SPACE ENGINEERING DEVELOPMENT Co.,Ltd.

SED-TEC21751

Hinode data processing service support (consignment)

JPEG data restoration module specifications

訂符	日付	関連	承認	点検	作成
B	2011/3/1		清野	石川	真鍋
A	2009/7/15		下平		石川
	2009/3/27		下平	石川	前原

配付				作成部署
社外		社内		第一事業部 宇宙システム技術部

宇宙技術開発株式会社

このページは、故意に空白にしている。

改訂履歴

訂符	日付	改訂ページ	改訂理由
ー	2009/03/27	ー	初版
A	2009/7/15	1	概要から契約に関する文章を削除。 適用文書を削除。
		2	不要な情報 (初期化する内部変数の一覧) を削除。 science header の記述を実験計画書に合わせ science data header に変更。 モジュールの呼出し形式に、パケットの SIRIUS 時刻、JPEG データ時刻、バイナリ情報を追加。 引数の説明を詳細化。
		3	引数の説明に JPEG データ時刻、バイナリ情報を追加。
		5	誤記修正。 欠損フラグテーブルの参照方法についての補足追加。
		6	構造体定義が増えたことに伴い、説明を変更。 メンバ Data type の説明を詳細化。
		8	構造体定義を追加。以降のページを 1 ページずつずらした。
		10	science header の記述を science data header に統一。 DepacketizeBody() 呼出しに引数を追加。
B	2011/03/01	i	誤記修正
		2	DePacketizeBody() を呼出す際の引数を変更。 packet, utime_pkt, jpg_data, chk_tbl の型を変更。
		3	戻り値のステータス名の誤記修正。
		9	マクロ定義の名前が間違っていたものを修正。 MAX_PIXELS, MAX_COMP_LEN を追加。
		10	DePacketizeBody() に渡す引数の型を変更。また、それに伴い DepacketizeBody() の呼出し形式を変更。 説明文を修正

有効ページ一覧

有効ページ

i	B 版
1	A 版
2	B 版
3	B 版
4	初版
5	A 版
6	A 版
7	初版
8	A 版
9	B 版
10	B 版

B

Contents

このページは、故意に空白にしている。

1. Overview

This document is the summary of the specification for the JPEG data restoration module and the usage of the module.

A

The JPEG data restoration module constructs the image data as a data which only lacks partial images where the packet losses occurred, by inserting the JPEG restart marker into the image data. The entire image data had been discarded when the packet losses occurred in the observational data from the *Hinode* satellite under the current processing method.

2. Associated document

A

(1) "Experimental plan for Scientific Satellite Solar-B" SES-TD-05-010

(2) "Investigation of JPEG format" SED-TEC21016

3. JPEG data restoration module

The JPEG data restoration module extracts or restores the JPEG data from the input CCSDS packet.

3.1 Module specification

The module specification is shown on the next page.

3.1.1 DePacketizeBody.c

Name	Language	Invoker
DePacketizeBody.c	C	―――

Functional overview

This module is composed of the following two functions.

(1) Initialization（DePacketizeBody_init）

・Initialize internal variables which are used in the JPEG restoration process. A

(2) JPEG data extraction and restoration processing（DePacketizeBody）

・extract or restore the JPEG data from the input CCSDS packet

・output the science data header of the extracted/restored JPEG data, the size of the JPEG data, APID, and the JPEG data loss flag table A

Input-output file:

――

Submodule:

――

Syntax:

(1) Initialization

```
void DePacketizeBody_init(void)
```

(2) JPEG data extraction and restoration processing

```
int DePacketizeBody(const unsigned char packet[],  const int64_t *utime_pkt,  int *apid,
                    struct sci_head_t *sci_head,  unseigned char jpg_data[MAX_COMP_LEN],
                    int *jpg_size,  unsigned char chk_tbl[MAX_PIXELS],  struct jpg_time_t *jpg_time,
                    struct bin_head_t *bin_head)
```

A B

№	parameters	Type and size	Unit	Description	input/ output
1	packet	const unsigned char []	－	CCSDS packet to be processed	input
2	utime_pkt	const int64_t *	μs	packet edition time (elapsed micro-seconds from 2000/01/01 00:00:00 (UT)) This parameter will be sotred in jpg_time->utime if corresponding packet is the first packet of JPEG data. If this parameter is NULL, jpg_time->utime is set to 0.	input
3	apid	int *	－	APID of the corresponding packet	output
4	sci_head	struct sci_head_t *	－	the science data header of the extracted/restored JPEG image data (details of the structure are shown in section 3.2)	output
5	jpg_data	unsigned char []	－	the buffer where JPEG data is stored	output
6	jpg_size	int *	－	Size of JPEG data	output
7	chk_tbl	unsigned char []	－	buffer where JPEG data loss flag table is stored chk_tbl[i] =0: indicates the JPEG image data is extracted =1: indicates the JPEG image data is lost	output

B A B B B

№	parameters	Type and size	Unit	Description	input/ output	
8	jpg_time	struct jpg_time_t *	μ s –	packet edition time of the first packet for the extracted/ restored JPEG image data jpg_time->utime: elapsed micro-seconds from 2000/01/01 00:00:00 (UT) jpg_time->ti_time: Solar-B S/C low resolution TI time (the value in the CCSDS secondary header) This parameter can be set to NULL.	output	A
9	bin_head	struct bin_head_t *	–	following binary data (1) Version information of MDP compression tables (2) Packet information and Data compression information in the science data header (3) Data information in the science data header (details of the structure are shown in section 3.2) This parameter can be set to NULL.	output	
10	Return value	int	–	Processing status = BUFF_NOW(0): buffering JPEG data = JPEG_EXTRACT(1): The JPEG data is able to be extracted. = JPEG_RESTORED(2): The JPEG data is restored. = RESTORED_IRRETRIEVABLE(3): The part of the first half of the JPEG data was restored, and the latter half was discarded. =NOT_APID(-1): input packet is not a packet to be processed. =NOT_JPEG_DATA(-2): input packet is not a packet of the JPEG data. =JPEG_IRRETRIEVABLE(-3): The JPEG data is not able to be restored.	output	B

（Complement）

The JPEG data loss flag table (chk_tbl) is two dimensional array (Xsize and Ysize are same as image data size)* that indicates whether the data segments of the JPEG data are lost.

Each elements of the array corresponds to the pixel of the subimage.

The JPEG data segments in the observational data of the *Hinode* satellite have the following sizes, and the correspondence of the JPEG data segment and the image pixel position is as shown in Figure 3.1.1-1.

When the compress mode is DCT　　X size: 512 pixel
　　Y size:　8pixel

When the compress mode is DPCM　　X size: It is the same as X size of the subimage.
　　Y size: JPEG Restart Interval ÷ X size

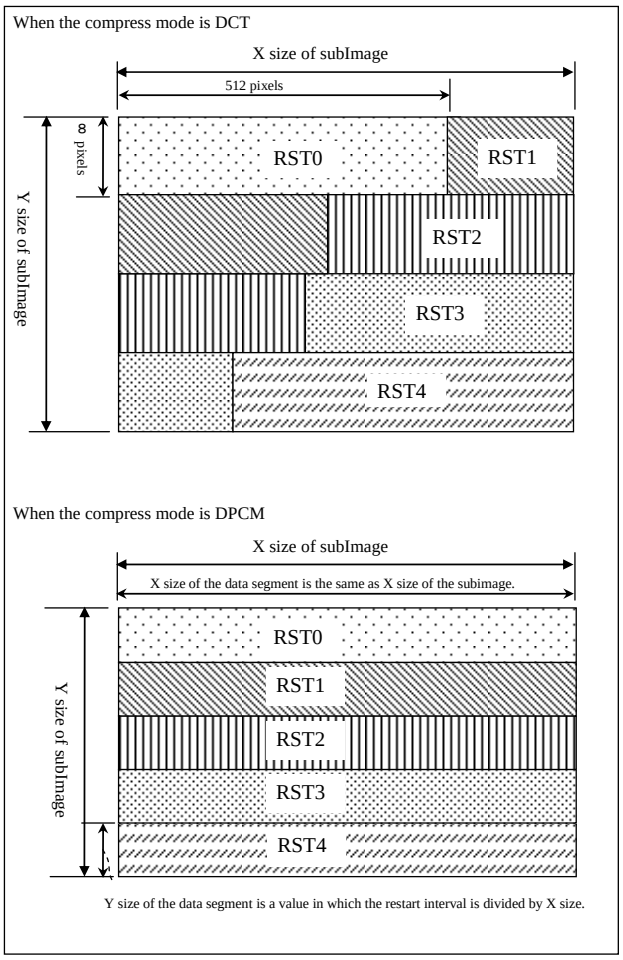

Figure 3.1.1-1　Correspondence of JPEG data segment and image pixel position

*) Because variable-length two dimensional array can not be passed in C language, chk_tbl is passed as one dimensional array of size X by Y.

If the data segment RST1 of the JPEG data is lost in the compress mode DCT, and X size of the subimage is assumed to be 600 pixels, the loss flag table is set as follows.

the following area is set to one (Figure 3.1.1-2). rest is set to zero.

0th line : chk_tbl[0][512]～chk_tbl[0][599]
1st line : chk_tbl[1][512]～chk_tbl [1][599]
2nd line : chk_tbl[2][512]～chk_tbl [2][599]
3rd line : chk_tbl[3][512]～chk_tbl [3][599]
4th line : chk_tbl[4][512]～chk_tbl [4][599]
5th line : chk_tbl[5][512]～chk_tbl [5][599]
6th line : chk_tbl[6][512]～chk_tbl [6][599]
7th line : chk_tbl[7][512]～chk_tbl [7][599]

8th line : chk_tbl[8][0]～chk_tbl [8][424]
9th line : chk_tbl[9][0]～chk_tbl [9][424]
10th line: chk_tbl[10][0]～chk_tbl [10][424]
11th line: chk_tbl[11][0]～chk_tbl [11][424]
12th line: chk_tbl[12][0]～chk_tbl [12][424]
13th line: chk_tbl[13][0]～chk_tbl [13][424]
14th line: chk_tbl[14][0]～chk_tbl [14][424]
15th line: chk_tbl[15][0]～chk_tbl [15][424]

A

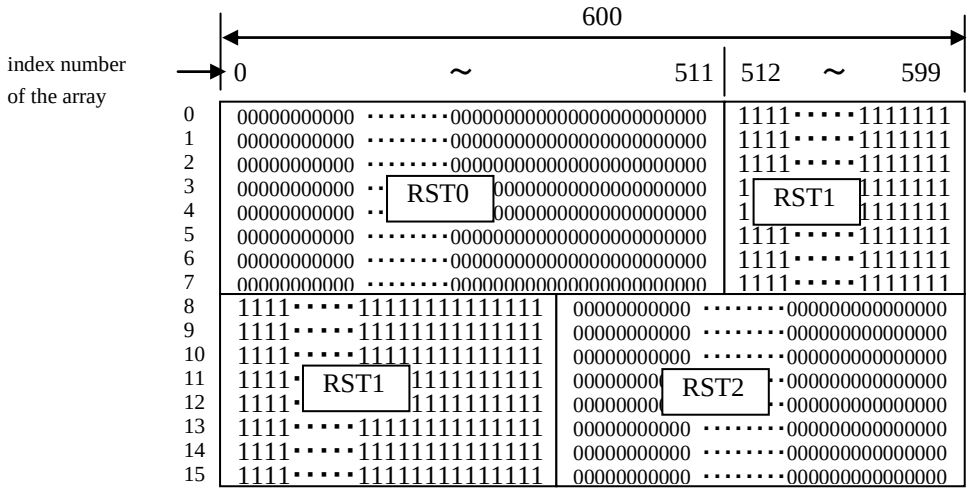

Figure 3.1.1-2 Example of the setting of JPEG data loss flag table

※ Because chk_tbl is an one-dimensional array, chk_tbl[0][512]～chk_tbl[0][599] are actually referred as chk_tbl[512]～chk_tbl[599], chk_tbl[1][512]～chk_tbl[1][599] are referred as chk_tbl[1112]～chk_tbl[1199], and so on.

A

3.2 Structure specification

This module defines following three structures.

(1) sci_head_t

(2) jpg_time_t

(3) bin_head_t

The specification of each structure is shown below.

A

Structure 1

Name	sci_head_t (1/2)		
Description	Science data header		
member	Type and size	Unit	description
DataType	int	—	Data type SOT 0x42, 0x43 FG Data 0x44, 0x45 SP Data 0x52, 0x53 CT Reference image 0x54, 0x55 CT Live Image 0x56, 0x57 CT Diagnostic Data 0x58, 0x59 Operating System Diagnostic Data 0x5E, 0x5F Memory Dump Data via Mission I/F XRT 0xA2, 0xA3 2sec Data ＋ Diagnostic Data EIS 0xC2, 0xC3 Memory Dump Data
PacketSize	int	—	Packet size
SerialPacketNo	int	—	Packet sequence Number
MainID	int	—	Main ID
MainSQFlag	int	—	Main sequence flag 0：Continuation Frame 1：First Frame 2：Last Frame 3：Stand_alone Frame
MainSQCount	int	—	Main sequence number
NumOfPacket	int	—	Number of packets in image frame
NumOfFrame	int	—	Number of image frames in data set
SubID	int	—	SubID
SubSQFlag	int	—	Sub sequence flag 0：Continuation Frame 1：First Frame 2：Last Frame 3：Stand_alone Frame
SubSQCount	int	—	Sub sequence number
FullImageSizeX	int	—	Width of full image data
FullImageSizeY	int	—	Height of full image data

Structure 1			
Name	sci_head_t (2/2)		
Discription	Science data header		
member	Type and size	Unit	description
BasePointCoorX	int	—	X position of partial image data in full image data
BasePointCoorY	int	—	Y position of partial image data in full image data
PartImageSizeX	int	—	Width of partial image data
PartImageSizeY	int	—	Height of partial image data
BitCompMode	int	—	Bit compression mode 0:no compression 1:16U to 12U 2:14U to 12U 3:16S to 12U 4:14.5S to 12U 5:13S to 12U 6:Low order 12U 7:EIS table
ImageCompMode	int	—	Image compression mode 0:noimage 3:DPCM 7:DCT
HTACNo	int	—	AC Haffman table number Table number : 1 or 2
HTDCNo	int	—	DC Haffman table number Table number : 1
QTNo	int	—	Quantization table number Table number : 0 ～ 7
DataInfo	unsigned char[256]	—	Data Information The data length is different according to the data type. (1) FLT_OBS1, FLT_OBS2, SPP_OBS1, SPP_OBS2 are 48 bytes (2) XRT_OBS1, XRT_OBS2 are 64 bytes (3) EIS_OBS1, EIS_OBS2 are 224 bytes

Structure 2			
Name	jpg_time_t		
Description	packet edition time of the first packet for the extracted/restored JPEG image data		
member	Type and size	Unit	description
utime	int64_t	μs	packet edition time expressed in micro-seconds elapsed from 2000/01/01 00:00:00 (UT).
ti_time	int	—	packet edition time expressed in Solar-B S/C low resolution TI. (the value in the CCSDS secondary header of the first packet)

Structure 3			
Name	bin_head_t		
Description	binary information of the extracted/restored JPEG image data		
member	Type and size	Unit	description
verinfo	unsigned char[8]	—	Version information of MDP compression tables in first CCSDS packet verinfo[0～1]: Bit compression lookup table version [2～3]: AC Huffman table version [4～5]: DC Huffman table version [6～7]: Quantization table version
mdphead	unsigned char[32]	—	first 32 byte of the science data header
datainfo	unsigned char[256]	—	Data information in the science data header
datainfo_len	int	—	length of the data information

3.3 Macro definition

The following macro definitions used in this module. B

Macro name	Value	Content	
BUFF_NOW	0	The processing status value.	buffering The JPEG data in the CCSDS packet.
JPEG_EXTRACT	1		The JPEG data in the CCSDS packet has been extracted.
JPEG_RESTORED	2		The JPEG data in the CCSDS packet was restored.
RESTORED_IRRETRIEVABLE	3		The part of the first half of the JPEG data in the CCSDS packet was restored, and the latter half was discarded.
NOT_APID	-1		APID in the CCSDS packet is not APID of the object.
NOT_JPEG_DATA	-2		The data in the CCSDS packet is not JPEG data.
JPEG_IRRETRIEVABLE	-3		The JPEG data in the CCSDS packet was discarded.
MAX_PIXELS	256x1024	Maximum image pixel from the telescopes. Maximum image size @MDP = 256k pixel (for EIS), 64k (for SOT and XRT)	
MAX_COMP_LEN	MAX_PIXELS x4	Maximum compressed size of a image.	

B

4. Use of the module

In this module, the DePacketizeBody_init() function should be called first, and the DePacketizeBody() function should be called after each input of the CCSDS packet.

In the DePacketizeBody() function, the JPEG data contained in the CCSDS packet is accumulated in an internal buffer, whenever extraction of one subimage is completed, the JPEG data and the status extracted are returned to the caller module.

The function prototype, the structure, and the macros of this module are defined in include file (DePacketizeBody.h).

The example of using this module is shown below.

```
#include <DePacketizeBody.h>              ← Specification of include file
#define MAX_PACKET 1024*1024;

int main() {
    unsigned char   packet[MAX_PACKET];   ← The storage buffer for the CCSDS packet
                                              must be prepared by the caller.                    B
    int   apid;
    unsigned char   jpg_data[MAX_COMP_LEN];     ← The storage buffer for JPEG data must be prepared
                                                    by the caller.
    int   jpg_size;
    unsigned char   chk_tbl[MAX_PIXELS];              ← The storage buffer for the loss flag table
                                                          must be prepared by the caller.
    struct sci_head_t sci_head;       ← The storage buffer for the science data header must be prepared   A
                                          by the caller.                                                  |
                                          The structure of the science data header is defined in the include file.
    int proc_apid;
    DePacketizeBody_init();           ← The initialization function should be called first.
    scanf ("%X", &proc_apid);
    while (recv_packet(&packet) != PACKET_END) {
        /*DePacketizeBody function should be called after each input of the CCSDS packet */
        iret=DePacketizeBody(recv_packet(&packet), NULL, &apid, &sci_head,                          A B
                                                    jpg_data, &jpg_size, chk_tbl, NULL, NULL);
         switch (iret) {                  ←branch on the return value.
            case BUFF_NOW:
                continue;
            case JPEG_EXTRACT:
            case JPEG_RESTORED:
            case RESTORED_IRRETRIEVABLE:
                if (apid == proc_apid) {
                    image_data = ImageProcessing (jpg_data, sci_head, jpg_size);
                    /*image data output*/
                    fwrite (image_data, sci_head.PartImageSizeX,
                                    sci_head.PartImageSizeY, fpOUT1);
                    fwrite (chk_tbl, sci_head.PartImageSizeX,
                                    sci_head.PartImageSizeY, fpOUT2);
                }
            default:
                continue;
        }
    }
    return 0;
}
```

The processing of the user program is executed by using the data that the JPEG restoration module outputs.