Extracting Raw SAR Data from the RADARSAT CD

This document explains how to read the RADARSAT data CD provided with the book "Digital Processing of Synthetic Aperture Data" by Ian Cumming and Frank Wong, Artech House, 2005.

On this web site, programs are provided in MATLAB to read the data and parameters from the CD. The structure of the programs is outlined in Figure 1.

There are two main programs and one function that are used to extract data from the CD. First run "`specify_parameters.m`" to define the parameters for the current experiment, then run "`extract_data.m`" to extract the data from the CD. The only parameters you need to change are in the first program. If desired, the call to "`extract_data.m`" can be placed at the end of "`specify_parameters.m`".

`specify_parameters.m`

Only one file on the CD is needed, "`dat_01.001`". This file can be copied to your hard disk, for convenience. The path to this file, whether it be located on the CD or on your hard disk, must be specified by the variable "`input_path`". The size of the data file is 364 MB.

Data extracted from the CD is written to files in a directory specified by the variable, "`output_path`". Another variable, "`output_prefix`", is provided in case you want to store more than one set of data in the same directory.

To define the limits of the data to be extracted, set the variables, `first_rg_cell`, `Nrg_cells`, `first_rg_line`, `Nrg_lines` and `Nrg_lines_blk` to the desired values for this experiment. The term "range cell" refers to samples in the range or "fast time" direction. The term "range line" refers to samples in the azimuth or "slow time" direction. The first range cell, `first_rg_cell`, and the number of range cells, `Nrg_cells`, can be set to any value within the boundaries of the data array (1 to 9288). The first range line, `first_rg_line`, should be set to a multiple of 8 plus 1, and the number of lines per block, `Nrg_lines_blk`, should be set to a multiple of 8. Also, the total number of lines to be extracted, `Nrg_lines`, should be set to a multiple of the number of lines per block. If these rules are not obeyed, the "`extract_data.m`" program will quantize the values accordingly.

The variable, `Nrg_lines_blk`, allows you to control the extracted file size, so that the data can be extracted into a number of smaller files, rather than one big file. This is helpful in limiting the memory size needed by the extraction and analysis programs.

The extracted data can be stored in two ways – in MAT files or general binary files. The binary files can be used with any program, such as programs written in C. But if you are using MATLAB for all your processing, it is more efficient to use MAT files for the extracted data. In the latter case, set `UseMATfiles` to 1 (true). If you want to use binary files, set `UseMATfiles` to 0 (false).

specify_parameters.m

Define location of files
Select data boundaries
Save parameters for this run

CD_run_params.mat

Run parameters

---

extract_data.m

Load run parameters
Check parameter limits
Set fixed parameters
Save run parameter file

CD_run_params.mat

Run parameters

Fixed parameters

---

read_CEOS_raw.m

Define file names
Set file data pointers
Read 8 lines at a time
Extract replica values
Extract SAR raw data
Write files to disk
Decode SAR data
Apply AGC correction
Write SAR data in MAT file

File header

Chirp replica

Line header

Line aux data

SAR raw data

raw SAR data MAT file

Your application program

Load run parameters

Load AGC values

Load SAR raw data

Process SAR data

CD_run_params.mat

load_AGC_block.m

Load AGC block

Plot AGC values

extract_AGC.m

Decode aux bytes
Convert to decimal

Line aux data

AGC_attenuation_values.mat

load_DATA_block.m

Read SAR raw data
Decode SAR data
Apply AGC correction

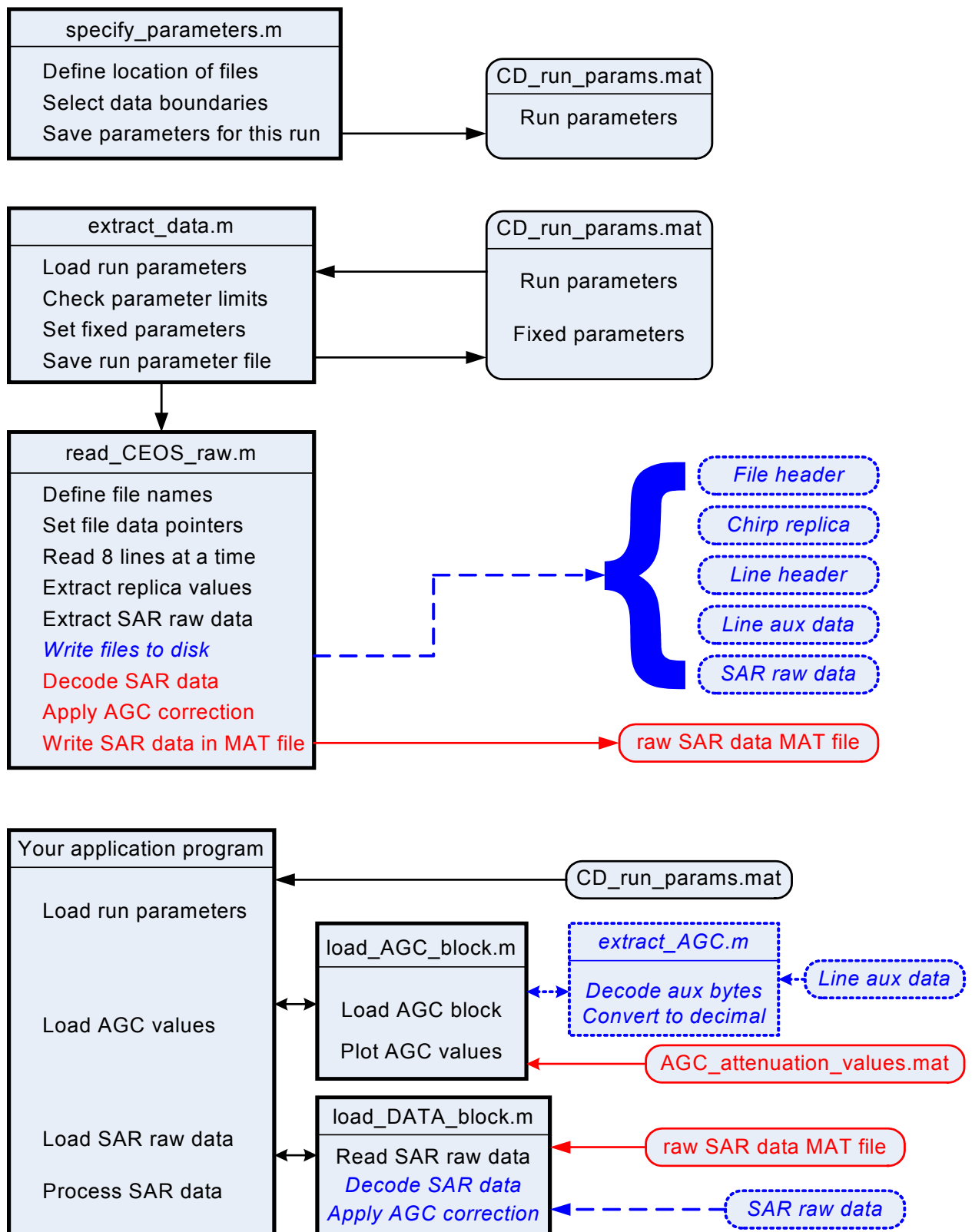raw SAR data MAT file

SAR raw data

Figure 1.   Structure of the MATLAB programs that read SAR data from the CD

These programs have been written in MATLAB 7.0, but have also been tested with MATLAB 6.5. The main difference is that MATLAB 7 allows the use of single precision floating point variables, which saves time and space in the data processing.

`SaveV6`

MATLAB 7 uses data compression when storing MAT files. This saves space, but means they cannot be loaded by a MATLAB 6 program. When using MATLAB 7 programs, the MAT files can be stored in MATLAB 6 format by setting to 1 (true). In this way, you can share files among different users running different versions of MATLAB.

After these parameters are specified, they are stored in the file, "`CD_run_params`" in the current directory, for use by the subsequent data extraction and processing programs.

extract_data.m

This program is run after "`specify_parameters.m`". It calls the function "`read_CEOS_raw.m`" to read the data from the CD, extract the desired data, decode the data (in the case of MAT files) and write the data into easy-to-access disk files.

Because the data on CD has a pulse replica stored every 8 range lines, it is convenient to read the CD 8 lines at a time. To simplify the read procedure, the first range line and number of range lines per block are restricted to be multiples of 8.

Next, a set of parameters are specified that pertain to the specific RADARSAT beam used and certain fixed parameters of the CD. These include radar parameters such as the PRF and sampling rate, and are defined in the in-line comments in the program. Do not change these values for this data set. These parameters are added to the parameters already stored in "`CD_run_params.mat`". These parameters are added in this program in order to keep the program "`specify_parameters.m`" as simple as possible.

Note that while the number of replica samples stored on the CD is 1440 complex samples, 1349 samples is the effective length of the transmitted replica. Its center frequency is zero and its FM rate is `0.72135e+12` Hz/s, rather than the lower value given in the book. The transmitted chirp is generated digitally and is a linear FM function, so it is more accurate to generate the chirp within your processing programs rather than use the 4-bit replica stored on the CD.

read_CEOS_raw.m

This is the function that reads the bits, bytes and words from the book CD and stores them in a form more easily read by your application programs.

There are two ways of storing the extracted data, as set by the parameter "`UseMATfiles`" in "`specify_parameters.m`". If this variable is `true`, the raw SAR data is stored in a MATLAB MAT file. No other data is stored. This is the easiest and quickest way of extracting and using the SAR data if you are going to use MATLAB for all the subsequent processing. In the block diagram on the preceding page, the operations shown in red solid lines pertain to this option.

If the "`UseMATfiles`" parameter is `false`, the data read from the CD is stored in binary disk files, that can be read by any program. In addition to the SAR data and auxiliary data, the file header, the line headers and the pulse replicas are also stored. For most processing, you will not need this extra data, especially as none of the variable parameters like the PRF and range gate delay change in this data set.   In the block diagram in Figure 1, the operations shown in blue dashed lines pertain to this option.

compute_azim_spectra.m

This is an example of an application program you might write. It illustrates how the SAR data is read and used, including adjusting the signal data for the automatic gain control (AGC) attenuation values.

It first loads the "`CD_run_params.mat`" file, which was created when the data was extracted. It then calls the functions "`load_AGC_block.m`" and "`load_DATA_block.m`" for each block of data to be processed, and proceeds with the application.

This application divides the range swath into `Nspect` subswaths. It then finds the azimuth or Doppler power spectrum of each swath, and fits a curve of Doppler centroid vs. range to the estimates. Fitting the Doppler centroid curve is most useful if the whole range swath is specified. In this case, the azimuth blocks should be small, up to 1024 range lines. If a larger azimuth block is desired and memory is critical, multiple blocks can be used.

load_AGC_block.m,   extract_AGC.m

If MAT files are used, `load_AGC_block` simply reads the MAT file, "`AGC_attenuation_values.mat`" that has the AGC attenuation values for the whole CD. This data file is stored on the web site. The program then picks out the values pertaining to the range lines currently being processed.   The AGC attenuation values for the whole CD are given in Figure 2.

Figure 2.   Nominal AGC attenuation values for the 19432 lines on the CD
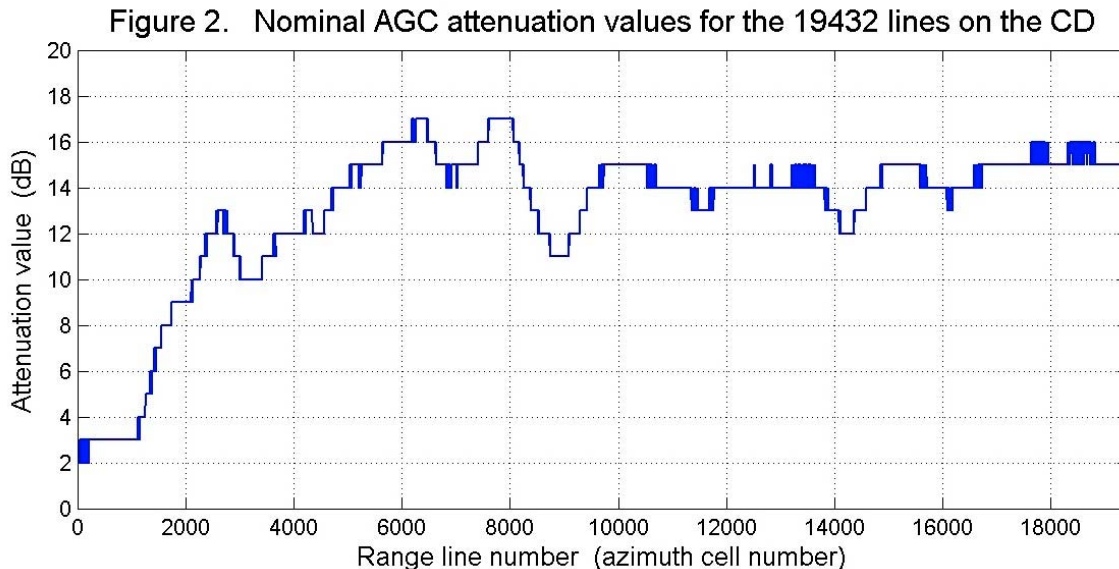
These values are the attenuation values used by the radar receiver for each range line. The AGC system is needed so that the received analog signal stays within a suitable range for the 4-bit A/D converter.   These values should be used in your application program to reverse the effects of the AGC gain changes.

If binary disk files are used, `load_AGC_block` calls the function `extract_AGC.m` to read the auxiliary file associated with the block of data currently being processed. The AGC attenuation values are extracted from the bit stream, converted to a byte then to a decimal value in units of dB.

load_DATA_block.m

If MAT files are used, `load_DATA_block` simply reads the MAT file `CDdatai.mat` that was saved by `read_CEOS_raw.m`. If binary files are used, the data is read from the SAR raw data file, and converted to single (v7) or double (v6) precision floating point data. The data is then decoded and converted to complex numbers (this step is already done in the case of MAT files). Finally, the data is compensated for the receiver AGC attenuation setting, which can change for every range line.

SAR Processing

After `load_AGC_block.m` and `load_DATA_block.m` are completed, the data are now ready for SAR processing. The image of Figure 3 gives a guide as to what starting line and cell number should be used to get an image located at the bottom left corner of the associated grid lines. Note that this image only covers about 65% of the data on the CD.

Good luck with your programs!

Ian Cumming
Kaan Ersahin
January 21, 2005

Figure 3.   Coordinates of Vancouver RADARSAT scene