

KAREN Altimeter

Processing chain and file format

Reference Code : MS-DTU-KAR-03- PFF-031
Issue : 3.1
Date : 10th January 2019

MetaSensing BV
Huygensstraat 44
2201DK Noordwijk, The Netherlands
Tel.: +31 71 751 5960
Email: info@metasensing.com

Document Status Log

Issue	Change description	Date
1.0	First version	13 Aug 2018
1.1	Additional sections included	20 Aug 2018
1.2	Editing correction and formatting	22 Aug 2018
2.0	Approval	29 Aug 2018
2.1	Implemented comments from DTU	25 Oct 2018
2.2	Implemented comments from ESA	5 Nov 2018
2.3	Implemented additional comments from ESA	12 Dec 2018
3.0	Approval	14 Dec 2018
3.1	Minor correction and approval	10 Jan 2019

Table of contents

1	Introduction.....	4
2	Backprojection algorithm.....	7
3	Processing chain.....	8
3.1	Raw-data unpacking.....	9
3.2	Range Compression	10
3.3	Zero-Doppler Filtering.....	11
3.4	Navigation data post-processing	12
3.5	GBP Focusing	12
3.5.1	Run-time performance	14
3.6	Multilooking	15
3.7	NetCDF Encapsulation	17
4	NetCDF File Content	18
4.1	File Name.....	18
4.2	Variables	20

1 Introduction

MetaSensing is a Dutch/Italian company offering radar sensors and services [Ref. 1]. MetaSensing's radar products cover a wide range of applications as mapping, deformation monitoring, weather monitoring, coastal surveillance and harbor management. During the past few years the MetaSensing Ka-band radar altimeter, KAREN, has been deployed in the framework of the ESA's CryoVEx validation and monitoring activities for sea ice and land ice. In particular, three airborne campaigns have been performed with KAREN in October 2016 (EGIG line, Greenland), Spring 2017 (Arctic) and Winter 2017/2018 (Antarctic).

The KAREN system is an interferometric frequency modulated continuous wave (FMCW) SAR altimeter working at the Ka frequency band. A block scheme of the radar instrument is shown in Figure 1. A baseband signal is up-shifted to the Ku band before being frequency-doubled to 34.5 GHz. The transmitting antenna continuously radiates linear chirps centered at this frequency, eventually amplified by the High Power Amplifier (HPA); the echoes scattered by the monitored scenario are received by two separate receiving antennas. The system works according to a deramping principle: a replica of the transmitted signal is mixed with the received signal which has been amplified by a Low Noise Amplifier (LNA). The Intermediate Frequency deramped signal (beat frequency) is sampled and stored as raw data, without any range windowing applied.

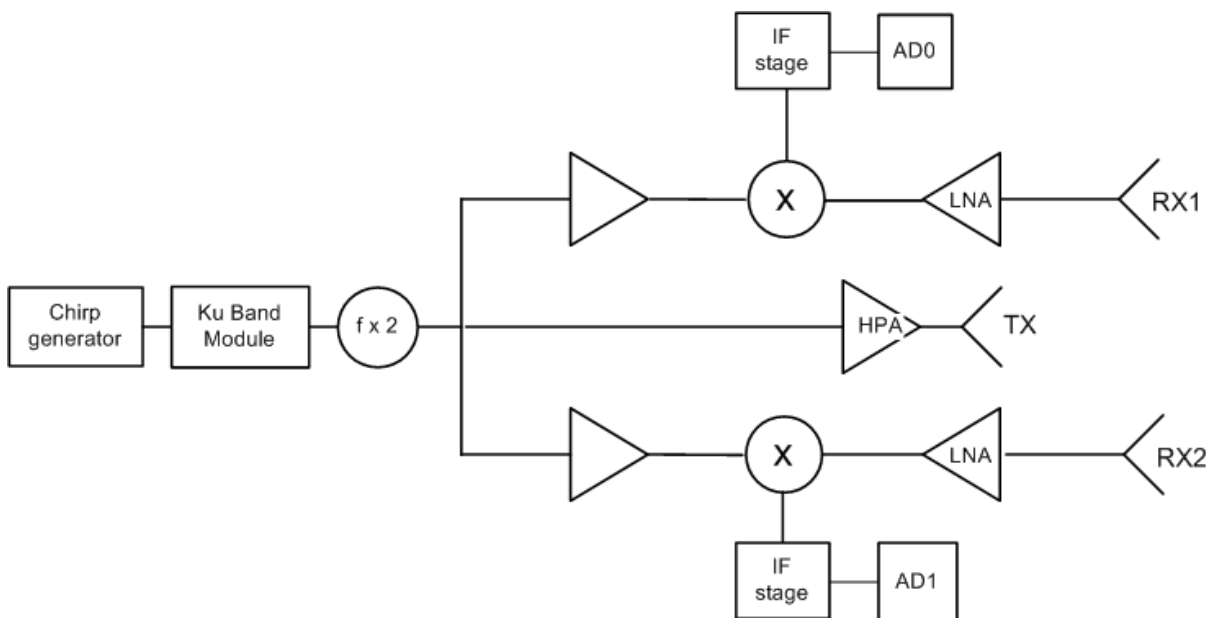


Figure 1: Block diagram of the KAREN system, the Ka band radar altimeter by MetaSensing.

The KAREN system is controlled by the user friendly MetaSAR GUI, see Figure 2. By that it is possible to configure the system parameters, to start and stop each acquisition and to monitor if eventual alerts or error messages occur.

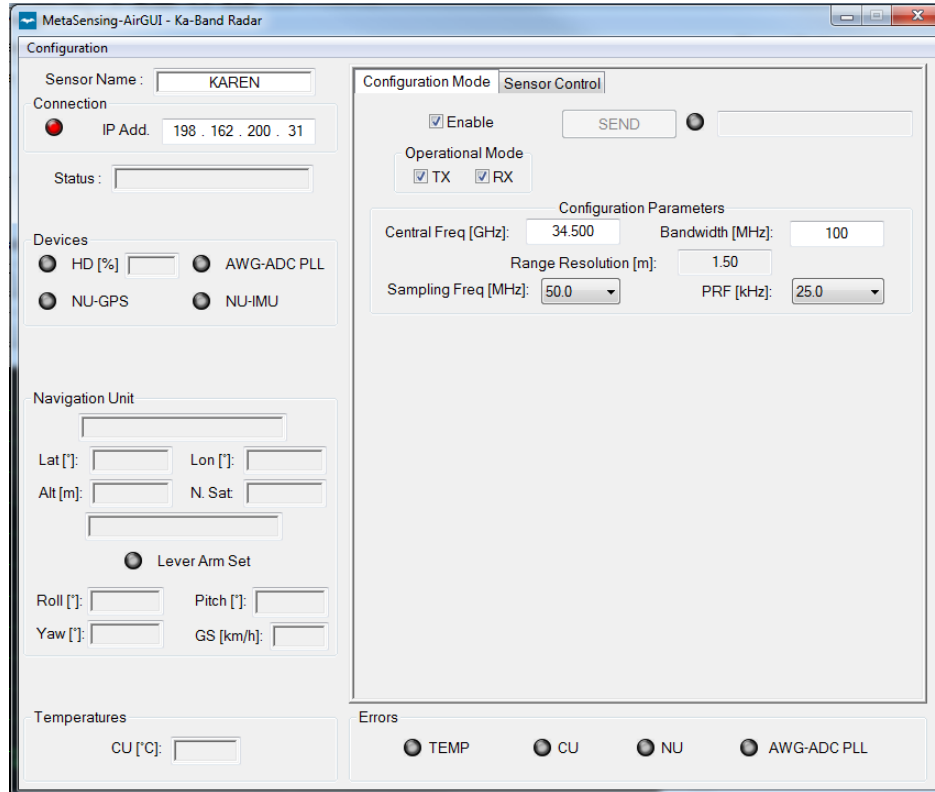


Figure 2: Configuration window of the MetaSAR AirGUI, the software tool to control the MetaSensing airborne radar sensors.

In particular on the right part of the GUI a configuration tab is selected, in which the main configuration parameters can be set.

Central frequency (F_c) Ideally, it can be tuned; however, it has been kept fixed to the value of 34.5 GHz during any KAREN campaign, as the one providing maximum transmitted bandwidth possible. The value of this parameter is provided within the delivered NetCFD file.

Bandwidth (B_w) 0-600 MHz. where 0 corresponds to a single tone transmission at the central frequency, any other specified transmitted bandwidth is centered at the central frequency. When possible, it has been kept to its maximum value of 600MHz, in order to maximize the slant range resolution to 25 cm (value shown in the GUI as a direct consequence of the set bandwidth). Sometimes, especially when high altitude acquisitions were performed, the bandwidth needs to be reduced to lower values, in order to increase the maximum operational range of the radar (in FMCW

systems the maximum range is inversely proportional to the transmitted bandwidth). The value of this parameter is provided within each delivered NetCFD file.

Pulse Repetition Frequency (PRF) The repetition rate of the linearly frequency modulated chirps can vary between 0.5 and 25 KHz (list of discrete values). This parameter should be traded off between maximum operation range and Doppler ambiguities. The value of this parameter is provided within each delivered NetCFD file.

Sampling frequency (F_s) The sampling frequency can be selected by GUI to be 10 MHz, 25 MHz or 50 MHz. However, the pre-sampling Nyquist filters should be manually changed according to the chosen F_s value, which is not a trivial operation during flights. Therefore, the sampling frequency has been selected to 25MHz and kept as such for all KAREN campaigns.

Once set, the values of these parameters are stored in the header of the performed acquisition files. Most of them have been included in the list of parameters within the delivered NetCDF files, see chapter 4.

As reference, a typical acquisition during the KAREN campaigns in Low Altitude Mode (LAM, the terrain is expected ~300 meters below the aircraft) is performed with the following parameters: $F_c = 34.5$ GHz, $B_w = 600$ MHz, $F_s = 25$ MHz, $PRF = 6.15$ KHz, to which corresponds a maximum operational range of 508 meters and an unambiguous Doppler bandwidth wide enough at the considered aircraft speed and antenna beamwidth.

The remaining of this document describes the steps implemented in the processing of data acquired by the KAREN system aiming at delivering level-1b data, i.e. focused multilooked data. Besides this introductory chapter, a short overview of the ground-backprojection (GBP) algorithm is given in chapter 2, while the processing chain is documented step by step in chapter 3; finally, chapter 4 describes the data format of delivered files.

2 Backprojection algorithm

The processor for the KAREN altimeter data is based on a time-domain ground-backprojection (GBP) method ([Ref. 2], [Ref. 3]). Opposite to the frequency-domain approach, separate motion compensation and range migration correction steps are not required because the GBP algorithm handles non-ideal motion/sampling implicitly and can precisely perform beam-steering [Ref. 7]. Therefore, the GBP algorithm can be used for any imaging geometry. This gives flexibility to use the same algorithm core to process data acquired at diverse configurations (side-looking, nadir-looking, bistatic), trajectories or modes.

The GBP algorithm interpolates and phase-compensates each received echo at the desired positions to be focused. Because the radar echo has been sampled according to the Nyquist criterion, it can be interpolated with arbitrary accuracy at any illuminated image position. By coherently adding the contributions of each echo to each desired position, the focusing is performed. The contribution of each echo is computed according to the acquisition geometry.

The drawback of time-domain GBP is that it is more computationally expensive than the frequency domain methods, because the geometry of each echo is taken into consideration. However, because technology is constantly increasing the speed and the ability to perform parallel computations (GBP is highly parallelizable), the computational expense of GBP is becoming less and less a concern if compared to its inherent advantages over frequency-domain approaches [Ref. 8]. GBP offers several opportunities for seamless parallelisation based in NVIDIA graphic cards. In Figure 3 it is shown a computer at MetaSensing laboratory busy processing KAREN data with eight graphic processing units (GPU) of last generation visible on top of the machine.



Figure 3: Processing machine at MS labs with 8 GPUs.

3 Processing chain

The MetaSAR Processor is the tool by MetaSensing to elaborate SAR images from the collected raw data. It is a generic software which can be used for any radar system by MetaSensing to generate both strip images (typical of side looking SAR instruments) and altitude profiles (typical of altimeters, as the ones discussed in this document). Figure 4 shows the block diagram implemented by the processor for the KAREN data, with fundamental steps leading from collected Level-0 data to the delivered Level-1B data. Each step is described in the following sub-sections.

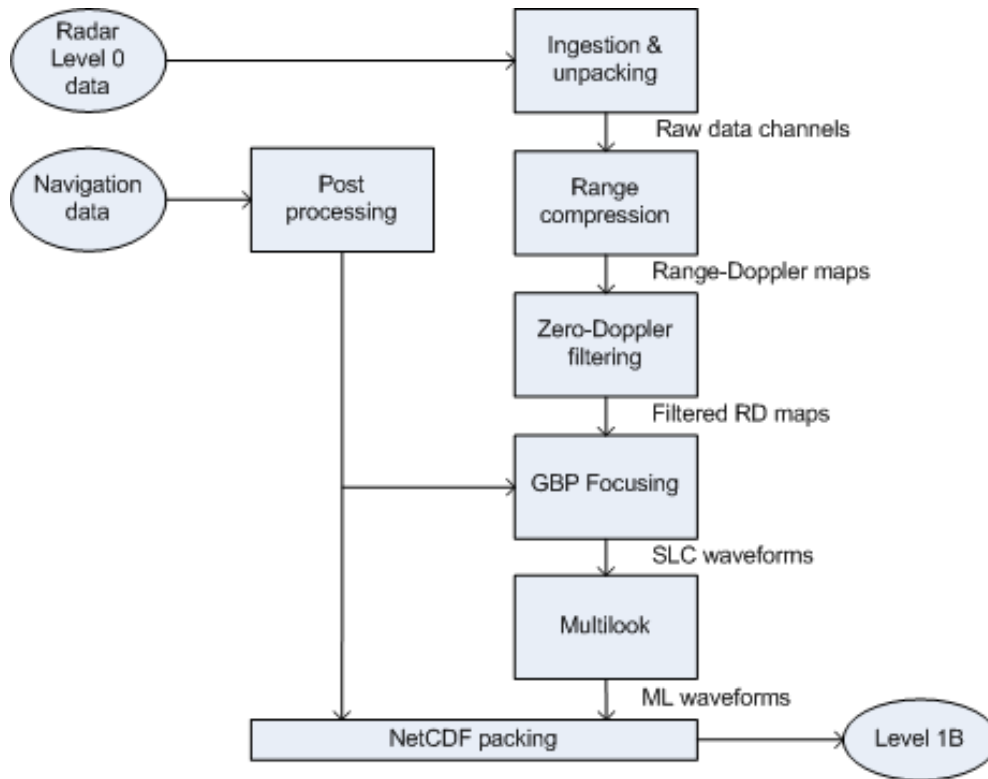
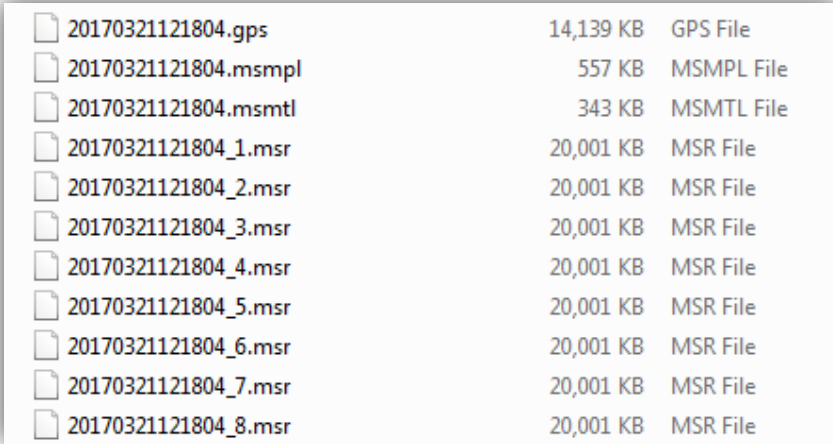


Figure 4: Block diagram for the KAREN processor.

3.1 Raw-data unpacking

Figure 5 shows the contents of a typical acquisition file from the KAREN altimeter. The deramped radar data for both channels are stored in fixed-dimensions *.msr* files, whose number depends on the length (in time) of the acquisition. These are joined together during the ingestion phase. Together with radar data, the corresponding configuration file (*.msmpl*) and time synchronization file (*.msmtl*) are provided. A real-time navigation file (*.gps*) is also associated to each acquisition. The GPS file contains both real time navigation information and GPS raw data which can be used during post processing in order to improve the accuracy of the navigation data.



20170321121804.gps	14,139 KB	GPS File
20170321121804.msmp1	557 KB	MSMP1 File
20170321121804.msmtl	343 KB	MSMTL File
20170321121804_1.msr	20,001 KB	MSR File
20170321121804_2.msr	20,001 KB	MSR File
20170321121804_3.msr	20,001 KB	MSR File
20170321121804_4.msr	20,001 KB	MSR File
20170321121804_5.msr	20,001 KB	MSR File
20170321121804_6.msr	20,001 KB	MSR File
20170321121804_7.msr	20,001 KB	MSR File
20170321121804_8.msr	20,001 KB	MSR File

Figure 5: Example of the contents of a KAREN acquisition file.

The KAREN instrument data extraction is organized in two steps:

- Header extraction.
- Level-0 data extraction.

Using the information included in the header, i.e. instrument configuration and user settings such as used Pulse Repetition Frequency (PRF), Sampling Frequency (F_s), etc, the time-tagged observation data is arranged in a raw-data matrix, one for each receiving channel: rows represent slow time domain, which parametrizes the along track position of the platform (sampled at PRF), while columns represent the fast time domain, which parametrizes the beat frequency (sampled at F_s).

3.2 Range Compression

Raw data of each channel (deramped frequency-modulated continuous waveforms) are the input for this step. The range compression is implemented as a simple Fast Fourier Transform (FFT) in range. Furthermore, a Hann window is applied to trade-off side-lobes level and resolution.

Figure 6 shows an example of Range-Doppler map for the KAREN range compressed data, which have been acquired at an altitude of ~ 365 m AGL during March 19th, 2017, for one of the two receiving channels.

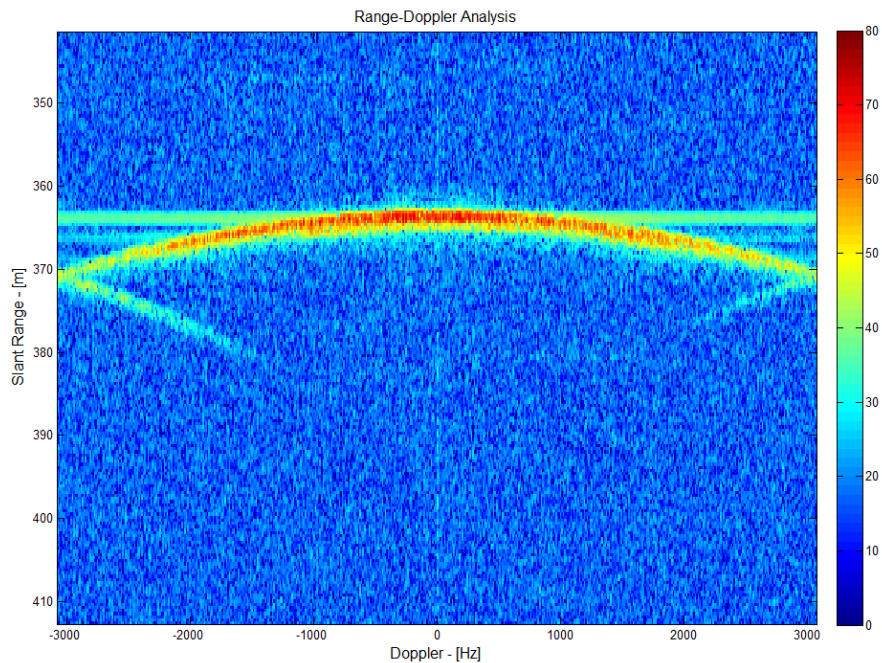


Figure 6: Example of processed Range-Doppler map. The colormap is expressed in dB.

To reduce the processing effort during further steps, the range compressed output is analyzed in the Range-Time domain to roughly estimate the range-delay of the signal and to determine the minimum and maximum slant range limits to the grid used for the backprojection beam steering.

3.3 Zero-Doppler Filtering

The main interest of altimeter data lies in the echoes coming from the nadir direction of the aircraft during its motion. Therefore, the range compressed data are filtered around the Zero Doppler in the Doppler dimension with a Doppler bandwidth corresponding to half the used PRF. Eventual folding effects at the extremities of the Doppler spectrum are avoided in this way. This is shown in Figure 7. By filtering half of the bandwidth, it is possible focusing the data with 100 looks and 5 m resolution along track.

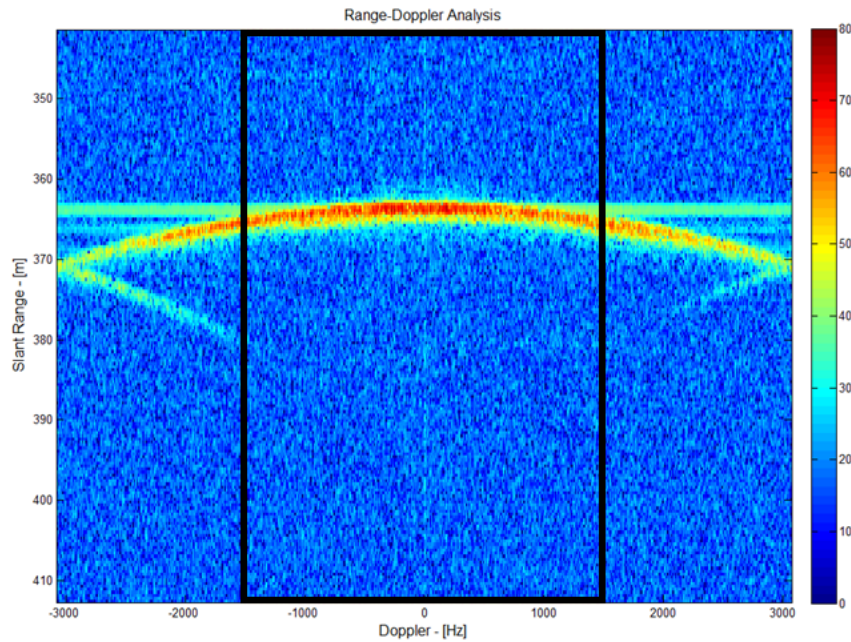


Figure 7: Zero-Doppler filtering (black rectangle) by half of the spectrum width, corresponding to $PRF/2$.

3.4 Navigation data post-processing

The KAREN system is equipped with a dedicated navigation unit, a compact, single enclosure GNSS+INS receiver. The accuracy of the real-time solution computed by the unit can be improved by post-processing. This is usually done in the KAREN processing flow.

However, to have consistency with other on-board remote sensing instruments (ASIRAS Ku-band altimetry, Airborne Laser Scanner, etc) during the campaigns, it has been agreed with final users to adopt a common navigation solution for all provided datasets. The navigation units deployed by DTU is the most accurate one among the on-board units during the KAREN acquisitions. Different models have been used in different campaigns, so the reader is invited to refer to dedicated campaign reports. An importing step has been implemented in the KAREN processing to properly convert the DTU navigation data into an acceptable format to the KAREN processor. The same navigation data is provided in the delivered NetCDF files.

3.5 GBP Focusing

As introduced in chapter 2, the backprojection algorithm implements the beam steering for the Doppler-filtered range compressed data for each sample on any arbitrary surface grid. The geometry of the output grid can be set to follow any direction: for example, the processed data can be directly projected on the Universal Transverse Mercator (UTM) system. The post-processed navigation information is used to determine the vectors joining the position of the beam center and the position of the point-target in the grid. With the knowledge of the ranges and angles relative to each point-target the vectors are phase compensated and then coherently summed up to form the focused beam. This is repeated for each point-target of the grid. As a result, a 2D image is obtained, as the one shown in Figure 10, characterized by a resolution of 0.05 m X 0.2 m (along track X nadir).

The main features of the backprojection algorithm are described in detail in the following:

- 1) The target function in the spatial domain $f[i, j]$ is created with the same dimensions of the reconstruction grid (output matrix).
- 2) The Nadir Coordinate System (NCS) T_{xyz} is defined for each output pixel. The NCS is aligned with the x-axis pointing in the direction of the aircraft forward motion, the z-axis pointing downward to the centre of the earth, and the y-axis pointing to the right direction to form an orthogonal right-handed system, as shown in Figure 8

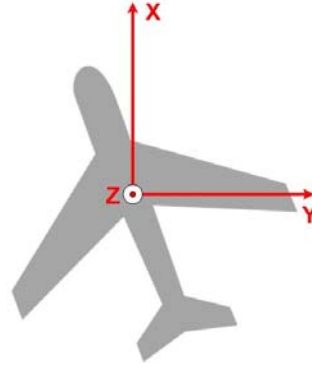


Figure 8: Nadir Coordinate System. Note, in this figure the aircraft is supposed flying with some yaw angle: the aircraft velocity vector is not parallel to the aircraft longitudinal direction.

3) The slant-range values $r(u, i, j)$ to the target are calculated for each pixel point $[i, j]$ on the output grid (NCS) and for each antenna position $P(u)$ on the synthetic aperture:

$$r(u, i, j) = \sqrt{(T_x - P_x(u))^2 + (T_y - P_y(u))^2 + (T_z - P_z(u))^2}$$

Note, as mentioned in chapter 2, the range cell migration and the motion compensation steps are implicitly performed in the backprojection, as the actual antenna positions are accounted for each pixel in the determination of the slant range to the target.

4) A coherent summation is performed by iteratively using the following formula along the synthetic aperture:

$$f[i, j] = f[i, j] + S_D[u] e^{j \frac{4\pi}{\lambda} [r(u, i, j) - r_{ref}(i, j)]}$$

where S_D refers to the range compressed data. In this way, the portion of each range compressed pulse that corresponds to the range to a given pixel from each aperture position is multiplied by the expected phase for that range and it is summed up across the entire collection.

It is worth to underline that no Fourier transform is involved in the GBP focusing: for every single pixel in the raw data the above coherent summation is performed to integrate the pulses in the slow-time domain. The number of integrated pulses for each pixel depends on the length of the synthetic aperture and it is computed based on the desired resolution and on the actual trajectory.

3.5.1 Run-time performance

As mentioned in chapter 2, KAREN data is processed by using multiple video cards in a single computer. The parallel computing is now implemented on the only focusing step. As a reference, Table 1 shows the overall run-time performance to process 1 minute of acquisitions by KAREN, represented by 6 GB of raw data (2 interferometric channels, 25MHz sampling frequency, 2 bytes per sample). The mentioned GPU's are Geforce GTX 1080 model by Asus.

Nr. of GPUs	Range Compression	Doppler Filtering	GBP Focusing	Total
1	4 min	27 min	40 min	72 min
6			7 min	38 min

One of the current developments undergoing at MetaSensing is the implementation of parallel processing also in the Doppler Filtering step, to further improve the overall run-time performance. Additionally, an adaptative range gating filter is being designed, so that only the actual range bins with meaningful data will be processed along the nadir direction, rather than the entire range profiles.

3.6 Multilooking

To reduce the inherent speckle noise of radar images, either spatial filtering or multi-look processing can be applied, both at the expense of spatial resolution ([Ref. 4], [Ref. 5]). The KAREN processor implements the multi-look method. Figure 9 shows the principle. The radar beam is divided into several sub-beams (three in the figure), each one providing an independent “look” at the illuminated scene. The final output image is obtained by summing and averaging together the output of each look, in which the amount of speckle is reduced.

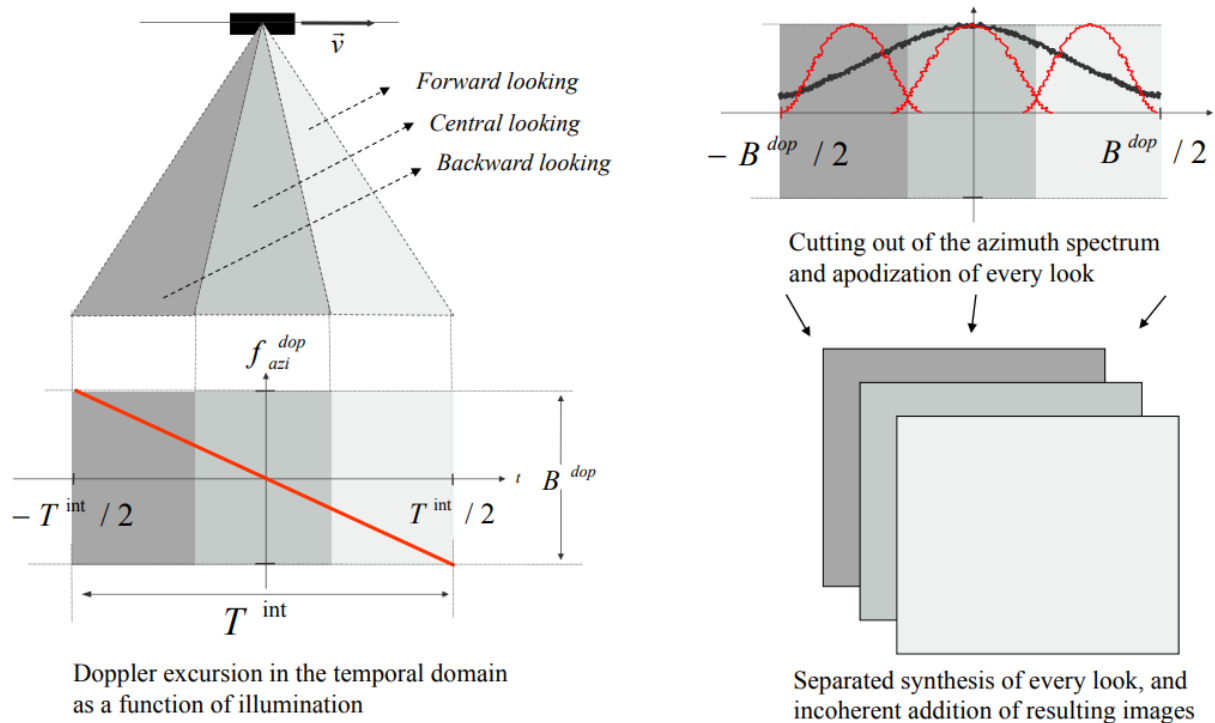


Figure 9: Principle of multi-look processing [Ref. 6].

The KAREN processor implements the multilooking by applying a FFT in the Doppler dimension to the back-projected data and dividing the spectrum in the desired number of looks. Hamming windowing is performed in the along-track direction and an IFFT is applied to each look to restore the backprojected data domain.

In the delivered Level-1b data, the power (*pwr*), the phase (*pha*) and coherence (*coh*) are computed according to:

$$pwr(rg) = \frac{1}{L} \sum_{l=1}^L \left(\frac{|S(l, rg, 1)|^2 + |S(l, rg, 2)|^2}{2} \right)$$

$$X(rg) = \frac{1}{L} \sum_{l=1}^L \left(\frac{S(l, rg, 1) \cdot conj(S(l, rg, 2))}{\sqrt{|S(l, rg, 1)|^2 \cdot |S(l, rg, 2)|^2}} \right)$$

$$pha(rg) = \arg(X(rg))$$

$$coh(rg) = \text{abs}(X(rg))$$

Where $S(l, rg, 1)$ and $S(l, rg, 2)$ refer to the back-projected data for look number l at range rg of channel 1 and channel 2, respectively, where L represents the total number of looks.

As mentioned, a loss of resolution is obtained within this step. Typically, the backprojected data (SLC) have a grid spacing and resolution of 0.05 m, while the delivered multilooked data have 5 m resolution and 100 looks.

3.7 NetCDF Encapsulation

The final step of the KAREN processing chain is represented by the NetCDF encapsulation of the processed data and of all the relevant ancillary data such as time tags, geographical position, and other radar/processing parameters. The required fields to be included have been agreed with final users.

During this step the transformation from NCS to geographical coordinate system is also performed. Additionally, the time reference grid is converted from the GPS time to UTC (Universe Time Coordinates) in seconds since the midnight of the 1st of January 2000.

Figure 3 shows an example of a delivered KAREN waveforms plotted in terms of range-delay versus time acquired on the 31st of March 2017, beginning at 15:32:53 (GPS time). For convenience, the time axes is shown in time since the beginning of acquisition.

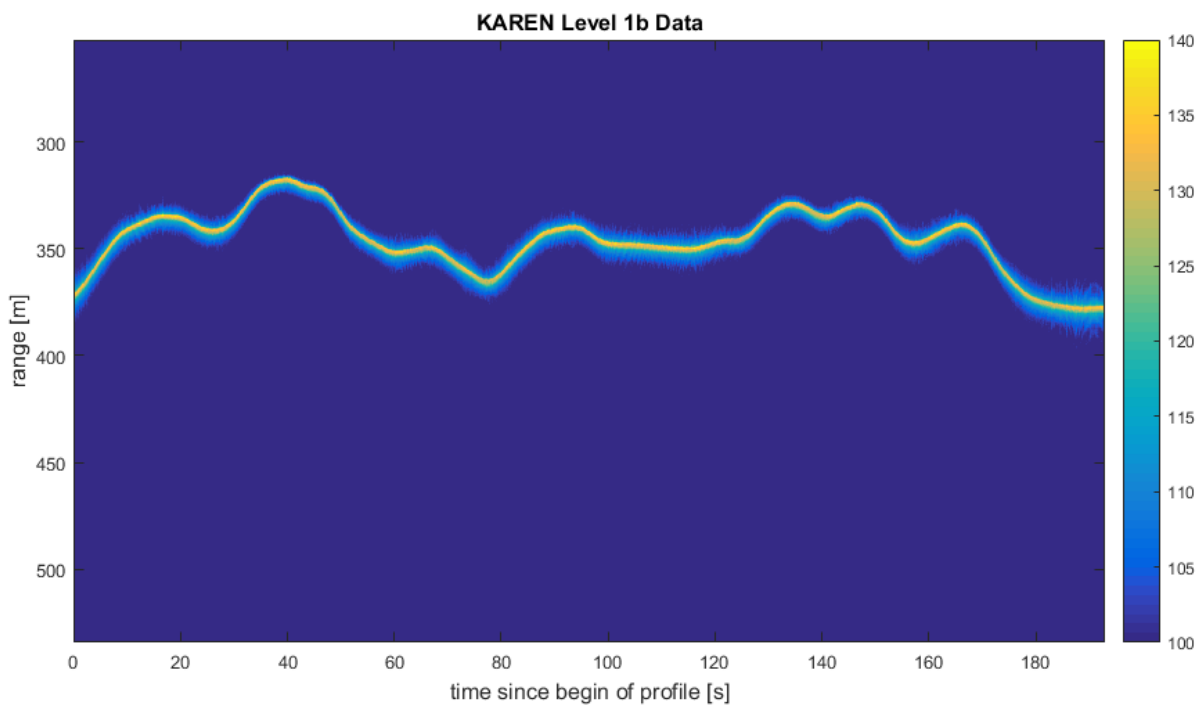


Figure 10: Example of KAREN Range-delay versus time dataset provided in the delivered NetCDF files.
The colormap is expressed in dB.

4 NetCDF File Content

KAREN data are provided in NetCDF format (*.nc* extension). Each NetCDF file contains:

- Focused multilooked altimeter waveforms
- Multilooked phase differences between the two channels
- Coherence between the two channels
- Waveform position
- Sensor attitude data
- Radar and processing parameters.

The NetCDF files are three dimensional, with *Range* (size 247), *Time* (4767) and *space_3d* (3). The *range* and *time* dimensions are related to the pre-defined grid in which the focused SAR pixels are projected. This grid is regular in space.

The *Range* dimension refers to the range delays to the pixels, in nadir direction, equally spaced.

The *Time* dimension refers to the UTC time since the midnight of the 1st of January 2000, during which the pixel data have been acquired, equally spaced.

The *space_3d* dimension is used to write the three-dimensional position and velocity vector coordinates of the sensor.

The file name format and the detailed contents of the NetCDF files are explained in the following paragraphs.

4.1 File Name

The delivered NetCDF files are named according to a defined structure, as in the following example:

`KAR_OPER_Level1b_20170331T104652_20170331T105245_hamh.nc`

KAR_OPER: Indicates that that data have been acquired by the KAREN system in its full operational mode.

Level1b: Processing level, i.e. focused multilooked SAR data.

20170331T104652: Date and time of the start of acquisition, in the format YYYYMMDDTHHMMSS, where YYYY is the year, MM the month, DD the day, *T* is a separator character indicating Time, HH is the hour, MM the minutes and SS the seconds, in UTC time.

20170331T105245: Date and time of the end of acquisition in the format YYYYMMDDTHHMMSS where YYYY is the year, MM the month, DD the day, *T* is a separator character indicating time, HH the hour, MM the minutes and SS the seconds, in UTC time.

hamh: Additional info. The last four digits before the *.nc* file extension are used to differentiate the various dataset KAREN datasets versions which have been delivered. In Table 2 an overview is given about the different versions, including a short description of the kind of processing which has been differently done with respect to previous version. It is advised to use the most recent one.

Table 2 – Versions of KAREN datasets		
Version	Info	Delivery date (KAREN dataset)
*_vvvv	First version processed with Hann window.	Jun 2018 (Spring 2017)
*_hamh	Changed the Hann by the Hamming window ($\alpha=0.54$)	Aug 2018 (Fall 2016, spring 2017)
*_leva	The RF antenna position has been corrected, measured lever arms have been introduced w.r.t. IMU reference point	Nov 2018 (Fall 2016, Spring 2017 and Antarctica 2018)
*_levb	A bug in the processing has been corrected, which was causing sudden jumps in longitude values in the <i>_leva</i> version	Dec 2018 (Fall 2016, Spring 2017 and Antarctica 2018)

4.2 Variables

range

Size: 247x1

Dimensions: range

Datatype: double

Attributes: long_name = range-delay in nadir direction. The difference between the samples of the range variable gives the range spacing.

units = '[m]'

time_ka

Size: 4767x1

Dimensions: time

Datatype: double

Attributes: long_name = Time to the instant the L1B waveform touches the surface. Time in UTC representing the seconds since the midnight of the 1st of January 2000. The difference between the samples of the time_ka variable gives the time spacing.

units = '[s]'

com_altitude_ka

Size: 4767x1

Dimensions: time

Datatype: double

Attributes: long_name = Altitude of the aircraft navigation unit above the reference ellipsoid (WGS84)

units = '[m]'

com_altitude_rate_ka

Size: 4767x1

Dimensions: time

Datatype: double

Attributes: long_name = Instantaneous altitude rate at aircraft navigation unit with respect to the reference ellipsoid (WGS84)

units = '[m/s]'

com_position_vector_ka

Size: 3x4767

Dimensions: space_3d, time

Datatype: double
Attributes: long_name = Position vector (x, y, z) at the aircraft navigation unit
units = '[m]'

com_velocity_vector_ka

Size: 3x4767
Dimensions: space_3d,time
Datatype: double
Attributes: long_name = Velocity vector (x, y, z) at the aircraft navigation unit
units = '[m/s]'

hr_power_waveform_ka

Size: 247x4767
Dimensions: range,time
Datatype: double
Attributes: long_name = 'level-1B multi-looked power waveform'(2 channels)
units = '[]'

latitude_ka

Size: 4767x1
Dimensions: time
Datatype: double
Attributes: long_name = 'Latitude of measurement [-90, +90]: Positive at North, Negative at South'
units = '[deg]'

longitude_ka

Size: 4767x1
Dimensions: time
Datatype: double
Attributes: long_name = 'Longitude of measurement [0, 360]'
units = '[deg]'

off_nadir_pitch_angle_pf_ka

Size: 4767x1
Dimensions: time
Datatype: double

Attributes: long_name = 'Pitch angle with respect to the nadir pointing direction'
units = '[deg]'

Off_nadir_roll_angle_ka

Size: 4767x1
Dimensions: time
Datatype: double
Attributes: long_name = 'Roll angle with respect to the nadir pointing direciton'
units = '[deg]'

Off_nadir_yaw_angle_pf_ka

Size: 4767x1
Dimensions: time
Datatype: double
Attributes: long_name = 'Yaw angle with respect to the forward velocity vector'
units = '[deg]'

Heading_angle_ka

Size: 4767x1
Dimensions: time
Datatype: double
Attributes: long_name = 'Heading angle with respect to the true North'
units = '[deg]'

hr_coh_waveform_ka

Size: 247x4767
Dimensions: range,time
Datatype: double
Attributes: long_name = 'Waveform coherence between the 2 channels'
units = '[]'

hr_phase_waveform_ka

Size: 247x4767
Dimensions: range,time
Datatype: double
Attributes:

long_name = 'Waveform phase between the 2 channels'
units = '[]'

TxBw

Size: 1x1
Dimensions:
Datatype: double
Attributes:
long_name = 'Transmitted Bandwidth'
units = 'hertz [Hz]'

Fc

Size: 1x1
Dimensions:
Datatype: double
Attributes:
long_name = 'Central Frequency'
units = 'hertz [Hz]'

PRF

Size: 1x1
Dimensions:
Datatype: double
Attributes:
long_name = 'Pulse Repetition Frequency'
units = 'hertz [Hz]'

AzBw

Size: 1x1
Dimensions:
Datatype: double
Attributes:
long_name = 'Azimuth bandwidth prior to multilooking'
units = '[Hz]'

MeanForwardVelocity

Size: 1x1
Dimensions:
Datatype: single
Attributes:
long_name = 'Velocity in the flight direction'
units = '[m/s]'

Looks

Size: 1x1
Dimensions:
Datatype: int32
Attributes:
long_name = 'Number of looks'
units = '[]'

BaselineHor

Size: 1x1
Dimensions:
Datatype: single
Attributes:
long_name = 'Horizontal baseline (half of the physical baseline length)'
units = 'units [cm]'

BaselineVer

Size: 1x1
Dimensions:
Datatype: single
Attributes:
long_name = 'Vertical baseline (half of the physical baseline length)'
units = 'units [cm]'

StartYearUTC

Size: 1x1
Dimensions:
Datatype: int32
Attributes:

long_name = 'UTC Year of the Start of Acquisition'
units = '[year]'

StartMonthUTC

Size: 1x1

Dimensions:

Datatype: int32

Attributes:

long_name = 'UTC Month of the Start of Acquisition'
units = '[month]'

StartDayUTC

Size: 1x1

Dimensions:

Datatype: int32

Attributes:

long_name = 'UTC Day of the Start of Acquisition'
units = '[day]'

StartHourUTC

Size: 1x1

Dimensions:

Datatype: int32

Attributes:

long_name = 'UTC Hour of the Start of Acquisition'
units = '[hour]'

StartMinUTC

Size: 1x1

Dimensions:

Datatype: int32

Attributes:

long_name = 'UTC Minutes of the Start of Acquisition'
units = '[min]'

StartSecUTC

Size: 1x1
Dimensions:
Datatype: single
Attributes:
long_name = 'UTC Seconds of the Start of Acquisition'
units = '[sec]'

FinalYearUTC

Size: 1x1
Dimensions:
Datatype: int32
Attributes:
long_name = 'UTC Year of End of Acquisition'
units = '[year]'

FinalMonthUTC

Size: 1x1
Dimensions:
Datatype: int32
Attributes:
long_name = 'UTC Month of End of Acquisition'
units = '[month]'

FinalDayUTC

Size: 1x1
Dimensions:
Datatype: int32
Attributes:
long_name = 'UTC Day of End of Acquisition'
units = '[day]'

FinalHourUTC

Size: 1x1
Dimensions:
Datatype: int32
Attributes:

long_name = 'UTC Hour of End of Acquisition'
units = '[hour]'

FinalMinUTC

Size: 1x1

Dimensions:

Datatype: int32

Attributes:

long_name = 'UTC Minutes of End of Acquisition'
units = '[min]'

FinalSecUTC

Size: 1x1

Dimensions:

Datatype: single

Attributes:

long_name = 'UTC Seconds of End of Acquisition'
units = '[sec]'

Dummy

Size: 1x1

Dimensions:

Datatype: int32

Attributes:

long_name = 'Dummy value'
units = '[]'

References

- [Ref. 1] <https://www.metasensing-group.com/>
- [Ref. 2] L.M. H.Ulander, H. Hellsten, and G.Stenström, "Synthetic-aperture radar processing using fast factorized back-projection," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, pp. 760–776, Jul. 2003.
- [Ref. 3] L.M. H. Ulander, H. Hellsten and G. Stenstrom, "Synthetic-aperture radar processing using fast factorized back-projection," in *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 3, pp. 760-776, July 2003.
- [Ref. 4] J. M. Durand. et al., "Speckle in SAR images: An evaluation of filtering techniques," *Adv. Space Res.* vol. 7, no. 11, pp. 301-304, 1987.
- [Ref. 5] F. K. Li, C. Croft, and D. N. Held, "Comparison of several techniques to obtain multiple-look SAR imagery," *IEEE Trans. Geosci. Remote Sensing*, vol. GE-21. July 1983.
- [Ref. 6] F. Sarti, Remote sensing and SAR images processing characterization and speckle filtering in radar images, ESA radar remote sensing course, Tartu, Estonia, 16-20 April 2012
On line: https://earth.esa.int/c/document_library/get_file?folderId=226458&name=DLFE-2125.pdf
- [Ref. 7] A. Egido and W. H. F. Smith, "Fully Focused SAR Altimetry: Theory and Applications," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 1, pp. 392-406, Jan. 2017.
doi: 10.1109/TGRS.2016.2607122
- [Ref. 8] Doerry, A.W., Bishop, E.E. Miller, J.A., "Basics of backprojections algorithm for processing SAR images", Sandia National Laboratories Albuquerque, New Mexico 87185 and Livermore, California 94550, Feb 2016.