Analytical Approach for identification of Orbits from SCATTEROMETER Level-0 Noise Images

Mudit J Dholakia
Dharmasinh Desai University
Nadiad
Gujarat-India

C K Bhensdadia, PhD
Dharmsinh Desai University
Nadiad
Gujarat-India

Anuja Sharma
SAC-ISRO
Ahmedabad
Gujarat-India

ABSTRACT

Considering the raw data extraction algorithms designed for different sensor data this paper focuses on the SCATTEROMETER Level-0 data extracted from raw data products. An analysis of extracted noise images is shown. Then a very simple technique but a very important concept is described and implemented on such noise images. Paper shows the approach which can be help full for identification of number of orbits in to the raw data products. The main objective of this approach is to provide the statistics to the application specific user. The approach is based on analyzed behavior of noise images extracted from the raw product. This approach defines a very basic standard to identify number of orbits based on SCATTEROMETER noise images. An implementation algorithm with its time complexity is shown with the corresponding implementation in C language. The proposed approach is for noise images of Data Quality Evaluation Level-0 SCATTEROMETER noise images, but it can also be extended for other noise images. The assumption for the paper is, the scanning geometry of noise image has been established in terms of scans and pixels i.e. header or interpretation format of the noise data.

General Terms

Scatterometer, Data Products

Keywords

Spatial Data Products, Orbit-Identification, Noise-Images, Image Data Analysis

1. INTRODUCTION

One of the important class of Microwave radars employed for remote sensing applications is the SCATTEROMETER which is mainly used for deriving the wind velocity over the ocean surface [6]. A SCATTEROMETER sensor is designed to precisely measure normalized backscatter cross-section σ^0 of the target surface[9]. Back scattered power is a function of wind speed, wave length of the radiation used, polarization, angle of incidence and azimuth angle[1]. Hence, when any given point on the ocean surface is viewed from multiple azimuth angles from a radar, the surface wind vector can be derived from such observations. OCEANSAT-II SCATTEROMETER with dual beam operation in Ku-band has two scanning pencil beams of VV (vertical) and HH (horizontal) polarization. Pencil beam configuration reduces mechanical complexity as well as provides four measurements, i.e. Inner fore, Inner aft, Outer fore and Outer aft measurements. It helps in ambiguity removal and to measure wind vector precisely[3]. Taking the ideal setup in consideration and scanning geometry of SCATTEROMETER, important thing to note is, it takes 281 measurements with HH beam (i.e. Inner beam) and 282 with VV beam (i.e. Outer beam). In both beams due to conical scanning geometry it takes 32 measurements of a single footprint window. Hence a single scan-line contains 8992 and 9024 samples in HH and VV respectively for the total viewing window size of 50km x 50km. To understand and explore more about SCATTEROMETER one has to go through ([1],[2],[3] & [4]). Also, before proceeding towards the background of the proposed approach it is mandatory to understand data packaging and other relative remote sensing terms from ([7] & [10]). [8] describes importance of Data Quality Evaluation in remote sensing and various techniques have been shown by some genius minds in ([5] & [8]).

2. BACKGROUND

Data Quality Evaluation refers to evaluation of various sensor parameters at different levels. A well established and scientifically driven system is capable of analyzing data behavior as well as generating alarms as and when required. Such mechanisms help application specific user to modify the satellite parameters on board (i.e. if required). Now to receive the data establishment of receiving mechanisms around the globe is a very daunting task. Hence a reception system when detected, satellite dumps the data to that reception system and moves on. Then a team packages them according to scientific standards and passes the packed data[13], i.e. data product to evaluation team. An evaluation team extracts the information from the data and evaluates it to the conceptual merits. A level wise procedure is shown in [1], i.e. ideal DQE system. In [4] a clear understanding of the information to be evaluated at Level-0 has been given. This paper focuses on Level-0 extracted data, i.e. Sensor Noise images. Level-0 consists the extraction of following information:

- Time Related Parameters
- Orbit Attitude Data Analysis & Flag-Modes
- Doppler & Encoder Related Parameters
- Antenna Related Parameters
- On-board Calibration Data

Detailed description of evaluation of Level-0 and detailed parameters of the above listed parameters have been given in [1]. One of the important tasks of DQE system at Level-0 is to identify the number of orbits covered in a single data product, i.e. raw data. This paper initiates a very simple but an important technique to calculate number of orbits using the Sensor Noise Data Images. Rest of the paper shows the proposed technique for orbit identification. Section-3 describes an analysis of Sensor Noise Data. In Section-4 an algorithm is proposed with the implementation in C language [12]. A program for implementation has also been given in Section-4. Section-5 concludes the paper followed by acknowledgement.

3. ANALYSIS OF SENSOR NOISE DATA

From the scanning geometry of SCATTEROMETER the interesting fact that puts a mark on researcher is the size of the image. Real world satellite dumps data of around 2 plus orbits to the receiver. Hence, if it is to be displayed using standard display soft-wares, the result may be a failure. So, to display such huge data there is a strong need of scientific scalable display software. One of the very good software to display such large stellar images has been designed by experts. Using such software the image is scaled down to the real world setup and windowing for full resolution can be done. Using such power full display software analysis for the proposed approach has been carried out.

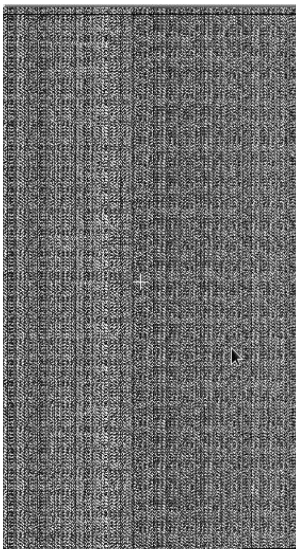


Fig 1: A raw sensor noise image data scaled and displayed using Stellar Display Software for HH beam.

An HH beam image has been displayed using stellar display software is shown in the (Fig. 1). For any human being it seems to be a lost signal into a television set. But once a clear understanding of these images has been made, one would come to know how these images will tell the world about the weather forecast as well as climate reasoning. At an extent this image is a scaled version of 281x32x8136 pixels, i.e. from 281x32x8136 to the 512x512 where 281 is number of footprints, 32 is number of samples each footprint have, 8136 is number of scan-lines. Corresponding VV beam image is shown in (Fig. 2) which is a 512x512 scaled image of 282x32x8136pixels of a raw VV image. These images are raw images extracted at Level-0 and no additional pre-processing has been carried out. Hence, they look like disturbance or noisy images. Moreover considerations for the scanning configuration can be determined from the Equation- 1 and Equation-2.

$$\text{Scansize} = 32 * \text{footprints in Polarization} \quad (1)$$

$$\text{Number of scans} = \frac{\text{Data Size}}{\text{Scan Size}} \quad (2)$$

Now if (Fig. 1) is analyzed with a window (shown in (Fig. 3)), the full resolution image of that window would be a vertical stitched image of four segments shown in Table-1. This image contains the actual grey values of Sensor Noise Data. The important thing to note in the representation of these segments is the red marked rectangle in all four segments. These are the 0 grey values which create a black chirp onto the image. This analysis has been carried out on several images of sensor data and the same black chirp pattern has been found on each of the images. These pattern indicates nothing but the change in orbit. Whenever satellite crosses north-pole it starts scanning the next orbit corresponding to scanning geometry.

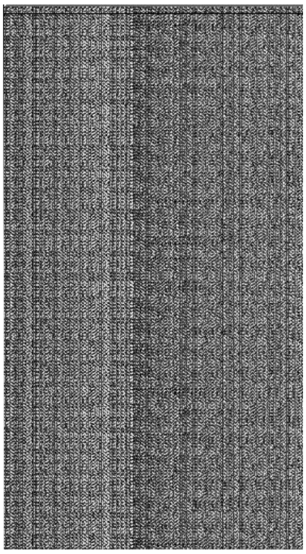


Fig 2: A raw sensor noise image data scaled and displayed using Stellar Display Software for VV beam.

But before that it misses at least one scan line or may be more than one according to the considerations. This will result as a black chirp in the sensor noise image. The data quality evaluators use this event and marks the north-pole crossing time for the retrospective analysis. But according to the modern computation approach this black chirp can be use full in identification of change. An algorithm can be proposed with predefined scanning considerations to identify the change in orbit and represent the statistic to the application specific user. Moreover it is not the only HH beam that misses a scan, the same is true for the VV beam also. In the Section-4 an algorithm is proposed to identify the change in orbit and count the number of orbits in dump data at Level-0. Corresponding implementation with result and time complexity is shown the same section.

4. PROPOSED ALGORITHM AND IMPLEMENTATION

4.1 Algorithm

To identify orbits and represent an exact orbit count one algorithm with the dynamic capability of determining the polarization can be proposed. Moreover based on dynamic scan consideration and geometry this algorithm can be used for other spatial data products. There are various approaches to accomplish the task of an ideal algorithm generation but to design the most basic algorithm a procedural approach is followed. To propose the approach at infant stage a procedure may lead to an efficient implementation. An algorithm to find the number of orbits from the dump can be proposed as follows:

Algorithm 1 Orbit Counting Algorithm

```
Procedure: Orbit–Counter(Sensor Noise Data File)
Determine polarization of file
Determine size of file
HHfootprints=281
VVfootprints=282
scansize='polarization'footprints*32
N = filesize/(scansize);
for each scan i:1 to N do
Pick $N^{th}$ scan in an array greyvalues[1::scansize].
        Check scan greyvalues
        if greyvalues[1..N]=0 then
        orbitcount=orbitcount+1;
        else continue;
        end if
        end for
        Check for true orbit count
        end procedure
```

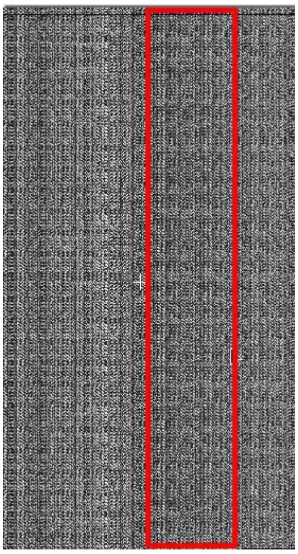
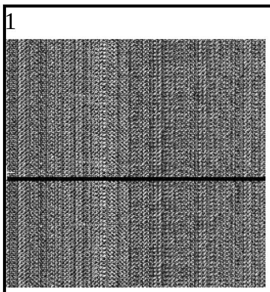
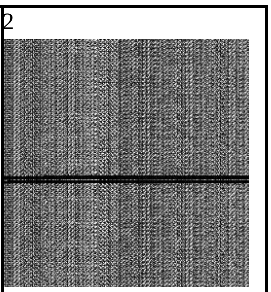
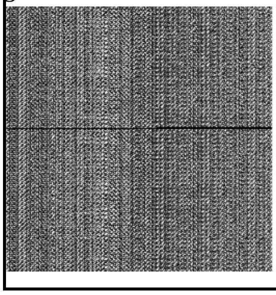
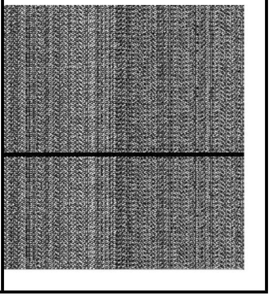


Fig 2: A raw sensor noise image data scaled and displayed using Stellar Display Software for HH beam and a window marked with red boundary on the image

It is straightforward that the approach is based on the data hence an algorithm is proposed which has more influence of basic input-output operations of the processing. Algorithm- 1 firsts reads the file, then it determines the polarization of the file and adjust the scanning considerations accordingly. Then one by one whole scan has been picked and checked whether it is empty or not. If it is found empty then that scan has to be considered as orbit boundary. Thus this algorithm determines the number of orbit boundaries based on which number of orbits can be found. Thus according to a standard notion in [14] & [15] , time complexity can be shown asymptotically as (m*n), where m stands for number of scans (i.e. N in the described case) and n stands for number of pixels in each scan. This algorithm can be implemented using any of the programming languages which support file and I/O operations. Moreover Section- 4.2 depicts the implementation of the proposed approach.

Table 1. Full resolution images for windowed portion in (Fig. 3)

1 	2 
3 	4 

4.2 Implementation

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
#include<sys/types.h>
#include<sys/stat.h>
#include<fcntl.h>
#include<string.h>
main(int argc,char *argv[]) {
int fp;
int i,j,k;
int empty_count;
int file_size;
int orbit_count;
int true_orbit_count;
char input_file[2048];
unsigned char datavalueHH[8992];
unsigned char datavalueVV[9024];
long int cutHH[100],cutVV[100];
strcpy(input_file,argv[1]);
fp=open(input_file,O_RDONLY);
file_size=lseek(fp,0,SEEK_END);
printf("Estimating file geometry.\n");
if(input_file[strlen(input_file)-5]=='H')
{
```

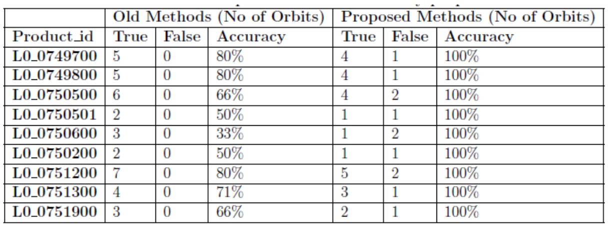
```
	printf("This is an HH file.\n");
	lseek(fp,0,SEEK_SET);
	for(i=0;i<(file_size/8992);i++)
	{
	read(fp,datavalueHH,sizeof(unsigned char)*8992);
	empty_count=0;
	for(j=0;j<8992;j++)
	{
	if((int)datavalueHH[j]==0)
	{
	empty_count=empty_count+1;
	}
	}
	if(empty_count>4000)
	{
	cutHH[orbit_count]=(long int)i;
	printf("An Orbit found at scan:%d\n",i);
	orbit_count=orbit_count+1;
	}
	}
for(k=0;k<orbit_count;k++)
{
	if(k==0)
	{
	true_orbit_count=true_orbit_count+1;
	}
	else
	{
		if(abs(cutHH[k]-cutHH[k-1])>3000)
		{
		true_orbit_count=true_orbit_count+1;
		}
	}
}
printf("This product contains data of %d
orbit.",true_orbit_count);
}
if(input_file[strlen(input_file)-5]=='V')
 {
 printf("This is a VV file.\n");
 lseek(fp,0,SEEK_SET);
 for(i=0;i<(file_size/9024);i++)
 {
	read(fp,datavalueVV,sizeof(unsigned char)*9024);
	empty_count=0;
	for(j=0;j<9024;j++)
	{
	if((int)datavalueVV[j]==0)
	{
	empty_count=empty_count+1;
	}
}
if(empty_count>4000)
{
	cutVV[orbit_count]=(long int)i;
	printf("Orbit found at scan:%d\n",i);
	orbit_count=orbit_count+1;
}
}
for(k=0;k<orbit_count;k++)
{
	if(k==0)
	{
	true_orbit_count=true_orbit_count+1;
	}
	else
	{
	if(abs(cutVV[k]-cutVV[k-1])>3000)
	{
	true_orbit_count=true_orbit_count+1;
	}
	}
}
printf("This product contains data of %d
orbit.",true_orbit_count);
}
}
```

Orbit Counting Program

Like the normal C program above code implements the idea of orbit counting and prints the number of orbits found as well as the scan number at which new orbit may exist. First it reads the sensor noise data file then based on its size it determines the polarization and scanning specifications. Then it collects the scan line data to identify whether there are 0 grey values or not. Based on this check it increments the empty count and checks it against the threshold, i.e.4500 in the proposed case. The threshold may vary according to scanning and sensor situations. It again resets the empty count to determine the state of next scan. Thus this approach counts the number of orbits. Now there may be a case when for a single orbit change it may record 0 values for more than 2 scans near to each other. Hence to check the true orbit change scan line it also identifies the relative order and position of the identified scans. From the sequence of the program it is clear that the asymptotic time complexity is O(i*j), where i is number of scans, i.e. N in Algorithm- 1 and j is number of pixels that may be 8992 or 9024 considering HH and VV beams in sensor data respectively. Hence the black chirps in the sensor images shown in Table- 1 have been very useful in terms of retrospective analysis. A sample run and code have been put at [11] for the reference. Data products can be downloaded from[16].

Experiments have been carried out on some randomly selected products from NRSC databse. Table below shows the same.

Table 1: Experimental Results



	Old Methods (No of Orbits)			Proposed Methods (No of Orbits)		
Product_id	True	False	Accuracy	True	False	Accuracy
L0_0749700	5	0	80%	4	1	100%
L0_0749800	5	0	80%	4	1	100%
L0_0750500	6	0	66%	4	2	100%
L0_0750501	2	0	50%	1	1	100%
L0_0750600	3	0	33%	1	2	100%
L0_0750200	2	0	50%	1	1	100%
L0_0751200	7	0	80%	5	2	100%
L0_0751300	4	0	71%	3	1	100%
L0_0751900	3	0	66%	2	1	100%

Analytically proposal shows 100% accuracy where as old methods shows the less accuracy due to overheads.

5. CONCLUSION

A very simple approach of identifying orbits from Level-0 data has been proposed. Considering the analysis of sensor noise data files the windowing on specific images have been carried out. Some of them presents a dark chirp as an indication of orbital change where a threshold of such grey values have been analyzed. Using C language the approach is implemented and time complexity has been computed using the standard asymptotic notations. Thus the proposed approach shows the technique to identify number of orbits which can be helpful to an application specific user as well as DQE team. In traditional systems this redundancy leads to waste of computational efforts. Instead, this approach provides the knowledge, using which the new data can be segmented as well as analyzed (using further integration) and improvements to the upcoming observation systems can be carried-out.

6. FUTURE SCOPE

Using such approach of analysis an automated system with dynamic capability of scanning geometry can be developed. This also initiates the orbit identification in SAR images which can be used for any of such sensors. It can be extended to optical images also.

7. REFERENCES

[1] SCATTEROMETER Payload Design Team-ISRO.OCEANSAT-II SCATTEROMETER PAYLOAD PRELIMINARY DESIGN REVIEW DOCUMENT. SAC-ISRO,2005

[2] Spencer, Michael W., Chialin Wu, and David G. Long. Improved resolution backscatter measurements with the Sea-Winds pencil-beam scatterometer. Geoscience and Remote Sensing, IEEE Transactions on 38.1 (2000): 89-104.

[3] Parmar, R. M., et al. OCEANSAT 2: mission and its applications. Asia-Pacific Remote Sensing Symposium. International Society for Optics and Photonics, 2006.

[4] Misra, Tapan, et al. Oceansat-II Scatterometer: Sensor Performance Evaluation, Analyses, and Estimation of Biases. Geoscience and Remote Sensing, IEEE Transactions on 52.6 (2014): 3310-3315.

[5] Singh, Randhir, Prashant Kumar, and Pradip Kumar Pal. Assimilation of Oceansat-2-scatterometer-derived surface winds in the weather research and forecasting model. Geoscience and Remote Sensing, IEEE Transactions on 50.4 (2012): 1015-1021.

[6] Calla, O. P. N. Study of Ocean using Indian Space Borne Microwave Sensors. IJCST 2.4 (2011).

[7] Joseph, George. Fundamentals of remote sensing. Universities Press, 2005.

[8] Palsule, S. S., et al. Resourcesat-1 data quality evaluation system. International Journal of Applied Earth Observation and Geoinformation 10.2 (2008): 147-158.

[9] Tummala, Kimeera. Synthetic Aperture Radar (SAR) Data Simulation for Radar Backscatter Cross-section Retrieval. Diss. ISRO, 2014.

[10] Tobler,W. R. Cellular geography. Philosophy in geography. Springer Netherlands, 1979. 379-386.

[11] Dholakia, Mudit. Muditdholakia/muditdholakia.github.io. GitHub. 1 Oct. 2015. Web. 30 Oct. 2015

[12] Balagurusamy, E. programming in ANSI C. Tata McGraw- Hill Education, 2004.

[13] Folk, Mike, Robert E. McGrath, and Nancy Yeager. HDF: an update and future directions. Geoscience and Remote Sensing Symposium, 1999. IGARSS99 Proceedings. IEEE 1999 International. Vol. 1. IEEE, 1999.

[14] Cormen, Thomas H. Introduction to algorithms. MIT press, 2009.

[15] Brassard, Gilles, and Paul Bratley. Fundamentals of algorithmics. Prentice-Hall, Inc., 1996.

[16] ISRO, NRSC. National Remote Sensing Centre. NRSC. ISRO. Web. 25 Nov. 2015