

# De-encapsulation and Decompression Algorithms for Level 0 Files of the ENVISAT Satellite in Image Mode

I. Franchi, and A. Zozaya

**Abstract**—This document describes in detail the structure of the level 0 data of the ENVISAT satellite and, in addition, presents the de-encapsulation and decompression algorithms of the raw ASAR data, specifically the IM operation mode files. These algorithms were developed in order to facilitate and optimize the task of retrieving the source data. As a result, a raw-data-based image of a region in the Falcón state of Venezuela was obtained after applying the aforementioned algorithms to the level 0 files downloaded by the ESA Simple Online Catalogue's.

**Index Terms**—Raw data, De-encapsulation, Decompression, ASAR, ESA, ENVISAT, FBAQ.

## I. INTRODUCCIÓN

LA mayor misión de observación de la Tierra se llevó a cabo en el año 2002 con la puesta en órbita del satélite ENVISAT de la Agencia Espacial Europea –*Europe Space Agency* (ESA)–. Con un peso de 8 toneladas y 10 instrumentos de medición, el satélite ENVISAT permitió durante 10 años la monitorización de los cambios ambientales y climáticos de prácticamente todo el planeta. Entre su carga útil se encontraba un radar de apertura sintética avanzado –*Advanced Synthetic Aperture Radar* (ASAR)–, capaz de obtener imágenes de alta resolución bajo cualquier condición climática. Como es bien sabido un radar de apertura sintética permite estimar diferentes parámetros o características del terreno, como por ejemplo la rugosidad y la constante dieléctrica, a partir del eco electromagnético retrodispersado por el terreno mismo [1].

Los radares de apertura sintética han formado y forman parte de una amplia variedad de misiones de investigación, constituyendo uno de los sensores de mayor importancia en la observación remota de la tierra o teledetección [2], [3], [4], [5]. En la teledetección basada en imágenes SAR el insumo inicial son los datos crudos, es decir, los datos sin procesar que el radar captura directamente de la escena observada y que dependiendo del fabricante son encapsulados en un gran archivo de acuerdo a una estructura determinada. Las diferentes áreas de aplicación de la observación remota de la tierra parten de la capacidad de acceder y procesar estos datos crudos para focalizar las imágenes de las áreas sondeadas por el sensor [6], [7], [8].

El radar de apertura sintética avanzado del ENVISAT era capaz de operar en cinco modos diferentes: *Image Mode* (IM), *Alternating Polarisation Mode* (AP), *Wide Swath Mode* (WS),

*Global Monitoring Mode* (GM) y *Wave Mode* (WV) [1]. De todos ellos, es de interés en el presente trabajo el *Image Mode*.

Actualmente, solo es posible la desencapsulación y descompresión mediante herramientas o *software* comerciales, lo que dificulta en gran medida la obtención de los datos crudos de las escenas mapeadas por los satélites. Estos datos son indispensables en investigaciones relacionadas con la invención, comparación o adaptación de algoritmos de enfoque.

Debido a esta problemática, en este artículo se propone la creación de unos algoritmos propios de desencapsulación y descompresión de datos crudos del instrumento de medición ASAR, específicamente en el modo de operación IM, para facilitar su adquisición y contribuir en futuras investigaciones. Además, este trabajo constituye el paso inicial de un sistema de procesamiento de imágenes *Open Source* del sensor ASAR para el satélite ENVISAT, en sus cinco modos de operación, promoviendo la creación y publicación de este tipo de algoritmos.

El resto de este artículo se organiza como sigue: el proceso de obtención de datos crudos se relata en la Sección II, la estructura de los datos de nivel 0 se describe en la Sección III; el algoritmo de desencapsulación es discutido en la Sección IV; en la Sección V se describe la estructura del archivo auxiliar; la compresión y descompresión de los datos de nivel 0 se presentan en la Sección VI; en la Sección VII se describe el algoritmo de descompresión; la implementación de los algoritmos se presenta en la Sección VIII; y finalmente se concluye en la Sección IX.

## II. PROCEDIMIENTO PARA LA OBTENCIÓN DE DATOS CRUDOS

En general, para conseguir la pseudo-imagen de una escena específica y analizarla para diversos propósitos, es necesario obtener los datos crudos del satélite en cuestión y enfocarlos mediante algún algoritmo de focalización, como por ejemplo el algoritmo *Range-DOPPLER*, el algoritmo  $\omega - \kappa$ , o el algoritmo *Chirp Scaling* [9].

En el satélite ENVISAT los datos crudos se originan en el sensor ASAR tal como se muestra en la figura 1. Inicialmente el sensor emite pulsos de ondas electromagnéticas hacia la superficie terrestre que, al alcanzar el objetivo sobre el terreno, son dispersados en diferentes direcciones de las que el radar es capaz de capturar únicamente aquellos ecos retrodispersados en la misma dirección de iluminación del radar [1].

Una vez recibidos en el sensor, los ecos de la escena iluminada sobre el terreno son organizados en una matriz en

I. Franchi, Escuela de Ingeniería de Telecomunicaciones, Universidad de Carabobo (UC), Valencia, Venezuela, ifranchire@gmail.com.

A. Zozaya, Departamento de Electricidad, Universidad Tecnológica Metropolitana (UTEM), Santiago, Chile, a.zozayas@utem.cl.

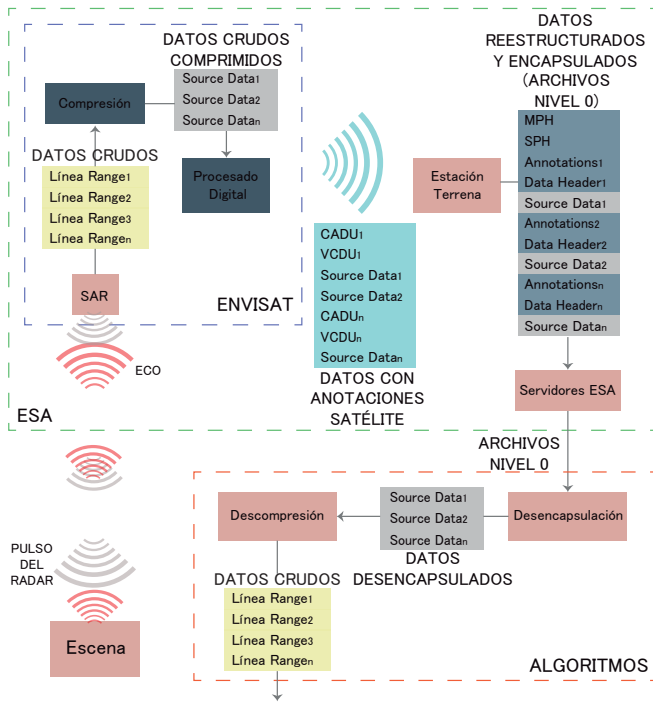


Fig. 1. Diagrama del procedimiento para la obtención de datos crudos.

la que cada fila corresponde a una posición acimutal distinta del satélite. Cada posición acimutal del radar se registra a su vez en correspondencia de la transmisión de cada pulso desde el radar. Al finalizar su recorrido o sondeo, el satélite realiza la compresión de los datos crudos y los procesa digitalmente, encapsulándolos en contenedores VCDU y CADU (codificaciones de canal, etiquetas de sincronización de paquetes, entre otros) para su posterior envío a una de las estaciones terrenas de la ESA [10] (ver figura 1).

Al ser recibidos en la estación terrestre, los datos son reestructurados y nuevamente encapsulados, dando lugar a un tipo de archivo denominado de nivel 0 (ver Sección III). Los archivos de nivel 0 son almacenados en los servidores de la ESA quedando disponibles para diferentes usuarios y aplicaciones. De estos datos, mediante procesamiento digital, se obtienen las imágenes de radar de nivel 1 y 2 [1].

Usualmente para obtener las imágenes de nivel 1 o 2 de la escena deseada, sin embargo, es necesario esperar a que los datos crudos sean procesados y las imágenes liberadas por la ESA misma, o recurrir a algún software comercial, en general muy costoso, para extraer los datos crudos de los archivos de nivel 0 y realizar la focalización de los mismos.

Por esta razón hemos desarrollado unos algoritmos de desencapsulación y descompresión propios para ser aplicados de manera sucesiva a los archivos de nivel 0, cuya descripción y aplicación constituyen el objeto principal de este artículo. Estos algoritmos constituyen el primer paso en el contexto de una iniciativa de investigación en la que posteriormente desarrollaremos códigos igualmente abiertos para la focalización de la imagen usando el algoritmo  $\omega - \kappa$ , y cuyos resultados también serán divulgados en el futuro cercano.

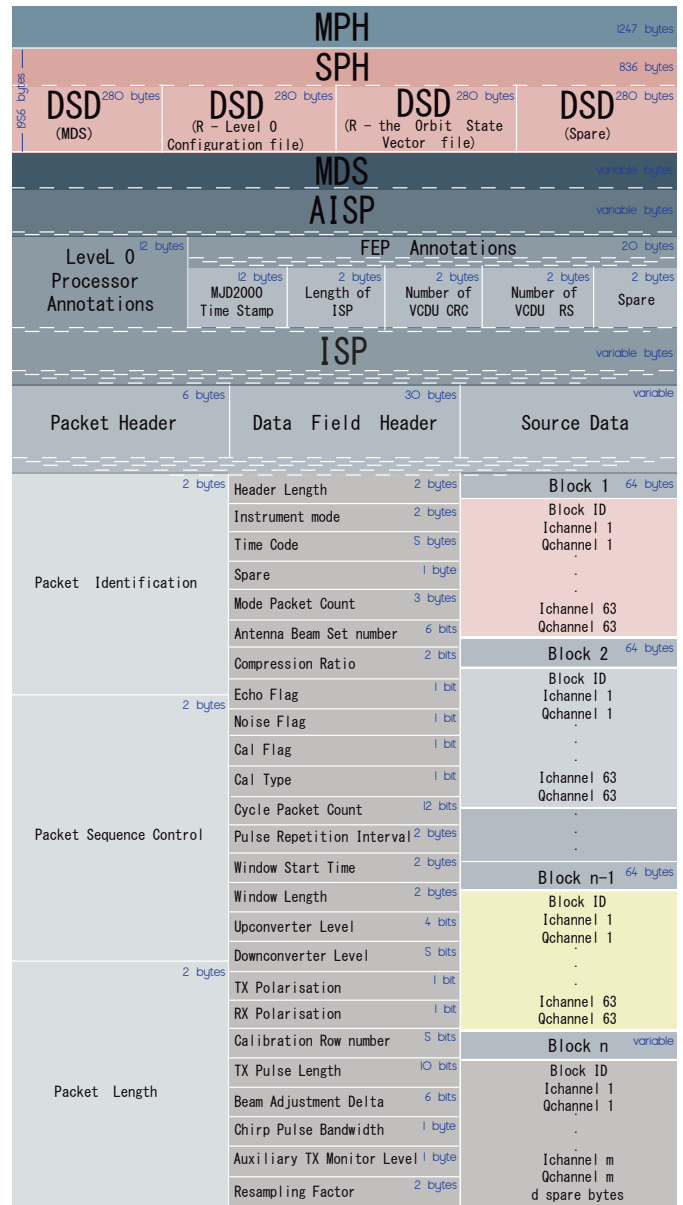


Fig. 2. Estructura de los datos nivel 0 del modo imagen.

### III. ESTRUCTURA DE DATOS NIVEL 0 DEL MODO IMAGEN

Para extraer los datos crudos es primordial conocer de antemano la estructura de los datos de nivel 0. Estos datos, en el modo de operación *Image Mode* del instrumento de medición ASAR (identificados como ASA\_IM\_OP) [1], se encuentran estructurados por cabeceras y contenedores tal como se muestra en la figura 2. Enseguida, se describen cada una de ellas:

#### A. Encabezado Principal del Producto –Main Product Header (MPH)–

El MPH contiene información necesaria para todos los sensores ENVISAT, tal como: identificación del producto, información sobre la adquisición y procesamiento de los datos, tiempo de detección, información de la órbita, conversión de tiempo binario satelital –*Satellite Binary Time* (SBT)– a

tiempo universal coordinado –*Coordinated Universal Time* (UTC)–, datos de confianza del producto –*Product Confidence Data* (PCD)– y la información sobre el tamaño del producto. El MPH tiene una longitud fija de 1247 bytes y un formato idéntico para todos los instrumentos del ENVISAT [1], [10]. En la figura 3 se observa un fragmento del contenido del MPH de un producto de nivel 0 de modo imagen.

```
ACQUISITION_STATION="PDHS-E"
PROC_CENTER="PDHS-E"
PROC_TIME="19-AUG-2007 03:18:09.000000"
SOFTWARE_VER="KSPT_L0/4302"

SENSING_START="19-AUG-2007 02:26:04.077150"
SENSING_STOP="19-AUG-2007 02:26:28.018921"

PHASE=2
CYCLE=+060
REL_ORBIT=+00476
ABS_ORBIT=+28586
STATE_VECTOR_TIME="19-AUG-2007 02:24:02.986654"
DELTA_UT1=-.163767<s>
X_POSITION=+2917060.051<m>
Y_POSITION=-6544906.838<m>
Z_POSITION=+0002181.430<m>
X_VELOCITY=-1493.430212<m/s>
Y_VELOCITY=-0653.700544<m/s>
```

Fig. 3. Fragmento del MPH de un archivo de nivel 0 de modo imagen.

### B. Encabezado Específico del Producto –*Specific Product Header* (SPH)–

El SPH proporciona información adicional y específica aplicable a todo el producto, relacionada con el procesamiento y los PCD. Una porción de estos datos se puede observar en la imagen 4. Posee una longitud fija de 1956 bytes para todos los archivos de nivel 0 del ASAR y al igual que el MPH, su formato es idéntico para todos los instrumentos ENVISAT [10]. Además, los SPH de nivel 0 contienen 4 descriptores de conjunto de datos –*Data Set Descriptor* (DSD)– de 280 bytes cada uno, los cuales son:

```
SPH_DESCRIPTOR="ASA_IM_0P SPECIFIC HEADER"
START_LAT=+0007203449<10-6degN>
START_LONG=-0067561723<10-6degE>
STOP_LAT=+0008624016<10-6degN>
STOP_LONG=-0067877940<10-6degE>
SAT_TRACK=+3.47503648E+02<deg>

ISP_ERRORS_SIGNIFICANT=0
MISSING_ISPS_SIGNIFICANT=0
ISP_DISCARDED_SIGNIFICANT=0
RS_SIGNIFICANT=0

NUM_ERROR_ISPS=+000000000
ERROR_ISPS_THRESH=+0.0000000E+00<%>
NUM_MISSING_ISPS=+000000000
MISSING_ISPS_THRESH=+0.0000000E+00<%>
NUM_DISCARDED_ISPS=+000000000
DISCARDED_ISPS_THRESH=+0.0000000E+00<%>
NUM_RS_ISPS=+000000000
```

Fig. 4. Fragmento del SPH de un archivo de nivel 0 de modo imagen.

- a) **DSD (M) para el conjunto de datos de medición –*Measurement Data Set* (MDS)–:** describe los paquetes

```
DS_NAME="ASAR_SOURCE_PACKETS"
DS_TYPE=M
FILENAME=""
DS_OFFSET=+0000000000000003203<bytes>
DS_SIZE=+0000000000285867894<bytes>
NUM_DSR=+0000050580
DSR_SIZE=-000000001<bytes>
```

Fig. 5. DSD del MDS de un archivo de nivel 0 de modo imagen.

fuentes del MDS, en él se indica el tamaño en bytes del MDS y la cantidad de paquetes fuente del instrumento –*Instrument Source Packets* (ISP)– que hay en el mismo, tal y como se muestra en la figura 5 [11].

- b) **DSD (R) para archivos de configuración de nivel 0 (level 0 configuration file):** es un descriptor de referencia, contiene el nombre del archivo de configuración del procesador de nivel 0 utilizado para crear el producto [11].
- c) **DSD (R) para el archivo del vector de estado de órbita (orbit state vector file):** al igual que el DSD anterior, es un descriptor de referencia y contiene el nombre del archivo de vector de estado de órbita utilizado para crear el producto [11].
- d) **DSD vacío (spare):** como su nombre lo indica, es un descriptor vacío, contiene 279 espacios en blanco seguidos por un carácter de salto de línea. Está reservado para permitir adiciones futuras al producto si es necesario [11].

### C. Conjunto de Datos de Medición –*Measurement Data Set* (MDS)–:

Aquí se encuentran los datos crudos capturados por el ASAR, comprimidos y colocados en múltiples contenedores que reciben el nombre de paquetes fuente del instrumento, que a su vez, poseen unas cabeceras de anotaciones [11]. Todo este conjunto de datos son denominados paquetes fuentes de instrumentos con anotaciones –*Annotated Instrument Source Packets* (AISP)–. El MDS posee un tamaño variable y depende de la cantidad de ISP que posea el archivo de nivel 0, sus datos son binarios mixtos, ya que dentro del mismo se encuentran varios tipos de datos (enteros con y sin signo, MJD 2000<sup>1</sup>, banderas, magnitud con signo, entre otros). La estructura tanto de los AISP como los ISP es la siguiente.

- 1) **Annotated Instrument Source Packets:** poseen los siguientes elementos:
- a) **Anotaciones del procesador de nivel 0 (level 0 processor annotations):** posee 12 bytes de información en formato MJD 2000, referente al tiempo en el que fue procesado el paquete [11].
- b) **Anotaciones del procesador frontal (FEP annotations):** con una longitud de 20 bytes, contiene 4 elementos: MJD 2000 *time stamp* (12 bytes), *length of ISP* (2 bytes), *num VCDU CRC* (2 bytes) y *num VCDU RS* (2 bytes). Siendo el elemento más

<sup>1</sup>El MJD 2000 es el número decimal del día desde el 1 de enero del 2000 a las 00:00 horas [12].

importante el *length of ISP*, ya que éste posee la longitud en bytes menos uno del ISP, sin incluir los bytes de la cabecera del paquete. Adicional a estos elementos, el *FEP annotations* contiene 2 bytes de *spaces*, que son espacios en blanco de relleno [11].

- c) **Paquetes fuente del instrumento –Instrument Source Packets (ISP)–**: dentro de él se encuentran los datos crudos comprimidos. Seguidamente se detalla su estructura [11].

2) *Instrument Source Packets*: dentro se encuentran dos cabeceras y un contenedor, estos son:

- a) **Encabezado de paquete (packet header)**: contiene información general sobre los datos contenidos en el ISP y tiene una longitud de 6 bytes. Esta información se detalla en tres elementos de 2 bytes cada uno con los siguientes nombres: *packet identification*, *packet sequence control* y *packet length*. A pesar de indicar al igual que el *FEP annotations* la longitud en bytes del ISP, se recomienda utilizar el elemento del *FEP annotations* ya que el *packet header* puede contener errores causados durante el *downlink* [13].
- b) **Encabezado del campo de datos (data field header)**: posee 30 bytes que brindan detalles de la configuración del radar durante la operación, desglosados en 24 elementos entre los cuales se destacan: *compression ratio* (2 bits), *echo flag* (1 bit), *pulse repetition interval* (2 bytes), *window length* (2 bytes) y *chirp pulse bandwidth* (1 byte), por poseer información de ayuda tanto para la desencapsulación, como para la futura focalización de los datos crudos [13].
- c) **Datos fuentes (source data)**: como su nombre lo indica, contiene los datos de medición que genera el instrumento y además se encuentran comprimidos. Existen 4 tipos de datos fuentes: de calibración inicial, de ruido, de eco y de calibración. Su longitud es variable por cada ISP y su estructura difiere dependiendo del tipo de datos fuente. En la figura 2, se puede observar en detalle la estructura de los datos fuentes de eco [13].

#### IV. ALGORITMO DE DEENCAPSULACIÓN

Se ha diseñado un algoritmo (ver figura 6) para realizar la desencapsulación de datos de nivel 0. Enseguida se detallan sus instrucciones:

Al inicio se solicita al usuario la introducción del archivo de nivel 0 que desea focalizar, proporcionado por la ESA y previamente descargado con el Simple Online Catalogue. Seguidamente se guarda este archivo en la variable “arch” para luego abrirlo y adentrarse entre cabeceras y contenedores, hasta obtener los datos crudos.

La primera cabecera que se encuentra es la del MPH, para extraerla es necesario realizar la lectura de los bytes que posee.

De esta manera se desplaza hasta la segunda cabecera, la del SPH y leyendo sus bytes correspondientes se visualizan los campos DSD. De los cuatros campos que tiene, es el DSDm

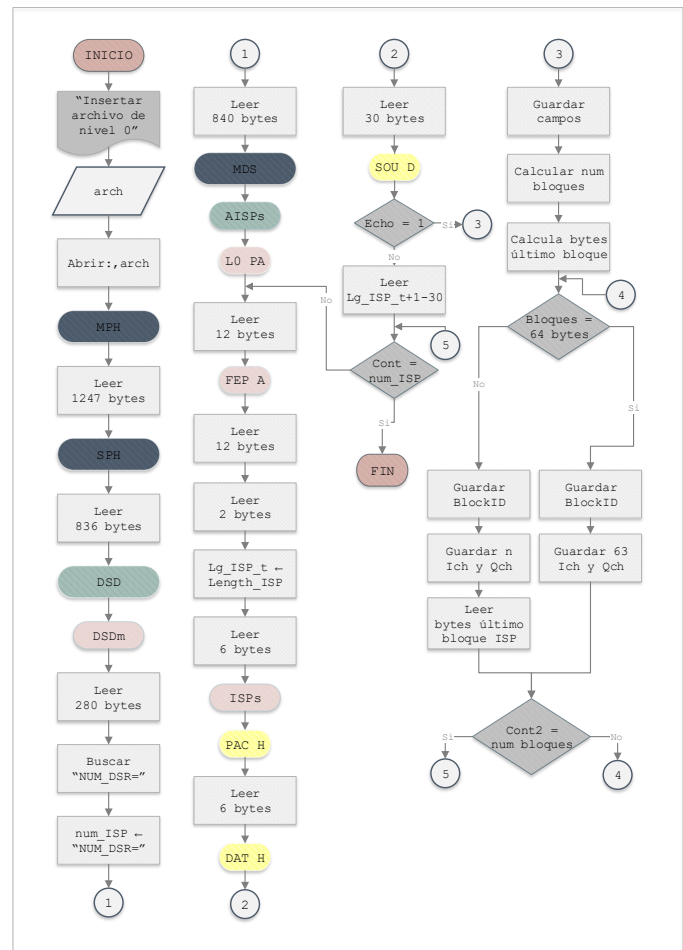


Fig. 6. Diagrama de flujo para la desencapsulación de los datos nivel 0.

quien brinda la información de los números de ISP existentes en el archivo, por ello, se leen sus bytes, se busca la palabra clave “NUM\_DSR=”, se almacena el valor y se leen los DSD restantes.

```

1 # Lectura de los num_ISP ISPs
2 for i in range(num_ISP):
3
4     #L0 Processor Annotations
5     archivo.read(12)
6
7     #FEP Annotation
8     archivo.read(12) #MJD 2000 Time Stamp
9     Length_ISP_temp = (int.from_bytes(archivo.read(2),
10                                bytearray='big')) #Length of ISP
11     archivo.read(6) #Resto del FEP annotation
12
13     #Packet Header
14     PacIdent_temp = (int.from_bytes(archivo.read(2),
15                                bytearray='big')) #Packet ID
16     PacSequeC_temp = (int.from_bytes(archivo.read(2),
17                                bytearray='big')) #Packet Seq. Control
18     PacLeng_temp = (int.from_bytes(archivo.read(2),
19                                bytearray='big')) #Packet Length

```

Fig. 7. Fragmento de código de desencapsulación.

El siguiente nivel es del contenedor MDS, en él se encuentran otros contenedores, los AISP. Dentro de estos últimos se encuentran las cabeceras de anotaciones *level 0 processor annotations* y *FEP annotations* y los contenedores ISP. Se



leen los bytes correspondientes a cada una de las cabeceras y además, se extrae y almacena la longitud menos uno del ISP (extraída del *FEP annotations*).

Dentro de los ISP se encuentran dos cabeceras: la *packet header* y la *data field header*, se extraen cada una leyendo sus bytes correspondientes.

El último nivel corresponde al contenedor *source data*, en donde es necesario comprobar previamente que estos datos fuente son de tipo eco, ya que contienen los datos crudos comprimidos.

Si no se trata de este tipo de datos fuentes, se lee todo lo que resta del ISP y, dependiendo de si ya se ha leído el número de ISP completo, el programa llegará a su final o continúa leyendo el siguiente AISP.

Si los *source data* son de tipo eco, se guardan los campos desde el *packet header* hasta el *data field header* y se calculan los bytes de relleno del último bloque.

Ya que el *source data* está estructurado en múltiples bloques de 64 bytes y un último bloque que puede tener 64 bytes o menos, se recorren uno a uno los bloques y se verifica si el mismo es de 64 bytes.

Si es de 64 bytes, entonces posee 1 *block ID* y 63 muestras tanto del canal I como del Q. De lo contrario, se trata del último bloque del ISP y este tendrá un *block ID*, “n” muestras del canal I y Q y además “d” bytes de relleno (calculados previamente).

Finalmente se verifica si se leyeron todos los bloques del *source data*. De no ser así, se repiten los 2 pasos previos y de lo contrario, se lee el siguiente AISP (de haberlo).

#### V. ESTRUCTURA DEL ARCHIVO AUXILIAR DE CARACTERIZACIÓN DE INSTRUMENTO (*Instrument characterization file*)

Los archivos auxiliares tal y como su nombre lo indica, son archivos extras que ayudan en la construcción de un producto distinto a las mediciones directas del instrumento. Estos pueden incluir datos de calibración medidos a bordo, archivos de calibración externos de fuentes distintas al satélite, archivos de configuración del procesador y cualquier otro archivo que necesiten los procesadores de instrumentos [14].

Al igual que los archivos de nivel 0, los archivos auxiliares constan de un MPH y un SPH con uno o más DSD. Estas cabeceras siguen la misma estructura y convenciones de los datos de nivel 0 del modo imagen. Sin embargo, a diferencia de los archivos de nivel 0, los MDS no están presentes, en su lugar se encuentran los conjuntos de datos auxiliares globales –*Global Auxiliary Data Set* (GADS)–. Estos son datos en binario mixto que contienen la información relevante de los archivos auxiliares [14].

El archivo auxiliar de caracterización de instrumento (identificado como ASA\_INS\_AX) contiene los parámetros clave necesarios para el procesamiento en la estación terrena. Esto incluye tablas de búsqueda –*Look Up Tables* (LUT)–, valores nominales de amplitud y fase para la calibración de los distintos modos del ASAR, múltiples tablas de configuración de valores predeterminados, entre otros [15].

El *Instrument Characterization File* al no crearse a bordo del satélite, sus cabeceras no poseen mucha información

relevante. El MPH, consta de 1247 bytes, sin embargo la única información de interés es el rango de tiempo para el que es válido el archivo auxiliar. El SPH por su parte, tiene un tamaño de 378 bytes y contiene básicamente solo el DSD del conjunto de datos auxiliares globales del archivo [15].

Los GADS en cambio poseen 171648 bytes y están en binario mixto, ya que contienen diversos tipos de datos. Estos si poseen información relevante que sirve para procesar los datos de medición y crear un producto distinto, tal y como se mencionó anteriormente [15].

#### VI. COMPRESIÓN Y DESCOMPRESIÓN DE LOS DATOS DE NIVEL 0

El ENVISAT ASAR, produce 3 tipos de datos diferentes. Estos pueden ser datos de calibración, de ruido o de eco. Dependiendo del tipo de dato obtenido, el satélite puede generar 3 tipos de compresiones de datos distintas, siendo estas *ful-8* (*full-8*), signo + magnitud –*sign + magnitude* (S+M)– y cuantización adaptativa de bloques flexibles –*Flexible Block Adaptive Quantization* (FBAQ)– respectivamente. Los paquetes fuentes de eco del modo imagen, están comprimidos con la técnica de FBAQ, más específicamente FBAQ-4 [16].

El FBAQ-4 comprime las muestras de 8 bits a muestras de 4 bits. Este es un tipo de algoritmo de compresión con pérdida, por lo que el mismo, introduce distorsión en los datos en forma de ruido de cuantificación. Para descomprimir los datos de eco, es necesario utilizar las LUT que contienen los valores normalizados de punto flotante de 4 bytes con corrección de ADC incluida [16].

Las LUT de reconstrucción del algoritmo FBAQ-4 poseen 4096 valores de punto flotantes de 4 bytes, distribuidos en 16 niveles de reconstrucción, cada uno con 256 valores de reconstrucción normalizados. Cada valor de reconstrucción correspondiente a un nivel de reconstrucción específico, está ligado a 2 variables. Una es el identificador de bloque (*block ID*) y la otra es el valor del canal (ya sea canal I o Q). Las LUT están indexadas en un orden específico que será explicado en detalle en la sección siguiente [15], [16].

Para descomprimir los datos crudos de un archivo de nivel 0, hace falta un archivo auxiliar, el *Instrument Characterization File*, ya que éste contiene las LUT de reconstrucción del algoritmo FBAQ-4 con corrección de ADC, tanto del canal I como del canal Q del radar [16]. Los pasos a seguir para realizar con éxito la descompresión de los datos crudos se explican subsecuentemente.

#### VII. ALGORITMO DE DESCOMPRESIÓN

Se ha diseñado un algoritmo (ver figura 8) para realizar la descompresión de datos de nivel 0. Enseguida se detallan sus instrucciones:

Al inicio se realiza la selección del archivo auxiliar de caracterización de instrumento que proporcione una cobertura completa del producto de nivel 0 en cuestión. Para ello se utiliza el formato presente en el nombre del archivo: ASA\_INS\_AXVIECyymmdd\_hhmmss\_YYYYMMDD\_HHMMSS\_YYYYMMDD\_HHMMSS, donde: YYYYMMDD\_HHMMSS representa la fecha y hora de inicio de la cobertura

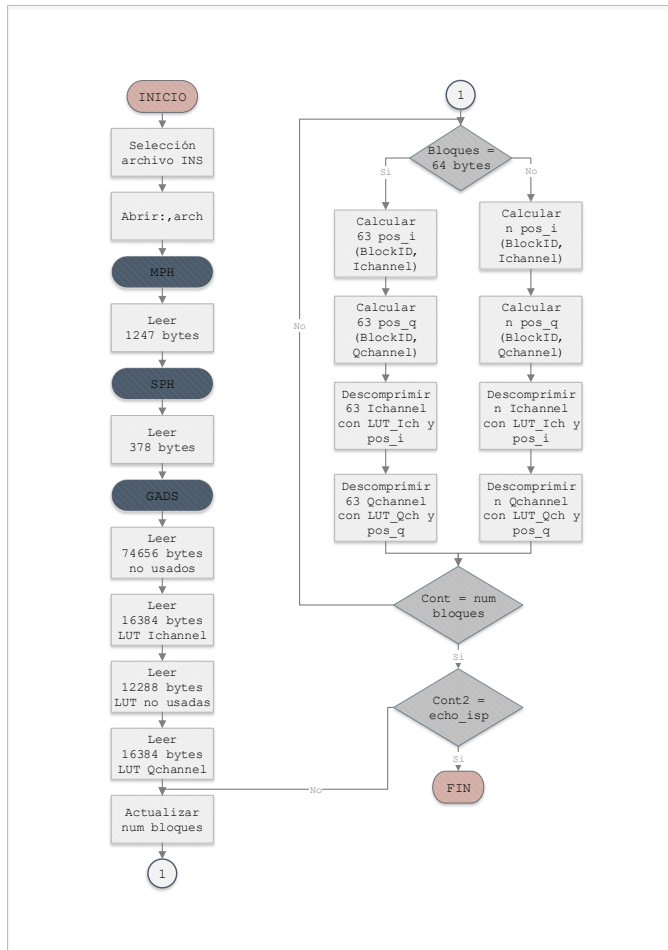


Fig. 8. Diagrama de flujo para la descompresión de los datos nivel 0.

del archivo, y `YYYYMMDD_HHMMSS` representa la fecha y hora de finalización de la cobertura del archivo.

Si hay más de un archivo que cubre el producto de nivel 0, se utiliza el más reciente, de igual forma, con el formato del nombre de archivo del producto: `ASA_INS_AXVIECyyyyymmdd_hhmmss_YYYYMMDD_HHMMSS_YYYYMMDD_HHMMSS`, donde la fecha de creación del archivo está dada por la primera fecha y hora (`yyyyymmdd_hhmmss`).

Posteriormente, se extraen las cabeceras MPH y SPH del archivo auxiliar leyendo sus bytes correspondientes.

Al llegar al contenedor de los GADS, se leen y se descartan los bytes no utilizados y enseguida se leen los bytes que poseen las LUT del canal I. Seguidamente, se descartan nuevamente los bytes no utilizados y después se leen los bytes correspondientes a las LUT del canal Q.

Una vez leídas las LUT, se procede a hallar en ellas los valores descomprimidos de cada una de las muestras adquiridas tanto del canal I como del Q. Tomando en cuenta, que esto debe hacerse a cada ventana de muestreo (línea *range* de datos) por separado, que para el modo imagen, equivale a su vez a un paquete fuente [16], [13].

Se comienza a calcular, por cada ISP, por cada uno de sus *Block ID* y a su vez por cada una de las muestras de los

```

1 #Leer GADS
2 archivo.seek(1625+74656) #Desplazamiento hasta las LUT
3
4 #Leer LUT del Ichannel
5 LUT_Ich = archivo.read(16384)
6
7 archivo.seek(12288,1) #Desplazamiento hasta las LUT
8
9 #Leer LUT del Qchannel
10 LUT_Qch = archivo.read(16384)
11 archivo.close
12
13 Des_Ich = [] #Matriz de I channel descomprimidos
14 Des_Qch = [] #Matriz de I channel descomprimidos
15
16 #Descomprimiendo uno a uno los ISPs
17 for i in range(len(Length_ISP)):
18     Des_Ich.append([])
19     Des_Qch.append([])

```

Fig. 9. Fragmento de código de descompresión.

canales I y Q correspondientes a cada *Block ID*, la posición en la LUT del valor descomprimido. Sabiendo que las mismas se encuentran estructuradas como se muestra en la tabla I.

TABLA I  
ESTRUCTURA DE LAS LUT

Posición	Block ID decimal (binario)	Channel decimal (binario)
0	0 (00000000)	-7 (1111)
1	1 (00000001)	-7 (1111)
	:	
255	255 (11111111)	-7 (1111)
256	0 (00000000)	-6 (1110)
	:	
1792	0 (00000000)	-0 (1000)
	:	
2048	0 (00000000)	+0 (0000)
	:	
4095	255 (11111111)	+7 (0111)

Para conocer la cantidad de muestras de canales I y Q que corresponde a cada *Block ID*, se verifica que éste, no sea el último. Si no es el último *Block ID*, entonces posee asociado 63 muestras tanto del canal I como de Q, por el contrario si se trata del último *Block ID* del ISP, entonces le competen las muestras de I y Q restantes del ISP.

Finalmente, luego de calcular las posiciones de los valores descomprimidos de las muestras, se ubica dicha posición en la LUT correspondiente, se extrae el valor de coma flotante y se almacena.

## VIII. IMPLEMENTACIÓN DE LOS ALGORITMOS DE DEENCAPSULACIÓN Y DECOMPRESIÓN

Para realizar la desencapsulación y descompresión de los datos crudos ASAR, es fundamental solicitar los archivos de nivel 0 ante la ESA. Para ello, es necesario registrarse e inscribir un proyecto en su plataforma y una vez aprobado, se podrá acceder en línea para visualizar y descargar los distintos productos solicitados previamente [17]. Anteriormente, este

proceso se realizaba mediante la herramienta interactiva EOLISA [17], sin embargo, a partir del 15 de febrero del 2019 EOLISA fue dada de baja por la ESA y en su sustitución, se lanzó en enero del mismo año el catálogo en línea simple (*Simple Online Catalogue*) [18]. En la figura 10 se observa el *Simple Online Catalogue*. En el mismo se detalla la zona de Venezuela seleccionada para descargar los datos de nivel 0.

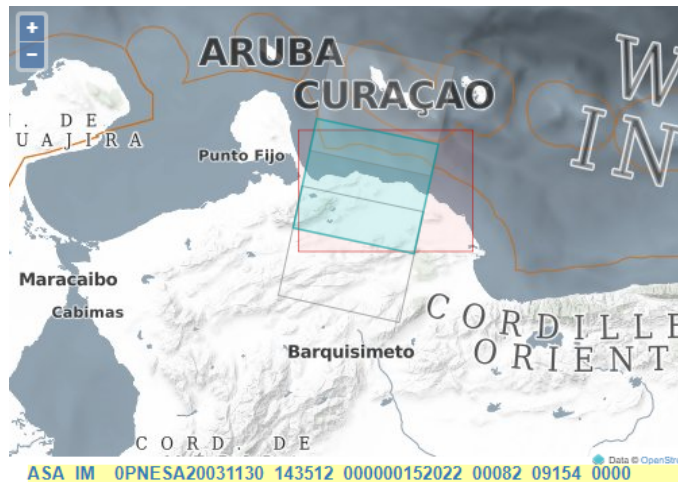


Fig. 10. Selección del producto de nivel 0 mediante el *Simple Online Catalogue*.

La imagen presente en la figura 11 representa la zona escogida de la región de Falcón de Venezuela, ya focalizada por la ESA. Posee un nivel de procesamiento de tipo 1 (datos crudos enfocados) y una resolución espacial de 30m x 30m aproximadamente [1].

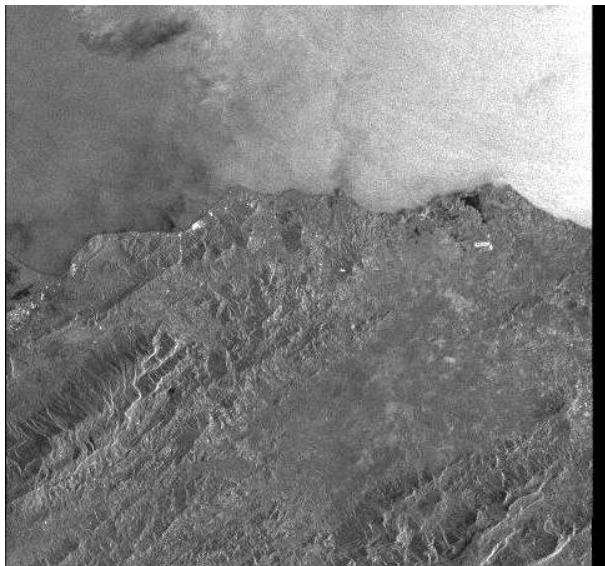


Fig. 11. Imagen de nivel 1 proporcionada por la ESA.

Los datos de nivel 0 descargados, se utilizaron para aplicar los algoritmos de desencapsulación y descompresión descritos en este documento. En la figura 12 se muestra una representación de la magnitud de los datos crudos desencapsulados y descomprimidos mediante el lenguaje de programación Python.

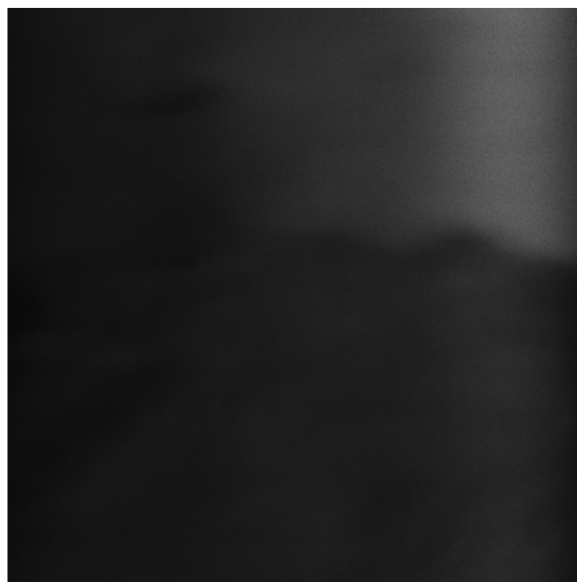


Fig. 12. Descompresión de los datos crudos extraídos del archivo de nivel 0 ASA\_IM\_\_0PNESA20031130\_143512\_000000152022\_00082\_09154\_0000.N1, correspondiente a una porción del estado Falcón de Venezuela.

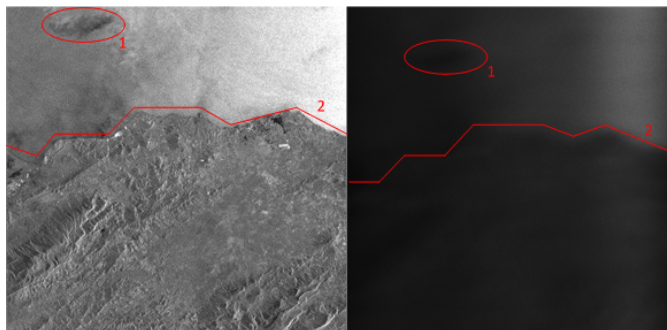


Fig. 13. Comparación de la imagen de nivel 1 de la ESA con la imagen de datos crudos procesada con los algoritmos desarrollados.

En la figura 13, se evidencia una comparativa entre la imagen nivel 1 de la ESA y la imagen obtenida con los algoritmos implementados. En ambas se resalta con líneas de color rojo los bordes que conforman el estado Falcón, comprobando así, que los algoritmos logran extraer con éxito los datos crudos (imagen sin procesar o desenfocada) del archivo nivel 0.

El desempeño computacional de los algoritmos, tomando en cuenta que el archivo descargado desde el *Simple Online Catalogue* posee un tamaño de 145 MB y que el equipo utilizado cuenta con un procesador intel core i5-2310 y 4GB de RAM, fue de 230s de tiempo de ejecución para el algoritmo de desencapsulación y de 1810s para el de descompresión.

## IX. CONCLUSIÓN

Conocer los datos de medición del instrumento ASAR en el modo imagen de nivel 0 del satélite ENVISAT, es un paso fundamental para la focalización de las imágenes provenientes de este radar. Los datos crudos están organizados dentro de cabeceras y contenedores, algunos de los cuales aportan



información relevante para la focalización y otros proveen información adicional y complementaria del instrumento.

Desencapsular, no es más que eliminar cada una de las cabeceras y contenedores que poseen los datos fuentes del modo imagen. Una vez que se llega a su contenedor, es necesario realizar la descompresión de los datos crudos. Los datos que aportan información de interés, son los de tipo eco, ya que contienen las muestras de los ecos electromagnéticos retrodispersados por los blancos sobre el terreno y recibidas en el emisor. Estos datos se encuentran comprimidos mediante el algoritmo con pérdida FBAQ-4 y para la reconstrucción del algoritmo FBAQ-4 es necesario utilizar las LUT contenidas en el *instrument characterization file*.

La realización de un algoritmo para la desencapsulación y descompresión de los datos crudos, es una herramienta que facilitará y optimizará la tarea de obtener los *source data* de los distintos archivos de nivel 0 de la ESA, para su posterior procesamiento y la obtención de forma autónoma de imágenes de nivel 1.

#### A. Trabajo a Futuro

Los datos crudos obtenidos luego de aplicar los algoritmos de desencapsulación y descompresión descritos en éste trabajo, son de vital importancia para el posterior proceso de focalización de la imagen de radar. Los datos crudos, junto con otros parámetros contenidos en algunas de las cabeceras del archivo de nivel 0, constituyen el insumo principal de los algoritmos de enfoque en general, y en nuestro caso específico del algoritmo  $\omega - \kappa$ . En el futuro cercano procederemos en efecto a procesar de manera autónoma estos datos crudos para obtener imágenes de radar para diferentes potenciales áreas de investigación. A mediano plazo este trabajo, además, servirá de base para la adaptación e implementación de los algoritmos de desencapsulación y descompresión, a los 5 modos de operación del instrumento ASAR del satélite ENVISAT.

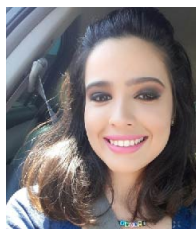
#### AGRADECIMIENTOS

A la Agencia Espacial Europea por facilitar la información necesaria para culminar con éxito esta investigación.

#### REFERENCIAS

- [1] E. S. Agency, *ASAR Product Handbook*, 2nd ed., Feb. 2007.
- [2] H. M. Braun, H. Baessler, and C. Jonas, "Daily monitoring of the mediterranean sea by geosynchronous SAR," in *Proc. IEEE Radar Conf. (RadarConf)*, May 2016, pp. 1–4.
- [3] T. Le Toan, A. Beaudoin, J. Riom, and D. Guyon, "Relating forest biomass to SAR data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 30, no. 2, pp. 403–411, Mar. 1992.
- [4] U. Wegmüller, M. Santoro, C. Werner, T. Strozzi, and A. Wiesmann, "Estimation of ice thickness of tundra lakes using ers - envisat cross-interferometry," in *Proc. IEEE Int. Geoscience and Remote Sensing Symp*, Jul. 2010, pp. 316–319.
- [5] L.-. Soh and C. Tsatsoulis, "Texture analysis of SAR sea ice imagery using gray level co-occurrence matrices," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 2, pp. 780–795, Mar. 1999.
- [6] J.-. Ovarlez, L. Vignaud, J.-. Castelli, M. Tria, and M. Benidir, "Analysis of SAR images by multidimensional wavelet transform," *Sonar and Navigation IEE Proceedings-Radar*, vol. 150, no. 4, pp. 234–, Aug. 2003.

- [7] W. Tian, Y. Shao, and S. Wang, "A system for automatic identification of oil spill in envisat asar images," in *Proc. IGARSS 2008 - 2008 IEEE Int. Geoscience and Remote Sensing Symp*, vol. 3, Jul. 2008, pp. III – 1394–III – 1397.
- [8] J. Ignacio Fernandez Michelli, M. Hurtado, J. Areta, and C. Muravchik, "Polarimetric SAR image segmentation using cem algorithm," *IEEE Latin America Transactions*, vol. 12, no. 5, pp. 910–914, Aug. 2014.
- [9] I. G. Cumming, Y. L. Neo, and F. H. Wong, "Interpretations of the omega-k algorithm and comparisons with other algorithms," *IEEE*, vol. 3, pp. 1455–1458, Jul. 2003.
- [10] M. Cardaci, "Envisat-1 products specifications," Aug. 2015, volume 5: Product Structures.
- [11] I. McLeod, "Envisat-1 products specifications," Oct. 1998, volume 6: Level 0 Products Specification.
- [12] M. Cardaci, "Envisat-1 products specifications," Aug. 2015, product Data Conventions.
- [13] D. Demuth, "Measurement data format definition," Apr. 1999.
- [14] G. Sotis, "Envisat-1 products specifications," Nov. 2007, volume 16: Auxiliary Data Files.
- [15] A. Kult, "Envisat-1 products specifications," Jan. 2012, volume 8: ASAR Products Specification.
- [16] I. McLeod, "Envisat asar data decoding," Jan. 2000.
- [17] I. Franchi, J. Santander, and A. Zozaya, "Procedimiento para el acceso a los datos de las misiones ers y envisat a través del portal de la esa y la herramienta interactiva eoli-sa," *Revista INGENIERIA UC*, vol. 24, no. 2, pp. 269–278, Aug. 2017.
- [18] ESA, "Eoli-sa dismissed - earth online - esa," 2019, accessed: 07-Jun- 2019. [Online]. Available: <https://earth.esa.int/web/guest/eoli-sa-dismissed>



**Indira Franchi** Telecommunication Engineering student of the University of Carabobo (UC), Valencia, Venezuela. Currently, she is finishing her thesis degree titled "Development of a computational tool for the processing of level 0 ASAR files of the satellite mission ENVISAT". Her research areas of interest are synthetic aperture radars, cybersecurity, networking, antennas and propagation.



**Alfonso Zozaya** received the B.Sc. degree in Electronic Engineering, with a major in Telecommunication, from the Polytechnic Institute of the National Armed Forces of Venezuela (I.U.P.F.A.N.), Maracay, Venezuela, in 1991, and his PhD degree from the Polytechnic University of Catalonia (UPC), Barcelona, Spain, in the area of Signal Theory and Communications in 2002. He worked as a Professor at the University of Carabobo, Valencia, Venezuela from 1994 to 2014. He worked as a senior researcher at the Ecuadorian Space Institute, Quito, Ecuador,

in the area of synthetic aperture radars in the periods from September 2014 to September 2015 and from August 2016 to August 2017. Currently, he is with the Universidad Tecnológica Metropolitana (UTEM), Santiago de Chile, where he works as a Full Professor at the Department of Electricity. His research areas of interest are applied electromagnetic, computational electromagnetic, digital signal processing, RF circuits design, and synthetic aperture radars.