Satellite Data Decompression Software Framework for High Data Rate Missions

Yugandhar Reddy S. [1*], Md Irfan Salauddin.[2], Manikumar Vedantam.[3], Prakash Chauhan[4]

[1] Head, Real-time Data Ingest & Pre-Processing Software Division

[2] Scientist/Engineer SF

[3] Group Head, Satellite Real-Time Data Processing Group

[4] Director, National Remote Sensing Centre

[1,2,3,4] National Remote Sensing Centre, ISRO, India

yugandhar_reddy@nrsc.gov.in

ABSTRACT

Technological advancements in the fields of sensor, optical and electro-mechanical lead to increase the spatial, spectral, radiometric and temporal resolution of space borne missions. It eventually results into generation of tremendous amount of data onboard satellite. Various compression techniques have been evolved to handle this large amount of data for efficient usage of channel bandwidth and storage space. CCSDS(Consultative Committee for Space Data Systems) has recommended DWT(Discrete Wavelet Transform) based image data compression standard 122.0-B-1. ISRO's Cartosat-3 satellite provides panchromatic imagery with a spatial resolution of 0.28 m using 24 onboard detectors and Multispectral imagery in four spectral bands with a spatial resolution of 1.12 m using 96 detectors, with a nominal swath of ~17 Km. High Data Rate (HDR) of 2.88 Gbps in Ka band transmission and image data compression technique of CCSDS 122.0-B-1 is employed in Catrosat-3 mission. To handle the large volume of data (10 minutes of satellite pass generates ~200 GB data) and computationally intensive processing of decompression algorithm, a frame work is designed and implemented as part of ground segment establishment to improve the turnaround time for decompression. It is implemented using multi processing techniques with data sharing. Synchronization between processes is achieved through POSIX Inter Process Communication(IPC). In this framework, multiple processes are spawned for performing detector wise decompression of both panchromatic and multispectral data , followed by detector-wise merging of data. Various tasks related to fetching of input data, file writing is performed in parallel. Framework supports configurable number of compressed data segments to be passed to decompression engine to avoid frequent spawning of processes. Turnaround time for decompression is improved by a factor of 2x by reading of compressed data , performing decompression, merging of data and writing of data to disk in parallel for each detector.

Keywords: *parallel, framework, image data compression, decompression , turnaround time*

1. Introduction

The earth observation satellites collect data from various instruments and sensors depends on mission requirements. The collected data is digitized and undergoes various processing steps as per the digital communication system shown in fig 1. In this figure the processing steps shown are onboard as well as on ground. Few of the processing blocks are not mentioned like modulation & demodulation for the simplification purpose.

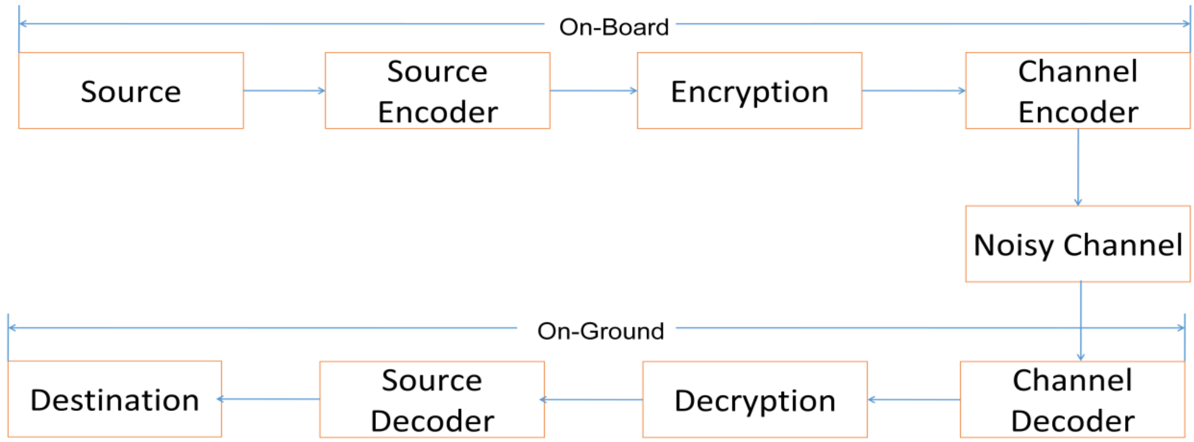


Figure 1: Digital Communication System

The amount of data getting generated from earth observation satellites increasing to meet ever increasing demand from user community w.r.t higher spatial, temporal, spectral and radiometric resolution. Technological advancements in the fields of sensor, optical and in electro-mechanical fields contributed to aid user demands and ultimately resulted in generating huge amount of data onboard. Earth observation satellites from ISRO (Indian Space Research Organization) is no exception to this hence the EO(Earth Observation) missions from ISRO are generating data from few mbps(mega bits per secs) to few Gbps (Giga bits per sec) to cater mission requirements. To handle such a tremendous data generated onboard effectively, various compression techniques have been employed onboard in terms of source coding and decompression is performed on ground as part of source decoder. Encryption is performed onboard to make the data secure by converting data to cyphertext and decryption is performed on-ground to get the data back. Channel encoder and channel decoder is used to make the data transmission error free through noisy channel by incorporating error correction capability by adding redundancy to the data in a user controlled manner. This paper mainly deals with the source encoder & source decoder part hence remaining digital communication blocks are not covered.

One of the recent ISRO's high resolution satellites Catrosat-3 provides panchromatic imagery with a spatial resolution of 0.28m and Multispectral imagery in four spectral bands with a spatial resolution of 1.12 m with a nominal swath of ~17 Km. Cartosat-3 satellite provide the downlink data rate of 2880 Gbps in ka band transmission. The satellite specifications are mentioned in table 1.

Table 1: Cartosat-3 Specifications

PARAMETERS	PAN	MULTISPECTRAL
Ground Sampling Distance (GSD)	0.28 m	1.12 m
Swath	~17 km	~17km
Spectral Band width (µm)	0.45 - 0.9	B1:0.45 - 0.52 B2:0.52 - 0 59 B3:0.62 - 0.68 B4:0.77 - 0.86
Quantization	11 Bits	11 Bits
Nominal TDI Stages	45	45
No. of Detectors	24	24 * 4
SSR Capacity(3Nos)	3.6 TB	

Cartosat-3 satellite employs on-board compression as per the CCSDS 122.0-B-1 standard to support sub-meter resolution and high swath. It is a DWT based image data compression standard applied on band wise and detector wise data. Integer DWT coefficients are calculated hence it supports lossless compression. On ground, Decompression software is used for detector-wise decompression of data and then merge to form payload lines. This paper describes the decompression software framework and its architecture along with the compression and decompression implementation algorithm.

2. Literature Review

Source coding techniques are essential to remove the redundancy in the acquired data for efficient usage of channel bandwidth. Various source coding techniques are existed[4,7] and wavelet based compression technique will be very effective for multirate analysis and non stationary signals[8,10]. CCSDS address common problems of space data systems and adapted wavelet based source coding technique in terms of Image data compression standard[1].

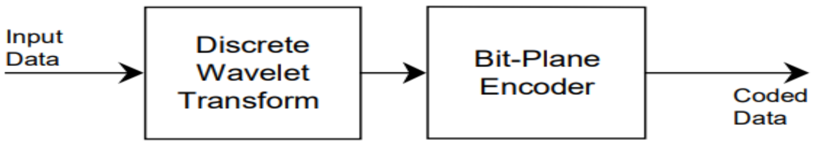
To implement the on ground decompression software framework, the following goals are kept i.e. it should be able to perform configurable number of independent tasks in parallel and easily extendable to any other mission which is having any kind of compression standard. The ultimate goal of the framework is to complete the decompression job as quick as possible by performing the parallel I/O operations(read & write) & input data need to be kept available to decompression engine readily to take of the next chunks of data to decompress.

The compression algorithm need to understand thoroughly to implement the decompression algorithm accurately and effectively. CCSDS bluebook[1] (Recommended standard 122.0-B-1) and green book[2](CCSDS 120.1-G-2) are referred for implementation of the decompression library and modification. The compression algorithm is DWT based and it can be effectively implemented using lifting scheme proposed by Wim Swldens[10] . One of the most complex part of the decompression algorithm bit plane decoder and the open source library provided by University of Nebraska is modified to according to the Cartosat-3 data compression scheme. The source code and the user manual is referred to understand the complex data structures in the code. Blue book is referred for integer inverse DWT equations. The bit plane encoder uses the different encoding schemes for different data like DC coefficients are encoded using variable length coding of RICE coding; AC bit depth & DC bit Depth are encoded using modified RICE coding. The data compression blue book[3] CCSDS 121.0-B-1 is referred for decoding logic.

The data parallelism, Task parallelism and interprocess communication are the key elements of the framework. Data parallelism is designed to perform the decompression within the detector to effective utilization of the system resources and speedup. Task parallelism is done to perform the different independent tasks in parallel like I/O(reading & writing of the data) and performing decompression. Interprocess communication is achieved through shared memory and synchronization between the processes is achieved through POSIX semaphores[6]. The framework is designed to avoid the race conditions.

3. CCSDS 122.0-B-1 algorithm

The source coding theorem selected for Cartosat-3 is CCSDS recommended integer DWT based 122.0-B-1. It has mainly two blocks as shown in figure 2.



Source: Blue book of CCSDS 122.0-B-1

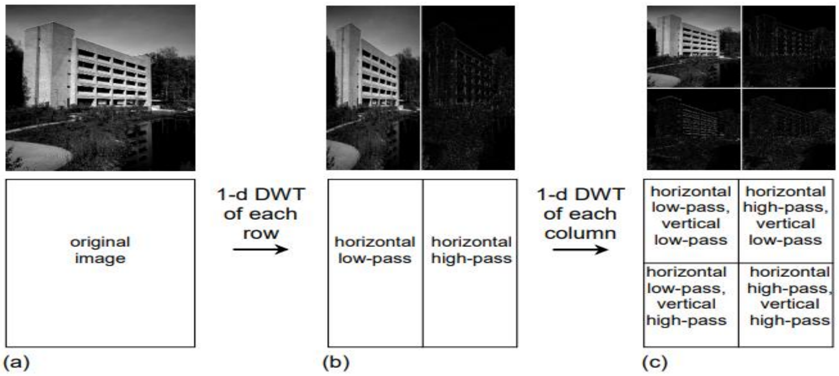
Figure 2 : General schematic of compressor

Input data undergoes discrete wavelet transform through which the data is decorrelated and the decorrelated data undergoes encoding using bit-plane encoder. In the decorrelation step , the data will be performed 3 level, two dimensional separable discrete wavelet transform using nine and seven taps of low pass and high filters respectively. Three level transform is applied reputedly on single level two dimensional DWT. Single level DWT calculation on the two dimensional (Rows & Columns of the image) image is mentioned below steps.

Step 1. The one dimensional dwt is applied as per the equations mentioned blue book, to each row of the image segment (Here the segment is the size of the image, on which the compression is applied) row wise resulted in to generation of DWT coefficients of both low pass and high pass of each half wide as the original image array. This step generates horizontall low pass and horizontal high pass sub bands.

Step 2. Then one dimensional dwt is applied to each columns of the image segment on the coefficients generated in step -1 column wise , resulted in to generation of DWT coefficients of both low pass and high pass each half wide as the original image array. This step generates four sub bands namely horizontal low pass vertical low pass(LL_1), horizontal high pass vertical low pass(HL_1), horizontal low pass vertical high pass(HL_1) and horizontal high pass vertical high pass(VV_1) sub bands. Subscript 1 indicates the 1st level.

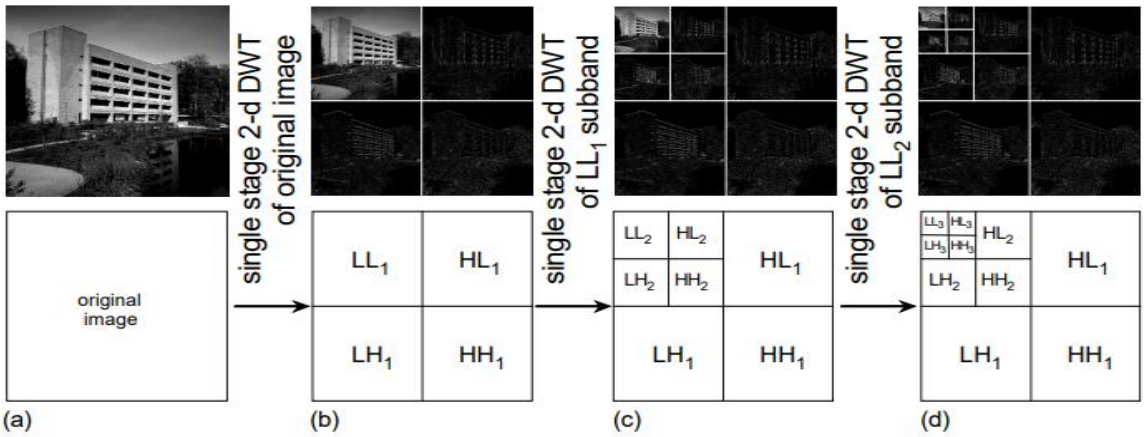
Schematic diagram of the above two steps are mentioned in the figure 3.



Source: Blue book of CCSDS 122.0-B-1

Figure 3 : Schematic of single level two dimensional DWT

As we can make out the low pass coefficients capture global or smooth variations of the image whereas the high pass coefficients give details/edges/discontinuities of the image. We need to preserve the high pass coefficients and hence second level DWT is applied on the horizontal low pass vertical low pass(LL_1) sub band in a similar manner of single level DWT calculation mentioned in the step 1 & 2. The second level DWT calculation to LL1 sub band generates four sub bands namely LL_2 , HL_2 , LH_2 and HH_2 sub bands. The third level DWT is applied to LL_2 coefficients in a similar manner of single level DWT calculation mentioned in the step 1 & 2. The third level DWT calculation to LL_2 sub band generates four sub bands namely LL_3 , HL_3, LH_3 and HH_3 sub bands. Schematic diagram of the three level 2-d DWT is shown in figure 4.



Source: Blue book of CCSDS 122.0-B-1

Figure 4 : Schematic of three level two dimensional DWT

Inverse DWT is applied to reconstruct the image data from the multi level 2d DWT coefficients as follows.

1. First LL_2 sub band needs to be reconstructed from LL_3, LH_3 , HL_3 and HH_3 sub bands. Column wise inverse DWT to be applied by taking LL3 and LH3 coefficients to get

intermediate low pass and similarly to intermediate high pass can be derived by applying inverse DWT to the column wise for HL3 and HH3 coefficients. Row wise inverse DWT to be applied for the intermediate low pass and intermediate high pass coefficients to get LL2 sub band. Schematic diagram of figure 5 shows the retrieval of sub band.

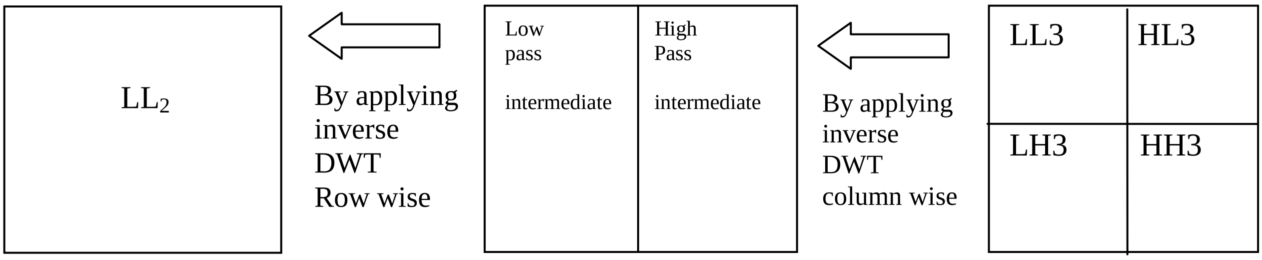
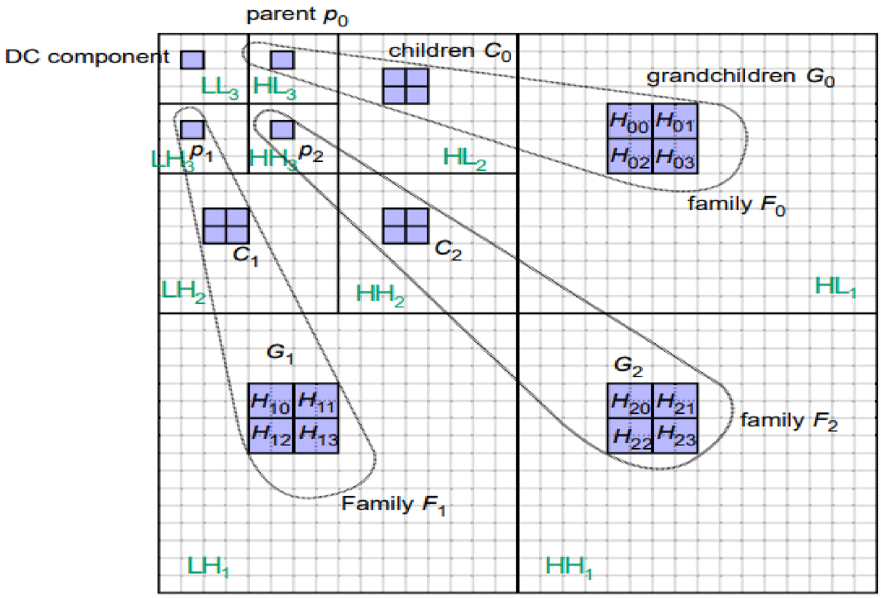


Figure 5 : Schematic of single level two dimensional inverse DWT

2. LL_1 sub band can be derived from the sub bands of LL2 , LH2, HL2 and HH2 by applying inverse DWT column wise followed by row wise as men tied in the step-1

3. Similarly original image can be retrieved from the sub bands of LL1, LH1, HL1 and HH1 by applying DWT column wise followed by row wise as mentioned in step-1

The bit plane encoder (BPE) encodes the coefficients in the group of 64 coefficients referred to as blocks. One block of 64 coefficients consists of one coefficient from LL3 referred to as DC and 63 AC coefficients from three families F0, F1 and F2. Each family contains 20 coefficients consists of one parent coefficient, four children coefficient and 16 grand children coefficient as shown in figure 5.

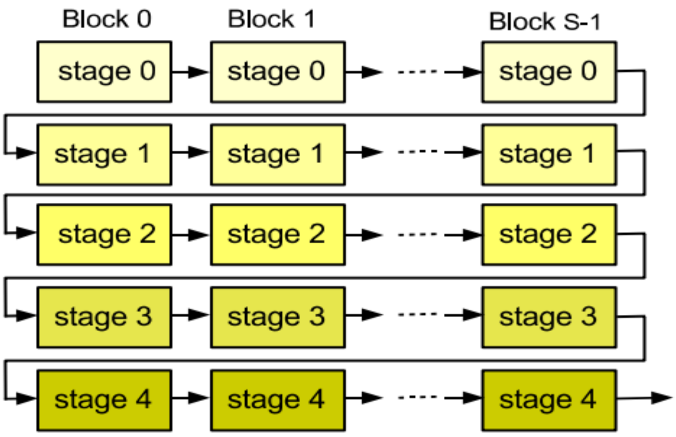


Source: Blue book of CCSDS 122.0-B-1

Figure 6 : Schematic of single level two dimensional inverse DWT

The DWT coefficients are divided in to number of segments and the segment size is defined as group of consecutive blocks. The coding of DWT coefficients done segment by segment and each segment is independently coded. Each segment is coded stages wise as stage-0, stage-1, stage-2, stage-3 and stage-4. Stage-0 consists of left out DC bit representation of that bitplane if any ; stage-1-4 represents the AC coefficients of the parent, child and grandparents.

The DWT coefficients encoded successively biplane wise in a given segment. Bit plane 'b' consists of the b^{th} bit of the two's-complement integer representation of each DC coefficient, and the b^{th} bit of the binary integer representation of the magnitude of each AC coefficient. The BPE encodes in sequence of highest bitplane 'b' to lowest bitplane '0' as the coding progresses hence it is also called as progressive coding. Encoder calculates the highest bit depth amongst of the all the AC coefficients of all the blocks of the segment and bitplane coding starts from that bitplane onwards for that segment and continue to encode till the 0^{th} bitplane. The BPE can stop the encoding of the coefficients either the quality limit is reached or the segment byte limit is reached. Quality limit indicates up to which bit plane it is encoded and the segment byte limit is set based on the fixed rate of the data which in turn depends on the compression ratio. The figure 7 shows the structure of single bitplane and figure 8 shows the compression segment details of all the biplanes.



Source: Blue book of CCSDS 122.0-B-1

Figure 7 : Schematic of compressed data structure of single bitplane

Segment Header
Coding of DC Coefficients
Coding of AC Coefficient bit depth

Coding of bit plane = BitDepthAC-1
Coding of bit plane = BitDepthAC-2
. . .
Coding of bit plane = 0

Source : Blue book of CCSDS 122.0-B-1

Figure 8 : Schematic of compressed data structure of all biplanes

Segment Header

Each compressed segment begins with a segment header consisting of the following parts in the following order:

a) Part 1 (three or four bytes, mandatory);

b) Part 2 (five bytes, optional);

c) Part 3 (three bytes, optional);

d) Part 4 (eight bytes, optional)

By CCSDS convention, all reserved bits in each header part shall be set to ‘zero’.

Part 1A	Part 1B	Part 2	Part 3	Part 4
3 Bytes	1 Byte	5 bytes	3 Bytes	8 Bytes

The details of the coding scheme of DC coefficients, AC bit Depths and details of coding of bit planes can be found in CCSDS 122.0-B-1 blue book.

2.1 Cartosat-3 compression

Detector wise and band wise compression is enabled for Cartosat-3 and it has rate fixed compression. The compression segment is derived by taking eight payload lines along the track direction and three level two dimension DWT is applied as per the compression standard. The compression is performed for the next eight lines by taking the actual image values from the previous image segment instead of duplicating the values from the current image segment. Hence the image continuity can be seamless and the end flag (Start and End flags are indicative of start and end of compressed segments available in segment header) will not be set throughout the image strip. on ground Decompression process to handle this. It means that every time whenever the data is passed to decompression engine, sufficient overlap needs to be ensured to decompress the data seamlessly. As in the figure 9 to decompress the 10 compressed segments of data the previous 3 segments of data and next 3 segments of data need to pass to decompression library and in the output first & last 24 lines of decompressed data need to be discarded. Please note that for one compressed segment of Crtosat-3 data resulted into 8 payload lines after decompression.

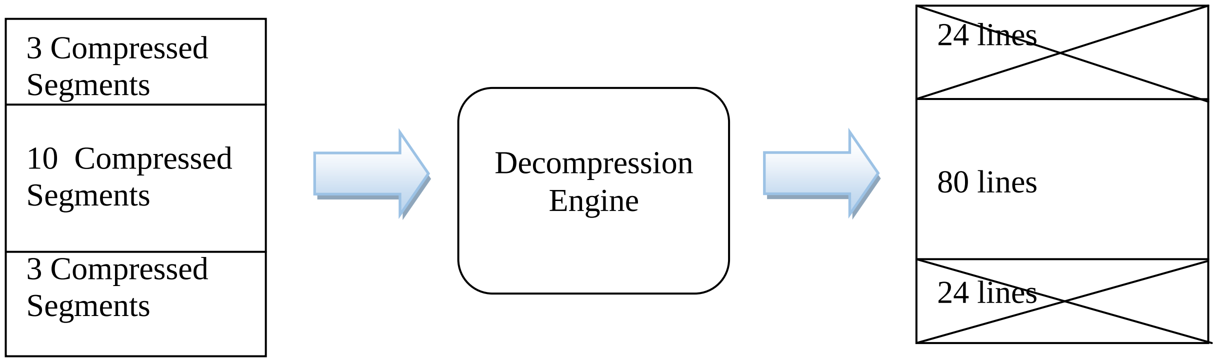


Figure 9 : Schematic of data parsing to decompression engine

4. Methodology

3.1. Process of generation of Decompressed Image

Figure 10 depicts the steps involved in the generation of decompressed image from data acquired from the satellite.

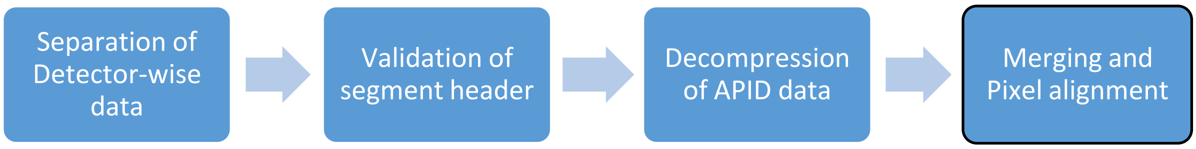


Figure 10. Process of generation of Decompressed Image

The following is the detailed description of steps:

3.2 Decompression and Merging of data:

The compressed payload line of PAN/MX data consists of 24 detectors. Each detector data should be separated and then decompressed independently. The following are the steps for conditioning the data before passing to decompression framework:

3.2.1 Separation of detector-wise APID(Application Process Identifier) data

3.2.2 Validation of segment header in each APID data:

Segment header of each APID packet contains the information required for decompression of each packet as explained in the previous sections.

3.2.3 Passing data to decompression library:

Open-source library provided by University of Nebraska is modified for reading multiple lines of compressed data in compatible to the compression scheme employed in Cartosat-3 data.

3.2.4 Merging of decompressed data from all APIDs to form payload lines

Each compressed line for Cartosat-3 after decompression gives 8 lines. These data from each detector should be merged to form full-swath payload lines.

3.2.5 Pixel alignment as per ground projection of the sensor

3.3 Design of Framework

This framework consists of multi-processing design using fork and wait system calls in C programming language. Detector-wise processes(referred to as child processes) are spawned to handle decompression of each APID data. Each detector-wise process in-turn spawns multiple processes (referred as grand-children processes) to decompress the data in chunks. The data is shared across multiple processes using POSIX Inter-process communication (IPC) namely Shared memory and Semaphores are used for signaling across processes for seamlessly performing the decompression, merging of payload data and writing the decompressed data to disk. Figure 11 shows the activity diagram of the framework.

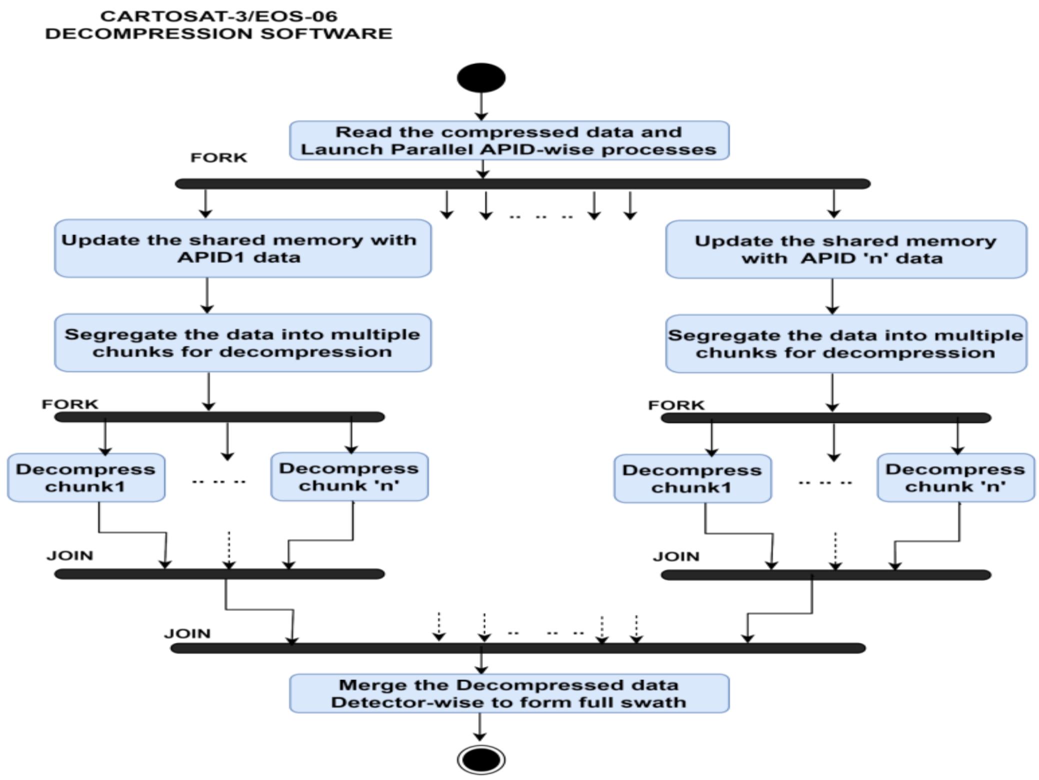


Figure 11. Process of generation of Decompressed Image

The software starts with reading a configurable number of compressed data lines from a file on SAN (Storage Area Network). The framework spawns parallel APID-wise processes(24 processes) where each process handles single APID data via a shared memory. Each APID data is further divided into multiple chunks (configurable value) and each chunk is decompressed independently by spawning processes. Upon successful decompression, all processes are joined and the APID data from all detectors are merged. Pixel alignment is performed on the merged data and then the decompressed data is written to a file on disk. This flow of processing is continuously performed in cycles on all the compressed lines available in the file.

The signaling of the action to be performed by each process in the framework is very critical to avoid any deadlock scenario. So, meticulous care is taken in signaling the processes to ensure smooth flow of interactions between different processes to avoid any overwriting of compressed or decompressed data available in the shared memory.

The following figure 12 explains the signaling mechanism used in this framework:

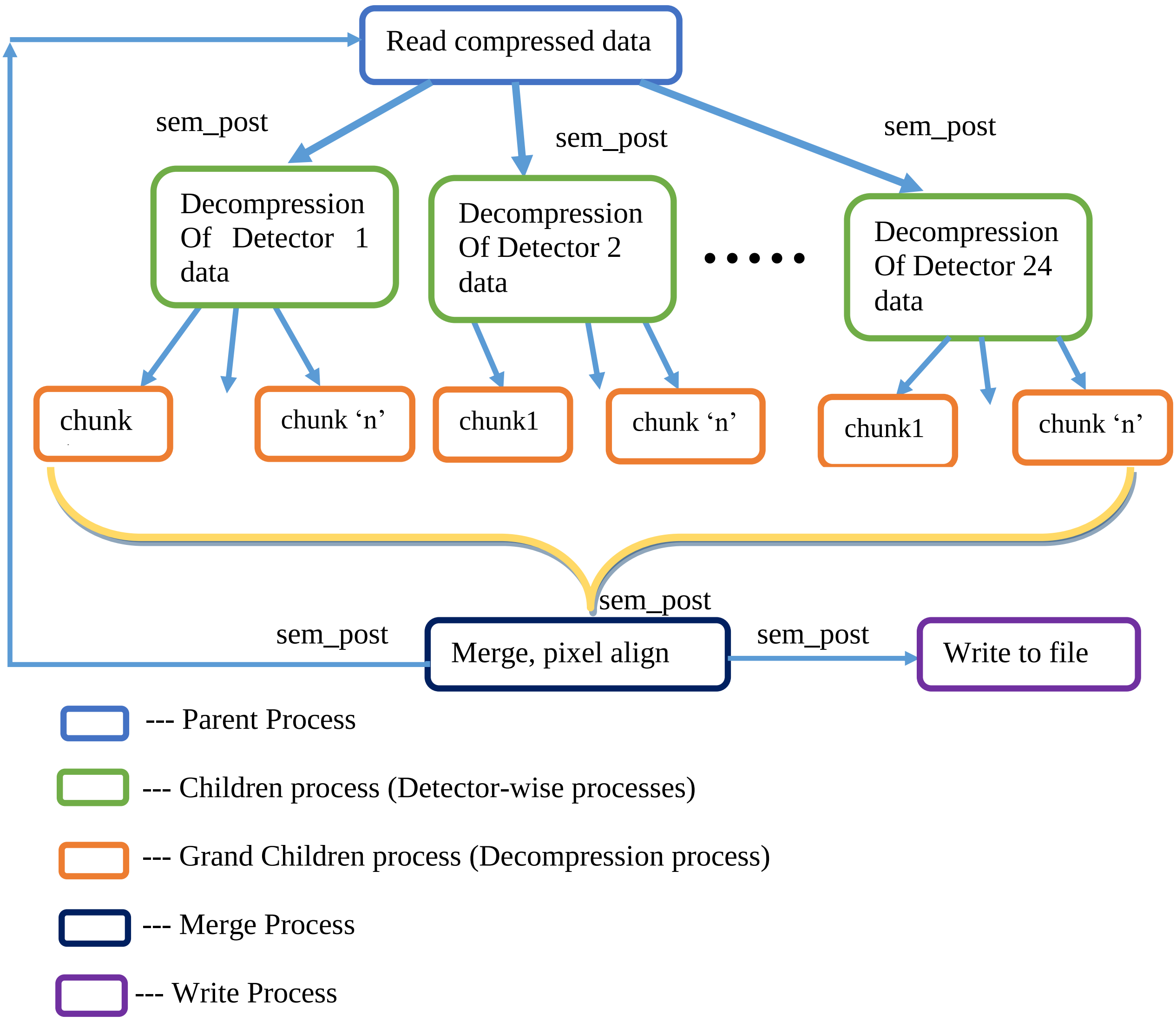


Figure 12. Process of generation of Decompressed Image

Initially when the framework starts, all the processes spawned, shared memory and semaphores are initialized. All processes are in waiting state to start the processing job which are assigned to them. “sem_post” provides a trigger which unblocks the process in waiting state and the processing of data starts. When the data is read, the semaphore post signal is issued to perform decompression by detector-wise processes, which in - turn signal the grand-children processes. After completion of decompression, these children processes send signal to merge process to merge all the detector-wise data to form single decompressed payload line. After completion of merging, signal is sent to read process to continue with the next read of data. This cycle of interaction continues till completion of decompression for all the read data. Upon successful completion, all the processes are exited and IPC objects are gracefully deleted.

5. Results

The decompression framework mentioned in figure 10 & 11 is successfully implemented in 'C' language as a part of Cartosat-3 ground segment establishment. Decompression process is improved in terms of the time taking to decompress the data from the inception of the ground segment. In the process of development of decompression process, it has been observed that the timelines will not be met unless a framework is established to perform independent tasks in parallel, due to the computationally intensive nature of the decompression algorithm. Hence the framework is developed. In one of the intermediate version of framework, sem_post is used to signal the read process to read the next compressed data, after write process but it is observed that the timelines can be improved by writing the decompressed data and reading the next chunks of the compressed data for reading in parallel after merging of the data is over as shown in figure 11.

The framework is tested and made operational in the following configuration of the system.

Processor	Intel Xeon(R) Gold 6154 @3.0Ghz
Number of Cores	4 CPU 18 core (144 logical cores)
RAM	512 GB
OS	Red Hat Enterprise Linux version 7.5

The following time lines are observed for the Cartosat-3 data with and without framework

SNO	Data Size	Decompression Time without Framework (minutes)	Decompression Time With Framework (Minutes)	Speed up
1	55 GB	65	35	~1.85
2	100 GB	122	63	~1.93
3	160 GB	155	85	~1.82

6. Conclusion & future scope

The propound framework is simple, scalable and adaptive for an on ground decompression software. It is provided the ~2X speed in terms of the timelines compared to the decompression implementation without the framework. It also segregates the decompression design to the framework so that the complexities involved in the design of decompression engine is abstracted with respect to the framework implementation.

The present framework further improved by implementing the parallel tasks in GP-GPU like applying inverse discrete wavelet transform etc. One can also attempt to do bit plane decoding of different segments parallel if the segments are all independently encoded in a given compression segment.

7. References

1. Image Data Compression. Issue 1. Recommendation for Space Data System Standards (Blue Book), CCSDS 122.0-B-1. Washington, D.C.: CCSDS, November 2005. https://public.ccsds.org/Pubs/122x0b1c3s.pdf
2. Image data compression. Issue 2 . Report concerning Space Data System Standards (Green Book) CCSDS 120.1-G-2. Washington, D.C.: CCSDS https://public.ccsds.org/Pubs/120x1g2s.pdf
3. Lossless Data Compression. Recommendation for Space Data Systems Standards. CCSDS 121.0-B-1. Blue Book. Issue 1: CCSDS, May 1997.

https://public.ccsds.org/Pubs/121x0b2ec1s.pdf

4. Shen-En Qian, Optical Satellite Data Compression and Implementation (ISBN 978-0-8194-9787-1), SPIE Press, Bellingham, Washington USA (2013)
5. Prof Khalid Sayood, Introduction to data compression (ISBN 13: 978-0-12-620862-7) University of Nebraska Third Edition (2006)
6. Mark Mitchel, Jeffrey oldham. Advanced Linux Programming (ISBN 0-7357-1043-0) First Edition, 2001.
7. NPTEL video Lecture on Information Theory, Coding and Cryptography by Prof: Ranjan Bose https://archive.nptel.ac.in/courses/108/102/108102117/
8. NPTEL video Lecture on Introduction to Time-Frequnecy Analysis and Wavelet Transforms by Prof: Arun K Tangaria https://archive.nptel.ac.in/courses/103/106/103106114/
9. Wim Swldens. The Lifting Scheme: A new philosophy in Biorthogonal Wavelet Constructions. https://cm-bell-labs.github.io/who/wim/papers/spie95.pdf
10. Robi Polikar . The Wavelet Tutorial, Second Edition https://web.iitd.ac.in/~sumeet/WaveletTutorial.pdf