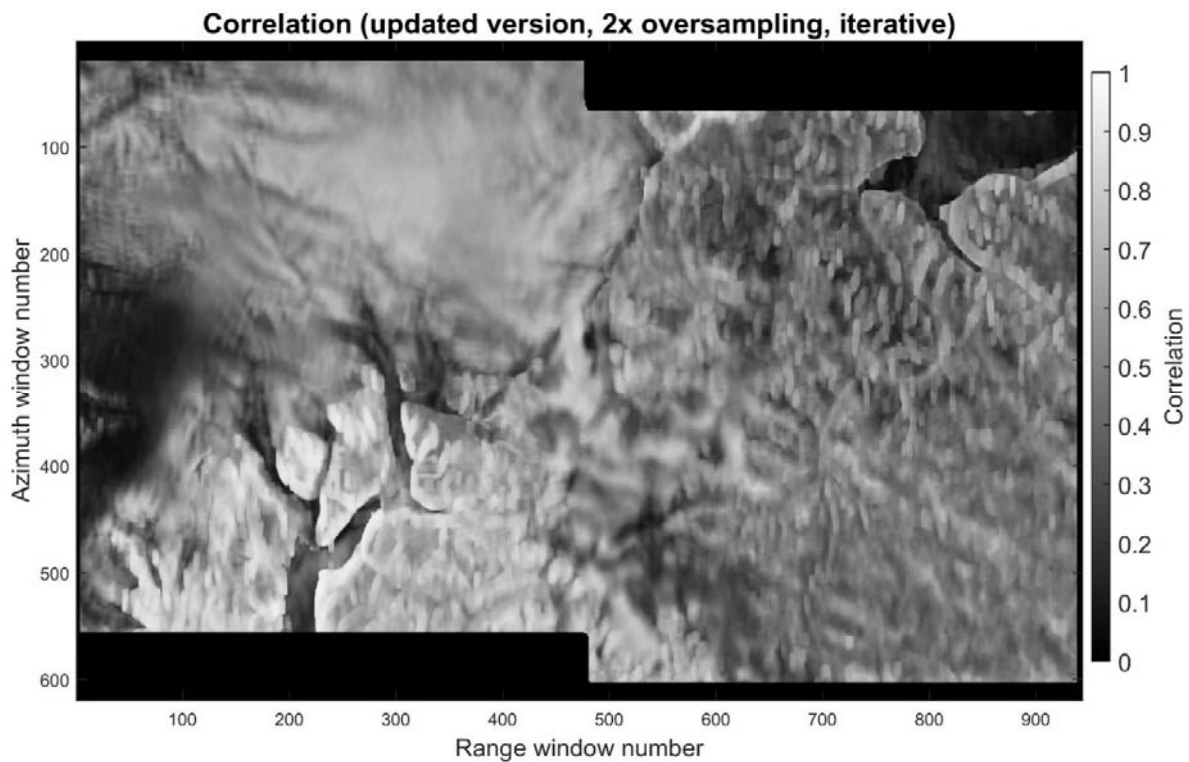


Fig. 29 Correlation estimated using updated offset_pwr_tracking2 program: 256x128 patch size, 2x SLC oversampling, no deramp, 1.0 bandwidth fraction.

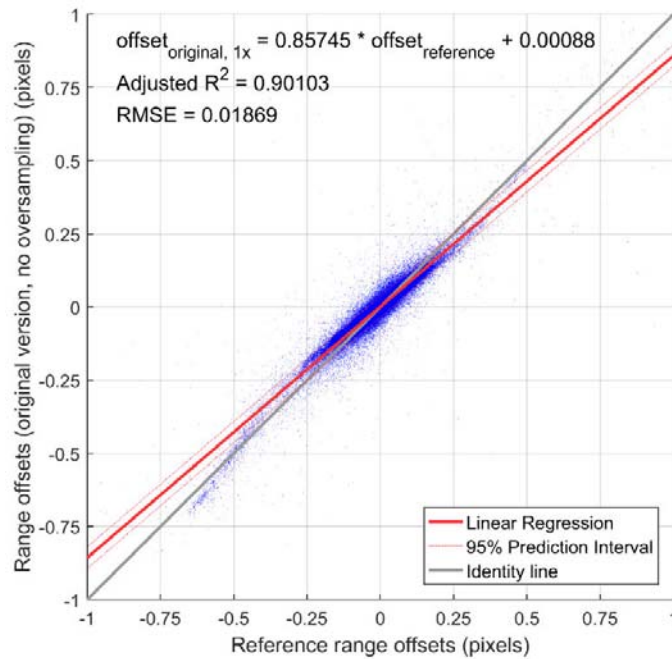


Fig. 30 Scatter plot comparing the range offsets obtained using the original offset_pwr_tracking program without oversampling vs. the updated version of offset_pwr_tracking2 using 2x oversampling.

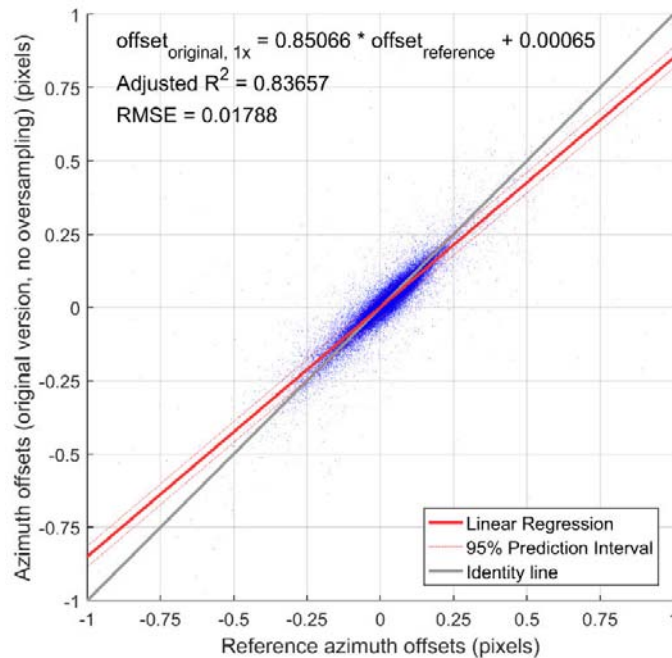


Fig. 31 Scatter plot comparing the azimuth offsets obtained using the original offset_pwr_tracking program without oversampling vs. the updated version of offset_pwr_tracking2 using 2x oversampling.

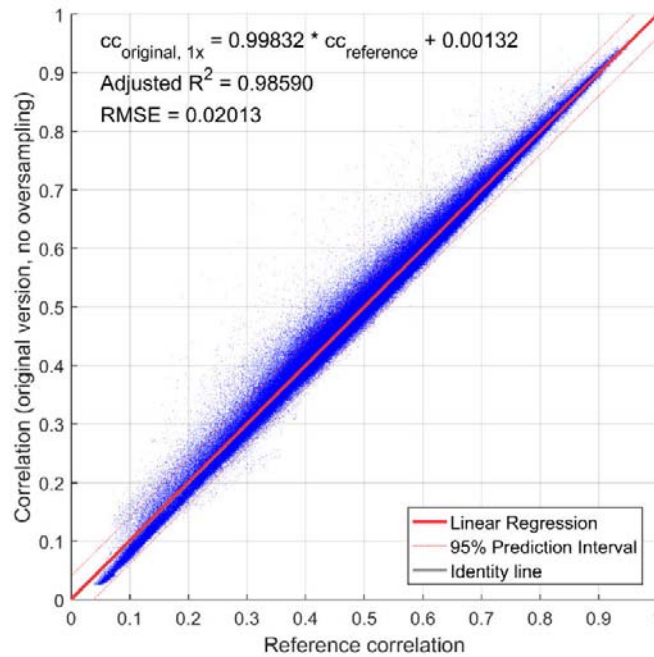


Fig. 32 Scatter plot comparing the correlation obtained using the original offset_pwr_tracking program without oversampling vs. the updated version of offset_pwr_tracking2 using 2x oversampling.

4 DISCUSSION

The offset estimation using the GAMMA software was already very effective before the update. The improvements brought by this update concerns biases in the estimated offsets typically lower than 1/10 of a pixel. For the offset estimation on SLCs, when large patches and 2x oversampling are used, the differences between the updated version and the previous one should be barely noticeable.

The update successfully addresses the shortcomings identified in Section 1.

4.1 Sinusoidal biases

Biases varying between plus and minus 0.05 pixels when no oversampling is used can now be reduced using low-pass filtering of the patches implemented using a window function in the Fourier domain.

Results on both synthetic and actual data show that the bandwidth reduction of the intensity data (80% of original sampling frequency) decreased these biases by $\sim 3/4$. This sinusoidal bias reduction can be seen by comparing Fig. 4 to Fig. 5, Fig. 8 to Fig. 9, the series from Fig. 10 to Fig. 13, or, in the case of actual data, by comparing Fig. 21 to Fig. 23. As a drawback, the filtering increases the offset uncertainty in some cases. This can be observed in Tab. 3 where the RMSE was higher for the updated version compared to the original version, despite an improved R^2 value and a regression line closer to the identity line.

4.2 Offset underestimation

While not removing it completely, the offset underestimation was considerably reduced through the use of a window applied to the intensity data. (In the case of SLC data, the intensity is obtained after the optional oversampling step by multiplying each sample by its complex conjugate.) This can be clearly seen by comparing the results shown in Fig. 2 and Fig. 18. We also observe a decrease of the measurement uncertainty (lower spread of the offset estimates). The efficiency of the windowing to reduce the offset underestimation decreases with the offset magnitude.

As a drawback, the correlation of decorrelated areas is slightly increased. This can be observed in Fig. 32 where values < 0.3 are typically ~ 0.02 higher than using the original program.

4.3 Iterative processing

The iterative processing using *offset_pwr_tracking2* and *offset_pwr_trackingm2* effectively removes the offset underestimation. Hence, it is very useful when using relatively small patches (64x64 or smaller). Results obtained using larger patches are only marginally improved, as the offset underestimation is expected to be very limited in that case (see Fig. 2 and last column in Tab. 3-5).

In the case of SLC data, using iterative processing, no sinusoidal bias is observed even when estimating offsets without oversampling. However, in the case of MLI data, iterative processing does not cancel sinusoidal biases caused by aliasing already present in the intensity data.

4.4 Interpolation changes - issues using very small patches

The large inaccuracies using very small patches in combination with oversampling have been solved by the new / updated interpolation methods (see Fig. 16 and 17). Improved interpolation was also partially responsible for the general improvement of the offset estimation results, as can be observed by comparing Fig. 2 to Fig. 3 and Fig. 6 to Fig. 7.

4.5 Peak detection

The peak detection algorithm was successfully replaced by the gradient descent algorithm. While the effect on the accuracy of the peak detection is barely noticeable at the scale reported in the figures of Section 3, the new algorithm is efficient and accurate, and delivers a much-improved estimation of the correlation value. The patterns seen in Fig. 3, 5, 7, 9 and 13 were much reduced. This improvement can be best seen by comparing Fig. 10 to Fig. 13, since other program modifications were minimal there. In many tested cases, thanks to the other program modifications, these patterns completely disappeared.

4.6 Processing time

The processing time was reported in the results for the tested scenarios. As a general remark, the processing time is typically higher. Depending on the options, processing time should range between

~0.8 times and ~2 times the original processing time. Note that we used the Lanczos 9 interpolation method for several tests. Using a Lanczos 5 interpolation method would decrease the processing time, while still keeping a good interpolation quality.

Several parameters influence the processing time of the updated version relative to the original version:

- For SLC input, the deramping is now optional, hence when no deramping is carried out, it speeds up the processing.
- In *offset_pwr*, *offset_pwr_tracking*, *offset_pwr_m*, and *offset_pwr_trackingm*, oversampling is now carried out using the Lanczos interpolation method over the whole range width. This interpolation method is slower than FFT-based method for small data, such as when interpolating only patches. However, it is faster for the large range widths typical of spaceborne SAR images. As an overall result, the interpolation step is now slower for sparsely spread patches, and faster for patches that overlap (i.e. when the step size is smaller than the patch size).
- In *offset_pwr_tracking2*, *offset_pwr_trackingm2*, and *offset_pwr_list*, oversampling is carried out as before, i.e. separately on each patch. However, a buffer was added at the edges of the patches to improve the interpolation quality. Hence the interpolation step will be slightly slower.
- The optional low-pass filter on SLC data increases the processing time.
- The low-pass filter on intensity data requires an additional FFT and IFFT on the patches, which slows down the processing. For time-critical processing, it can be disabled in *offset_pwr*, *offset_pwr_tracking*, *offset_pwr_tracking2*, and *offset_pwr_list* using the *int_filt* option. Disabling this filter is not recommended when no oversampling is applied.
- The data shift in *offset_pwr_tracking2* and *offset_pwr_trackingm2* includes sine and cosine computations that are computationally expensive. Although the number of computations is limited, it still noticeably slows down the processing.
- The other modifications only have minor influence on the processing time.

4.7 Outlook

The bias observed when estimating offsets on aliased data might be improved by characterizing the bias using the data itself. A set of shifts between 0 and 1 pixels would be applied to a patch; these shifts would be estimated against the non-shifted patch using the same parameters as selected by the user, and their deviation from the nominal value would be used to generate a correction function. The correction would then be applied to the effective offset estimates. Such an algorithm could be used when no oversampling is applied to the data, providing results without sinusoidal bias even when no low-pass filter is applied to the intensity data. It would also be applicable when the SLC data are not available as occurs with geocoding refinement and when the input data products are already detected.

The most important point in this update was to provide more accurate offset estimates, while keeping computing requirements within limits. Some optimizations are nevertheless probably possible, such as limiting the number of FFTs and improving block processing to reduce multiple interpolation of the same pixels, at the cost of possible greater memory usage.

5 REFERENCES

- [1] W. Burger, and M.J. Burge, Principles of digital image processing: Core Algorithms, London: Springer, 2009, *pp.* 231–232.
- [2] J. Snyman, Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms (Vol. 97), Springer Science & Business Media, 2005, *pp.* 40–42.

6 APPENDIX: PROGRAMS USAGE

The cross-correlation function oversampling factor option was removed (c_ovr command line parameter), because peak detection now uses a gradient descent algorithm that incorporates high order interpolation. New options (Lanczos interpolation order, bandwidth fraction, deramp flag, intensity low-pass filter flag) are marked in blue.

6.1 Offset_pwr

usage: offset_pwr <SLC1> <SLC2> <SLC1_par> <SLC2_par> <OFF_par> <offs> <ccp> [rwin] [azwin] [offsets] [n_ovr] [nr] [naz] [thres] [lanczos] [bw_frac] [deramp] [int_filt] [pflag] [pltflg] [ccs]

input parameters:

SLC1 (input) single-look complex image 1 (reference)
 SLC2 (input) single-look complex image 2
 SLC1_par (input) SLC-1 ISP image parameter file
 SLC2_par (input) SLC-2 ISP image parameter file
 OFF_par (input) ISP offset/interferogram parameter file
 offs (output) offset estimates in range and azimuth (fcomplex)
 ccp (output) cross-correlation of each patch (0.0->1.0) (float)
 rwin range patch size (range pixels, enter - for default from offset parameter file)
 azwin azimuth patch size (azimuth lines, enter - for default from offset parameter file)
 offsets (output) range and azimuth offsets and cross-correlation data in text format, enter - for no output
 n_ovr SLC oversampling factor (integer 2**N (1,2,4), enter - for default: 2)
 nr number of offset estimates in range direction (enter - for default from offset parameter file)
 naz number of offset estimates in azimuth direction (enter - for default from offset parameter file)
 thres cross-correlation threshold (0.0->1.0) (enter - for default from offset parameter file)
 lanczos Lanczos interpolator order 5 -> 9 (enter - for default: 5)
 bw_frac bandwidth fraction of low-pass filter on complex data (0.0->1.0) (enter - for default: 1.0)
 deramp deramp SLC phase flag (enter - for default)
 0: no deramp (Doppler centroid close to 0) (default)
 1: deramp SLC phase
 int_filt intensity low-pass filter flag (enter - for default)
 0: no filter
 1: low-pass filter of intensity data, highly recommended when no oversampling used (default)
 pflag print flag (enter - for default)
 0: print offset summary (default)
 1: print all offset data
 pltflg plotting flag (enter - for default)
 0: none (default)
 1: screen output
 2: screen output and PNG format plots
 3: output plots in PDF format
 ccs (output) cross-correlation standard deviation of each patch (float)

NOTE: ScansAR and TOPS data need to be deramped when oversampling the SLCs (n_ovr > 1)

6.2 Offset_pwr_tracking

usage: offset_pwr_tracking <SLC1> <SLC2> <SLC1_par> <SLC2_par> <OFF_par> <offs> <ccp> [rwin] [azwin] [offsets] [n_ovr] [thres] [rstep] [azstep] [rstart] [rstop] [azstart] [azstop] [lanczos] [bw_frac] [deramp] [int_filt] [pflag] [pltflg] [ccs]

input parameters:

SLC1 (input) single-look complex image 1 (reference)
 SLC2 (input) single-look complex image 2
 SLC1_par (input) SLC-1 ISP image parameter file
 SLC2_par (input) SLC-2 ISP image parameter file

OFF_par (input) ISP offset/interferogram parameter file
 offs (output) offset estimates in range and azimuth (fcomplex)
 ccp (output) cross-correlation of each patch (0.0->1.0) (float)
 rwin range patch size (range pixels, enter - for default from offset parameter file)
 azwin azimuth patch size (azimuth lines, enter - for default from offset parameter file)
 offsets (output) range and azimuth offsets and cross-correlation data in text format,
 enter - for no output
 n_ovr SLC oversampling factor (integer 2**N (1,2,4), enter - for default: 2)
 thres cross-correlation threshold (0.0->1.0) (enter - for default from offset parameter file)
 rstep step in range pixels (enter - for default: rwin/2)
 azstep step in azimuth pixels (enter - for default: azwin/2)
 rstart offset to starting range pixel (enter - for default: 0)
 rstop offset to ending range pixel (enter - for default: nr-1)
 azstart offset to starting azimuth line (enter - for default: 0)
 azstop offset to ending azimuth line (enter - for default: nlines-1)
 lanczos Lanczos interpolator order 5 -> 9 (enter - for default: 5)
 bw_frac bandwidth fraction of low-pass filter on complex data (0.0->1.0) (enter - for default: 1.0)
 deramp deramp SLC phase flag (enter - for default)
 0: no deramp (Doppler centroid close to 0) (default)
 1: deramp SLC phase
 int_filt intensity low-pass filter flag (enter - for default)
 0: no filter
 1: low-pass filter of intensity data, highly recommended when no oversampling used (default)
 pflag print flag (enter - for default)
 0: print offset summary (default)
 1: print all offset data
 pltflg plotting flag (enter - for default)
 0: none (default)
 1: screen output
 2: screen output and PNG format plots
 3: output plots in PDF format
 ccs (output) cross-correlation standard deviation of each patch (float)

NOTE: ScansAR and TOPS data need to be deramped

6.3 Offset_pwr_tracking2

usage: offset_pwr_tracking2 <SLC1> <SLC2> <SLC1_par> <SLC2_par> <OFF_par> <offs> <ccp> [OFF_par2] [offs2] [rwin] [azwin] [offsets] [n_ovr] [thres] [rstep] [azstep] [rstart] [rstop] [azstart] [azstop] [bw_frac] [deramp] [int_filt] [pflag] [pltflg] [ccs]

input parameters:

SLC1 (input) single-look complex image 1 (reference)
 SLC2 (input) single-look complex image 2
 SLC1_par (input) SLC-1 ISP image parameter file
 SLC2_par (input) SLC-2 ISP image parameter file
 OFF_par (input) ISP offset/interferogram parameter file
 offs (output) offset estimates in range and azimuth (fcomplex)
 ccp (output) cross-correlation of each patch (0.0->1.0) (float)
 OFF_par2 (input) ISP offset/interferogram parameter file of the offset map to determine initial offsets (enter - for none)
 offs2 (input) input range and azimuth offset map to determine initial offsets (enter - for none)
 rwin range patch size (range pixels, enter - for default from offset parameter file)
 azwin azimuth patch size (azimuth lines, enter - for default from offset parameter file)
 offsets (output) range and azimuth offsets and cross-correlation data in text format,
 enter - for no output
 n_ovr SLC oversampling factor (integer 2**N (1,2,4), enter - for default: 2)
 thres cross-correlation threshold (0.0->1.0) (enter - for default from offset parameter file)
 rstep step in range pixels (enter - for default: rwin/2)
 azstep step in azimuth pixels (enter - for default: azwin/2)
 rstart offset to starting range pixel (enter - for default: 0)
 rstop offset to ending range pixel (enter - for default: nr-1)
 azstart offset to starting azimuth line (enter - for default: 0)
 azstop offset to ending azimuth line (enter - for default: nlines-1)

```

bw_frac  bandwidth fraction of low-pass filter on complex data (0.0->1.0) (enter - for
         default: 1.0)
deramp    deramp SLC phase flag (enter - for default)
         0: no deramp (Doppler centroid close to 0) (default)
         1: deramp SLC phase
int_filt  intensity low-pass filter flag (enter - for default)
         0: no filter
         1: low-pass filter of intensity data, highly recommended when no oversampling used
         (default)
pflag     print flag (enter - for default)
         0: print offset summary (default)
         1: print all offset data
pltflg    plotting flag (enter - for default)
         0: none (default)
         1: screen output
         2: screen output and PNG format plots
         3: output plots in PDF format
ccs       (output) cross-correlation standard deviation of each patch (float)
  
```

NOTE: ScansAR and TOPS data need to be previously deramped

6.4 Offset_pwr

```
usage: offset_pwr <MLI-1> <MLI-2> <DIFF_par> <offs> <ccp> [rwin] [azwin] [offsets] [n_ovr]
         [nr] [naz] [thres] [lanczos] [bw_frac] [pflag] [pltflg] [ccs]
```

input parameters:

```

MLI-1     (input) real valued intensity image 1 (reference)
MLI-2     (input) real valued intensity image 2
DIFF_par  (input) DIFF/GEO parameter file
offs      (output) offset estimates in range and azimuth (fcomplex)
ccp       (output) cross-correlation of each patch (0.0->1.0) (float)
rwin      range patch size (range pixels, enter - for default from offset parameter file)
azwin     azimuth patch size (azimuth lines, enter - for default from offset parameter file)
offsets   (output) range and azimuth offsets and cross-correlation data in text format,
         enter - for no output
n_ovr     MLI oversampling factor (integer 2**N (1,2,4), enter - for default: 1)
nr        number of offset estimates in range direction
         (enter - for default from offset parameter file)
naz       number of offset estimates in azimuth direction
         (enter - for default from offset parameter file)
thres     cross-correlation threshold (0.0->1.0) (enter - for default from offset parameter
         file)
lanczos  Lanczos interpolator order 5 -> 9 (enter - for default: 5)
bw_frac  bandwidth fraction of low-pass filter on intensity data (0.0->1.0)
         (enter - for default: 0.8)
pflag     print flag (enter - for default)
         0: print offset summary (default)
         1: print all offset data
pltflg    plotting flag (enter - for default)
         0: none (default)
         1: screen output
         2: screen output and PNG format plots
         3: output plots in PDF format
ccs       (output) cross-correlation standard deviation of each patch (float)
  
```

6.5 Offset_pwr_trackingm

```
usage: offset_pwr_trackingm <MLI-1> <MLI-2> <DIFF_par> <offs> <ccp> [rwin] [azwin] [offsets]
         [n_ovr] [thres] [rstep] [azstep] [rstart] [rstop] [azstart] [azstop] [lanczos] [bw_frac]
         [pflag] [pltflg] [ccs]
```

input parameters:

```

MLI-1     (input) real valued intensity image 1 (reference)
MLI-2     (input) real valued intensity image 2
DIFF_par  (input) DIFF/GEO parameter file
offs      (output) offset estimates in range and azimuth (fcomplex)
  
```

ccp (output) cross-correlation of each patch (0.0->1.0) (float)
 rwin range patch size (range pixels, enter - for default from offset parameter file)
 azwin azimuth patch size (azimuth lines, enter - for default from offset parameter file)
 offsets (output) range and azimuth offsets and cross-correlation data in text format,
 enter - for no output
 n_ovr MLI oversampling factor (integer 2**N (1,2,4), enter - for default: 1)
 thres cross-correlation threshold (0.0->1.0) (enter - for default from offset parameter file)
 rstep step in range pixels (enter - for default: rwin/2)
 azstep step in azimuth pixels (enter - for default: azwin/2)
 rstart offset to starting range pixel (enter - for default: 0)
 rstop offset to ending range pixel (enter - for default: nr-1)
 azstart offset to starting azimuth line (enter - for default: 0)
 azstop offset to ending azimuth line (enter - for default: nlines-1)
 lanczos Lanczos interpolator order 5 -> 9 (enter - for default: 5)
 bw_frac bandwidth fraction of low-pass filter on intensity data (0.0->1.0) (enter - for default: 0.8)
 pflag print flag (enter - for default)
 0: print offset summary (default)
 1: print all offset data
 pltflg plotting flag (enter - for default)
 0: none (default)
 1: screen output
 2: screen output and PNG format plots
 3: output plots in PDF format
 ccs (output) cross-correlation standard deviation of each patch (float)

6.6 Offset_pwr_trackingm2

usage: offset_pwr_trackingm2 <MLI-1> <MLI-2> <DIFF_par> <offs> <ccp> [DIFF_par2] [offs2] [rwin]
 [azwin] [offsets] [n_ovr] [thres] [rstep] [azstep] [rstart] [rstop] [azstart] [azstop]
 [bw_frac] [pflag] [pltflg] [ccs]

input parameters:

MLI-1 (input) real valued intensity image 1 (reference)
 MLI-2 (input) real valued intensity image 2
 DIFF_par (input) DIFF/GEO parameter file
 offs (output) offset estimates in range and azimuth (fcomplex)
 ccp (output) cross-correlation of each patch (0.0->1.0) (float)
 DIFF_par2 (input) DIFF/GEO parameter file of the offset map to determine initial offsets
 (enter - for none)
 offs2 (input) input range and azimuth offset map to determine initial offsets
 (enter - for none)
 rwin range patch size (range pixels, enter - for default from offset parameter file)
 azwin azimuth patch size (azimuth lines, enter - for default from offset parameter file)
 offsets (output) range and azimuth offsets and cross-correlation data in text format,
 enter - for no output
 n_ovr MLI oversampling factor (integer 2**N (1,2,4), enter - for default: 1)
 thres cross-correlation threshold (0.0->1.0) (enter - for default from offset parameter file)
 rstep step in range pixels (enter - for default: rwin/2)
 azstep step in azimuth pixels (enter - for default: azwin/2)
 rstart offset to starting range pixel (enter - for default: 0)
 rstop offset to ending range pixel (enter - for default: nr-1)
 azstart offset to starting azimuth line (enter - for default: 0)
 azstop offset to ending azimuth line (enter - for default: nlines-1)
 bw_frac bandwidth fraction of low-pass filter on intensity data (0.0->1.0) (enter - for default: 0.8)
 pflag print flag (enter - for default)
 0: print offset summary (default)
 1: print all offset data
 pltflg plotting flag (enter - for default)
 0: none (default)
 1: screen output
 2: screen output and PNG format plots
 3: output plots in PDF format
 ccs (output) cross-correlation standard deviation of each patch (float)

6.7 Offset_pwr_list

usage: offset_pwr_list <SLC-1> <SLC-2> <SLC1_par> <SLC2_par> <OFF_par> <clist_RDC> <clist_MAP>
 <offs> <ccp> <nx> <ny> [rwin] [azwin] [offsets] [n_ovr] [thres] [bw_frac] [deramp]
 [int_filt] [pflag] [pltflg] [ccs]

input parameters:

SLC-1 (input) single-look complex image 1 (reference)
 SLC-2 (input) single-look complex image 2
 SLC1_par (input) SLC-1 ISP image parameter file
 SLC2_par (input) SLC-2 ISP image parameter file
 OFF_par (input) ISP offset/interferogram parameter file
 clist_RDC (input) list of x,y pixel coordinates in the reference SLC image geometry
 (Range-Doppler Coordinates) (text format)
 clist_MAP (input) list of x,y pixel coordinates in the map projection geometry (text format)
 offs (output) offset estimate 2D map (fcomplex)
 ccp (output) cross-correlation of each patch (0.0->1.0) in map coordinates (float)
 nx width of 2D offset map in MAP geometry
 ny height of 2D offset map in MAP geometry
 rwin range patch size (range pixels, enter - for default from offset parameter file))
 azwin azimuth patch size (azimuth lines, enter - for default from offset parameter file))
 offsets (output) range and azimuth offsets and cross-correlation data in text format,
 enter - for no output
 n_ovr SLC oversampling factor (integer 2**N (1,2,4), enter - for default: 2)
 thres cross-correlation threshold (0.0->1.0) (enter - for default from offset parameter
 file)
 bw_frac bandwidth fraction of low-pass filter on complex data (0.0->1.0) (enter - for
 default: 1.0)
 deramp deramp SLC phase flag (enter - for default)
 int_filt intensity low-pass filter flag (enter - for default)
 0: no filter
 1: low-pass filter of intensity data, highly recommended when no oversampling used
 (default)
 pflag print flag (enter - for default)
 0: print offset summary (default)
 1: print all offset data
 pltflg plotting flag (enter - for default)
 0: none (default)
 1: screen output
 2: screen output and PNG format plots
 3: output plots in PDF format
 ccs (output) cross-correlation standard deviation of each patch
 in map coordinates (float)

NOTE: ScansAR and TOPS data need to be previously deramped