2007 年度 芝浦工業大学 システム工学部

電子情報システム学科

総合研究論文

ROSAT衛星による X線プラネタリウムの基礎開発 The basic development of the X-rays planetarium by the ROSAT satellite

p03086

西尾 光史

指導教員: 久保田 あや

目 次

第1章 1.1 1.2	はじめに X線1 研究目的
笠の辛	
弗 2 早 ○ 1	\mathbf{A} 線大乂開生 $ROSAI C 三大リーハイ 3 DOG(AT) 第日の期間$
2.1	ROSAT 衛星の慨安
2.2	各載機器4
	2.2.1 HRI
	2.2.2 PSPC
	2.2.3 XRT
2.3	全天サーベイ \dots \dots 7
第3章	The Flexible Image Transport System $: FITS$
3.1	FITS とは?
3.2	<i>FITS</i> ファイルの構造
3.3	<i>FITS</i> の誕生とその後の動き 10
∽∡辛	
弗 4 早 4 1	World Coordinates System · WCS
4.1	
4.2	WCSの誕生と経緯
4.3	基本 $FTTS(県始 FTTS) での変換 14$
4.4	WCS での基本変換 (WCS Paper I より) $\dots \dots \dots$
	4.4.1 変換の概要 変換の概要 14
	4.4.2 変換行列の表し方 2.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1
4.5	天球座標の変換 (WCS Paper II より)
	4.5.1 天球座標 (α, δ) から射影平面座標 (x, y) への変換
	4.5.2 射影平面座標 (x, y) から天球座標 (α, δ) への変換
4.6	さまざまな投影法
	4.6.1 ZEA:Zenithal equal-area 投影法
	4.6.2 CAR:Plate Carrée 投影法 23
	4.6.3 AIT:Hammer-Aitoff 投影法 23
	4.6.4 PCO:Polyconic 投影法 25
	4.6.5 CSC:COBE(\underline{CO} smic <u>Background Explorer</u>) Quadrilateralized Spherical
	Cube 投影法 26
第5音	開発 90
5.1	開発に使用するソフトウェア 90
0.1	511 CFITSIO 20
	0.1.1 0711510

		5.1.2 WCSLIB
		5.1.3 STIFF
		5.1.4 farith $\ldots \ldots 30$
5	5.2	<i>ROSAT</i> 全天サーベイデータの入手 31
5	5.3	各 fits データの構造
5	6.4	全天画像 fits ファイルの作成
		5.4.1 個々の fits ファイルの全天への投影
		5.4.2 画像データの重なりの削除 34
5	5.5	プラネタリウム投影用カラー画像の作成 38
		5.5.1 プラネタリウム投影用カラー画像を作る上での注意点
		5.5.2 画像の作成 38
5	6.6	上映

笛	6章	まとめと今後の課題
21	$\mathbf{v} =$	

43

付録 A	45
A.1 <i>FITS</i> のヘッダレコード	. 45
A.2 <i>FITS</i> ファイルを扱うソフトウェア	. 46
A.2.1 主なソフトウェアの種類	. 46
A.2.2 主なソフトウェアの入手先	. 48
A.3 cfitsio のインストール	. 48
A.4 WCSに関係するその他のキーワード	. 48
A.4.1 非線形アルゴリズムに伴うパラメータのキーワード:PV <i>i_m</i> , PS <i>i_m</i>	. 48
A.4.2 世界座標の次元に関するキーワード:WCSAXES	. 49
A.4.3 単位に関するキーワード:CUNITi	. 49
A.4.4 キーワード値のデフォルト	. 49
A.4.5 軸の代替記述	. 50
A.4.6 座標の不確定性に関するキーワード	. 50
A.5 STIFFのConfigurationファイルとパラメータ	. 51
A.6 プラネタリウム投影用画像の作成	. 51
A.6.1 fits $\mathcal{O} \land \mathcal{P} \not \mathcal{P} = (931049 \text{p-fp1.fits}) \dots \dots$. 51
A.6.2 プログラム (convertWCS.c)	. 53
A.6.3 プログラムに使用されている関数の解説	. 57
A.6.4 画像サイズの変更	. 60

第1章 はじめに

1.1 X線

X線は可視光の1000~1万倍も高いエネルギーの電磁波であり、ブラックホールや超新星残骸、 銀河団などの高温・高エネルギーの天体を起源としている。

その X 線を X 線カメラで捉えてみると普段我々が見ている可視光の天体とは違う映像を見ることが出来る。その例として図 1.1 に可視光の太陽と X 線カメラで捉えた太陽を示す。図を見てわかるように可視光で見るとおとなしい太陽でも X 線では激しく活動している様子が見て取れる。



図 1.1: 可視光の太陽 (左) とようこう衛星軟 X 線望遠鏡で捉えた太陽 (右)

このように、X線で見た宇宙は、普段見慣れている宇宙とは違う様子を見せてくれている。

1.2 研究目的

近年、XMM-Newton(ヨーロッパ)やROSAT(ドイツ、アメリカ、イギリス)、すざく(日本)な どのX線天体観測衛星が次々と登場し、より鮮明なX線で捉えた天体の画像を得られるように なった。そこで本総合研究ではこの画像データを用い、普段見かける一般的な可視光のプラネタ リウムのようなX線デジタルプラネタリウムを開発することを目的とした。

今回そのX線プラネタリウムに用いる画像は、公開されているデータが全天をカバーしている ことから ROSAT 衛星による全天サーベイ観測によって得た画像データを用いることにした。そ して、プラネタリウム投影用のフォーマットでつなげるプログラムを作り、実際にプラネタリウ ムに投影することを目指し、研究を進めた。

第2章 X線天文衛星 ROSATと全天サーベイ

2.1 *ROSAT*衛星の概要

ROSAT(<u>RO</u>entgen <u>SAT</u>ellite)¹[1][2] は、ドイツ、アメリカ、イギリス間の協同プログラムにより開発された X 線観測衛星である。この衛星の設計と運用はドイツにて行われた。1990 年 6 月 1 日 にアメリカより打ち上げられ、1999 年 2 月 12 日に役目を終えるまでに PSPC(<u>Position Sensitive</u> <u>Proportional Counter</u>) による全天サーベイなどの様々な観測、調査をした。

図 2.1、図 2.2 はそれぞれ ROSAT の外観と簡単な ROSAT の図である。動力は、太陽の光が当 たっている間は 3 枚のソーラーパネルから供給され、1 軌道につき最高 40 分間のスペースクラフト (地球の裏側に回り太陽の光が当たらない時間帯)では充電式バッテリーにより供給された。観測は XRT(<u>X-Ray Telescope</u>) とその焦点面にある 2 種類の検出器 PSPC および HRI(<u>High R</u>esolution Imager)の組み合わせで行われる。表 2.1 に ROSAT の特性を示す。



図 2.1: ROSATの外観

¹http://heasarc.gsfc.nasa.gov/docs/rosat/rosat3.html



⊠ 2.2: ROSAT

2.2 搭載機器

2.2.1 HRI

HRI²は ROSAT に搭載されている高空間分解能映像器で、これは 1978 年~1981 年に稼働した Einstein 衛星の HRI ととてもよく似ており、クロスグリッド位置表示システムとして 2 つの滝状 のマイクロチャンネルプレート (MCPs) で構成されている。HRI は一辺 38 分角の正方視野につい て、FWHM で 2 秒角の高空間分解能を持つ。またエネルギー分解能は良くないものの 61 マイク ロ秒以下の時間分解能を提供する。表 2.2 に HRI の基本性能を示す。

2.2.2 PSPC

PSPC³[3] は位置検出型の比例計数管であり、MPE(<u>Max Planck Institute for Extraterrestrial</u> Physics:マックス・プランク研究所)によって開発された。図 2.3 に PSPC の外観を、および図 2.4 に PSPC の観測窓の概念図を示す。陰極ストリップ読み出しによって、検出位置を決定するマ ルチワイヤー比例計数器である。エネルギーEに大使、 $\Delta E/E = 0.43(E/0.93)^{-0.5}$ で表されるエ ネルギー分解能を持ち、2 度以上の直径視野で高空間分解能 (1keV で 25 秒角)と、130 マイクロ秒 までの時間分解能を有する。表 2.3 に *ROSAT* PSPC の基本性能を示す。[4]

²http://heasarc.gsgc.nasa.gov/docs/rosat/hri.html

 $^{^{3}} http://heasarc.gsfc.nasa.gov/docs/rosat/pspc.html$

打ち上げ	1990年7月1日
ミッション終了	1999年2月12日
初期高度	$580 \mathrm{km}$
軌道周期	96分
歳差	$66~^\circ$
方向転換周期	402秒
エネルギーバンド	0.1- $2.4 keV$
衛星重量	2400 kg

表 2.1: ROSAT の特性

視野	38 分角 (平方視野)
空間分解能	1.7 秒角 FWHM(Full Width Half Maximum)

表 2.2: ROSAT HRI の基本性能

PSPC は二つのカウンターから成っている。一つは位置決定のための二つの陰極 K1、K2 およ び陽極 A1 から成る、X 線信号を捕らえる X 線精密カウンターであり、もう一つは陽極 A2 から成 る、バックグラウンド除去用のアンチコインシデンスカウンターである。陽極は金メッキをされ た直径 10 µ m のタングステンワイヤーであり、A1 は 1.5mm、A2 は 2mm 間隔で並べられてい る。一方、陰極はプラチナイリジウムワイヤーでできており、その直径は 50 µ m、0.5mm 間隔で 並べられている。陰極は信号処理のために 7~8 本で一纏めにされている。陽極および陰極は混合 ガスで満たされた容器の中でグリッドを構成している。ガスの成分はアルゴンが 65 %、キセノン が 20 %、メタンが 15 %と成っている。細い入射窓からガスの中に入ってきた X 線は、ガスに光 電吸収され、光電子を作り出すことによって信号として検出される。[5]

PSPC は PSPC-B と PSPC-C による冗長系をなし、これらは *ROSAT*の XRT の焦点面タレットの中で回転するホルダーに収められた。2 つの予備の PSPC(PSPC-A と PSPC-D) は組み立てられ、いくつかの地上構成測定に使われた。

打ち上げ後、主に PSPC-C を用いて全天サーベイ等を行っていたが、1991 年1月25日の AMCS(Attitude Control System)の不具合によりおよそ 15 時間制御不能になり姿勢を崩した。その姿勢を崩している間に得られた管理データによると衛星が太陽の方向をスキャンしており、その結果、PSPC-Cが故障した。その後の観測には残された PSPC-B が使われ 1991 年 8 月 3~13 日の間にほとんどくまなく調査した。⁴そして、1994 年 9 月 11 日に 4 年間の活動が成功した後、PSPC は残留する検知器ガスを節約するためにシャットダウンされた。

⁴http://heasarc.gsfc.nasa.gov/docs/rosat/rass.html



図 2.3: PSPC の外観



Figure 3.6: A schematic diagram of the ROSAT PSPC window support structure showing the ribs, ring and wire meshes.

図 2.4: PSPC の観測窓の概念図

ウィンドウサイズ	直径 8cm
視野	直径 2°
混合ガス	アルゴン 65%
	メタン 15%
	キセノン 20%
運用圧力	1.466 bar at 22 °C
エネルギー分解能	43% at $0.93 \rm keV$
空間分解能	300 μ m (${\sim}25''$) at 1keV
入射窓	1 μ m polypropylene

表 2.3: ROSAT PSPC の基本性能

空間分解能	1arcsec
エネルギーバンド	0.1- 2.0 keV
有効面積 (@1keV)	$300{\rm cm}^2$

表 2.4: ROSAT XRT の基本性能

2.2.3 XRT

XRT は 4 組の Wolter-I 型ミラーから構成されており、その口径は 83.5cm、焦点距離は 240cm である。その視野は直径 2°、角分解能はおよそ 5 秒角である。表 2.4 に *ROSAT* XRT の基本性 能を示す。

2.3 全天サーベイ

*ROSAT*が行ったサーベイの一つで、1990年7月30日よりおよそ6ヶ月間 PSPC を用いて行われた。図 2.5 に全天サーベイ⁵のフィールド構成を示す。この図のように、このサーベイにより得られた PSPC による全天マップは幅 6.4 度×6.4 度ごとの 1378 個の RASS(<u>ROSAT All-Sky Survey</u>) フィールドに分けられており、隣り合ったフィールドは少なくとも 0.23 度重なり合っている [6][7]。PSPC のエネルギーバンドは 0.1-2.4keV で、全天サーベイのデータは 0.1-0.4keV、0.4-0.9keV と 0.9-2.4keV の 3 バンドに分けられて記録されている。

図 2.5 のように、それぞれのフィールドは連続する 1~33 までのゾーン番号(赤緯)と、ゾーン 番号に依存した連続する 1~64 までのセグメント番号(赤径)により特定される。

⁵http://www.xray.mpe.mpg.de/rosat/survey/rass-3/main/help.html

	-	N	ო	4	S	9	2	ω	ი	10	÷	4	<u>9</u>	4	15	16	17	1 8	19	20	5	22	23	24	25	26	27	28	29	8	<u>9</u>	33 33 33		
0	Γ				-/	-	-	7	-	-	-	F	٢	-	-	-	F	F	F	-	-	-	1	र	-	-	4	۰					0	
-			-	-			2	N	N	2	N	3 2	3 2	3 2	3 2	3 2	3 2	3 2	3	3 2	3 2	2 M	2	2	2	2	~		/	-	-		-	
		-			N		3	e	e	3	4	4	4	4	4	4	4	4	4	4	4	4	4 3	1 3	в	9	0	2	0			-	-	
N	_					e	4	4	4	5 4	5	5	65	6 5	6 5	6 5	6 5	6 5	6 5	6 5	6 5	5	5	5	4	4	4	3	3				ŝ	
			N	8		4	10	5	9	9	9	7 6	1	2	3 7	5 7	2	3 7	2	4	7	7 6	9	9	6 5	5	10	4		в	N			
ო	-				4	2	<u> </u>	9	2	~	8 7	8	98	8 6	9 6	8 6	8 6	9 6	6 0	9 8	8	8	8 7	2	4	ø		5	4	_			ŝ	
		2	3	4	5	-		-	8	8 6	6	6 0	10	1 10	1 10	1 10	1 10	1 10	1 10	1 10	10	6 0	6	8 6	8	~	H	/	5	4	þ	~		
4						9	2	ω	6	10	9	11 1	2 11	12 11	12 1	12 1	12 1	12 1	12 1	12 11	2 11	1	10	10 \$	6	8	7	9					4	
2	_	_		2	9	2	8	6	ę	Ŧ	12 11	3 12	13 1	4 13	4 13	4 13	4 13	4 13	4 13	4 13	13 1	3 12	12 11	FF.	10	6	Â	7	9	5			2	
			4		7	8	6	10	7	3 12	13	14 13	5 14	15 1-	15 1	15 1	15 1	15 1	15	15 1	5 14	14 13	13	3 12	11	위	ര	8	7		4			
9		в		9		6	10	÷	3 12	4	14	15	16 1	7 16	11 16	16	16	11 16	11 16	7 16	16 1	15	5 14	4 13	3 12	뒤	우	6	0	9		m	9	
			5		œ	_	÷	12	4	15 1	6 15	7 16	3 17	18 1	18 1	18	18	18	18	18 1	3 17	7 16	6 15	15 1	14	12	÷	-	8		2			
\sim	-			~	6	ę	12	13	15	16	17	18 1	19 18	0 19	20 19	20 19	20 19	20 19	20 19	0 19	19 18	18 1	17 1	16	15 1	13	1₽	10	9	7			· ~	
~				\square		F	13	14	16	17	18	19	20	21 2(21 2	21 2	21 2	21 2	21 2	21 20	20	19	18	24	16	14	é	11			1		~	
ω		4	9	°	9	12	4	15	17	9 18	0 19	21 20	22 21	3 22	23 22	23 22	23 22	23 22	23 22	3 22	22 21	21 20	0 19	9 18	17	15	4	12	10	8	þ	4	ω	
6	_		/	\square	÷	6	15	16	18	20 1	21 2	22 2	23 2	24 2;	5 24 3	5 24 3	5 24	5 24 3	5 24	24 2;	23 2	22	21 2	20 1	18	2	15	3	11				- の	
				6		7	16	3 17	0 19	21	22	4 23	24	6 25	26 25	26 25	26 25	26 25	26 25	6 25	25 24	4 23	22	21	0 19	4	16	7		6				_
10	-		7		Ř	14	17	15	1 20	22	4 23	25 24	26 2	27 21	3 27	3 27	3 27	3 27	3 27	27 20	26 2	25 24	4 23	22	1 20	Ž	7	14	12		7		- 9	D
		5	+	7	13	15	18	٥ ۱	22 2	4 23	25 2	26	8 27	9 28	29 2	29 28	29 2	29 2	29 2	9 28	8 27	26	25 2	4 23	22 2	÷ 0	œ	15	13	Ē		ы С		S <u>i</u>
÷	-		8	Ļ	-	16	6	4	53	25 2	26	28 27	29 2	30 2	1 30	1 30	1 30	1 30	1 30	30 2	29 2	28 27	26	25 2	23	4	6	16		-	8		÷	E U
N				7	4	~	0	5	24	26	8 27	29 2	1 30	2 31	32 3	32 3	32 3	32 3	32 3	2 31	1 30	29 2	8 27	26	24	5	6	7	4	-			2	õ
-				2	15	-	-	33	25	3 27	29 2	1 30	32 3	33 3	34 33	34 33	34 33	34 33	34 33	33 3	32 3	1 30	29 2	3 27	25	33	-	-	15	12			F	Ř
13	_	6	6	_		18	2	24	7 26	9 26	30	32 3	4 33	35 34	35 3	35 3	35 3	35 3	35 3	35 34	4 33	32 3	30	9 26	7 26	54	2	18		1	6		. 1 3	Ę
			_	Ω	Ę	19	3	25	8	30 2	32 31	t 33	35 3	36	37 36	37 36	37 36	37 36	37 36	36 3	35 3	t 33	32 31	30 2	8 2	25	~	19	16	13				g
4	-		10		17	20	Ň	26	29 2	31	33	35 34	7 36	38 37	38	38	9 38	38	38	38 37	7 36	35 34	33 3	31	29 2	26	Ň	20	44		10		. 4	ï
ю				4	~	-	24	27	30	3 32	34	36	38 3	39 0	40 35	40 35	40 35	40 35	40 35	39 3	38 3	36	34	3 32	30	27	54	1		14			ю	
÷				\vdash	7		25	28	31	34 3;	36 35	8 37	0 39	41 40	2 41	2 41	2 41	2 41	2 41	41 40	0 39	8 37	36 35	34 3;	31	28	55		Ŧ		/		÷	
16	_	7	7	15	19	2	26	29	32	35 3	37 3	39 3	41 4	3 42	43 4	43 4	43 4	43 4	43 4	3 42	41 4	39 3	37 3	35 3	32	29	26	22	19	15	11	~	10	
			\vdash	\square		23	12	30	4 33	36	38	1 40	3 42	44	5 44	5 44	5 44	5 44	5 44	44 43	3 42	1 40	38	36	4 33	30	27	23	0	_			-	
17	-		2	16	Ñ	24	8	31	5 32	8 37	0 36	42 4	44 4	3 45	46 4	46 4	46 4	46 4	46 4	3 45	44 4	42 4	0 36	3 37	5 3/	31	28	24	õ	16	2		17	
~			-		21	5	29	32	36 3	39 3	41 4	43	5 45	47 46	47 47	t8 47	t8 47	47 47	t8 47	47 46	6 45	6 4	41 4	39 3	36 3	32	59	5	21		-		~	
16	1	8		17	0	10	30	33	37	40	3 42	15 44	47 4(9 48	49 4	49 4	49 4	49 4	49 4	9 48	47 4	15 44	3 42	40	37	33	30	2	2	17		∞ -	- #	
ი			13	_		56	31	34	38	41	40	46 4	9 48	50 4	51 5C	51 50	51 50	51 50	51 50	50 4	9 48	46 4	43	41	38	34	3	26	Q	~	13		െ	
-				Ψ	23	27	Se la	35	39	3 42	45 4	3 47	50 49	2 51	52 3	52	52	52	52	2 51	50 49	3 47	45 4	3 42	39	35	32	27	23	₽			F	
20	_		Ţ	6	4	8	33	36	40	44	46	49 48	51	53 5	54 53	54 53	54 53	54 53	54 53	53 5	51	49 48	46	44 4	40	36	ŝ	28	94	6	+		50	
		6	r (Ê		6	34	37	2 41	45	8 47	50	53 52	5 54	55 55	55 5	55 5	55 5	55	5 54	53 52	50	8 47	45	2 41	37	34	6	Q	7	14	თ		
2	-		_	0	25	~	35	38	43 4	46	49 4	2 51	54	56 5	57 56	57 56	57 56	57 56	57 56	56 5	54	2 51	49 4	46	43 4	38	35	2	25	0			21.	
0			15		9	8	36	39	4	8 47	50	53 5	56 55	8 57	9 58	9 58	9 58	9 58	9 58	8 57	56 55	53 5	50	8 47	44	39	e e	30	90		15		01	
Ň				5		ы.	24	40	45	49 4	2 51	5 54	57	59 5	60 5	60 5	60 5	60 5	60 5	59 5	57	5 54	2 51	49 4	45	40	5	31	.,	21			Ň	
g	_	10			27	32	8	4	46	50	53 5	56 59	59 58	1 60	2 61	2 61	2 61	2 61	2 61	1 60	59 58	56 55	53 5	20	46	4	8	32	27			9	20	
			16	22	28	2	93	3 45	3 47	51	54	57	60 £	62 6	63 6	63 6	63 6	63 6	63 6	62 E	60 5	57	54	: 51	3 47	3 45	е 6	33	28	22	16			
$\begin{array}{c c c c c c c c c c c c c c c c c c c $																																		
90 80 70 70 80 80 80 80 80 80 80 90 90 90 90 90 90 90 90 90 90 90 90 90																																		
	Declination																																	

zone number

第3章 The Flexible Image Transport System :*FITS*

3.1 *FITS*とは?

FITS(The <u>F</u>lexible <u>Image Transport System)[8]</u>は天文分野で使われるファイルの代表的フォーマットである。天体スペクトル等のデータ、X線観測のイベントデータ、天文カタログを収めた 表データなどの天文分野で使われる一通りのデータを扱える汎用のフォーマットになっており、 IAU(<u>International Astronomical Union</u> = 国際天文連合)の commission5(天文学データを扱う第 5 委員会)の下の *FITS* ワーキンググループがフォーマット規約の管理をしている。

3.2 *FITS*ファイルの構造

単純な FITS ファイルの構造は ASCII テキストで書かれたヘッダとバイナリの (通常は多次元の) データ配列からできている。現在ではこの「基本」 FITS 要素に加えて同じデータ格納構造 (ヘッ ダ+データ) を持つ拡張された他の FITS 要素が付け加わっても良いことに成っている。

多くの *FITS*ファイルは単純な1つのヘッダと1つのデータ配列を持つファイルだが、もっと複 雑なデータを詰め込むこともでき、実際、最新の大型観測装置が吐き出すデータにはそのような ものも見られるようになっている。

FITS ファイルの構成をもう少し詳しく見ていくと

- (基本) FITS 要素
- (拡張) FITS 要素 1
- (拡張) FITS 要素 2
 - •••

のように (ヘッダ+データ)の FITS 要素が連なっており、いずれの FITS 要素も整数個の (論理) レコードからなる。論理レコードのサイズは 2880 バイト (23040 ビット=あらゆる計算機のワー ド長の最小公倍数) である。

1つの FITS 要素は大きく2つの部分に分けられる。前半部のヘッダはヘッダレコードとも呼ば れ、データの目的、種類、構造、バイト数、レコード数等のデータに関する解説部分となってい る。1 行が 80 文字からなるカードイメージの連なりで整数個の論理レコードに収められる(1 レ コードには 2880/80=36 行が収まるので、ヘッダが 36 行を超えるときは複数個の論理レコードが 必要となる。また、ヘッダが 1 レコードに満たない場合は空白行で埋められちょうど整数個のレ コードとされる。)1つの行のかく乱の使い方や用語には一定の規約がある。後半部のデータ(デー タ配列)はデータレコードとも呼ばれ、ヘッダレコードの直後のレコードから実際のデータが書き 込まれる。

・ヘッダレコード	ヘッダ 1(80 バイト・カードイメージ)
(2880八年下 × n(整数))	$(\sqrt{y}) = 2(80)(\sqrt{1} + \frac{1}{2})(\sqrt{y}) = (\sqrt{y})$
	ヘッダ 4(80 バイト・カードイメージ)
	•••
・データレコード	データ (バイナリ or アスキー)
(2880 バイト	•
× n(整数)	•

表 3.1: FITS の要素の構造

すなわち、1つの FITS 要素の構造は表 3.1 のようになっている。

3.3 *FITS*の誕生とその後の動き

- 1976年11月、NFRA(<u>Netherlands Foundation for Radio Astronomy</u>)のR.Harten と KPNO(<u>Kitt Peak National Observatory</u>)のD.Wells によるデータ交換システムの開発への着手が事の起こり。
- 1979年1月、KPNOで開かれた NSF(<u>National Science Foundation</u>)の会合でデータ交換 用フォーマットについて議論され、議長の P.Boyce(NSF) が NOAO と NRAO へ開発作業の 着手を要請し、R.Burns(NRAO),E.Groth(Princeton),Wells にタスクフォースを結成させ、 Burns は VLA で他のプログラマと共に会合を持った。
- 1979年3月27/28日、HartenとWellsの経験を元に、36時間にわたる議論を経て、E.Greisen(NRAO) とWellsが、Basic FITS Agreement を完成させた。
- 1979 年 5 月、NOAO と NRAO の間で *FITS* によるデータ交換が試みられ、その実用性が確認された。
- 1979年6月、Basic FITSがイタリアのトリエステにおける International Image Processing Workshop で Wells と Greisen により提案された。Harten がこれを是認した。FITS は即座 に受け入れられ、1年を待たずに天文コミュニティにおける国際標準フォーマットの地位を 確立した。
- 1979 年末から 1980 年初めにかけて、Greisen と Harten は小配列のグループを扱う形式を考 案した。
- 1982年 IAU 総会で FITS と共に、Random Groups は、バイナリデータの推奨交換フォーマットとして是認された。
- 1987 年初めに、W.Cotton(NRAO) によりバイナリ表形式が設計された'A3DTABLE' は AIPS(<u>A</u>stronomical <u>Image Processing System</u>)の一部としてリリースされた。
- 1988年、Generalized Extension(一般化拡張部)の規則は、IAUにより是認された。

- 1989年12月22日、IAUのFITS Working Group は浮動小数についての合意に達し、IEEE-754(IEEE1985)の32-bitと64-bit数をFITSでの標準的な浮動小数のタイプとして採用され、1990年1月1日より実施された。
- 1989年末、Green Bankでの単一電波望遠鏡での標準フォーマット開発のための会合で、 D.Wells はバイナリ表形式のフィールドに多次元配列を使用できるようにすることを提案 した。
- 1991年、異なるメディア上でのFITSファイルの物理的な表現が必要とされていたことから、全てのメディアでの一般的な規則と、特に、2ⁿ-byte物理ブロック上でのFITS 論理レコードの書き方に関する提案がWellsとGrosbøl(ESO)によってされ、マイナー変更後1994年春、IAU FITS WG によって是認された。
- 1991年4月、CottonはA3DTABLEを基として、'BINTABLE'と命名された標準バイナリ 表形式の最初の規約案を提示した。
- 1991年7月ころ、W.Pence(GSFC/HEASRC)は文字列の配列を単一の長い文字列と区別する点について疑義を提出し、議論の結果、副配列に関する規約が3番目の付録として付加され、改訂されたBINTABLEの提案はCotton, Tody, Penceにより、1993年5月に公開された。さらに1994年の春にはIAU FITS WG はこの提案の本文をFITSのスタンダードの一部として是認した。
- 1991 年 10 月に多次元フィールドや可変長配列のフォーマットや、それらのためのキーワード やフィールドフォーマットを記述した公式テキストが Cotton と Tody によって公開された。
- 1992 年初めに、J.D.Ponz, J.R.Muñoz(ESA IUE グループ) と R.Thompson(CSC, GSFC IUE グループ) は Image Extension についての詳細なドラフトを作成した。
- 1996年11月、RGOのP.Bunclarkは、FITSのキーワード(DATE-OBSなど)での日付の扱いが、年の部分が2桁しか取っていなかったため、2000年には破綻することを指摘し、それを解決するため、DATE-OBSキーワードの改訂を提案し、1997年11月13日にIAU FITS WGにて正式に是認された。
- 2000 年 10 月 12 日に NOST 100-2.0 は IAU FWG にて正式に標準規約 (FITS スタンダード) と認められた。

第4章 World Coordinates System WCS

4.1 WCSとは?

World Coordinate(世界座標)とは、多次元のパラメータ空間のうち何らかの物理的測量値、例 えばスペクトル中の波長値とか物理空間中の方向を表わす緯度経度とかを提供する座標のことを さし、世界座標と *FITS* ファイル中の N 次元データ配列の各データ値を対応させるためのキーワー ド等を含む規定集が WCS である。

データ配列がデジタル画像を表わしている場合、データ配列と物理画像との変換はピクセルのう ちのどこが (中心かコーナか) データ点か、ということを知る必要がある。コンピュータグラフィッ クスで一般的な、ピクセルの中心は .5 の点に対応する、という慣例とは異なり、FITS ファイル の中のインデックスはピクセルの中心を表わすと天文学者は仮定してきた。FITS ファイルでのピ クセルは通常、物理空間の体積要素と見なされ、変換や回転によっては別の視点から見られる可 能性がある。そのような操作をしたとき、要素の中心だけが不変である。現在は、天文のコミュ ニティではこれに関する標準の規約が決まっていないので、FITS ファイル作成者は適切なコメン トによって、そのファイルがどのような規約に従っているかを読み取りソフトウェアでわかるよ うにすべきとされている。

データ配列の中の順序と、表示されたイメージの中の位置との関係もまた規約の問題である。 WCS Paper I[9] によると、*FITS*ファイル作成者は、最初のピクセルが画像の左下隅であり、続 くピクセルは画像の右方向 (直交座標の x-軸のように) へ並び、それが順次上方向 (y-軸) へと続く ように並べることを推奨している。¹

4.2 WCSの誕生と経緯

D.Wells は 1981 年頃から、天球座標とデータ配列 (天体イメージの x、y とかだけでなく、スペクト ルの波長軸やストークスパラメータのようなものも含めて) の間の対応を表現するためのシステムが 必要であることを認識し、必要なキーワードの提案を行っていた。これが<u>World Coordinate System</u> (WCS) の端緒である。その後、電波天文分野の製約ソフトウェアである AIPS(The <u>A</u>stronomical <u>Image Processing System</u>) の開発に関連して、Greisen はもう少し詳しい規定を提案し、これらは 電波天文分野をはじめ、他の分野 (X や光赤外など) にも波及していった。

WCSが正式に議論されるようになったのは、1988年1月にNRAO(the <u>National Radio Astronomy</u> Observatory) で開催された会合でのことであり、AIPS での規約をもとに、スケーリングや歪みを 取り入れた一般的な WCS の提案がなされた。この会合で提案された表記法のバリエーションが HST(<u>Hubble Space Telescope:ハッブル宇宙望遠鏡</u>) を運用する STScI(<u>Space Telescope Science</u> Institute) や IRAF(The <u>Image Reduction and Analysis Facility</u>) を開発する NOAO(the <u>National</u> Optical <u>Astronomy Observatory</u>) などで取り入れられ発展していった。

¹この規約は、現行の CRCALn などのキーワードを使った座標軸の表現をおきかえるものではない。

1992 年の ADASS(<u>A</u>stronomical <u>D</u>ata <u>A</u>nalysis <u>S</u>oftware and <u>S</u>ystems) ミーティングでの議論 をふまえて、Greisen と Calabretta が 1992 年 12 月に WCS のドラフトを作成し、1993 年 6 月に Berkeley で行われた AAS の会合で提示した。ここでの D. Tody (NOAO) との議論をふまえて改 訂されたバージョンが 1993 年 8 月に配布され、その後、1996 年には Binary Table と歪みを持っ た実イメージの変換法について追加した WCS が提案された。

ここからの数年は標準化の動きにあまり進展がなかったが、1997 年、1998 年の ADASS で引き続き議論され、1999 年には Calabretta と Greisen がその結果を提示した。1999 年の ADASS で WCS の標準化を投票しようとする動きがでたが果たせず、2001 年 6 月 30 日に NOAO の F. Valdes, D. Tody, L. Davis らによる一般化の提案を受けて改訂された 3 つの Paper として提示された。この 3 つの WCS Paper はさらに機器関係の部分を 4 つ目の Paper に分離することとなり、WCS Paper I - III[11] が 2001 年の ADASS で提示された。その後、WCS Paper III(スペクトル関係) にはまだ議論の余地があるということで、WCS Paper I, II についてアメリカの地域委員会で是認され、あと 2 つの地域委員会も通って、最終的に 2002 年 12 月 18 日に IAU FWG(International Astronomical Union *FITS* Working Group) で標準として是認された。

その後、Paper Ⅲ についても改訂が進み、2004 年 10 月の公開コメント募集から半年今日の手 続きを経て、2005 年 8 月 18 日に IAU FWG で正式に認められた

また、Paper II にはその後 spherical projection の一部として HEALPix(Hierarchical Equal Area and isoLatitude Pixelisation) projection が 2006 年 4 月 27 日に取り入れられた。

4.3 基本 *FITS*(原始 *FITS*)での変換

当初の *FITS*(基本 *FITS*) では、データ配列のインデックス (i, j, k,) から物理量である座標値 $(x_i, x_j, x_k,)$ への変換のために以下のキーワードが定義されている。

- CRVALn 参照点での座標値
- CRPIXn 参照点でのインデックス
- CDELTn 参照点での座標値の増分
- CTYPEn 座標軸の種類 (8 文字)
- CROTAn 回転角
- (n は座標軸の番号、単位は、SI系と角度の「度」)

これにより、CROTAn = 0.0の場合、座標値 x_n はインデックスnから次式で計算される。

$$x_n = \operatorname{CRVAL} n + \operatorname{CDELT} n \times (n - \operatorname{CRPIX} n)$$

$$(4.1)$$

これはあまりにも単純であり、もっと一般的な表現方法として WCS が提案された。

4.4 WCSでの基本変換 (WCS Paper I より)

4.4.1 変換の概要

WCSの提案 (WCS Paper I[9]) では、ピクセル座標から世界座標 (World Coordinate) への変換 は、3つのステップを踏んで変換されることになる。ピクセル座標 $(p_j) \rightarrow$ 中間ピクセル座標 $(x_i) \rightarrow$ 世界座標である。このステップの流れ図と簡単な説明は次のようになる。 [ピクセル座標 *p*_{*i*}]

↓ (step1) ← 線形変換する (CRTIXjs, PCi_js or CDi_js キーワード) ↓ 行列をかけ回転、歪み、(オプションで) スケールの補正 [中間ピクセル座標 q_i]

↓ (step2) ← 物理単位へ再スケーリングする (CDELTis キーワード)

 \downarrow

[中間世界座標 x_i]

```
↓ (step3) ← 座標変換 (CTYPEis, CRVALis, PVi_ms キーワード)
```

```
」 球面から平面への射影と、実世界座標への変換
```

[世界座標 (World Coordinate)]

ここで最初のステップ (step1) は、ピクセル座標から中間ピクセル座標への線形変換である。 このためにはピクセル座標ベクトル *p_i* に対して以下の行列をかける。

$$q_i = \sum_{j=1}^n m_{ij}(p_j - r_j)$$
(4.2)

ここで、 r_j は CRPIX *j* で与えられる参照点でのピクセル座標であり、 m_{ij} が変換行列、 q_i が 中間ピクセル座標である。これ以降、添字の *j* はピクセル軸を、*i* は世界軸を表わす。 m_{ij} は $n \times n$ の正方行列であり、NはNAXIS キーワードで与えられる。ただし、この点は WCSAXES キーワードによってより一般化される。変換結果の q_i は、中間世界座標軸と一致する方向の 中間ピクセル座標軸ベクトルであり、無次元のピクセル単位での参照点からのオフセットで ある。

次に2番目のステップ (step2) では、q_i を対応する中間世界座標の x_i に変換するために、単 に次のようなスケーリングをすればよい。

$$x_i = s_i q_i \tag{4.3}$$

 m_{ij} や s_i などを FITS ヘッダーでどう表わすかは後で触れる。

最後に3番目のステップ (step3) は中間世界座標から世界座標への変換である。具体的には、 球面から平面への射影法と平面と天球面の接点での世界座標の値から決まる変換により実際 の世界座標に変換する。この変換は CTYPE*i* に依存する。単純な線形軸では、*x_i* は CRVAL*i* で与えられる参照点における座標値に加えるオフセットと解釈される。それ以外の場合には、 CTYPE*i* は *x_i*、CRVAL*i* と他のパラメータの関数を規約に従って定義することになる。規 約にない CTYPE*i* は線形と解釈される。非線形座標は CTYPE*i* に"4-3"形式で記述され る。これは例えば、VOPT-F2W、のようなもので、最初の4文字が座標の種類を表し、5番 目の文字は、-、で、残りの3文字が中間世界座標から世界座標に変換するアルゴリズムを 指定する。座標の種類が4文字に満たない場合は'-'で補い、アルゴリズムが3文字に満 たない場合は空白を補う。例えば'RA---UV'のように表す。ただし、アルゴリズムのコー ドは3文字にすることを推奨する。

4.4.2 変換行列の表し方

上記のステップ2の変換行列 m_{ij} には PC i_j と、CD i_j の2つのキーワードのどちらかが使われる。²

 $CDi_j \ge PCi_j$ のデフォルトの振る舞いは異なり、もし1枚以上の CDi_j カードが存在すれ ば、存在しない CDi_j は0 とみなされる。一方、 CDi_j が存在しない場合は、 PCi_j カードが存 在しなくても、 PCi_j 形式が仮定される (これは Wells らによる FITS の原論文 (FITS Paper I) の CDELTiの解釈と一致する)。 $PCi_j \ge CDi_j$ の混在は禁止される。このため、 CDi_j 形式と PCi_j 形式の変換は単純にキーワードの置き換えで行われ、 CDi_j は PCi_j と同値で CDELTi は 1 と見なされ、逆の場合は PCi_j と CDELTiから CDi_j が計算される。

1. PCi_j 形式での記述

 PCi_j 形式では、変換行列の要素 m_{ij} はヘッダーの PCi_j (浮動小数) で表され、 s_i は CDELTi で表される。 $i \approx j$ は 1 から始まり (例えば PC1_1 とか CDELT1)、デフォルトの PC i_j の 値は i = j に対して 1.0 それ以外は 0.0 である。PC i_j 行列は正則行列で逆行列を持たなけ ればならず、CDELTi は 0 であってはならない。

$$\begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ \vdots \end{pmatrix} = \begin{pmatrix} \text{PC1_1} & \text{PC1_2} & \text{PC1_3} & \dots \\ \text{PC2_1} & \text{PC2_2} & \text{PC2_3} & \dots \\ \text{PC3_1} & \text{PC3_2} & \text{PC3_3} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} p_1 - r_1 \\ p_2 - r_2 \\ p_3 - r_3 \\ \vdots \end{pmatrix}$$

中間世界座標の x_i は前述の通り $x_i = s_i q_i = \text{CDELT} i q_i$ で計算される。

2. CDi_j 形式での記述

CDi_j形式では、式(4.2)と(4.3)は一緒になって、

$$x_{i} = \sum_{j=1}^{n} (s_{i}m_{ij})(p_{j} - r_{j})$$
(4.4)

となり、積 $s_i m_{ij}$ が CD i_j (浮動小数) で表される。 $i \approx j$ は 1 から始まる (例えば CD1_1)、 CD i_j 行列は正則行列で逆行列を持たなければならない。

$\langle x_1 \rangle$		$(CD1_1)$	$CD1_2$	$CD1_3$)	$(p_1 - r_1)$
x_2		$CD2_1$	$CD2_2$	$CD2_{-}3$		$p_2 - r_2$
x_3	=	$CD3_1$	$CD3_2$	$CD3_3$		$p_3 - r_3$
(:/			:		·)	$\left(\begin{array}{c} \vdots \end{array} \right)$

 $^{^2}$ これは WCS の Paper をまとめる過程で、CDELT*i* とキーワード PC で表される PC 行列で記述する案に対して、HST と IRAF ではキーワード CD で表される CD 行列が既に使われていたことから、両方を併記することになったのである。

4.5 天球座標の変換(WCS Paper Ⅱより)

前節で座標の変換手順について概説したが、実際の変換にあたっては step3 で扱うのが天球座標の場合と分光座標の場合が想定される。このために WCS Paper II[10] と WCS Paper III が分離され、先に天球座標に関する WCS Paper II が正式に認められ、議論の収束を待って、後から分光座標についての WCS Paper III が正式に認められた。

step1,step2で、ピクセル座標から中間世界座標までの変換 $(p_j \xrightarrow{r_j, m_{ij}, s_i} (x, y))$ をした後、この中間 世界座標から天球座標への変換 (step3) を 2 つのサブステップに分割する。この 2 つのサブステップ は平面から球面への変換と球面回転に対応しており、 $(x, y) \xrightarrow{(\phi_0, \theta_0)} (\phi(経度), \theta(緯度)) \xrightarrow{(\alpha_0, \delta_0), \phi_p, \theta_p} (\alpha($ 天球経度), $\delta($ 天球緯度)) という変換をすることになる。

[step3 の部分の詳細図]

[中間世界座標]

(射影平面座標 (x, y))

- ↓ ← 座標変換 (CTYPEia, PVi_ma キーワード)
- ↓ 平面から球面への射影

(局所 (native) 球面座標 (ϕ , θ))

- ↓ ← 球面回転 (CRVALia, LONPOLEa, LATPOLEa キーワード)
- ↓ 3つの Euler 角を決定して回転

(天球座標 (α, δ))

[世界座標 (World Coordinate)]

2つのサブステップでの変換はやや複雑であり、本節では通常の観測から得られたデータを *FITS* ファイルに書くような場合を想定して、天球座標 (例えば、 α , δ) からピクセル座標 (*i*,*j*) への変換 手順を概説する (§4.5.1)。次にその逆に、WCS を使った *FITS* ファイルのデータのピクセル座標 (*i*,*j*) から天球座標 (α , δ) を求めるための *FITS* パラメータについて述べる (§4.5.2)。

4.5.1 天球座標 (α, δ) から射影平面座標 (x, y) への変換

ある天域を観測したとき、天球座標 (α, δ) がどのようにピクセル座標 (i, j) に変換されるかを順 を追って考えると次のようになる。

1) 天球座標 (α, δ) から局所球面座標 (*native coordinate*) (ϕ, θ) への変換

2) 局所球面座標から射影平面座標への変換 (射影) $(\phi, \theta) \Rightarrow (R_{\theta}, \phi)$ or 一般に (x, y)

3) (x, y)から回転や歪みの変換などをへてピクセル座標 (i, j) へ

3) については前節に詳しく述べてあり、ここでは1)と2) について詳しく述べる。

1) 天球座標 (α, δ) から局所球面座標 (ϕ, θ) への変換

天球座標から局所 (native) 球面座標への変換として、天球面からこれに接する平面への射影を 行う。天球面と平面の接点の天球座標を (α_P, δ_P) とし、天球上でこの点を極とする新たな座標系 を設定する。天球上のある点 (α, δ) が新しい座標系で (ϕ, θ) (ϕ は経度、 θ は緯度) になるとすると、 次式が成り立つ。

$$\sin \theta = \sin \delta \sin \delta_P + \cos \delta \cos \delta_P \cos(\alpha - \alpha_P)$$

$$\cos \theta \sin(\phi - \phi_P) = -\cos \delta \sin(\alpha - \alpha_P)$$

$$\cos \theta \cos(\phi - \phi_P) = \sin \delta \cos \delta_P - \cos \delta \sin \delta_P \cos(\alpha - \alpha_P)$$
(4.5)

ここで φ_P は、元の座標系での極点の、新しい座標系における経度である。

2) 局所球面座標 (ϕ , θ) から射影平面座標 (R_{θ} , ϕ) or (x, y) への変換

図 4.1 で示しているのは、局所球面座標の (x, y) 座標に対する位置関係を 2 つの典型的なケース (平面が極で接する場合と赤道面 (の基準経度) で接する場合) である。

射影平面上に球面との接点を中心をする極座標 (R_{θ}, A_{ϕ}) を設定する。 A_{ϕ} は軸を適当に取ること により $\phi = A_{\phi}$ とすることが出来るので、 $\theta \ge R_{\theta}$ の関係として射影を記述できることとなる。

ここでは接平面への射影 (Zenithal projection) の一般形について述べる。

一般形 (AZP) では、射影の投影中心は球面と投影面の接点と球面の中心を通る直線上にあり、 その球面中心から距離を μ とすると、 R_{θ} と θ の関係は、

$$R_{\theta} = \frac{180^{\circ}}{\pi} \frac{(\mu+1)\cos\theta}{\mu+\sin\theta}$$
(4.6)

となる。投影中心の位置 (μ の値) により射影の性質が完全に決まる。 ここまでに出てきた変数を (関係する *FITS* キーワードを含めて) 表 4.1 にまとめておく。

4.5.2 射影平面座標 (*x*, *y*) から天球座標 (*α*, *δ*) への変換

各変換を記述するために特に重要な FITS パラメータは次の通りである。

- 追加 LONPOLEa 天球座標の極の局所球面座標での経度 ϕ_P
- 追加 LATPOLEa 天球座標の極の局所球面座標での緯度 θ_P
- 追加 PVi_ma 局所球面座標各軸のパラメータ
- 定義拡張 CTYPEia 局所球面座標各軸のタイプ

定義拡張 CRVALia 局所球面座標各軸についての参照点の物理値



図 4.1: 参照点を極とした局所 (native) 球面座標 (左、(ϕ_0, θ_0) = (0,90°))と、参照点を赤道と基準経度の交点とした局所座標 (右、(ϕ_0, θ_0) = (0,0))



図 4.2: zenithal 投影の R_{θ}, θ, μ の関係図 (左)、と3つの特別なケース (右)

変数	意味	関係する FITS キーワード
i	世界座標のインデックス	
j	ピクセル座標のインデックス	
a	代替記述コード (空白か A - Z)	
p_j	ピクセル座標	
r_j	参照ピクセル座標	CRPIX ja
m_{ij}	線形変換行列	CDi_ja or PCi_ja
s_i	座標スケール	CDELTia
x_i	中間世界座標 (一般形)	
(x,y)	射影平面座標	
$(\phi, heta)$	局所 (native) 経度、緯度	
$(lpha,\delta)$	天球経度、緯度	
$(\phi_0, heta_0)$	接点の局所 (native) 経度、緯度	PVi_1a, PVi_2a
$(lpha_0,\delta_0)$	接点の天球経度、緯度	CRVALia
$(\phi_P, heta_P)$	天球の極の局所 (native) 経度、緯度	$LONPOLEa(=PVi_3a), LATPOLEa(=PVi_4a)$
(α_P, δ_P)	局所 (native) 極の天球経度、緯度 ($\delta_p = \theta_p$)	
arg()	正確な象限を返す tangent の逆関数	

表 4.1: これまでに出てきた変数

これら新パラメータおよび旧来のパラメータを用いて、前述の2つのサブステップに従って、射 影平面座標を実際の天球座標に変換する。

1. $(x, y) \rightarrow (\alpha, \delta)$

もし座標軸が線形なら、真の座標は単に CRVALia によって与えられる参照点からのオフ セットを加えるだけで良い。そうでない場合は、オフセット量と CRVALia、それに他のパ ラメータを使って真の座標値を決める関数の規約について合意が必要となる。

ここでは、天文学的な極座標のペア (天球上の経度と緯度) と様々なシステム (様々な球面 投影法で表現されるもの) との変換の規約について扱うことにする。この場合には線形座標 でのオフセット値 (x, y) を特定の球面投影法を使った局所 (native) 球面座標 (ϕ, θ) に変換す る計算をすることになる。球面投影のタイプは CTYPEia キーワードの6から8桁目で特定 され、座標のペアの両方の軸に対して同じでなければならない。

例えば、投影面が平面の代表的な投影法である zenithal(または azimuthal) 投影の場合に はパラメータ μ を指定するために新しいキーワード PV*i*_ma と投影タイプとして AZP を使 う。特に $\mu = 0$ の場合は投影タイプは TAN、 $\mu = \infty$ の場合の拡張された投影タイプは SIN となる。

これらの関係式を使うと、

 $(x, y) \to (R_{\theta}, \phi) \to (\phi, \theta)$

のように、局所 (native) 球面座標が計算できる。

最後にこうして得られた球面座標のペア (ϕ, θ) を球面上で回転させて天球座標に変換すれば良い。天球座標のタイプは CTYPEia キーワードの最初の4桁で表され、AIPS の慣例か

[RADECSYSa が存在する場合]				[RADECSYSa が存在せず		
RADECSYSa 値	equinox EQUINOXa, EPOCH			EQUINOX が存在する場合]		
		両者がない時の分点		EQUINOX の値	RADECSYSa として	
'FK4' or	Besselian	1950.0			想定される値	
'FK4-NO-E'				<1984.0	'FK4'	
'FK5'	Julian	2000.0		$1984.0 \ge$	'FK5'	

表 4.2: RADECSYSa が存在する場合 (左) としない場合 (右)

ら赤道座標形では 'RA--' と 'DEC-' (赤経・赤緯)を使い、その他の天球座標では 'xLON' と 'xLAT' を使う。例えば銀河座標形では x=G として、'GLON' と 'GLAT' (銀経・銀緯) とする。他にも黄道座標は x=E、日心座標は x=H、超銀河座標系は x=S が決まっている が、惑星や月などを表す場合には 'yzLN' と 'yzLT' を使う記法も許される。CRVAL*ia* キー ワードは局所 (native) 球面座標での参照点 (上の zenithal 投影の場合は北極点、すなわち $(\phi, \theta) = (0, 90^{\circ})$ の点)の天球座標での座標値を表す。球面上での回転を完全に表すための3 番目の角度パラメータ ϕ_P は、新キーワード LONPOLE*a* で記述し、デフォルトでは0°ま たは 180°である。これらから式 (4.5) により必要な変換が得られる。

これらによると原始 *FITS* で定義されていた CROTA*i* キーワードは必要でなくなるが、古 いキーワードを使ったファイルは新しいキーワードで表現しなおすことが出来る。

2. 座標の準拠フレーム

赤道座標などの場合 equinox や基本座標システムを与えないと厳密な定義が出来ないが、 このうち基本座標システムを表す新しいキーワードとして RADESYSa が提案されており、 次のような値を持つ。

<u>RADESYSa</u> 定義

'ICRS'	平均位置、International Celestial Reference System
'FK5'	平均位置、新しい (IAU 1984 以後) システム
'FK4'	平均位置、古い (Bessel-Newcomb) システム
'FK4-NO-E'	平均位置、古いシステム、ただし e-terms なし
'GAPPT'	<u>G</u> eocentric <u>APP</u> aren <u>T</u> place, IAU 1984 以後のシステム

(以前提案されていた RADECSYS と同じ意味だがキーワード名が若干違うことに注意)

EQUINOXa キーワード (浮動小数) も使うことが出来る (EPOCH は今後は使わない)。 EQUINOXa キーワードが存在する場合には RADECSYSa キーワードも伴うべきであるが、 もし RADECSYSa が伴わない場合は表 4.2 の右側の表のように解釈される。

RADECSYSa も EQUINOX もない場合は 'ICRS' がデフォルトとなる。

正確な観測時刻が必要な場合には、時刻の記述を統一するために連続的で扱いやすい MJD-OBS キーワード (浮動小数値で DATE-OBS に対応する <u>Modified Julian Date</u> (JD-2400000.5) を表す) を使うことも提案されている。

4.6 さまざまな投影法

4.6.1 ZEA:Zenithal equal-area 投影法

ZEA とはランベルト正積方位図法とも呼ばれる地図投影法の一つ。極を基準点 (中心) とした場合、経線は中心から放射状に、緯線は基準点を中心とする同心円上に描かれる (図 4.3)。緯線の間隔は特に図の外側で狭くなっている。中心付近の歪みは比較的小さい。今回は、横軸法 (赤道方向から見た図)を用いて画像を作成した。この投影法の式を以下に示す。

$$R_{\theta} = \frac{180^{\circ}}{\pi} \sqrt{2(1 - \sin\theta)} \tag{4.7}$$

$$=\frac{360^{\circ}}{\pi}\left(\frac{90^{\circ}-\theta}{2}\right) \tag{4.8}$$

$$\theta = 90^{\circ} - 2\sin^{-1}\left(\frac{\pi R_{\theta}}{360^{\circ}}\right) \tag{4.9}$$

4.6.2 CAR:Plate Carrée 投影法

経緯線が直角、等間隔である正距円筒図法の一つで、標準緯度を0度に置いたものである。こ の図法は緯度、経度をそれぞれ図の縦と横にそのまま読み替えたもので、標準緯線上と縦方向に 関しては正距である。標準緯線から離れるほど横方向の長さが拡大されるため、角度は正しくな い。定義式を以下に示す。

$$x = \phi(\& \mathcal{B}) \tag{4.10}$$

この投影法のイメージは、図 4.4 である。

4.6.3 AIT:Hammer-Aitoff 投影法

ランベルト正積方位図法の横軸図を変形して擬円筒図法のような形状にしたものである。角の 歪みが非常に大きくなっている。Aitoff が ARC(Zenithal equidistan 投影法)を元に考えだしたも のに、Hammer が修正を加えたものである。この投影法の定義式を以下に示す。

$$x = 2\gamma\cos\theta\sin\frac{\phi}{2} \tag{4.12}$$

$$y = \gamma \sin \theta \tag{4.13}$$

$$\gamma = \frac{180^{\circ}}{\pi} \sqrt{\frac{2}{1 + \cos\theta\cos\left(\frac{\phi}{2}\right)}} \tag{4.14}$$



図 4.3: Zenithal equal-area 投影法



図 4.4: Plate Carrée 投影法

$$\phi = 2\arg\left(2Z^2 - 1, \frac{\pi}{180^{\circ}}\frac{Z}{2}x\right)$$
(4.15)

$$\theta = \sin^{-1} \left(\frac{\pi}{180^{\circ}} yZ \right) \tag{4.16}$$

$$Z = \sqrt{1 - \left(\frac{\pi}{180^{\circ}} \frac{x}{4}\right)^2 - \left(\frac{\pi}{180^{\circ}} \frac{y}{2}\right)^2}$$
(4.17)

$$=\sqrt{\frac{1}{2}\left(1+\cos\theta\cos\frac{\phi}{2}\right)}\tag{4.18}$$

4.6.4 PCO:Polyconic 投影法

正規多円錐図法とも呼ばれ、擬円錐図法の一種である。緯線は中央経線の延長線上に中心を持 つ異心円群として表される。経線は緯線を等間隔に区切った点を結んだ曲線になる。緯線の長さ は正しくなるが、経線の長さは中央経線から離れるにつれて引き延ばされていく。各緯線上と中 央経線上の距離は正しくなっている。

この投影法の定義式を以下に示す。

$$x = \frac{180^{\circ}}{\pi} \cot\theta \sin(\phi \sin\theta) \tag{4.19}$$

$$y = \theta + \frac{180^{\circ}}{\pi} \cot \theta [1 - \cos(\phi \sin \theta)]$$
(4.20)



図 4.5: Hammer-Aitoff 投影法

$$x^{2} - \frac{360^{\circ}}{\pi}(y - \theta)\cot\theta + (y - \theta)^{2} = 0$$
(4.21)

$$\phi = \frac{1}{\sin \theta} \arg\left(\frac{180^{\circ}}{\pi} - (y - \theta) \tan \theta, x \tan \theta\right)$$
(4.22)

4.6.5 CSC:COBE(<u>CO</u>smic <u>B</u>ackground <u>E</u>xplorer) Quadrilateralized Spherical Cube 投影法

この投影法は、図 4.6 のように極方向の図とそれ以外 (赤道方向) とに分けられており、この展 開図を組み立てると、ちょうど立方体の内側に全天画像を貼付けたようになる。この投影法は、プ ロジェクタが中央に1台と、四方に1台ずつあるプラネタリウムで使用される。

定義式を以下に示す。

$$x = \phi_c + 45^\circ F(\chi, \psi)$$
 (4.23)

$$y = \theta_c + 45^\circ F(\psi, \chi) \tag{4.24}$$

$$F(\chi,\psi) = \chi\gamma^* + \chi^3(1-\gamma^* + \chi\psi^2(1-\chi^2) \left[\Gamma + (M-\Gamma)\chi^2 + (1-\psi^2)\sum_{i=0}^{\infty}\sum_{j=0}^{\infty}C_{ij}\chi^{2i}\psi^{2j}\right] + \chi^3(1-\chi^2) \left[\Omega_1 - (1-\chi^2)\sum_{i=0}^{\infty}D_i\chi^{2i}\right]$$
(4.25)



図 4.6: COBE Quadriateralized Spherical Cube 投影法

第5章 開発

本章ではプラネタリウムの開発について、データの取得から、WCSLIB を用いた全天への投影 プログラムの開発、画像処理、さらにカラー合成して最終的な画像が得られるまでを述べる。

5.1 開発に使用するソフトウェア

5.1.1 CFITSIO

CFITSIO¹は C 言語で書かれた FITSIO のことで、*FITS*ファイルを読み書きするためのサブ ルーチン・ライブラリである。これを用いると、*FITS*ファイルについてあまり詳しくなくても、 データにアクセスすることが可能となる。

CFITSIO は SunOS, Solaris, OSF/1, HP-UX, VAX/VMS, IRIX, Linux, MS-DOS, Windows95, MacOS など多くの OS 上での動作が確認されている。また、かつての FITSIO を利用したプログ ラムも利用できるように FORTRAN 用のラッパーも提供されている。

CFITSIO のソースコードは HEASARC(<u>H</u>igh <u>Energy</u> <u>A</u>strophysics <u>S</u>cience <u>A</u>rchive <u>R</u>esearch <u>C</u>enter) で公開されており、以下の ftp サイトから入手することが出来る。(v3.060 が 2007 年 12 月 14 日時点で最新版)

ftp://heasarc.gsfc.nasa.gov/software/fitsio/c/cfitsio3060.tar.gz : (UNIX 版)

cfitsioの詳しい仕様は、C ならば

ftp://heasarc.gsfc.nasa.gov/software/fitsio/c/cfitsio.ps

Fortran ならば

ftp://heasarc.gsfc.nasa.gov/software/fitsio/c/fitsio.ps

を参照すれば良い。なお、インストールの詳細を A.3(48 ページ) に示した。

5.1.2 WCSLIB

WCSLIB は ANTF(<u>A</u>ustralia <u>T</u>elescope <u>N</u>ational <u>F</u>acility) の M. Calabretta によりリリースさ れた WCS を扱うライブラリで、WCS の Paper II で提案されている座標変換をインプリメントし たルーチン集である。

WCSLIB のソースコードは ANTF で公開されており、以下の ftp サイトから入手することが出 来る。(2007 年 12 月 27 日時点で WCSLIB 4.3 が最新版)

¹http://chunjiao.astro.ncu.edu.tw/ daisuke/ja/Research/Astronomy/ImageProcessing/cfitsio/

ftp://ftp.atnf.csiro.au/pub/software/wcslib/wcslib.tar.gz

また、WCSLIB の詳細については以下のサイトを参照すると良い。 http://www.atnf.csiro.au/people/mcalabre/WCS/

5.1.3 STIFF

STIFF は専門的な *FITS* イメージをよりポピュラーな TIFF フォーマットに変換するプログラ ムである。入力するイメージが1つだとグレースケールを出力し、3つだと赤、緑、青のカラーイ メージを出力する。

STIFF は TERAPIX のサイト²よりダウンロードすることが出来る。

実際に動かす場合は以下のようにコマンド入力する。

入力イメージが1つの場合:

% stiff 入力画像 -c configuration-file [-Parameter1 Value1] [Parameter2 Value2]

入力イメージが3つの場合:

% stiff 入力画像 1(赤) 入力画像 2(緑) 入力画像 3(青) -c configuration-file [-Parameter1 Value1] [-Parameter2 Value2]

Configuration parameter のリストは表 A.2(51 ページ) に示した。 また、A.5 に記されている方法を使えば、以下のようにコマンド入力することも可能となる。

% stiff 入力画像 1(赤) 入力画像 2(緑) 入力画像 3(青) -OUTFILE_NAME 出力ファイル名.tif

5.1.4 farith

HEASARC にて公開されている HEAsoft の中の fits を扱うツール群である FTOOL の一つで、 2つの *FITS* ファイルを読み込み四則演算を行うプログラムである。

% farith infile1 infile2 outfile operation

このプログラムを使うには HEAsoft をインストールする必要がある。HEAsoft のダウンロード や詳細については以下のサイトを参照すると良い。

http://heasarc.gsfc.nasa.gov/lheasoft/

インストールの詳細については、「すざく」ファーストステップガイド第 3.03 版³を参照すると わかりやすい。

 $^{^{2}} http://terapix.iap.fr/rubrique.php?id_rubrique=178$

 $^{^{3}} http://cosmic.riken.jp/suzaku/help/guide/fstep/suzaku_fstep070803.pdf$

5.2 *ROSAT*全天サーベイデータの入手

全天サーベイデータはすべて世界中に公開 (アーカイブ) されており、マックスプランク研究所 の以下の ftp サイトより取得することが出来る。

ftp.xray.mpe.mpg.de/rosat/archive/rra/ddp/s/

ここには、9301~9333 までのフォルダがあり、それぞれ図 2.5 に対応したディレクトリが存在 する。

ここで、ftp.xray.mpe.mpg.de/rosat/archive/rra/ddp/s/9310/931049p/のフォルダを例に挙げる。このフォルダの中には以下のファイルが含まれている。

931049p-fp1.fits.gz 931049p-fp2.fits.gz 931049p-fp3.fits.gz 931049p-fsr.fits.gz 931049p-fsrd.fits.gz 931049p-p10.fits.gz 931049p-p20.fits.gz 931049p-p30.fits.gz

今回使用するファイルは、

931049p-fp1.fits.gz(図 5.1) 931049p-fp2.fits.gz(図 5.2) 931049p-fp3.fits.gz(図 5.3)

の3つであり、fp1, fp2, fp3がそれぞれ0.1-0.4keV, 0.4-0.9keV, 0.9-2.4keV に対応している。

5.3 各 fits データの構造

図 5.1~図 5.3 に、931049p-fp(1-3).fits に保存されている画像データを示す。これらの画像は §2.3 で述べたように各画像は 6.4°×6.4°の視野を持ち、サイズは 512×512 である。

以下に本論文で扱う RASS データの fits ヘッダーの重要なキーワードがどのように設定されて いるか説明をする。ヘッダー全体の例として 930101p-fp1.fits のヘッダーを §A.6.1(51 ページ) に のせている。

- **NAXIS1, NAXIS2** この画像のサイズを示す。使用する RASS データでは 512 × 512 であるこ とが読み取れる。
- RADESYS 基本座標システムを表し、FK5が用いられている。FK5はIAU 1984 以降の新しい システム
- **CRPIX1, CRPIX2** 参照ピクセル座標。それぞれ値が 256.5 になっているので、NAXIS より画 像データの中心が参照ピクセル座標となっていることが読み取れる。



図 5.1: 931049p-fp1



図 5.2: 931049p-fp2



図 5.3: 931049p-fp3

CRVAL1, CRVAL2 接点の天球経度、緯度。つまり天球座標においてのこの画像データの位置 を示す。

CTYPE1, CTYPE2 局所球面座標各軸のタイプ。

もとの RASS データでは CTYPE1, CTYPE2 がともに EQU--CAR と書かれているが、こ れは WCS 規約に準拠しておらず、wcslib は正常に動作しない。そこで、値を規約に準拠 した値に書き換える必要があるため、開発に先立って CTYPE1, CTYPE2 の値をそれぞれ RA---CAR, DEC--CAR と書き換えた。fits キーワードの値を書き換えるには farith 同様 に FTOOL の一つで fits キーワードを書き換えるツールの fparkey を用いる。

5.4 全天画像 fits ファイルの作成

5.4.1 個々の fits ファイルの全天への投影

個々の fits ファイルを AIT などの投影法で全天に射影するプログラムを作成する。プログラム の第 0 バージョンは宇宙科学研究本部の海老沢教授により開発され、海老沢教授のホームページ に公開されている⁴。このプログラムは、入力パラメータとして、§5.2 で取得した全天サーベイの ファイルリスト、投影法 (AIT, CAR, CSC, PCO, ZEA の 5 つをサポート)、X 方向および Y 方向 の出力画像ピクセル数 (解像度)、参照ピクセル座標 (X, Y)、ピクセルサイズ (degree/pix)、出力 ファイル名を与え、全天画像を得るものである。本研究ではこのプログラムをもとに画像処理を 施した。

⁴http://plain.isas.jaxa.jp/~ebisawa/Planetarium/

投影法	解像度	参照ピクセル座標 (crpix1, crpix2)
AIT	3240×1620	(1620.5, 810.5)
CAR	3600×1800	(1800.5, 900.5)
PCO	3600×2760	(1800.5, 1380.5)
ZEA	2290×2290	(1145.5, 1145.5)
CSC	3600×2700	(3150.5, 1350.5)

表 5.1: 入力パラメータ

プログラムの全文は A.6.2 に示す。このプログラムでは、まず作成したいサイズに従って出力画 像ファイルを用意し、必要な WCS キーワードを書き込む。AIT 投影の場合の作成した fits ヘッダー を図 5.4 に示す。AIT の例では画像サイズ (NAXIS1 × NAXIS2) を 3240 × 1620、参照ピクセル座 標 (CRPIX1, CRPIX2) を (1620.5, 810.5)、座標スケール (CDELT1, CDELT2) を 0.1degree/pix としたことがわかる。また基本座標系 (RADSYS)、春分点 (EQUINOX) はそれぞれ FK5, j2000 を用いている。他の投影法についても基本座標系、春分点、座標スケールは同じとし、画像サイ ズと参照ピクセル座標は表 5.1 のようにした。次にリストされた 1378 個のファイルから個々の画 像 (サイズは 512 × 512)を開き、fits ヘッダーを確認し、512 × 512 の各ピクセルに格納されたピ クセル値、即ちそのピクセルの X 線カウント数を配列に書き込む。これを WCSLIB を用いて世界 座標に変換し、さらにその世界座標を希望の投影形式にしたがってピクセル座標に投影している。

AIT 投影法についてプログラムを実行し、得られた全天画像を図 5.5 に示す。

SIMPLE = T / file does conform to FITS standard
BITPIX = -32 / number of bits per data pixel
NAXIS = 2 / number of data axes
NAXIS1 = 3240 / length of data axis 1
NAXIS2 = 1620 / length of data axis 2
EXTEND = T / FITS dataset may contain extensions
COMMENT FITS (Flexible Image Transport System) format is defined in
'Astronomy
COMMENT and Astrophysics', volume 376, page 359; bibcode: 2001A&A
376359H
RADESYS = 'FK5 ' / Coordinate system
EQUINOX = 2.000E+03 / Equinox
CTYPE1 = 'RAAIT' / DEC projection
CTYPE2 = 'DECAIT' / DEC projection
CRPIX1 = 1.620500000000E+03 / X reference pixel
CRPIX2 = 8.105000000000E+02 / Y reference pixel
CRVAL1 = 0.0000000000000E+00 / RA of reference pixel
CRVAL2 = 0.0000000000000E+00 / DEC of reference pixel
CDELT1 = -1.0000000000000E-01 / X pixel increment (degree)
CDELT2 = 1.000000000000E-01 / Y pixel increment (degree)
END

図 5.4: AIT(低エネルギー)のヘッダー



図 5.5: 1378 枚の画像をつなぎ合わせた画像 (AIT)

5.4.2 画像データの重なりの削除

図 5.5 にも見られるように §5.4.1 で作成したプログラムだけでは、画像データの重なりが生じてしまう。これを取り除くためのプログラムが必要となってくる。

考え方としては、重なっていない部分のピクセル値に対して、画像が2枚重なっている部分のピ クセル値は2倍、4枚重なっている部分のピクセル値は4倍となっていると考えられる。したがっ て余分に加算されたカウント数を規格化するためには、もとのfitsファイルの各ピクセルに保存 されたピクセル値(X線のカウントレート)にかわって、定数1を入力し、それを全天の各座標の ピクセル値として出力させてやれば良い。こうすることで、データの重なりのない部分には1が、 2枚画像が重なった部分には2が、3枚、4枚にはそれぞれ3と4がピクセル値として出力される。 その後に元の画像データ(図 5.6)からこの規格化用の画像データを割ると全体のピクセル値が均 ーとなり、重なりが取り除かれた画像になる。

入力を1にするには A.6.2 のプログラムで、各データに格納されたピクセル値を世界座標に変換してそのピクセル値を置換している箇所、具体的にはプログラムの後半にある下線部の部分を 以下のように変更し、ピクセル値を入れるかわりに規格化の定数1を入力してやればよい。

 $inimage[i*512+j] \implies 1$

変更後のプログラムを実行した結果、図 5.6 の規格化用の画像が得られた。

§5.1.4 で説明した farith を用いて、元の画像 (図 5.5) を個々で作成した規格化用画像 (図 5.6) で 割ることで、重なりを除去した画像が得られる。低エネルギー、中間エネルギー、高エネルギー について得られた画像をそれぞれ図 5.7、図 5.8、図 5.9 に示す。



図 5.6: 規格化用に作成した画像 (AIT)



図 5.7: 重なりを取り除いた画像 (低エネルギー)(AIT)



図 5.8: 中間エネルギーの画像 (AIT)



```
図 5.9: 高エネルギーの画像 (AIT)
```

5.5 プラネタリウム投影用カラー画像の作成

5.5.1 プラネタリウム投影用カラー画像を作る上での注意点

プラネタリウムに投影する上で以下のことに留意しなければならない。

- 画像の周りの余分な部分(黒い部分)を出来るだけ少なくする。
- 解像度が縦横共に2の階乗であることが望ましい

これを実現するには §5.4.1 や §5.4.2 でプログラムを動かす際の解像度の入力値を変えれば良い。 しかし、入力値を初めから 2 の階乗としてもうまく余分な部分がなくならない。そこで、とりあ えず余分な部分を取り除くことが出来る数値を入力し、その後解像度を変更できるフリーソフト ウェア (A.6.4:60ページ)を用いて解像度を変更することにした。以下に示すのが、それぞれの投 影法で余分な部分をできるだけ取り除いた画像のプログラム実行時の解像度と変更後の解像度で ある。

AIT $3240 \times 1620 \implies 4096 \times 2048$ CAR $3600 \times 1800 \implies 4096 \times 2048$ PCO $3600 \times 2760 \implies 4096 \times 3140$ ZEA $2290 \times 2290 \implies 4096 \times 4096$ CSC $3600 \times 2700 \implies 4096 \times 3072$

PCO と CSC については変更後の解像度が2の階乗ではないことがわかるが、これは PCO と CSC の元々の形の縦と横の比率が2の階乗倍ではないためである。そこで便宜的に横の解像度を 2の階乗の値に設定し、元々の形の比率から縦の解像度を設定してある。

5.5.2 画像の作成

低エネルギー、中間エネルギー、高エネルギーのそれぞれのバンドについての画像を得た後、 STIFF を用いてカラーイメージを得る。低エネルギーを赤、中間エネルギーを緑、高エネルギー を青として合成した画像を図 5.10 に示す。

他の投影法についても同様の作業を行いカラーイメージを得た。CAR を図 5.11、PCO を図 5.12、 ZEA を図 5.13 に示す。

5.6 上映

2008年1月16日に平塚市博物館⁵のプラネタリウム(ドーム径10m、五藤光学研究所製G1014) で実際に投影した。AIT 射影で作成した全天画像と星座、天体のカタログを重ねてドームに投影 し、画像の位置情報が正しいことが確認できた。図 5.14 は実際に上映した時の写真である。

⁵http://www.hirahaku.jp/



図 5.10: カラー合成した画像 (AIT)



図 5.11: CAR のカラーイメージ



図 5.12: PCO のカラーイメージ

図 5.14 の左の図は天の川付近の X 線放射を示している。青で示した高エネルギーの放射 (高温のガス) が天の川全体に光っていることがわかる。



図 5.13: ZEA のカラーイメージ



図 5.14: 上映風景

第6章 まとめと今後の課題

可視光のプラネタリムのように X 線観測衛星により得た画像をプラネタリムに投影するという 目的で研究を進めた。今回は ROSAT 衛星の全天サーベイによって得られた画像をプラネタリウ ム投影用のフォーマットに合成することにした。

この全天サーベイのデータは 0.1-0.4keV, 0.4-0.9keV, 0.9-2.4keV の 3 つのエネルギーバンドそ れぞれに 1378 個の fits ファイルがあり、それで全天をカバーしている。隣り合った画像は最低で 0.23 度ずつ重なっている。

全天サーベイのデータをマックスプランク研究所のサイトより取得した後、AIT などのプラネ タリウム投影用のフォーマットにつなげるプログラムを作成し、実行した。しかし、このプログ ラムは画像をただつなげただけなので画像が重なっている部分が生じていた。そこで、この重な りを取り除くために入力を1にしたプログラムを作り、それによって出来た画像で最初に作った 画像を farith を用いて割った。これを3つのエネルギーバンド全てに対して行い、それぞれの画 像を得た。その後3つエネルギーバンドの画像を STIFF を用いて3 色合成した。

2008年1月16日に平塚市博物館にて実際にプラネタリウムを投影してみた。そこで画像の位置 情報が正しいことを確認できたが、色々な課題も見つかった。

当初の目的であるプラネタリウムへの投影までは出来たが、所々が粗かったり、ROSAT が全天 サーベイをした時のスキャンのラインが入ってしまっていたりとまだ課題が残っている。今後は これらの問題を解決し、より鮮明な画像を作ることと、他のX線天文衛星で観測した画像を同じ ようにプラネタリウムで上映できるようにするということが考えられる。

付録A

A.1 FITSのヘッダレコード

ヘッダレコードは80バイトのカードイメージの並びである。その数は無制限であり、最後のヘッ ダレコードはENDというキーワードで示される。

FITS ヘッダのカードイメージは次の形式に従う。(80 バイト= 80 文字を 80 桁と表示している。 ただし HISTORY、COMMENT、"空白"キーワードは例外でこれに従わない。)

キーワード = 値/注釈

ここで 1-8 桁目:キーワード、8 文字以下の左詰めされた ASCII 文字列

9 桁目: = (等号)

10 桁目:空白 (ASCII のブランク、16 進の 0x20 または 20H)

キーワードには小文字は使用しない。値については一定のフォーマットに従う。いくつかの必 須パラメータは一定のフォーマットが要求され、その他のパラメータについてもフォーマットを 固定しておくことが推奨される。

推奨(場合によっては要求)される固定フォーマットは以下の通り。

- 論理型の変数:T 又は F を 30 桁目に置く。
- ・
 整数型の変数:11-30 桁目に右揃え。(虚数部分は 31-50 桁目に右揃え)
- 実数型の変数:11-30 桁目 (小数部が有る場合は小数点必須)。指数表現なら右揃え。(虚数部は 31-50)
- 文字型の変数:標準8文字(長くても可)。11桁目に 'を置き、続けて文字列を書き、20桁
 目以降に終端の 'を置く。

最小限必要なキーワードは以下の通りで順序は固定されており、フォーマットも上述の通りで ある。

SIMPLE 論理型:ファイルが単純 FITS になっているかどうかを示す。

BITMAP 整数型:各データの値を何ビットで表現しているかを示す。

NAXIS 整数型:データ配列の座標軸の本数を示す。

NAXISn 整数型:nは1からNAXISの値まででおのおの第n軸に沿ったデータの本数。

END 値を持たない。9-80 桁は空白でヘッダレコードの終了を示す。

このうち SIMPLE を除くキーワードはすべての *FITS* 要素のヘッダに必要である。また、SIMPLE キーワードは Basic *FITS* 要素のヘッダの最初に現れなければならない。NAXIS=0の場合、NAXISn はあってはならない。キーワードの二重定義はしない。

/(スラッシュ)は後ろに注釈がある場合は必須である。/はパラメータ値の後ならどこでも良いが最低1つの空白を直前に置く。

A.2 *FITS*ファイルを扱うソフトウェア

A.2.1 主なソフトウェアの種類

*FITS*ファイルを表示するためのソフトは多数あるが、無償で使える代表的なものについて簡単にまとめたものを表 A.1 に示す。¹

表 A.1 からもわかるように、それぞれのブラウザは特徴を持っているので、使用目的に応じて 最適なものを選ぶのが良いことになる。以下に例を示す。

- IRAF(Image <u>Reduction and Analysis Facility</u>)の画像ブラウザに使用したい ⇒ ds9
- データベース矢カタログと一緒に使用したい ⇒ JSky, fv, ds9
- 機種やOSを問わず使用したい ⇒ FITSview, fv, JSky
- 色々な *FITS* ファイル (WCS や Extension 含む) を見たい \Longrightarrow fv
- とにかく軽いソフトウェアが欲しい \implies FITSview
- 色々な画像形式を扱いたい ⇒ Makali'i, fv, JSky
- 日本語でヘルプや説明がある ⇒ Makali'i, Qfits, BeSpec
- 分光スペクトルを扱いたい \implies Bespec, Makali'i

本論文には、ds9を使用した。

¹2006年12月末時点

	Makali'i	Qfits	BeSpec	FITSview	fv	ds9	JSky
開発 (配布) 元	国立天文台	理研	美星天文台	NRAO	Gsfc	SAO	ESO
配布形態	В	В	В	S/B	S/B	S/B	S/B
Windows 版	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc
Macintosh 版	×	×	×	\bigcirc	\bigcirc	×	\bigcirc
Linux 版	×	×	×	\bigcirc	\bigcirc	\bigcirc	\bigcirc
Solaris(Sparc)版	×	×	×	\bigcirc	\bigcirc	\bigcirc	\bigcirc
他の UNIX 版	×	×	×	HP,SGI	OSF	MacOS X	java
				Alpha	MacOS X	Darwin	環境
最新版	1.3e	0.80	0.9	2.02(Win)	44.4	4.0b10	2.6
最新版の日付	'06/11/29	'04/12/6	'02/5/10	'99/6/30	'06/7/19	'06/9/27	'05/11/1
ヘッダ表示	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc	\bigcirc
WCSsupport	\bigcirc	×	×	\bigcirc	\bigcirc	\bigcirc	\bigcirc
tableFITS 対応	×	×	×	×	\bigcirc	×	×
Profile 抽出	\bigcirc	\bigcirc	\bigcirc	×	\bigcirc	\bigcirc	\bigcirc
Database 連携	×	×	×	×	\bigcirc	×	\bigcirc
画像形式の対応	BMP	PNG	×	BMP	BMP	PNG	BMP
	GIF,PNG				PNG	JPEG	PNG
	TIFF				JPEG	TIFF	JPEG
	SBIG				PS 他		TIFF
その他	日本語	日本語	日本語	-	Tcl 対応	IRAF 対応	-
	ヘルプ	ヘルプ	ヘルプ				

表 A.1: 代表的な *FITS* ブラウザソフトの比較

A.2.2 主なソフトウェアの入手先

Makali'i http://makalii.mtk.nao.ac.jp/

Qfits http://jahou.riken.go.jp/ koike/QFITS/

BeSpec http://www.bao.go.jp/soft/bespec/

FITSview http://www.cv.nrao.edu/ bcotton/fitsview.html

fv http://heasarc.gsfc.nasa.gov/ftools/fv/

ds9 http://hea-www.harvard.edu/RD/ds9/

Jsky http://archive.eso.org/jsky/

A.3 cfitsioのインストール

まずは、ソースコードを解凍する。GNUの tar を利用できるなら

%tar xzvf cfitsio
3060.tar.gz

GNUの tar が利用できないならば

%gunzip -
c cfitsio3060.tar.gz — tar xvf -

とする。そうすると、cfitsioというディレクトリが出来ているので、そのディレクトリに移る。 移動した後は以下の通りに入力すると、コンパイルは完了する。

% ./configure

% make

無事にコンパイルが終了したら、できあがった libcfitsio.a と*.h を適当な場所にコピーします。 例えば、

%su

cp libcfitsio.a /usr/local/lib/

cp *.h /usr/local/include/

などとする。これでインストールは完了。

A.4 WCS に関係するその他のキーワード

A.4.1 非線形アルゴリズムに伴うパラメータのキーワード:PVi_m, PSi_m

非線形アルゴリズムを使う場合にはパラメータが必要になる場合がある。このためには次のキー ワードを使う。

PVi_m (浮動小数)

ここで*i*は中間世界座標、*m*はパラメータの番号である。もうひとつ、非線形アルゴリズムの 場合に文字型のパラメータが必要になる場合もある (Table などの参照の場合など)。このための キーワードとしては、

PSi_m (文字列)

を使う。ここで i は中間世界座標、m はパラメータ番号である。

A.4.2 世界座標の次元に関するキーワード:WCSAXES

世界座標の要素数はピクセル座標の要素数を超過する場合がある。この問題は初期の FITSでは 縮退した軸、即ち NAXISj = 1 と AIPS などで使われた CROTAi によって表現された。しかし、 こうした縮退した軸を使うやり方は、例えば 2 次元のイメージを表示するソフトウェアが NAXIS = 3 を読んで表示できなかったりする問題がある。そこで、こうした世界座標の次元に関する問 題に、縮退した軸を使わずに対応するために次のキーワードを予約する。

WCSAXES (整数)

これはヘッダーにある WCS 関係のキーワード (CRPIX*j*, PC*i*_*j* 又は CD*i*_*j*, CDELT*i*, CTYPE*i*, CRVAL*i* 又は CUNIT*i* など) のうち最もインデックスの大きなものの値を特定する。デフォルト の値は最も大きな NAXIS であり、*FITS* ヘッダーにでてくるこの種のキーワードは最大値である。

A.4.3 単位に関するキーワード:CUNIT*i*

原始 *FITS* では各軸の単位は CTYPE*i* キーワードで示すことが出来ると考えられていたが、一般的にはこれは正しくない。そこで新しいキーワードとして、

CUNITi (文字列)

を導入し、これで CRVALi, CDELTi の単位を特定できるようにした。

A.4.4 キーワード値のデフォルト

キーワードについてはデフォルト値をが以下のように決められている。

WCSAXES NAXIS 又は最も大きい *i* 又は *j*

- $CRVALi \quad 0.0$
- CRPIX j = 0.0
- CDELTi = 1.0
- CTYPE*i* ''(線形の特に定められていない軸)
- CUNITi ''(定められていない)
- $PCi_{-j} \qquad 1.0 \ (i = j \ \mathcal{O} \ \mathcal{E} \ \mathcal{E})$
- $PCi_j = 0.0(i \neq j \mathcal{O} \wr \mathfrak{E})$
- $ext{CD}i_{-j}$ 0.0

しかし、これらのデフォルト値は WCS Paper で決めたものなので *FITS* ファイルを書くときは 常に完全な WCS を書くようにしてデフォルトに頼るべきではないことが推奨されている。

A.4.5 軸の代替記述

ある座標軸が2つ以上の座標の種類を持っているような場合 (例えば分光軸の周波数・速度・波 長のように) に、各軸に対して最大 26 までの追加の記述が出来るように、オプションキーワード が予約されている。

WCSAXESa WCS に関する記述の軸の数 (整数)

- CRVALia 世界座標各軸についての参照点での物理量(浮動小数)
- CRPIXja ピクセル座標の各軸について参照点となるピクセル位置(浮動小数)
- PCi_ja 線形変換行列 (浮動小数)
- CDELTia 座標値の増分(浮動小数)
- CD*i_ja* スケールを伴う線形変換行列 (浮動小数)
- CTYPEia 座標各軸のタイプ (8 文字)
- CUNITia CRVALia と CDELTia の単位 (文字列)
- PVi_ma 座標のパラメータ m(浮動小数)
- PA*i_ma* 座標のパラメータ m(文字列)

i, jは各々ピクセル、中間世界座標の軸の番号であり、aはA~Zのアルファベットの1文字で 座標のバージョンを表す。この規約によって、軸の番号は1-99に制限され、パラメータmは0-99 に制限される。軸の主記述(最初の記述)はaを空白にしたものになる。例えば最初の軸の記述が CRVAL1, CRVAL2なら2つ目はCRVAL1A, CRVAL2A、3つ目はCRVAL1B, CRVAL2Bのよう になる。これらの軸の代表記述はオプションであり、主記述がある場合に限って記述できる。も う一つのオプションキーワードとして、

WCSNAMEa (文字列)

を定義する。これは WCS の記述の様々なバージョンの名前を特定するのに使われる。(注: CTYPEia では前述のように物理量を表す4文字と射影方法を表す4文字で構成することになって いるが、次の2つのものについては Wells et al. (1981) 以来広く使われてきたので今後も使うこ ととしている。)

CTYPEia = 'COMPLEX' & 'STOKES'

A.4.6 座標の不確定性に関するキーワード

ピクセルの座標軸にはランダムエラーやシステマティックエラーが含まれることがある。このた め次の2つのオプションキーワードを定義する。

CRDERia 座標軸のランダムエラー (浮動小数)

CSYERia 座標のシステマティックエラー (浮動小数)

どちらも CUNITia の単位でデフォルト値は0である。

Parameter	default
OUTFILE_NAME	stiff.tif
BINNING	1
GAMMA	2.22
GAMMA_FAC	1.0
COLOUR_SAT	1.0
NEGATIVE	Ν
SKY_TYPE	AUTO
SKY_LEVEL	0.0
MIN_TYPE	GREYLEVEL
MIN_LEVEL	0.0
MAX_TYPE	QUANTILE
MAX_LEVEL	0.0
VERBOSE_TYPE	NORMAL

表 A.2: Configuration パラメータリストとデフォルト値

A.5 STIFFのConfigurationファイルとパラメータ

コマンドで configuration-file を入力しないと、うまく動かない。なくても動かないことはない がコマンドに全てのパラメータに付いて入力する手間が生じる。そこで、 configuration ファイル を作る必要がある。

一番簡単な方法として、デフォルト値が入っている configuration ファイルを作れば良い。コマン ドで configuration-file を空白にすると、システムは stiff.conf というファイルを探すので、デフォ ルト値を表示する"-d"オプションを使って、デフォルト値の入った stiff.conf を作る。

% stiff -d > stiff.conf

と入力すれば、デフォルト値の入っている configuration ファイルが出来る。

A.6 プラネタリウム投影用画像の作成

A.6.1 fitsのヘッダー (931049p-fp1.fits)

SIMPLE	=		Т						
BITPIX	=		-32						
NAXIS	=		2						
NAXIS1	=		512						
NAXIS2	=		512						
TIMESYS	=	'UTC '		/	POS	time sca	le sp	ecific	ation
RADESYS	=	'FK5 '		/	P2S	WCS for	this	file	
EQUINOX	=	2.000000	000000E+03	/	POR	equinox			
OBS_ID	=	'WG931049P	_N1_SI01.N1	,	/ RDF	7 observ	vation	ID	

```
OBJECT = 'RASS 3/10/49'
                             / POR| name of object
OBSERVER= 'MPE
                  ,
                             / POR| observer(s)
TELESCOP= 'ROSAT
                  ,
                             / POR| telescope
OBS_MODE= 'SURVEY '
                              / RDF | observing mode
INSTRUME= 'PSPCC
                 ,
                              / POR| instrument name
FILTER = 'NONE
                  ,
                              / RDF| filter ID
DATE-OBS= '1990-12-03T13:38:26' / POR| time of first event in image (UTC)
DATE_END= '1990-12-18T04:11:56' / RDF| time of last event in image (UTC)
ONTIME =
                        23438 / RDF| On time (s)
ONTIMED =
                        20192 / JER | sum of OBI times in RDA data (s)
ONTIMEI =
                        20177 / JER | sum of OBI times in image (s)
EXPMIN =
                          436 / JER | minimum exposure (s)
EXPMAX =
                          647 / JER| maximum exposure (s)
PIMIN
                           11 / RDF | Min PI channel
      =
                           41 / RDF| Max PI channel
PTMAX
       =
                        78222 / JER| tot.nr. of acc.evt. in RDA data
NEVTTA =
NEVTTR =
                        11525 / JER| tot.nr. of rej.evt. in RDA data
                 5.659841E-01 / JER | minimum pixel value
ZMIN
       =
ZMAX
                 3.204206E+01 / JER | maximum pixel value
       =
                 4.574612E+05 / JER| sum of pixel values
7.SUM
       =
BSCALE =
                 1.000000E+00 / POR| Real = TAPE*BSCALE+BZERO
BZERO =
                 0.000000E+00 / POR|
BUNIT = 'cts/s/deg**2'
                              / POR|
CTYPE1 = 'RA---CAR'
                              / POR| axis type for dim 1
CTYPE2 = 'DEC--CAR'
                              / POR| axis type for dim 2
CDELT1 = -1.2500000000E-02 / POR | X degrees per pixel
CDELT2 = 1.2500000000E-02 / POR| Y degrees per pixel
CRPIX1 =
           2.56500000000E+02 / POR| pix coord of x origin ('pix 1 from 0.5 to
CRPIX2 =
           2.56500000000E+02 / POR| pix coord of y origin (1.5' convention
CRVAL1 =
           3.357692000000E+02 / POR| sky coord of x origin (deg)
CRVAL2 =
           3.93750000000E+01 / POR| sky coord of y origin (deg)
CROTA1 =
           0.0000000000E+00 / POR| ccw rotation of crd system (deg)
           0.0000000000E+00 / POR| ccw rotation of crd system (deg)
CROTA2 =
ORIGIN = 'JRFI_I0|2001-02-26|JER' / program|version|author
       = '2001-04-11T05:03:10' / file creation date (YYYY-MM-DDThh:mm:ss UTC)
DATE
                 0.000000E+00 / JER| FOV cut radius for input fields (deg)
FOV_CUTR=
COL_TRF =
                            0 / JER| rgb transfer function (0-no transfer)
COL_GAM =
                 1.000000E+00 / JER| rgb gamma index
IMGFLT = '5__GAU NSL=1 NBP=400 RB1=30 RB2=120' / JER | image filter
                 1.000000E+01 / JER | likelihood threshold
LH_THR =
COMMENT Keyword types:
COMMENT * POR - reserved in Paper 0
COMMENT * POS - suggested in Paper 0
COMMENT * P2S - suggested in Paper 2
```

```
COMMENT * RDF - present in ROSAT RDF FITS files

COMMENT * JER - introduced by JER

COMMENT References:

COMMENT * Paper 0: NOST 100-2.0, FITS Definition, http://fits.gsfc.nasa.gov/

COMMENT * Paper 2: Paper II in http://fits.gsfc.nasa.gov/documents.html#WCS

END
```

A.6.2 プログラム (convertWCS.c)

```
#include<stdio.h>
  #include<stdrib.h>
  #include<string.h>
  #include<fitsio.h>
  #include<longnam.h>
  #include<wcshdr.h>
  #include<wcs.h>
  #include<math.h>
 main(int argc, char **argv){
    FILE *fp;
    char infile[256], inputlist[256], PRJ[4];
    char outfile[256], RAproj[9], DECproj[9];
    long outimagesizeX, outimagesizeY; /*出力画像のサイズ*/
    fitsfile *fptr_in, *fptr_out; /*入力および出力の fits ファイル*/
    int status=0, ncards, relax, nreject, nwcs, ctrl, i, j, anynul;
    long naxes [2];
    float *inimage, *outimage; /*入力画像および出力画像のピクセル値 (flux) を書き出す配列*/
    float X_in, Y_in, X_out, Y_out;
    char *header;
    struct wcsprm *wcs_in, *wcs_out;
    double pixcrd_in[1][2], imgcrd[1][2], phi[1], theta[1], world[1][2], alpha, delta;
    double crpix1, crpix2, pixcrd_out[1][2];
    int i_out, j_out, nfile;
    double pixelsize;
    if (argc != 9)
      printf("usage: convertWCS inputfilelist PRJ outimagesizeX outimagesizeY crpix1 crpix2
pixelsize outfitsname\n");
      printf("where PRJ must be one of the following; CAR, AIT, ZEA, CSC, PCO.\n");
      exit(1);
    }else{
      strcpy(inputlist, argv[1]);
      strcpy(PRJ, argv[2]);
      sscanf(argv[3], "%ld", &outimagesizeX);
      sscanf(argv[4], "%ld", &outimagesizeY);
```

```
sscanf(argv[5], "%lf", &crpix1);
     sscanf(argv[6], "%lf", &crpix2);
     sscanf(argv[7], "%lf", &pixelsize);
     strcpy(outfile, argv[8]);
   }
   printf("input file list = \%s\n", inputlist);
   if(strcmp(PRJ, "CAR")!=0 &&strcmp(PRJ, "AIT")!=0 &&strcmp(PRJ, "ZEA")!=0 &&str-
cmp(PRJ, "CSC")!=0 &&strcmp(PRJ, "PCO")!=0){
     printf("PRJ must be one of the folloing: CAR, AIT, ZEA, CSC, PCO\n");
     exit(1); \}
   /* 入力画像の ASCII リストを読み込む */
   if(NULL==(fp=fopen(inputlist, "r"))){
     printf("cannot open %s\n", inputlist);
     exit(1);
   }
   inimage=malloc(512*512*4); /*入力画像のピクセル値 (4byte) のためのメモリ確保*/
    outimage=malloc(outimagesizeX*outimagesizeY*4); /*出力画像のピクセル値 (4byte)の
ためのメモリ確保*/
   /*出力画像の各ピクセル値を0にリセット*/
   for(i=0; i < outimagesizeX; i++)
     for(j=0; j < outimageseizeY; j++)
       outimage[i*outimagesizeY+j]=0.0;
     }
   }
   /* 構造の中にアウトプット WCS キーワードを置く*/
   relax=1; /*0 は公式の WCS スタンダードに定義されている FITS キーワードだけを認める。
1は0より緩い*/
   ctrl =0; /* もし0なら、認められていないヘッダーカードを報告しない。値が1, 2, 3なら詳
細なエラーメッセージをこの順序で与える。*/
   /* 出力用の fits ファイルを作成する。 */
   if(fits_create_file(&fptr_out, outfile, &status)){
     fits_report_error(stderr, status);
     return 1;
   }
   naxes[0]=outimagesizeX;
   /*用意した出力ファイルのヘッダーにキーワードを書き込む*/
   fits_create_img(fptr_out, FLOAT_IMG, 2L, naxes, &status);
   /*基本座標システムに FK5 を使用する*/
   fits_write_key_str(fptr_out, "RADESYS", "FK5", "Coordinate system", &status);
```

```
/*2000 年の春分点 (equinox) を利用する*/
fits_write_key_dbl(fptr_out, "EQUINOX", 2000.0, 3, "Eauinox", &status);
```

```
/*CTYPE に局所球面座標のタイプを入力している。PRJ はプログラムを動かすときに入力
するパラメータで、AIT などの投影法を指定している。*/
    sprintf(RAproj, "RA---%s", PRJ);
    fits_write_key_str(fptr_out, "CTYPE1", RAproj, "DEC projection", &status);
    sprintf(DECproj, "DEC--%s", PRJ);
    fits_write_key_str(fptr_out, "CTYPE2", DECproj, "DEC projection", &status);
    /*参照ピクセル座標を入力したパラメータの crpix1, crpix2 にする。*/
    fits_write_key_dbl(fptr_out, "CRPIX1", crpix1, 13, "X reference pixel", &status);
    fits_write_key_dbl(fptr_out, "CRPIX2", crpix2, 13, "Y reference pixel", &status);
    /*出来た画像は天球全体の画像なので、接点の天球経度、緯度を 0.0 に初期化している。*/
    fits_write_key_dbl(fptr_out, "CRVAL1", 0.0, 13, "RA of reference pixel", &status);
    fits_write_key_dbl(fptr_out, "CRVAL2", 0.0, 13, "DEC of reference pixel", &status);
    /*座標スケール値を入力パラメータの pixelsize の値にする。*/
    fits_write_key_dbl(fptr_out, "CDELT1", -pixelsize, 13, "X pixel increment (degree)", &sta-
tus);
    fits_write_key_dbl(fptr_out, "CDELT2", pixelsize, 13, "Y pixel increment (degree)", &sta-
tus);
    /*出力ファイルに書き込んだヘッダーを読み込み0でないことを確認*/
   if(fits_hdr2str(fptr_out, 1, NULL, 0, &header, &ncards, &status)){
     fits_report_error(stderr, status);
     return 1;
    }
    /* 構造の外にアウトプット WCS キーワードを置く */
   if(status = wcspih(header, ncards, relax, ctrl, &nreject, &nwcs, &wcs_out)){
     fprintf(stderr, "wcspih ERROR %d: %s.\n", status, wcs_errmsg[status]);
    Ĵ
    if(status = wcsset(wcs_out))
     fprintf(stderr, "wcsset ERROR %d: %s.\n", status, wcs_errmsg[status]);
    wcsprt(wcs_out);
    free(header);
    nfile=0;
    while (\text{fscanf}(\text{fp}, "\%\text{s"}, \text{infile}) = = 1)
     nfile++:
      /* 入力 fits ファイルを開く */
     if(fits_open_file(&sptr_in, infile, READONLY, &status)){
        fits_report_error(stderr, status);
```

```
return 1;
     }
     printf("###### file %d: %s\n", nfile, infile);
      /* 入力ファイルのヘッダーを読み込む*/
     if(fits_hdr2str(fptr_in, 1, NULL, 0, &header, &ncards, &status)){
        fits_report_error(stderr, status);
        return 1;
     }
      /* 入力画像 (512×512) の各ピクセル値を 2 次元配列に書き込む*/
     fits_read_2d_flt(fptr_in, 0, 0.0, 512, 512, 512, 512, inimage, &anynul, &status);
      /* 構造の中にインプット WCS キーワードを置く */
     if(status = wcspih(header, ncards, relax, ctrl, &nreject, &nwcs, &wcs_in)){
        fprintf(stderr, "wcspih ERROR %d: %s.\n", status, wcs_errmsg[status]);
      }
                  /*獲得したメモリの解放*/
     free(header);
     if(status = wcsset(wcs_in))
        fprintf(stderr, "wcsset ERROR %d: %s.\n", status, wcs_errmsg[status]);
      }
      /*入力画像 (512 × 512) を読み込み、元のピクセル座標を世界座法に変換する */
     for(i=0; i<512; i++){
       for(j=0; j<512; j++){
     /* (1, 1) が左下で X は右に行くほど増加し、Y は上にいくほど増加し、X=i+1, Y=j+1 と
表せる時のイメージ座標 (X, Y) を定める。*/
         X_{in} = (float)j+1.0;
         Y_{in} = (float)i+1.0;
      /* wcslib を使って World Coordinates(赤径と赤緯)を算出する*/
         pixcrd_in[0][0] = (double) X_in;
         pixcrd_in[0][1] = (double) Y_i;
         wcsp2s(wcs_in, 1, 2, pixcrd_in[0], imgcrd[0], phi, theta, world[0], &status);
         alpha = world[0][0]:
         delta = world[0][1];
      /* wcslib を使って (X, Y) 座標のピクセル値 (flux) を算出する*/
         wcss2p(wcs_out, 1, 2, world[0], phi, theta, imgcrd[0], pixcrd_out[0], &status);
         X_out = (float) pixcrd_out[0][0];
         Y_out=(float) pixcrd_out[0][1];
         i_out = floorf(X_out)-1;
         j_{out} = floor(Y_{out})-1;
    /*入力画像の (j, i) ピクセルの flux を対応する出力画像の (i_out, j_out) ピクセルに書き込
む。*/
         if(i_out>=&&i_out<outimagesizeX&&j_out>=0&&j_out<outimagesizeY){
```

```
outimage[i\_out+j\_out*outimagesizeX] = outimage[i\_out+j\_out*outimagesizeX] + inimage[i*512+j];
```

```
}
}
/* 入力ファイルを閉じる */
fits_close_file(fptr_in, &status);
wcsfree(wcs_in);
}
```

imagesize= outimagesizeX*outimagesizeY;

 $fits_write_2d_flt(fptr_out, 0L, outimagesizeX, outimagesizeX, outimagesizeY, outimage, \&status);$

```
/* 出力ファイルを閉じる*/
fits_close_file(fptr_out, &status);
wcsfree(wcs_out);
```

A.6.3 プログラムに使用されている関数の解説

「CFITSIO の簡単なメモ」²と CFITSIO と WCSLIB のヘッダーファイルを基に、本論文で開発したプログラムに使用している cfitsio および wcslib の関数を解説する。

• CFITSIO

}

fitsfile 基本的な HDU(Header and Data Unit) 情報を示す。

一般に、fitsfile *fptr と使い、コラムに含まれている HDU の位置を示すのに使われる。

fits_open_file ファイルを開く

fits_open_file(fitsfile **fptr, char *filename, int iomode, int *status);

fptr CFITSIO で定義されている構造体で、あらかじめ宣言しておく必要がある。 filename オープンしたい *FITS*ファイルの名前。

iomode どんなモードでオープンするかを指定する。READONLY または READ-WRITE のどちらかを指定する。これはマクロになっている。 status オープンに成功したか失敗したかを返す変数である。

fits_create_file ファイルを作成する

fits_create_file(fitsfile **fptr, char *filename, int *status);

filename の名前を持つ fits ファイルを作成する。

 $^{^{2}} http://www.cr.scphys.kyoto-u.ac.jp/member/matumoto/XIS/CFITSIO/memo.html$

```
fits_create_img もし fptr が指すファイルが空だったら primary HDU が作成されて、存在
していたら新しい IMAGE extension が作られる
```

fits_create_img(fitsfile *fptr, int bitpix, int naxis, long *naxes, int *status);

bitpix

_	BYTE_IMG	8	unsigned char
_	SHORT_IMG	16	signed short integer
_	LONG_IMG	32	signed long integer
_	FLOAT_IMG	-32	float
_	DOUBLE_IMG	-64	double
の	いずれかを選ぶ。		
naxis	データの次元を	指定。	
naxes	この配列には各	次元(の長さを指定。

```
fits_write_key_str ファイルに文字列のキーワードを書き込む
```

fits_write_key_str(fitsfie *fptr, char *keyname, char *value, char *comment, int *status);

keyname キーワードの名前 value キーワードの値の文字列 comment コメントの文字列。必要なければ NULL

fits_write_key_dbl doubleの値のキーワードを書き込む

fits_write_key_dbl(fitsfile *fptr, char *keyname, double value, int decim , char *comment, int *status);

keyname キーワードの名前 **value** 書き込む値 **decim** 通常は、小数点以下の桁数。

fits_hdr2str ヘッダーキーワードを読み込む

ftis_hdr2str(fitsfile *fptr, int exclude_comm, char **exclist, int nexc, char **header, int *nkeys, int *status);

```
fits_report_error エラーを返す
fits_report_error(FILE *stream, int status);
```

fits_read_2d_flt 各ピクセルのデータを2次元配列に読み込む

fits_read_2d_flt(fitsfile *fptr, long group, float nulval, LONGLONG ncols, LONG-LONG naxis1, LONGLONG naxis2, float *array, int *anynul, int *status);

group 複雑な FITS ファイルを扱う場合には必要だが、とりあえず0を入れておいて も問題無し。 **nulval** 未定義のピクセル値を置き換えるために使用。 array データが格納される配列。 ncols 配列の最初の次元の長さを指定する。 naxis1, naxis2 *FITS* データの大きさを指定する。

fits_close_file ファイルを閉じる

fits_close_file(fitsfile *fptr, int *status);

fptr CFITSIO で定義されている構造体で、あらかじめ宣言しておく必要がある。 **status** クローズに成功したか失敗したかを返す変数である。

• WCSLIB

wcsprm 基本となる構造体。あらかじめ定義しておく必要がある。

wcs_errmsg エラーを返す

wcsprt wcsprm 構造体の内容をプリントする

wcsprt(const struct wcsprm*);

wcsset wcsprm 構造体の情報によって wcsprm 構造体をセットアップする。 wcsset(struct wcsprm*);

wcsp2s ピクセル座標を世界座標に変換する

wcsp2s(struct wcsprm^{*}, int ncoord, int nelem, const double pixcrd, double imgcrd, double phi, double theta, double world, int stat);

ncoord, nelem ベクトル長 pixcrd ピクセル座標の配列 imgcrd 中間世界座標の配列 phi, theta 元の座標形の緯度と経度 world 世界座標の配列

wcss2p 世界座標をピクセル座標に変換する

wcss2p(struct wcsprm^{*}, int ncoord, int nelem, const double world, double phi, double theta, doube imgcrd, double pixcrd, int stat);

wcsfree wcsprm に割り当てられるメモリを解く wcsfree(struct wcsprm*);

wcspih イメージヘッダーを解析する高水準 FITS WCS ルーチン

wcspih(char *header, int ncards, int relax, int ctrl, int *nreject, int *nwcs, struct wcsprm **wcs);

header ヘッダーの配列 ncards header の中のカード番号 relax 任意の角度 ctrl エラー報告し無効な WCS と他のヘッダーカードのオプションをコントロールする nreject WCS カードが構文エラー、不適切な値などを拒否する数 nwcs 見つかった座標表現の数 wcs wcsprm 構造体配列のポインタ

A.6.4 画像サイズの変更

ここでは画像の解像度を変更するために今回使用したフリーソフトウェアを紹介する。

• 解像度変更

かんたん画像サイズ ver1.1.0

このソフトは以下のサイトからダウンロードできる。

http://www.vector.co.jp/soft/win95/art/se374178.html

このソフトは JPEG ファイルにのみ対応しているので、TIFF ファイルを JPEG ファイルに 変換することが必要となる。

ファイル変換

SentTo-Convert 1.64

このソフトは以下のサイトからダウンロードできる。

http://www.vector.co.jp/soft/win95/art/se267970.html

BMP/PNB/JPEG/TIFF(無圧縮)/GIF などのファイルを/BMP/PNG/JPEG ファイルに変換することが出来る。

これらのソフトは Windows 上でのみ動作する。

関連図書

- [1] Trümper, J. 1983, Adv. Space Res., 2(4) 241
- [2] Trümper, J. 1992 QJRAS, 33. 165
- [3] Axchenbach et al. 1985
 ROSAT PSPC scientific Data Processing Requirement Document, MPE Internal Report 44, Max-Planck-Institut f
 ür extraterrestrische Physik, D 85740 Garching near Munich, Germany
- [4] 川埜 直美 修士論文 (広島大学) 2003
- [5] 大戸 彰三 修士論文 (広島大学)2003
- [6] Snowden, S.L., & Schmtt, J.H.M.M. 1990, Ap&SS, 171, 207
- [7] Voges, W. 1992, in Proc, Satellite Symp. 3, Space Science With Particular Emphasis on High-energy Astrophysics, ed. T.D.Guyenne & J.J.Hunt(Noordwijk:ESA), 9
- [8] *FITS*の手引き ~第5.1版~
- [9] "Representations of world coordinates in *FITS*", (WCS paper I)
 E.W.Greisen and M.Calabretta, Astronomy & Astrophysics, 395, 1061-1075, 2002
- [10] "Representation of celestial coordinates in *FITS*"' (WCS paper II)

M. Calabretta and E.W. Greisen, Astronomy & Astrophysics, 395, 1077-1122, 2002

[11] "Representations of spectral coordinates in *FITS*", (WCS paper III)
 E.W. Greisen, M.R. Calabretta, F.G. Valdes, and S.L. Allen, Astronomy & Astrophysics, 446, 747-771, 2006

表目次

2.1	ROSAT の特性	5
2.2	ROSAT HRIの基本性能	5
2.3	ROSAT PSPC の基本性能	7
2.4	ROSAT XRT の基本性能	7
3.1	<i>FITS</i> の要素の構造	10
4.1	これまでに出てきた変数	21
4.2	RADECSYSa が存在する場合 (左) としない場合 (右)	22
5.1	入力パラメータ	34
A.1	代表的な <i>FITS</i> ブラウザソフトの比較	47
A.2	Configuration パラメータリストとデフォルト値	51



1.1	可視光の太陽 (左) とようこう衛星軟 X 線望遠鏡で捉えた太陽 (右)
2.1	<i>ROSAT</i> の外観
2.2	ROSAT
2.3	PSPC の外観
2.4	PSPC の観測窓の概念図
2.5	全天サーベイ 8
4.1	参照点を極とした局所 (native) 球面座標 (左、 $(\phi_0, heta_0) = (0, 90^\circ)$) と、参照点を赤
	道と基準経度の交点とした局所座標 (右、 $(\phi_0, \theta_0) = (0, 0)$)
4.2	zenithal 投影の R_{θ}, θ, μ の関係図 (左)、と3つの特別なケース (右) 20
4.3	Zenithal equal-area 投影法 24
4.4	Plate Carrée 投影法
4.5	Hammer-Aitoff 投影法 20
4.6	COBE Quadriateralized Spherical Cube 投影法 22
5.1	931049p-fp1
5.2	931049p-fp2
5.3	931049p-fp3
5.4	AIT(低エネルギー)のヘッダー 35
5.5	1378 枚の画像をつなぎ合わせた画像 (AIT)
5.6	規格化用に作成した画像 (AIT) 36
5.7	重なりを取り除いた画像 (低エネルギー)(AIT) 36
5.8	中間エネルギーの画像 (AIT)
5.9	高エネルギーの画像 (AIT)
5.10	カラー合成した画像 (AIT)
5.11	CAR のカラーイメージ 39
5.12	PCO のカラーイメージ 40
5.13	ZEA のカラーイメージ 41
5.14	上映風景