

# Chapter 4

## *AIRSAR Data Formats*

As described in the previous chapters, the AIRSAR system is an airborne SAR which operates simultaneously in a fully polarimetric mode at three frequencies. The data collected by the AIRSAR system are processed to polarimetric imagery at JPL and provided to the user community in digital and photographic form. This chapter describes the various products produced and the formats of the “standard product” digital data provided to the user community from AIRSAR processors with version numbers larger than 3.5.

The AIRSAR system produces several different output products. This includes the different data types, as well as the photo products accompanying some of these data types. The survey imagery (used for selecting areas for further processing) and the final processed digital products are described in some detail here. Two of the most common digital products are the *frame products* and the *synoptic products*. Frame products correspond to about 12 km along track of imagery by 10-15 km across track, while synoptic data covers about 62 km along track with the same across track swath as the frame product. A special digital data product, the so-called *compressed scattering matrix*, is also described.

Real-time images, processed in a low resolution survey mode, are provided to the principal investigator(s) for each SAR pass. The output product is a black and white single frequency, single polarization (typically LHH) image of the entire pass. No digital survey data are provided. Annotation on the image allows the investigator to select areas for further processing. In order to get data processed by the AIRSAR ground processor, the investigator must fill out a processing request form and submit this to JPL. Two types of processing may be chosen: frame or synoptic, with a selection of data products for each. The processed digital data is typically provided on computer compatible tapes (CCTs), with both 9-track and 8mm tapes supported as media. Each tape contains a number of files corresponding to image data (in a specific format with specific headers), followed by a number of files containing ancillary information.

### **4-1 Real-Time Imagery - Survey Data**

The AIRSAR real-time processor on-board the DC-8, known as the flight correlator, is used to produce low resolution imagery which is distributed to investigators. The purpose of this product is to provide the

investigators with a “quick look” at the data acquired. Also, using these survey images, requests for frame or synoptic processing can be made.

A typical flight campaign for the AIRSAR crew involves data collection followed by one or two down days on which no flights take place. During this down period, the flight correlator is operated in a real-time mode at one channel (typically LHH) as an entire raw data tape is played back. The resulting image is stored on disk, and printed to a continuous tone printer as a black and white image. (In the past, black and white transparencies have been used.) Several copies of this image are made so that they may be distributed to the investigator(s) and archived at JPL.

Due to the nature of the real-time data, no digital data are provided - only the photo-product is officially archived and available. Figure 4.1 shows an example image produced from the real-time processor. The annotation which appears on the real-time imagery is decoded from the header information which is stored with the raw radar data. The exact information which is provided on the images has evolved over the past few years, but as of the time this document is written includes the following five pieces of information:

**Run Name:** Name assigned by radar operator for this data acquisition pass. Typically includes the site name, the aircraft heading and an incrementing number which indicates the sequence of this run.

**GMT Time:** The day of year followed by the current GMT time.

**A/C LAT-LON:** The latitude and longitude of the aircraft - not the imaged swath. Decoded from the DC-8 inertial navigation unit which is updated frequently.

**Framecount:** Each transmitted pulse is coded with a incrementing number known as the framecount. This number is the most important for making processing requests.

**FREQ/POL:** The frequency and polarization being displayed.

The annotation is displayed in the far range portion of the imagery. The position of the aircraft corresponding to the latitude and longitude shown is approximately straight across from the annotation and just off the near range of the image shown.



**Figure 4.1.** *Example of survey imagery produced by the AIRSAR flight correlator.*

Other indicators present on the real-time imagery include small crosses at 25, 40 and 55 degree incidence angles. The positions of these crosses are calculated assuming a flat earth. *Note that the survey images are not radiometrically calibrated; in fact, the antenna patterns are not even removed. These images should only be used to select areas for further processing, and are not suitable for quantitative data analysis.*

#### 4-1-1 Real-Time Processing Algorithm

The algorithm used to produce the real-time imagery is known as *Doppler beam sharpening* or *unfocused SAR*. The particular implementation produces about a 15 km wide image at two looks and about 25-30 meter resolution. This is achieved by filtering the raw SAR data in azimuth to this resolution using a low-pass filter, and *then* performing two-look range processing using FFT correlation techniques.

Azimuth focusing is achieved by taking advantage of the fact that near zero Doppler (directly broadsides of the aircraft), the phase from targets on the ground does not change very quickly from pulse to pulse :

$$\phi(t) = 2\pi \frac{V^2}{\lambda R} t^2 + \phi_0 \quad (4.1)$$

where  $V$  is the aircraft velocity,  $R$  is the slant range to the target, and  $\lambda$  is the radar wavelength. From (4.1) it follows that for  $|t| < \sqrt{\lambda R} / (4V)$ , the phase can be considered a constant equal to  $\phi_0$ . Because of this *stationary phase*, azimuth points coherently added together will add in phase over a time corresponding

to  $\pm\sqrt{\lambda R} / (4V)$  to the time of closest approach. Data that are outside this time window have quickly changing phasors and will tend to sum to zero. Therefore, after the data are range compressed, a simple azimuth filter of the appropriate time constant will produce a focusing effect with a resolution of about  $\sqrt{\lambda R} / 2$ .

For the AIRSAR system, the theoretical resolutions (at 45 degrees incidence angle) are 13m, 26m and 44m for C, L and P bands, respectively. Under nominal operating conditions, the number of azimuth points corresponding to the ideal azimuth filter bandwidths are 17, 34 and 58, respectively for C, L and P bands. The azimuth processor in the flight correlator real-time processor is tuned to L-Band using 34 azimuth points in the filter, but still performs well, with some degraded performance, at C and P bands.

## 4-2 Tape Format for Digital Data

The AIRSAR ground processor is capable of producing two different types of data. The frame processor processes all twelve channels of radar data into three-frequency fully polarimetric images. The typical size of an image is about 12 km along track, and 10-15 km across track. For those investigators that are willing to trade radar channels for increase image size, the synoptic processor is used to process three (selected by the investigator) of the twelve radar channels into radar cross-section only images, *i.e.* no phases are preserved in the synoptic processor. The synoptic images cover about 62 km along track with the same across track swath as the frame product.

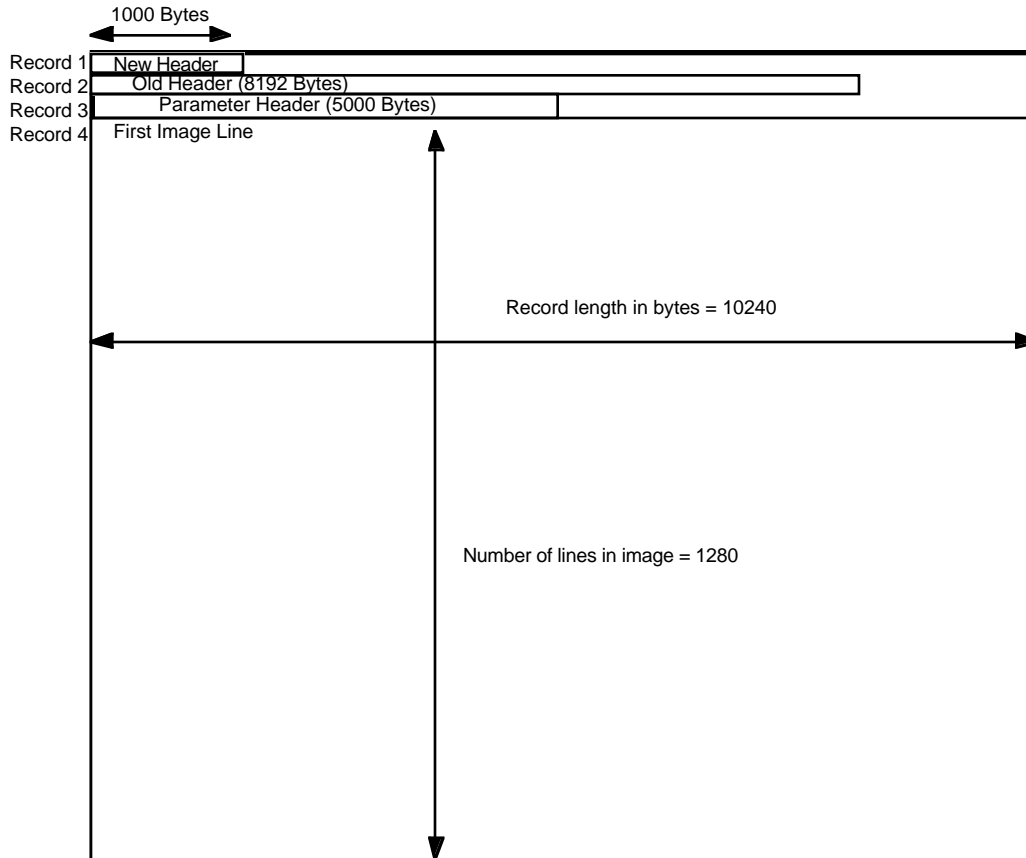
The standard AIRSAR data tapes produced by the ground processor are known as “compression tapes” or “CM” tapes for the frame product, “synoptic tapes” or “SY” tapes for the synoptic processor and “compressed scattering tapes” or “CS” tapes for frame data provided in compressed complex scattering matrix form. Two types of files are found on the tapes: data files and navigational files.

For the standard frame processor output, each CM tape will typically contain three data files corresponding to the SAR imagery. These will be the first three files on the tape. A single file contains the compressed Stokes matrix data of a scene at a particular frequency (P, L or C-band), plus some ancillary (header) information. The order of these files is always P, L and C. The SY tapes also contain three files, one for each channel processed - a channel being a particular frequency/polarization channel. The order of these files on the tapes is the order in which they are shown in the photo product from top to bottom, which is typically in P, L and C order and HH, HV, VH and VV order within a particular frequency. For example if PHH, LHH and CVV were processed during a synoptic run, the order would be PHH, LHH and CVV; if PHH, CHH and CVV were processed the order would be PHH, CHH and CVV. There are six navigational text files following the three data files on each of the tapes. The tape format is simply a series of files of fixed length records, and is not computer or operating system specific. The navigational data are not stored on CS tapes.

### 4-2-1 Data File Header Structure

The header structure for the data files produced by AIRSAR processors with version numbers larger than 3.5 was introduced in 1989, and was updated in 1991. This format is shown below. The first few records of each data file contain “header” information, and the remaining records contain the corresponding radar data from the AIRSAR scene.

The first record of the file (see Figure 4.2) is a header record describing the general characteristics of the file. This record is described in appendix A and is referred to as the *new header*. Note that one of the parameters in the new header (parameter number 13) indicates the location of the first image line. This allows one to add any number of additional headers, as long as this parameter is updated. *Users writing software to analyze AIRSAR data should use the information in the new header when reading data or header information.*



**Figure 4.2.** Typical CM file format with three header records. For SY data, the record length is typically 5120 (1280 real\*4 pixels) and there are 5088 lines of image data.

The second record in the file is the *old header*, which was found in record 1025 in data processed prior to 1989. The information in this header is stripped directly from the raw radar data on the HDDT and represents in many cases more information than is used by most investigators. This record is retained for use by users who have software that require this information. The most frequently used information found in this header has been included in the newer parameter header. *We urge all investigators to use the information in the parameter header when developing software to analyze AIRSAR data.*

The third header record is the *parameter header*. This header record was introduced in 1991 (processor version 3.50 and greater) and appears with all CM and SY data currently processed (and not in the CS data). Figure

4.2 shows the file layout of a CM data set. Appendix A defines the new header and parameter header entries and provides an example of the headers found on a typical data product.

The new header also contains a special field (field 12) which is a byte offset to a user header. This field allows users to add their own headers to AIRSAR data. If this is done, both fields 12 and 13 of the new header should be updated to have the correct offsets. *Users should not use the standard AIRSAR headers to store information.* Some of the fields currently not defined are to be used in the future, and some data or information may be lost if user software writes over these fields. *If at all possible, user software should create user headers to store information.*

#### 4-2-2 Detailed Data Formats: CM Data

Compressed Stokes matrix data (CM data) is the standard pixel format provided on the CM tapes. Each line, as indicated in the CM header, corresponds to data that are at a constant range, *i.e.* in the azimuth direction. The reasons for providing data in this compressed form are given below, along with the algorithm used to compress (decompress) the data.

In the AIRSAR frame processor, raw data are initially processed into scattering matrix form for each frequency and each of 16 looks. Scattering matrix form simply provides a complex backscatter measure for each pixel of each of the four sampled linear polarizations. The raw output from the processor is therefore twelve data sets with complex numbers, corresponding to each frequency and polarization. The size of each of these data sets is 160 MBytes, for a total of 1.92 GBytes of data per frame. (A frame corresponds to about 60 seconds of data acquisition.) This volume of data is quite large, typically larger than most investigators can handle. However, conventional multi-look techniques can not be applied to the polarimetric scattering matrix data because the phase information, required for polarimetry, is lost during the incoherent look summation of the data.

A compression scheme was introduced to reduce this data volume from 1.92 GBytes per scene to 37.5 MBytes (Dubois and Norikane, 1987). This scheme involves converting the scattering matrix data into Stokes matrix data, which retain the polarimetric information. The Stokes matrix is composed of various *real* cross products of the scattering matrix, and therefore, may be multi-looked. In the case of the AIRSAR data, the data are processed to 16 individual looks, which are converted to Stokes matrix format and summed together. Additional compression is gained by efficiently coding the matrix elements in a pseudo block floating point format. The CM data correspond to these compressed multi-look Stokes matrix pixels, retaining all the phase information necessary to perform polarimetric analysis. A total data volume reduction factor of 51.2 is achieved.

##### 4-2-2-1 Stokes Matrix Compression Algorithm

The Stokes matrix is a 4x4 real matrix (van Zyl and Ulaby, 1990). The most important assumption behind the AIRSAR data compression scheme is that the measured scattering matrix should be symmetric, *i.e.*  $S_{vh} = S_{hv}$ . This follows from reciprocity (van Zyl and Ulaby, 1990) and the fact that the AIRSAR operates in the backscatter mode. The resulting Stokes matrix will also be symmetrical. The expressions relating the elements of the Stokes matrix,  $\mathbf{M}$  to the elements of the scattering matrix,  $\mathbf{S}$ , in this case are:

$$M_{11} = \frac{1}{4} [S_{hh} \cdot S_{hh}^* + S_{vv} \cdot S_{vv}^* + 2S_{hv} \cdot S_{hv}^*] \quad (4.2)$$

$$M_{12} = \frac{1}{4} [S_{hh} \cdot S_{hh}^* - S_{vv} \cdot S_{vv}^*] \quad (4.3)$$

$$M_{13} = \frac{1}{2} \Re[S_{hh} \cdot S_{hv}^*] + \frac{1}{2} \Re[S_{hv} \cdot S_{vv}^*] \quad (4.4)$$

$$M_{14} = -\frac{1}{2} \Im[S_{hh} \cdot S_{hv}^*] - \frac{1}{2} \Im[S_{hv} \cdot S_{vv}^*] \quad (4.5)$$

$$M_{22} = \frac{1}{4} [S_{hh} \cdot S_{hh}^* + S_{vv} \cdot S_{vv}^* - 2S_{hv} \cdot S_{hv}^*] \quad (4.6)$$

$$M_{23} = \frac{1}{2} \Re[S_{hh} \cdot S_{hv}^*] - \frac{1}{2} \Re[S_{hv} \cdot S_{vv}^*] \quad (4.7)$$

$$M_{24} = -\frac{1}{2} \Im[S_{hh} \cdot S_{hv}^*] + \frac{1}{2} \Im[S_{hv} \cdot S_{vv}^*] \quad (4.8)$$

$$M_{33} = \frac{1}{2} S_{hv} \cdot S_{hv}^* + \frac{1}{2} \Re[S_{hh} \cdot S_{vv}^*] \quad (4.9)$$

$$M_{34} = -\frac{1}{2} \Im[S_{hh} \cdot S_{vv}^*] \quad (4.10)$$

$$M_{44} = \frac{1}{2} S_{hv} \cdot S_{hv}^* - \frac{1}{2} \Re[S_{hh} \cdot S_{vv}^*] \quad (4.11)$$

where  $\Re[\ ]$  and  $\Im[\ ]$  represent the real and imaginary parts of the subsequent quantities, respectively. The remaining elements are filled in using the assumption that the matrix is symmetric. It is easily shown using (4.2)-(4.11) that (van Zyl, 1985)

$$M_{11} = M_{22} + M_{33} + M_{44} \quad (4.12)$$

$$M_{13}M_{23} + M_{14}M_{24} = M_{12}(M_{11} - M_{22}) \quad (4.13)$$

$$M_{13}M_{14} + M_{23}M_{24} = M_{34}(M_{33} + M_{44}) \quad (4.14)$$

$$M_{13}^2 + M_{14}^2 + M_{23}^2 + M_{24}^2 = M_{11}^2 - M_{22}^2 \quad (4.15)$$

$$M_{13}^2 - M_{14}^2 - M_{23}^2 + M_{24}^2 = M_{33}^2 - M_{44}^2 \quad (4.16)$$

Once one averages a number of Stokes matrices, as is done during the multi-look procedure, only the first of these five relationships still remain. This means that a maximum of 9 independent numbers per Stokes matrix results after averaging. We therefore only need to store 9 numbers for each Stokes matrix.

The compression procedure originally introduced by Dubois and Norikane (1987), is implemented as follows. First, the average of the phase difference between the  $vh$  and  $hv$  components of the scattering matrix is measured using a subset of the scene. Then, the scattering matrix is symmetrized by performing the following calculation:

$$S_{hv} = \frac{1}{2} [S_{hv} + S_{vh} e^{i\phi_{vh-hv}}] \quad (4.17)$$

where  $\phi_{vh-hv}$  is the average of the phase difference between the  $vh$  and  $hv$  components of the scattering matrix. This symmetrized matrix is then used to calculate the symmetrical Stokes scattering operator. The value of  $\phi_{vh-hv}$  is saved in the parameter header of the data file, as described in appendix A.

Next, each data set is scaled by a single value for all pixels such that the dynamic range of the total power elements in the Stokes matrices ( the  $M_{11}$  element ) falls within values between  $2^{-128}$  and  $2^{127}$ . Since radar images typically exhibit a dynamic range less than 30 dB, this condition is easily met. The total power for each matrix is then coded into two bytes, one for the exponent in the above range and one

for the mantissa. The remaining eight elements are then normalized to the  $M_{11}$  element in the Stokes matrix.

Four of these elements (those related to the cross products of co- and cross-polarized channels) are observationally found to be much smaller than  $M_{11}$ ; many theories predict these elements to be small or zero for most natural surfaces. Therefore, before encoding, the square root of each of the four normalized elements is calculated. Each of the eight values is next truncated to eight bits (one byte) and the resulting eight bytes are saved. These eight bytes plus the two for the total power are then stored, requiring only 10 bytes per pixel.

The equations for the compression and decompression operations are given here. For data encoding from the Stokes matrix to the compressed 10 byte format, the following apply:

$$\text{byte}(1) = \text{Int} \left( \frac{\log_2 M_{11}}{\log_2 2} \right) \quad (4.18)$$

where  $\text{Int}()$  is the integer part of the subsequent quantity.

$$\text{byte}(2) = \text{Int} \left( 254 \cdot \left( \frac{M_{11}}{2^{\text{byte}(1)}} - 1.5 \right) \right) \quad (4.19)$$

We then compute a normalization factor  $x$  approximately equal to  $M_{11}$ . Choosing this factor for normalization results in slightly smaller errors than using  $M_{11}$  directly:

$$x = \left( \frac{\text{byte}(2)}{254} + 1.5 \right) \cdot 2^{\text{byte}(1)} \cdot \text{gen\_fac} \quad (4.20)$$

The general scale factor,  $\text{gen\_fac}$ , is the average of all the  $M_{11}$  values in the image, and is stored in field 133 of the *old header*. The remaining eight bytes are coded as follows, with  $\text{sign}() = +/- 1$ , reflecting the sign of the quantity in brackets:

$$\text{byte}(3) = 127 \cdot M_{12} / x \quad (4.21)$$

$$\text{byte}(4) = 127 \cdot \text{sign}(M_{13} / x) \cdot \sqrt{M_{13} / x} \quad (4.22)$$

$$\text{byte}(5) = 127 \cdot \text{sign}(M_{14} / x) \cdot \sqrt{M_{14} / x} \quad (4.23)$$

$$\text{byte}(6) = 127 \cdot \text{sign}(M_{23} / x) \cdot \sqrt{M_{23} / x} \quad (4.24)$$

$$\text{byte}(7) = 127 \cdot \text{sign}(M_{24} / x) \cdot \sqrt{M_{24} / x} \quad (4.25)$$

$$\text{byte}(8) = 127 \cdot M_{33} / x \quad (4.26)$$

$$\text{byte}(9) = 127 \cdot M_{34} / x \quad (4.27)$$

$$\text{byte}(10) = 127 \cdot M_{44} / x \quad (4.28)$$

To reconstruct the Stokes matrix from the reduced data, the following inverse operations are required:

$$M_{11} = \left( \frac{\text{byte}(2)}{254} + 1.5 \right) \cdot 2^{\text{byte}(1)} \cdot \text{gen\_fac} \quad (4.29)$$

$$M_{12} = \text{byte}(3) \cdot \frac{M_{11}}{127} \quad (4.30)$$



$$M_{13} = \text{sign}(\text{byte}(4)) \cdot \left( \frac{\text{byte}(4)}{127} \right) * * 2 \cdot M_1 \quad (4.31)$$

$$M_{14} = \text{sign}(\text{byte}(5)) \cdot \left( \frac{\text{byte}(5)}{127} \right) * * 2 \cdot M_1 \quad (4.32)$$

$$M_{23} = \text{sign}(\text{byte}(6)) \cdot \left( \frac{\text{byte}(6)}{127} \right) * * 2 \cdot M \quad (4.33)$$

$$M_{24} = \text{sign}(\text{byte}(7)) \cdot \left( \frac{\text{byte}(7)}{127} \right) * * 2 \cdot M_1 \quad (4.34)$$

$$M_{33} = \text{byte}(8) \cdot \frac{M_{11}}{127} \quad (4.35)$$

$$M_{34} = \text{byte}(9) \cdot \frac{M_{11}}{127} \quad (4.36)$$

$$M_{44} = \text{byte}(10) \cdot \frac{M_{11}}{127} \quad (4.37)$$

$$M_{22} = M_{11} - M_{33} - M_{44} \quad (4.38)$$

#### 4-2-3 Detailed Data Formats: Compressed Scattering Matrix

Starting with the AIRSAR Processor Version 2.3, all scattering matrix data are produced in a compressed format. The tape prefix is “CS” and each line, as indicated in the CS tape header, corresponds to data that are at a constant range, *i.e.* in the azimuth direction. This format is the 8-bit algorithm described in detail by van Zyl and Burnette (1991).

The advantages of this algorithm are the following:

- Since quantizations are done at the electric field as opposed to power level, the signal-to-compression noise ratio is better than 35 dB even for the cross-polarized components.
- The phase errors introduced by this method have standard deviations less than 0.6 degrees.
- These files have the same record length and number of bytes per sample as our current Stokes matrix compressed (CM) files. In fact, one can display the total power image for these files using exactly the same software one would use for the Stokes matrix compressed files.
- The data volume is reduced from 32 bytes per pixel (4 complex numbers) to 10 bytes per pixel.

Operationally, this compression algorithm is implemented in ground processors with version numbers larger than 3.5 to produce four compressed scattering matrix scenes, each consisting of one of four center looks of the original sixteen looks.

This compression algorithm is implemented as follows. First, the full data set is scaled by a single value for all pixels such that the average *total power*, *i.e.*:

$$TP = \frac{1}{4} \left( |S_{hh}|^2 + |S_{hv}|^2 + |S_{vh}|^2 + |S_{vv}|^2 \right) \quad (4.39)$$

of the image is unity. This scale factor, which is called the *general scale factor*, is supplied to the user in a header that forms part of the data file (field 133 of old header). The scaled total power for each scattering matrix is then coded into two bytes, one for the exponent, and one for the mantissa. Since we use the binary basis, and use a full byte for the exponent, it follows that we allow scaled total power values in the range between  $2^{-128}$  and  $2^{127}$ ; a total dynamic range of 765 dB! Since the scaled total power values

typically exhibit less than about 30 dB dynamic range, this condition is easily met. We note here that so far this algorithm is identical to the Stokes matrix compression algorithm. Since the same number of bytes per pixel are used to store the data, the same file structures could be used to store the data from both the 8-bit and the multi-look Stokes matrix compression algorithms.

From the expression of the total power given in (4.39), it follows that the square root of the *span* of the scattering matrix, which is four times the total power, is greater than or equal to the amplitude of any of the individual scattering matrix elements. Thus, if each of these scattering matrix elements is divided by the square root of the span, the real and imaginary parts of the resulting normalized scattering matrix elements will be in the range -1 to +1. During compression, we multiply these normalized values by 127 and code the real and imaginary parts each as one byte, resulting in eight bytes for the total normalized scattering matrix. Note that this is different from the Stokes matrix method where the elements of the Stokes matrix are normalized and saved. Also, we save both the normalized  $hv$  and  $vh$  elements, thereby preserving any possible asymmetry in the measured scattering matrix. We also note here that since we use one byte for each of the real and imaginary parts of each scattering matrix element, the dynamic range of the quantization process is 48 dB, i.e. the compression algorithm allows for a 48 dB range in values *inside each scattering matrix*. (This is in addition to the 765 dB dynamic range allowed for the total power values - the reader is reminded that each scattering matrix is normalized by its own span.)

The explicit equations for the 8-bit compression algorithm are:

$$\text{byte}(1) = \text{Int}\left(\frac{\log_2 TP}{\log_2 2}\right) \quad (4.40)$$

$$\text{byte}(2) = \text{Int}\left(254 \cdot \left(\frac{TP}{2^{\text{byte}(1)}} - 1.5\right)\right) \quad (4.41)$$

where , as before,  $\text{Int}(x)$  is the integer part of  $x$ . We then compute a normalization factor  $y$ , which is approximately equal to the square root of the span of the scattering matrix. Choosing this factor for normalization results in slightly smaller errors than using the square root of the span directly. The normalization factor  $y$  is calculated as follows using the general scale factor,  $gen\_fac$ , found in the old header (field 133):

$$y = 2 \cdot \sqrt{\left(\frac{\text{byte}(2)}{254} + 1.5\right) \cdot 2^{\text{byte}(1)} \cdot gen\_fac} \quad (4.42)$$

The remaining eight bytes are coded as follows:

$$\text{byte}(3) = 127 \cdot \Re(S_{hh}) / y \quad (4.43)$$

$$\text{byte}(4) = 127 \cdot \Im(S_{hh}) / y \quad (4.44)$$

$$\text{byte}(5) = 127 \cdot \Re(S_{hv}) / y \quad (4.45)$$

$$\text{byte}(6) = 127 \cdot \Im(S_{hv}) / y \quad (4.46)$$

$$\text{byte}(7) = 127 \cdot \Re(S_{vh}) / y \quad (4.47)$$

$$\text{byte}(8) = 127 \cdot \Im(S_{vh}) / y \quad (4.48)$$

$$\text{byte}(9) = 127 \cdot \Re(S_{vv}) / y \quad (4.49)$$

$$\text{byte}(10) = 127 \cdot \Im(S_{vv}) / y \quad (4.50)$$

As before,  $\Re[]$  and  $\Im[]$  represent the real and imaginary parts of the subsequent quantities, respectively.

To reconstruct the original scattering matrix from the compressed data, the following operations are required:

$$\Re(S_{hh}) = \text{byte}(3) \cdot y / 127 \quad (4.51)$$

$$\Im(S_{hh}) = \text{byte}(4) \cdot y / 127 \quad (4.52)$$

$$\Re(S_{hv}) = \text{byte}(5) \cdot y / 127 \quad (4.53)$$

$$\Im(S_{hv}) = \text{byte}(6) \cdot y / 127 \quad (4.54)$$

$$\Re(S_{vh}) = \text{byte}(7) \cdot y / 127 \quad (4.55)$$

$$\Im(S_{vh}) = \text{byte}(8) \cdot y / 127 \quad (4.56)$$

$$\Re(S_{vv}) = \text{byte}(9) \cdot y / 127 \quad (4.57)$$

$$\Im(S_{vv}) = \text{byte}(10) \cdot y / 127 \quad (4.58)$$

To change programs written to analyze the compressed Stokes matrix data to analyze the compressed scattering matrix data, one simply has to change the decompression part in the existing software. In the case of the compressed scattering matrix data, one first has to decompress the scattering matrix data and then calculate an equivalent Stokes matrix for each pixel. The equations describing these calculations can be found in (4.2)-(4.11). Note that in compressing the scattering matrix data, the scattering matrices are not symmetrized, i.e. the full scattering matrix is preserved. *Also note that the compressed scattering matrix data are not archived at JPL.*

#### 4-2-4 Detailed Data Formats: SY data

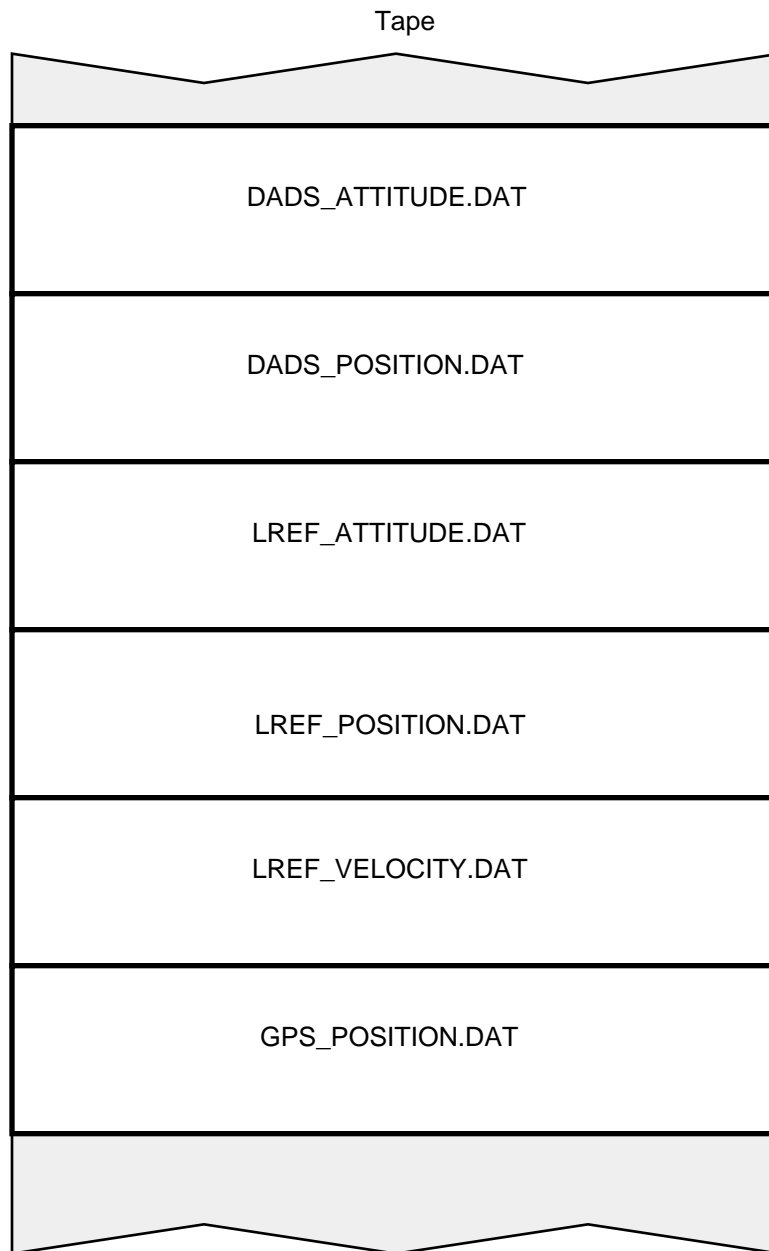
The synoptic data are processed to multi-look *amplitude* pixels. No phase information is retained in this product and can therefore not be used to synthesize polarimetric information. The product is intended for investigators who wish to view a larger area than supplied by the frame product, but only at selected frequencies and polarizations.

Synoptic data (SY) files use the same header structure as the CM data files. However, the format of the individual pixels is a single 4 byte floating point number. The data are in Digital Equipment Corporation VAX real\*4 format, so care should be taken when using the data on non-VAX computers. The data is supplied in units of the square root of power, nominally the square root of sigma zero directly. Each data line of an SY file is oriented in the range direction, *i.e.* it contains data of equal azimuth, with the first pixel in the line at the range closest to the radar, and the last pixel in the line at the range furthest from the radar. This is different from the frame product (CM) data, where each data line represents values of equal range. Field 15 of the new header specifies how the data lines are oriented-see Appendix A.

#### 4-2-5 Detailed Data Formats: Navigational Data Files

In response to requests from users who wish to geocode AIRSAR data for use in Geographical Information Systems, we now also provide six small files with relevant navigational data. These six data files contain navigational data derived from the DC-8 data acquisition and distribution system (DADS) system, the AIRSAR Laser Ref inertial navigational system, and the AIRSAR global positioning system (GPS) receiver. This navigational information is embedded in the AIRSAR raw data main and subcommutated headers during a data take. Each of the six files supplies information about the DC-8 on one second intervals during the time in which the raw radar data used to produce the images were recorded.

The six navigational data files all have the same format. In each case, the first line is of the form TITLE\_IDENTIFICATION.DAT, where TITLE indicates the title of the processed scene and IDENTIFICATION indicates the information summarized in each output column. For example, the DC-8 DADS system data file contains the identifier DADS\_ATTITUDE. The first line is then followed by a number of output lines containing the relevant navigational information. The number of output lines is a function of the length of data take processed. In the case of the frame processor, each file will contain about 70 lines, while in the case of the synoptic processor, each file will contain about 300 lines. The order of the files on tape is shown in Figure 4.3.



**Figure 4.3.** Order of the navigational data files on the tape. The first navigational data file is shown at the top (DADS\_ATTITUDE.DAT).

The output columns for each of the six data files are as follows:

**TITLE\_DADS\_ATTITUDE.DAT:**

GMT TIME	DADS_YAW	DADS_PITCH	DADS_ROLL
		.	
		.	
GMT TIME	DADS_YAW	DADS_PITCH	DADS_ROLL

**TITLE\_DADS\_POSITION.DAT:**

GMT TIME	DADS_LAT	DADS_LON	RADAR_ALT
		.	
		.	
GMT TIME	DADS_LAT	DADS_LON	RADAR_ALT
POSITION OF PLANE			

**TITLE\_LREF\_ATTITUDE.DAT:**

GMT TIME	LREF_YAW	LREF_PITCH	LREF_ROLL
		.	
		.	
GMT TIME	LREF_YAW	LREF_PITCH	LREF_ROLL

**TITLE\_LREF\_POSITION.DAT:**

GMT TIME	LREF_LAT	LREF_LON
		.
		.
GMT TIME	LREF_LAT	LREF_LON

**TITLE\_LREF\_VELOCITY.DAT:**

```
GMT TIME  LREF_N/S  LREF_E/W  LREF_VERT  LREF_HEADING
```

```
.
```

```
.
```

```
.
```

```
GMT TIME  LREF_N/S  LREF_E/W  LREF_VERT  LREF_HEADING
```

**TITLE\_GPS\_POSITION.DAT:**

```
GMT TIME  GPS_PDOP  GPS_LAT  GPS_LON  GPS_ALT
```

```
.
```

```
.
```

```
.
```

```
GMT TIME  GPS_PDOP  GPS_LAT  GPS_LON  GPS_ALT
```

Each file contains a maximum of five columns of formatted ASCII text, however, as can be seen from the definitions above, most of the data files do not make use of all five available fields. Those fields which are unused are filled with spaces (ASCII 32). Data for each of the six output files is formatted through a single format statement. This FORTRAN format statement is as follows:

```
c*****
c   Single Output Format Statement For All The Navigational   c   Data Files:
c
c           107   Format(a12,t15,a14,t31,a14,t47,a14,t63,a14)
c*****
```

To really use these data, one has to be able to connect the position of the DC-8 with the image data. This is accomplished by calculating the time at which the DC-8 illuminated (with the center of azimuth antenna pattern) the center pixel of the deskewed image. The latitude and the longitude of the plane, calculated from the DADS information, at this time is supplied as the last data line of the file containing the DADS positional information, and is specified as the POSITION OF PLANE in the file specification above. Note that this may not be the time at which the plane was at the point of closest approach to this pixel due to the plane's yaw and pitch pointing the antennas slightly forward or aft of perfect sidelooking.

**4-3 Photo Product**

Starting in September 1990, a new standard photo product accompanied all JPL AIRSAR standard Frame products. This product was decided upon using the following assumptions:

- The photo product is not intended to be used for quantitative data analysis. Its main purpose is to show the user, in photo format, what area was processed.
- The photo product is produced on a dye diffusion color printer, negatives are not produced.
- Photo products are produced the same day the data are processed so that photo production does not limit the product delivery speed.

Using these assumptions, it was decided that for the frame processor, the standard photo product will be a single total power color overlay with the P-band total power image displayed in red, the L-band total power image displayed in green and the C-band total power image displayed in blue. The images are produced in the same projection as the data to avoid any confusion. Since the current AIRSAR standard product is in slant range projection, this means that all photo products are currently produced in slant range projection. Figure 4.4 is a color photocopy of the standard color photo product for a scene imaged near Pellston, Michigan during the Summer of 1991

The synoptic photo product consists of four image strips. The top three are black-and-white images of the three individual channels selected for processing. The frequency and polarization of each channel is identified by the annotation immediately to the right of the image strips (see Figure 4.5). The color image strip in the bottom is made by displaying the top three strips in the colors indicated by the annotation immediately to the right of each black-and-white strip and is identified as a "THREE CHANNEL OVERLAY." Note that due to the limited number of pixels available during reproduction on the diffusion dye printer, the synoptic photo product is produced by averaging over a 4 x 4 box in the range and azimuth directions. Therefore, the Synoptic photo product is of lower resolution than the actual Synoptic data.

#### 4-3-1 General Layout

The photo product consists of an image with some annotation surrounding the actual image. Most of the annotation are on the right of the actual image and will be discussed in more detail below.

Shown at the top is annotation identifying the image. On the left is the JPL AIRSAR identification, while on the right the image is identified as a TOTAL POWER overlay for the frame processor and as SYNOPTIC for the synoptic processor. In the center is the SCENE TITLE assigned during processing of the data.

On the left of the image are tick marks with incidence angle values in degrees. These incidence angles are calculated assuming a flat earth, and using the same aircraft altitude as was used to remove the antenna patterns during processing (see A/C Altitude below). In the case of the synoptic processor, incidence angle marks are shown to the left of each of the image strips. On the top right are arrows indicating the direction in which the aircraft was moving (labeled velocity) and the direction of radar illumination. Below the text on the right-hand side is an arrow indicating the direction of TRUE NORTH. In the case of the frame processor, a scale bar showing 1 km in the azimuth direction is shown below the image. A scale bar showing 10 km is shown to the right of the color strip in the Synoptic processor photo product. Note that these scales do not apply to the range direction. Finally, in the case of the frame processor, the actual radar bands (with the wavelengths in centimeters in parentheses) are shown below the azimuth scale bar. These words, for example P-BAND (68 cm) are written in the color assigned to the image of that particular frequency. Therefore, P-BAND (68 cm) is written in red, L-BAND (24 cm) is written in green and C-BAND (5.7 cm) is written in blue.



**Figure 4.4.** *Example of the AIRSAR standard Frame photo product. The scene, Pellston, Michigan, was imaged during the Summer 1991 campaign.*





**Figure 4.5.** *Example of the AIRSAR synoptic photo product. The scene, the ocean off the coast of England, was imaged during the Summer 1991 campaign.*

### 4-3-2 Sensor Parameters

The parameters related to the radar and the aircraft are displayed in the first of three sub-blocks of annotation directly to the right of the image. These parameters are:

**Near Range** This is the slant range distance from the radar to the first image line and is quoted in meters.

**A/C Altitude** This is the altitude used during processing to remove the antenna patterns when performing radiometric calibration. For processor versions 3.50 and later, we use radar altimeter derived altitudes during processing.

**A/C Heading** This shows the heading of the aircraft with respect to True North during the data take and is given in degrees.

**A/C Yaw** The yaw of the aircraft (in degrees) at the start of the scene.

**A/C Pitch** The pitch of the aircraft (in degrees) at the start of the scene.

**A/C Roll** The roll of the aircraft (in degrees) at the start of the scene.

**Bandwidth** This shows which bandwidth the radar was operating in for the data take. Possible values are 20 and 40 MHz.

**Pulse Length** The length of the pulse transmitted by the radar during the data take. Possible values are 10 and 5 microseconds.

**Frequencies** This shows the frequencies of the radars that were operated simultaneously. For the current standard AIRSAR product, this can only be PLC.

**Quad Pol Mode** This shows the polarization mode of the radars as they were operated during the data take. For the current standard AIRSAR product, this can only be Quad Pol Mode, meaning that the full scattering matrix was measured for each pixel in the image.

### 4-3-3 Data Parameters

The parameters related to the radar data are displayed in the second of three sub-blocks of annotation directly to the right of the image. These parameters are:

**Run Name** The name assigned to the data run at the time of the data acquisition.

**Date Acquired** The date on which the data were acquired.

**Scene Start** The start time of the data for this scene in GMT.

**Approximate Scene Center** This is the approximate latitude and the longitude of the scene center. (For processor versions before 2.3, the longitude is placed in the wrong hemisphere.)

**No. Range Samples** The number of actual data (not counting headers) lines in the image.

**No. Azimuth Samples** The number of samples per data line.

**HDDT Input ID** The number of the HDDT on which the raw data were recorded during the radar overflight.

**JPL Output ID** The JPL identification number assigned to this data set.

**Projection** The projection in which the data are presented. Currently, the AIRSAR standard product is in SLANT range projection.

#### 4-3-4 Processing Parameters

The parameters related to the processor are displayed in the third of three sub-blocks of annotation directly to the right of the image. These parameters are:

**Processor Version** The version number of the AIRSAR processor used to process this data set. Note that the synoptic and frame processor version numbers are updated independently.

**Date Processed** The date on which the data were processed.

**Range Pixel Size** The range pixel spacing in the image (in meters), measured in the projection appropriate for this scene.

**Azimuth Pixel Size** The pixel spacing in the azimuth direction, and is quoted in meters.

**Frame Count** This number is the frame count at the start of the data transfer, and is about 20,000 frame counts less than the frame count at the center of the image for Frame products and about 50,000 frame counts less than the center frame count for the Synoptic processor.

**Processed Yaw, deg** The yaw angle used in the processor.

**Processed Pitch, deg** The pitch angle used in the processor.

**Processed Roll, deg** The roll angle used in the processor.

**Processed Alt, m** The actual altitude used in the processor to perform radiometric correction.