AN OVERVIEW OF XSAT DATA DOWNLINK AND IMAGE PROCESSING SYSTEM DESIGN AND DEVELOPMENT

Wee Juan Tan [a], Leong Keong Kwoh [b]

Centre for Remote Imaging, Sensing and Processing, National University of Singapore, Block S17 Level 2, 10 Lower Kent Ridge Road, Singapore 119076, Republic of Singapore

[a] Research Scientist, Tel: +65-65165567, E-mail: crstwj@nus.edu.sg

[b] Director, Tel: +65-65163220

KEY WORDS: Image processing, ground station operation

ABSTRACT: To produce images from a raw data bitstream from Earth observation satellites, a multi step process is employed. It includes reception via X-band antenna, down conversion, demodulation, disambiguation and frame formatting, error correction if any, data descrambling, image generation and timing recovery, radiometric and geometric correction, and finally archiving and distribution. Apart from antenna and demodulator hardware subsystems, XSAT's data ground station's design and development was an in-house effort at CRISP, National University of Singapore. In this paper, we present the design and operation of the various components of the XSAT data ground station. This includes the software based QPSK disambiguation algorithm, the image processing software for normalization and formatting, archival and distribution architecture and the automation process to increase operational efficiency.

1. XSAT Data Processing Ground Station Overview

The XSAT project provided CRISP with an opportunity to design and develop in-house a complete data processing system chain. The processing chain thus could be tuned to specific requirements such as the ability to run on generic hardware, efficiency and autonomous operation. The sequence of events that take place from reception of X-band signal to completion of image processing is detailed in the following sections.

2.1 RF Subsystems and Demodulation

XSAT has an X-band data downlink using QPSK modulation. The antenna, RF, demodulator and DIS subsystems are regular commercial off the shelf CRISP operated hardware components. These subsystems handle the tracking of XSAT, reception of the X-band signal, demodulation of the down converted signal, and the extraction of the I and Q channel bits. The demodulator does not perform any form of I/Q bit order correction or frame extraction. The raw bits are deposited onto a Direct Ingest System, whereupon the data processing chain starts.

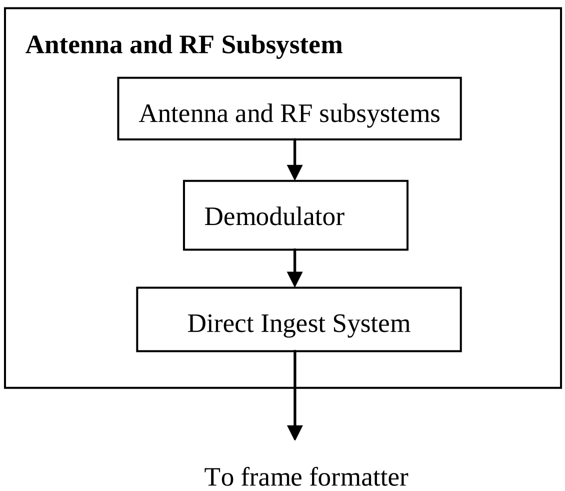

Fig. 1. Antenna and RF Subsystem.

2.2 Frame Formatting

The demodulator is instructed to perform a naïve I/Q bit merging to produce a raw data file. The bits in this data file are not byte aligned, nor are the bits corrected for flipping or swapping. The objective of the frame formatter is to identify situations of I/Q bit swapping or flipping, correct for them and produce byte align frames with the attached synch marker at the start of each frame. Suppose an arbitrary synch marker *m* of *n*-bytes is given by :

$$m = \{b_0, b_1, b_2, \dots b_{n-1}\}$$

and each byte in the marker is 8 bits in length:

$$b_i = \{x_0, x_1, x_2, \dots x_7\}$$

where x is a single bit. Then the bit-wise permutations for each byte in the synch marker would be:

I/Q normal, no inversion	$b_{i0} = \{x_0, x_1, x_2, \dots x_7\}$
I/Q normal, I and Q inverted	$b_{i1} = \{\sim x_0, \sim x_1, \sim x_2, \dots \sim x_7\}$
I/Q normal, I inverted	$b_{i2} = \{\sim x_0, x_1, \sim x_2, \dots \sim x_7\}$
I/Q normal, Q inverted	$b_{i3} = \{x_0, \sim x_1, x_2, \dots x_7\}$
I/Q reversal	$b_{i4} = \{x_1, x_0, x_3, \dots x_7, x_6\}$
I/Q reversal, Q inverted	$b_{i5} = \{\sim x_1, x_0, \sim x_3, \dots x_6\}$
I/Q reversal, I inverted	$b_{i6} = \{x_1, \sim x_0, x_3, \dots \sim x_6\}$
I/Q reversal, I and Q inverted	$b_{i7} = \{\sim x_1, \sim x_0, \sim x_3, \dots \sim x_6\}$

Table 1. I/Q Permutations in QPSK modulation systems.

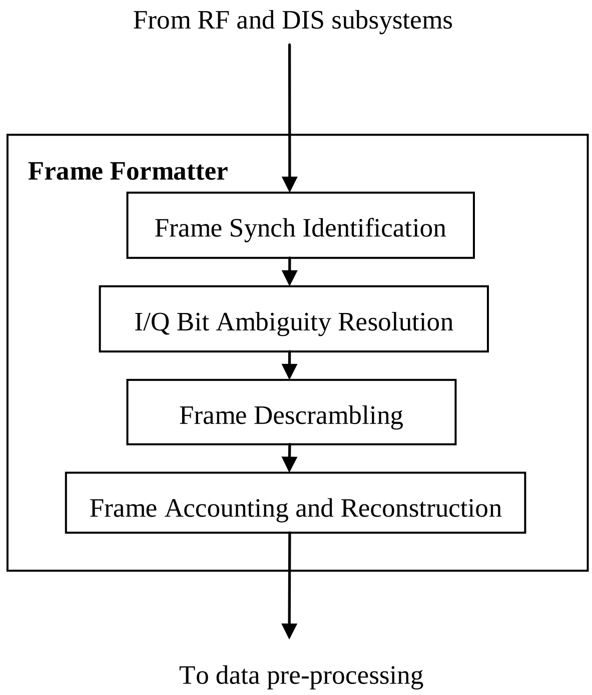

Fig. 2. Frame Formatter

We are therefore presented with a family of 8 markers $M = \{m_0, m_1, m_2, \dots m_7\}$ where each member *m* is a different combination of the constituent byte permutation. [1]

The unaltered, original synch marker would thus be represented as:

$$m_0 = \{b_{00}, b_{10}, b_{20}, b_{30}, \dots b_{n0}\}$$

Any arbitrary member m_j of the family of markers can be represented as:

$$m_j = \{b_{0j}, b_{1j}, b_{2j}, b_{3j}, \dots b_{nj}\}$$

It is this family of synch markers of length *n* that is searched for at each bit location in the raw stream. This is done by bit-slipping an *n*-byte search window through the raw file. At each bit position, the search window's nx8 bits is cast into *n*-bytes and byte-wise compared to family members. If an exact match is found, ambiguity resolution is performed on the following frame length of bytes (minus the synch marker length) in a manner dependent upon the marker permutation identified. The I/Q disambiguation process outputs frames that are ready for descrambling.

XSAT frames are scrambled to maximize bit transition density. The descramble pattern is generated according to [2] and each byte of the descramble pattern is XOR-ed with each corresponding byte of the disambiguated frame. Upon descrambling and any further decryption as needed, frame indices and counters are examined to determine if there are lost or mangled frames. Such frames are reconstructed using blank payload frames with relevant timing and PPS transitions reconstructed in order to keep frame accounting in strict compliance with the data frame format and file meta structure.

2.3 Data Pre-Processing

Once the frames have been disambiguated, descrambled and fully accounted for, pre-processing of payload frames begins. XSAT has various payloads with different processing requirements. CRISP is responsible for the processing of the data from the electro-optical payload known as IRIS. This is a multispectral camera with a designed ground sample distance of 10m. Data from the various payloads are interlaced together in the x-band data stream. They need to be separated based on internal frame sub-headers. Once the IRIS payload has been isolated, it is treated to a dark current compensation and a relative normalization process to remove CCD sensor striping.

This is done using the following equation:

$$O_i = I_i \times G_i + C_i$$

where I_i is an input digital number from sensor element at location *i*, and O_i is the compensated output digital number. G_i is the gain co-efficient and C_i the dark current offset. This vendor proposed correction technique is similar to the simple procedure detailed by Horn and Woodham [3]. More elaborate and potentially more effective techniques are being studied.

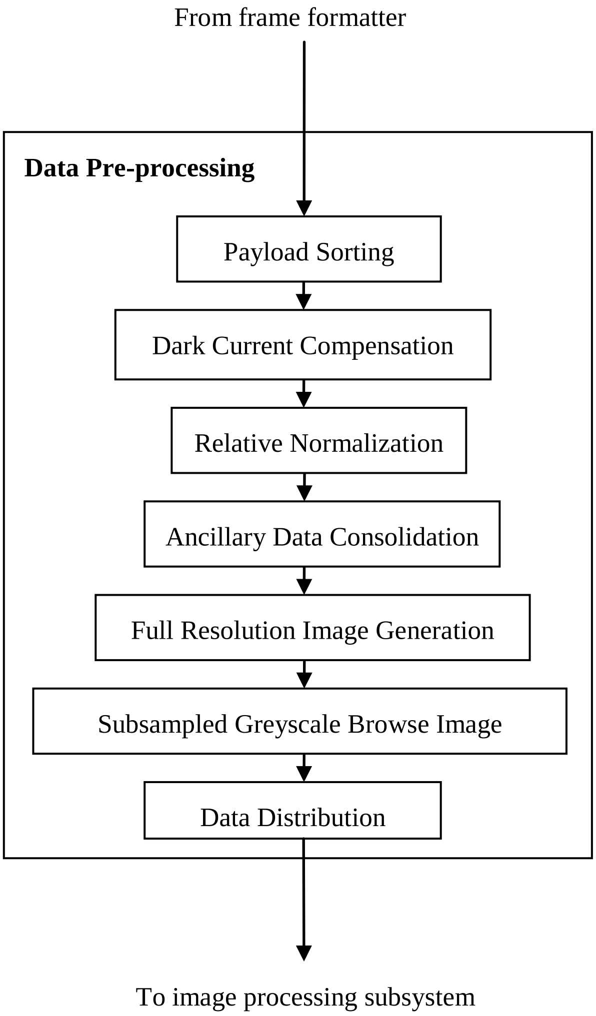

Fig. 3. Data Pre-processing.

Data necessary for extracting timing information is located and extracted from ancillary data frames and externalized into human readable format. At this point, three level 0 images at full resolution are produced in 8-bit uncompressed TIFF format, one for each IRIS channel. A compressed, sub-sampled version is also produced for convenient preliminary browsing. The output of the pre-processing stage is needed by various parties, both inside and outside CRISP. A data-push procedure is performed to get the data to the interested parties.

2.4 Image Processing

In order for XSAT images to perform useful tasks such as environmental monitoring, they must be processed to a higher level, and the various channels stacked to produce colour images. The XSAT is designed initially to provide GPS timing and location information and star tracker attitude information as ancillary data required for such processing. Unfortunately, due to a design error, the star tracker is not operable, thus no a-priori attitude information is available. A workaround implemented by CRISP is to make use of ground control points picked by operators from GoogleEarth.

The GCP and image-coordinate pairs are fed into a geometric processing software, which has the camera model as its core, to back-compute the attitude information (roll, pitch and yaw). This software module allows for the application of up to 4^{th} order polynomials of roll, pitch and yaw angles, although this is usually restricted to as low an order as is able to produce well registered images. The FOV of the camera can also be adjusted in the software. This process will allow the stacking of the channels to form full resolution colour images. A compressed, subsampled version of the colour images is prepared for distribution as browse images. At this point, images are ready for cataloguing and archiving.

2.5 Image Distribution and Archiving

Once the final geometrically corrected image has been produced a series of abutted browse images of roughly square aspect is produced. This is uploaded together with geo-information to a web catalogue server. With their corner and centre lat-lon known, a client browser will be provided with a wireframe representation of the XSAT image. When selected by the user, the browse image is displayed.

Alongside the dissemination via the web catalogue, the level-0 images and the raw files are archived in a network storage server.

This would mark the completion of the processing chain.

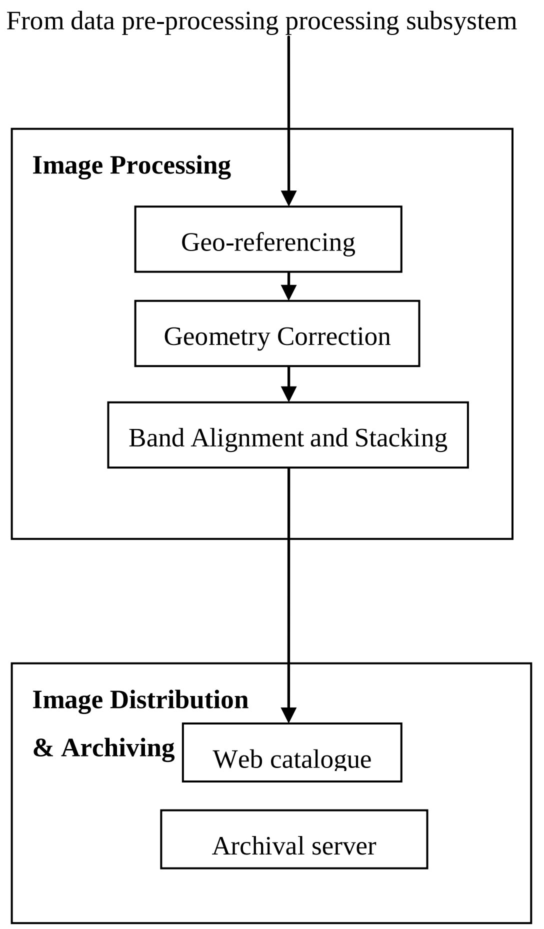

Fig. 4. Image Processing, Image Distribution and Archiving

2.6 Automation of Data Processing Operations

The processing chain is automated with the exception of ground control point selection. Once the antenna subsystem has been scheduled to receive data from XSAT, the process up to the point of level-0 ouput requires no further operator input. This is intentional in order to maximize operational efficiency of the ground station. XSAT and the data ground station at CRISP have the capability to schedule multiple imaging operations over a period of time, say a weekend. During this period, imaging missions are carried out fully autonomously, completely independent of operator input. This automation is accomplished through the use of Linux shell scripts, and by designing software executables to be small, modular and loosely coupled.

3 Platform, Technology and Software Design

The main criteria for platform and operating system selection was:

1) Avoid being locked into closed systems with little control over upgrade cycles. It is envisaged that the basic building blocks of XSAT's data ground processing system will form the foundation for any future projects. Thus future proofing of the platform is of importance.

2) Lower administration overheads. A platform less susceptible to intrusion and malware would result in lower administrative overhead.
3) Ease of interfacing with existing systems. The CRISP ground station infrastructure already has an installed base of machines running support systems such as tape backup systems, etc. To efficiently interface with this established infrastructure, it was important that XSAT's operating systems be compatible.

The main criteria for the selection of development environment, tools and software architecture were:

1) Flexibility. While satellite ground stations are not new entities and their operations are generally well understood, a degree of post-launch flexibility in operations is useful. A single full-function software application would not lend itself to easy functionality modification to support minor changes in operations. These minor changes could include a simple request to output intermediate data, or provide extra logging facilities or distribute more data to more clients. Small executables encapsulating core functionality scripted together would more easily allow for such changes.
2) Stability. While many new application development technologies have presented themselves as candidates over the past 10 to 15 years, it was decided that software stability and predictability would be a major requirement. As the development team is small and agile, there is no provision for long term continuous maintenance of software sets based on semi-mature, constantly evolving software libraries. Thus software languages and platforms of recent vintage are avoided. Some were compelling in their functionality, but due to their frequent update/deprecation cycle, had to be ruled out.

To satisfy all the above criteria, Linux was chosen for its extensive, mature shell scripting environment. This allowed quick and easy linking together of new and established, in-house and open-source modules to form greater functionality at minimal expense. For instance, to incorporate data distribution between processing stages, simple scripted ftp commands could be added. Had a monolithic software application been built, that may have required writing new code and a complete recompile and software test cycle. C/C++ was selected for the development of software executables, and Eclipse and CodeBlocks as the development environment. Software libraries were restricted to STL and the cross platform application framework wxWidgets, and the standard C lib. GCC and djcc compilers were used on account of stability and familiarity. Operational machines in the ground station are generic X86 machines running various versions of Linux.

4 Conclusion and Future Work

XSAT is an indigenously developed micro satellite whose space and ground segments are developed in-house. CRISP's data and image processing subsystems have been described in this paper. Ongoing work includes periodic update of the relative normalization and dark current data, and various efficiency optimisations.

References

1. Tien Manh Nguyen, JPL Publication 89-4, Part 1, *Phase-Ambiguity Resolution for QPSK Modulation Systems – Part 1: A Review*, May 15, 1989.

2. CCSDS Secretariat, Program Integration Division, NASA, *TELEMETRY CHANNEL CODING, 101.0-B-4*, May 1999.

3. B.K.P. Horn and R.J. Woodham, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, A.I. Memo No. 467, *Destriping Satellite Images,* March 1978.