

# CMOSイメージセンサを用いたX線撮像偏光計の 性能評価およびデータ処理系の開発

Development of an X-ray imaging polarimeter using a CMOS image  
sensor and its data processing system

東京大学大学院  
理学系研究科 物理学専攻  
馬場研究室

渡邊泰平

2022年1月27日

## 概要

X線天文学は、熱く激しい天体を対象として宇宙の高エネルギーの姿を探究する学問である。光の持つ情報としては時間・空間・エネルギーの他に偏光が挙げられるが、X線帯域での偏光観測は技術的な困難が伴うことから未だ発展途上の段階である。特に、非熱的放射が卓越し十分な光子統計量を得られる10-30 keVの硬X線帯域において、偏光観測と撮像を同時に行なうようなミッションはまだ実現していない。そこで我々は10-30 keVのエネルギー帯域での観測を実現する撮像偏光計の開発を進めており、超小型衛星CubeSatへの展開を目指している。コンパクトなサイズでの実現のため、撮像系には符号化開口マスクを、偏光X線検出器には微小ピクセルを持つCMOSイメージセンサを採用する。このプロジェクトの実現に向けて研究と開発が進められてきたが、センサからのデータ読み出しから、偏光撮像の最終結果を出力するまでの一貫したデータ処理フローが未完成であった。特に未着手であったのがX線検出に特化した実用的な読み出しシステムの構築と、偏光解析と撮像を統合する解析手法の確立である。

本修論では、撮像偏光計の一貫したデータ処理フローの完成を目指し、FPGA実装による新たな読み出しシステムの構築と、画像再構成法の精度向上、および偏光情報を含めた画像再構成手法の開発に取り組んだ。従来の読み出しシステムでは、多ピクセルCMOSセンサのデータ量の多さに由来する撮影時のデッドタイムが、フレームあたり約1s生じていた。この問題に対して本修論では、解析に必要な情報のみを取捨選択するアルゴリズムをFPGAに実装することによって、フレームあたりのデッドタイムを露光時間600msで38msまで減らすことが可能になった。これによって実質的な観測効率率は従来の約2倍になった。また、符号化開口では再構成に伴う偽像が生じるが、従来の相関法による再構成ではマスクのランダムパターンの特性による偽像が多く出てしまっていた。これに対し本修論では、ランダム性を保ったまま開口率が均等になるようなマスクパターンを考案することで、相関法の再構成で最適化していないパターンの約0.10倍に偽像を抑えられることを示した。また再構成手法として新たにExpectation-Maximization(EM)アルゴリズムを導入し、EMアルゴリズムに適用するための符号化開口イメージング用のモデルを作成した。このモデルを用いると、最適化していないパターンであっても相関法による再構成の約0.12倍に偽像を抑えられることが分かった。最後に、新たな読み出しシステムを用いて符号化開口マスクとCMOSセンサを含めた光学系に実際の偏光X線を照射するビーム試験を行なった。ステージ操作によって視野角420"に広がった天体を模したデータセットを取得し、EMアルゴリズムによる画像再構成を成功させた。またCrabを模した円環光源の画像再構成を行い、EMアルゴリズムの再構成で必要な光子統計量が3000イベント程度であることが分かった。さらに異なる偏光角の実験データを混ぜ合わせたデータに対してEMアルゴリズムを適用し、天球面領域を切り分けて場所ごとの偏光度を出せることを示した。

# 目次

第1章	はじめに	1
第2章	偏光 X 線とその検出	4
2.1	偏光	4
2.1.1	偏光と偏光度	4
2.1.2	ストークスパラメータ	5
2.2	偏光 X 線の放出過程	6
2.2.1	シンクロトロン放射	6
2.2.2	逆コンプトン散乱	7
2.3	偏光 X 線の検出	8
2.3.1	光電効果	9
2.3.2	コンプトン散乱	9
2.4	CMOS イメージセンサの原理	9
2.4.1	半導体検出器の原理	10
2.4.2	MOS と MOSFET	10
2.4.3	CMOS	11
2.5	2 章のまとめ	13
第3章	CMOS センサの X 線検出器としての性能評価試験	14
3.1	使用センサと読み出し系	14
3.1.1	CMOS センサ	14
3.1.2	読み出し系	15
3.2	センサ性能測定試験セットアップ	17
3.3	検出器のダーク特性評価	18
3.3.1	ペDESTAL とピクセル選定	18
3.3.2	測定と結果	18
3.3.3	ピクセル選定	19
3.4	検出器の X 線応答評価	20
3.4.1	解析フロー	20
3.4.2	測定と結果	22
3.4.3	エネルギー較正	25
3.4.4	エネルギー分解能	25
3.4.5	検出効率	26
3.5	温度応答	28

3.5.1	簡易温度変化試験 . . . . .	28
3.5.2	温度変化データの特徴 . . . . .	29
3.6	3章のまとめと次章への展望 . . . . .	30
<b>第4章</b>	<b>FPGAを用いた高速読み出しシステム開発</b>	<b>31</b>
4.1	背景 . . . . .	31
4.2	動的ペDESTAL . . . . .	32
4.2.1	概要 . . . . .	32
4.2.2	線源データへの適用 . . . . .	34
4.3	データ削減アルゴリズム . . . . .	35
4.3.1	アルゴリズム内容 . . . . .	35
4.3.2	動作例 . . . . .	38
4.3.3	タイミングチャート . . . . .	39
4.4	FPGA上の回路設計と実装 . . . . .	40
4.4.1	FPGA . . . . .	40
4.4.2	回路全体とデータ削減IPの設計 . . . . .	42
4.4.3	リングバッファ . . . . .	44
4.4.4	ステートマシン . . . . .	46
4.4.5	VHDLによる実装 . . . . .	48
4.5	動作確認テストと性能 . . . . .	48
4.5.1	データ削減モードでのデータ取得 . . . . .	48
4.5.2	解析手法 . . . . .	49
4.5.3	検出性能 . . . . .	49
4.5.4	読み出し性能 . . . . .	50
4.6	4章のまとめ . . . . .	53
<b>第5章</b>	<b>符号化開口のノイズ低減を目指した画像再構成法の検討</b>	<b>54</b>
5.1	符号化開口イメージング . . . . .	54
5.1.1	概要 . . . . .	55
5.1.2	撮像性能 . . . . .	55
5.2	モンテカルロシミュレーションによる調査 . . . . .	56
5.2.1	セットアップ . . . . .	56
5.2.2	データセット . . . . .	56
5.2.3	撮像ノイズ指標 . . . . .	57
5.3	画像再構成法: 相関法 . . . . .	58
5.3.1	バックプロジェクションによる再構成 . . . . .	58
5.3.2	実装 . . . . .	59
5.3.3	並行ランダムパターン . . . . .	60
5.3.4	ランダムパターン選定: 大局的な考察 . . . . .	61
5.3.5	ランダムパターン選定: 一様束縛条件の導入 . . . . .	63
5.3.6	ランダムパターン選定: 相互相関からの最適パターンの選出 . . . . .	64

5.3.7	最適パターンでの画像再構成	65
5.4	画像再構成法: EM アルゴリズム	68
5.4.1	観測データへの最尤法適用	69
5.4.2	符号化開口への適用	70
5.4.3	実装	71
5.4.4	シミュレーションデータの再構成	71
5.5	5章のまとめ	75
<b>第6章</b>	<b>ビームライン実験による偏光撮像実証</b>	<b>76</b>
6.1	セットアップ	76
6.1.1	照射ビーム	77
6.1.2	光学系	77
6.1.3	データフロー	78
6.1.4	アラインメント	80
6.2	データ取得	83
6.2.1	データセット	83
6.2.2	時間効率	84
6.3	検出器性能評価	85
6.3.1	イベント解析	85
6.4	偏光応答評価	86
6.4.1	モジュレーションファクターの算出	86
6.5	撮像性能評価	88
6.5.1	データ抽出手順	89
6.5.2	相関法によるスキャンデータ再構成	89
6.5.3	EM アルゴリズムによるスキャンデータ再構成	90
6.5.4	光子統計量が少ない場合の円環再構成	93
6.6	偏光撮像性能評価	94
6.6.1	重ね合わせによる偏光の表現	94
6.6.2	複数偏光源データの作成	96
6.6.3	EM アルゴリズムの偏光撮像への拡張	97
6.6.4	複数偏光源データの再構成	98
6.6.5	任意の偏光角での画像再構成	99
6.7	光子統計量からの <i>cipher</i> の実現性の考察	100
6.8	6章のまとめ	100
<b>第7章</b>	<b>結論</b>	<b>101</b>
<b>付録A</b>	<b>データ削減 IP の実装</b>	<b>105</b>
<b>付録B</b>	<b>ZDAQ ボード上読み出しソフトウェア</b>	<b>108</b>
B.1	ソフトウェアの実装概要	108
B.2	CPU と FPGA の処理速度比較	109

# 目次

1.1	Crab Nebula . . . . .	1
1.2	超小型衛星のモデル図 . . . . .	3
1.3	データ処理フロー概要 . . . . .	3
2.1	楕円偏光 . . . . .	5
2.2	シンクロトロン放射 . . . . .	6
2.3	逆コンプトン散乱の概念図 . . . . .	7
2.4	Si と光子の反応断面積 . . . . .	8
2.5	光電効果の概念図 . . . . .	9
2.6	半導体検出器の原理 . . . . .	10
2.7	MOS ダイオードの構造 . . . . .	11
2.8	MOSFET の構造 . . . . .	11
2.9	CMOS インバータの構造 . . . . .	12
2.10	CMOS イメージセンサの概略図 . . . . .	12
3.1	使用センサ . . . . .	14
3.2	読み出し系ハードウェア . . . . .	15
3.3	読み出し系全体図 . . . . .	16
3.4	性能測定実験セットアップ . . . . .	17
3.5	ダーク特性 . . . . .	19
3.6	X 線解析フロー . . . . .	20
3.7	イベント形状の種類 . . . . .	22
3.8	$^{109}\text{Cd}$ , $^{241}\text{Am}$ 検出スペクトル . . . . .	23
3.9	double イベントフィッティング . . . . .	23
3.10	$^{109}\text{Cd}$ データ解析結果 . . . . .	24
3.11	エネルギー較正直線 . . . . .	25
3.12	エネルギー分解能 . . . . .	27
3.13	温度変化 . . . . .	28
3.14	温度変化データの生波高値の時間変化 . . . . .	29
3.15	温度変化データ解析結果 . . . . .	29
4.1	従来の読み出し系 . . . . .	31
4.2	従来の読み出し系のタイミングチャート . . . . .	32
4.3	波高値の時間変化例 . . . . .	33
4.4	動的ペDESTALのスペクトル比較 . . . . .	34

4.5	温度変化データへの動的ペDESTAL適用	35
4.6	データ削減アルゴリズム動作例	38
4.7	タイミングチャート	40
4.8	AXI-Stream 概要	41
4.9	AXI-Stream タイミングチャート例	42
4.10	回路設計概要	43
4.11	データ削減 IP の概要	44
4.12	リングバッファ動作例	45
4.13	ステートマシン状態遷移図	46
4.14	送受信タイミングチャート例	47
4.15	データ削減モードでのスペクトル	50
4.16	露光時間依存性試験の結果	52
4.17	閾値依存性試験の結果	52
5.1	先行研究での再構成画像	54
5.2	Encode 過程	55
5.3	相関法概要	59
5.4	ランダムパターン	61
5.5	ランダムパターン再構成画像	62
5.6	マスクパターンに偏りがあるときの Encoding と Decoding	63
5.7	ランダムパターンの領域ごとの開口率	63
5.8	一様束縛条件を入れたランダムパターンセットの作成図	65
5.9	パターンスコアの比較	66
5.10	一様束縛条件を導入したランダムパターンセット	67
5.11	相互相関の分布	67
5.12	一様束縛条件導入後ランダムパターン再構成画像	68
5.13	EM アルゴリズム変数定義	69
5.14	EM アルゴリズムの再構成画像	72
5.15	EM アルゴリズムの逐次ステップ	74
5.16	$\Delta\hat{D}$ のステップ変化	75
6.1	ビームラインハッチ内	76
6.2	光学系	78
6.3	電気通信系統図	78
6.4	SPring-8 実験データフロー	79
6.5	クイックルックソフト	81
6.6	アラインメントスキャン例	82
6.7	撮像スキャン	83
6.8	$\theta_x = 0^\circ$ での 1 スキャン分のスペクトル	85
6.9	$\theta_x = 0^\circ$ での 1 スキャン分の weight 分布	86
6.10	モジュレーションカーブ	88
6.11	相関法による再構成画像	90

6.12	16 keV スキャンの EM アルゴリズムによる再構成画像 . . . . .	91
6.13	22 keV スキャンの EM アルゴリズムによる再構成画像 . . . . .	92
6.14	円環の再構成画像 . . . . .	93
6.15	再構成ノイズの光子統計量による依存性 . . . . .	94
6.16	偏光の重ね合わせ . . . . .	94
6.17	複数偏光源データの作成 . . . . .	96
6.18	複数偏光源データの EM アルゴリズムによる再構成画像 . . . . .	99

# 表 目 次

3.1	CMOS センサ仕様	15
3.2	使用線源の X 線帯域輝線	17
3.3	グッドピクセル選定条件	20
3.4	double イベントスペクトルフィッティング結果	24
3.5	エネルギー較正パラメータ	25
3.6	エネルギー分解能	26
3.7	$^{109}\text{Cd}$ による検出効率	28
4.1	4 章の演算定義	31
4.2	動的ペダスタル性能評価	34
4.3	ステートマシン状態リスト	46
4.4	データ削減 IP の入力信号と出力信号	48
4.5	読み出し性能測定試験でのパラメータ	49
4.6	データ削減モード性能評価	50
4.7	露光時間依存性フィッティングパラメータ	52
4.8	データサイズ依存性フィッティングパラメータ	53
5.1	シミュレーションパラメータ	57
5.2	5 章での計算環境	57
5.3	ランダムパターン性能比較	66
5.4	5 章の再構成性能まとめ	75
6.1	6 軸ステージ動作パラメータ	77
6.2	データ形式リスト	80
6.3	SPring-8 実験データセット	84
6.4	double イベントスペクトルフィッティング結果	86
6.5	モジュレーションファクター	88
6.6	16 keV スキャンデータの再構成のノイズ	92
6.7	混ぜ合わせの光量配分	97
6.8	偏光撮像結果	99
A.1	信号名定義	105
B.1	add 処理速度比較	110

# 第1章 はじめに

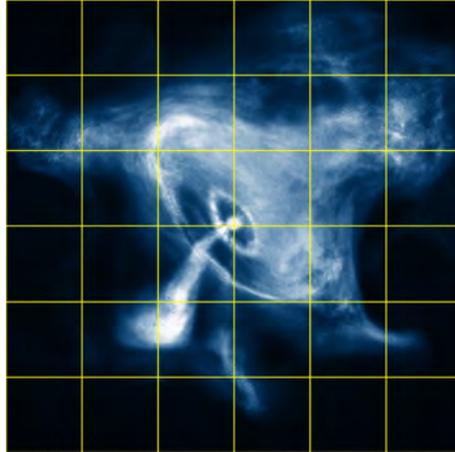


図 1.1: *Chandra* によって観測された Crab Nebula の X 線画像。NASA/CXC/SAO(<https://chandra.harvard.edu/photo/2018/crab/>) より引用した画像に 30'' 間隔格子を重ねている。

X 線天文学は、X 線を通じた観測によって宇宙の中でも激しい物理現象を探究する学問である。X 線は波長で言えば  $10^{-8}$  m 以下、エネルギーにすれば約 100 eV 以上、温度にすれば 100 万度を超える。そのため天体から放射される X 線からは、可視光観測では得られない熱く激しい天体の情報を得ることができる。X 線天文学では、ブラックホールや中性子星といったコンパクトな空間領域に高エネルギーが溜め込まれているような天体から、超新星残骸のように放射粒子が相対論的速度で移動するような構造まで、多くの天体をターゲットとして「宇宙の高エネルギーの姿」を探究している。

X 線天文学ではこれまで光の持つ情報として「時間」「空間」「エネルギー」を取り出し、宇宙でのさまざまな物理現象を解明してきた。光はこれら 3 要素に加えて「偏光」という第 4 の情報を含んでいる。偏光の情報を得ることが出来れば、散乱体の幾何構造やシンクロトロン光を放射する天体の磁場構造を直接的に解明することが可能になり、期待されるサイエンスは大きい。X 線帯域での偏光観測は技術的な困難が伴うことから、未だ発展途上の段階である。これまで X 線帯域での偏光観測には 20 keV 以上の硬 X 線帯域で *PoGO+* (Chauvin et al., 2017)、*Hitomi*/SGD (Hitomi Collaboration, 2018)、*X-Calibur* (Abarr et al., 2021) などのミッションが行われてきた。また 10 keV 以下の軟 X 線帯域をターゲットとした *IXPE* (Imaging X-ray Polarimetry Explorer) 衛星 (Weisskopf et al., 2016) が本修論執筆中の 2021 年 12 月に打ち上げられたところであり、X 線帯域では初となる撮像を含めた偏光観測に期待が寄せられている。しかしながら、非熱的な放射が卓越する 10 keV 以上のエ

エネルギー帯域において、撮像と偏光観測を同時に達成し、空間領域を切り分けて偏光度と偏光角を出すようなミッションは未だ実現していない。

我々は 10 keV から 30 keV のエネルギー帯域において初の偏光撮像を目指したプロジェクト *cipher* (Coded Imaging Polarimetry of High Energy Radiation) を提唱し (Odaka et al., 2020)、開発を進めている。*cipher* では超小型衛星 CubeSat 内に撮像偏光計を展開し、短期間かつ低コストのコンパクトな開発で観測の実現を目指している。図 1.2 にコンセプト模型図を示す。撮像偏光計は大きく 2 本の柱からなる。一つ目の柱は、X 線検出器として CMOS イメージセンサ (以下単に CMOS センサとも表記する) を採用することである。近年の技術進歩によって低ノイズで微小なピクセルピッチを持つ CMOS センサが登場してきており、X 線偏光を測定することが可能になってきている。二つ目の柱は、撮像システムとして符号化開口マスクを採用することである。X 線観測で主流であるミラーによる撮像と異なり、直接的に光源の情報を得ることが出来るほか、CubeSat サイズの小さい体積での実現が可能であるという利点がある。また *cipher* では、最も明るい X 線天体である Crab Nebula (図 1.1、以下単に Crab ともいう) を主なターゲット天体として考えている。Crab Nebula はシンクロトロン放射 (2.2 節で後述) で光る天体であり、X 線帯域で偏光していることが *PoGO+* などのミッションを通して示されているが (Chauvin et al., 2018)、Crab の空間領域を分解して場所ごとの偏光度を出した例はまだない。

*cipher* では先行研究 (春日, 2019; 會澤, 2020; 畠内, 2021) で述べられている通りの開発が進められてきた。ビーム照射試験によって複数の種類の CMOS センサが X 線の偏光感度を持つこと、また CubeSat サイズの符号化開口マスクでの撮像系による撮像が可能であることが実証されてきた。しかしながら、センサからのデータ読み出しから、偏光撮像の最終結果を出力するまでの一貫したデータ処理フロー (図 1.3) が未完成であった。特に未着手であったのが X 線検出に特化した実用的な読み出しシステムの構築と、偏光解析と撮像を統合する解析手法の確立である。従来の読み出しシステムは可視光用の製品を用いていたため、X 線検出の読み出し系としてカスタマイズされていなかった。畠内 (2021) では X 線検出に特化した読み出しボード ZDAQ-SoC の開発が進められたが、連続撮影時にフレームあたり約 1.0 s のデッドタイムが生まれてしまっていた。信号が微弱で時間変動を捉える必要がある天体観測ではこのデッドタイムは致命的であり、実効的な観測効率の観点からもデッドタイムを短縮する必要がある。また撮像と偏光解析それぞれが可能であることは先行研究で確かめられてきたが、それらを組み合わせ、空間領域を分解して場所ごとの偏光度を出す解析はまだ行われていない。この解析のためには天球面領域ごとの光源強度をある程度正確に出す必要があるが、広がった天体を模した撮像を行なう際に再構成手法に伴うノイズが信号の約 50 % 出てしまうことが判明している (図 5.1)。このノイズを低減するような画像再構成手法を考案し、偏光情報を撮像に取り入れることが撮像偏光計の解析手法上での重要な課題である。

*cipher* の一貫したデータ処理フローの完成を目指し、FPGA 実装による新たな読み出しシステムの構築と、画像再構成法の精度向上、および偏光情報を含めた画像再構成手法の開発に取り組んだ。そして新たに開発した読み出しシステムと符号化開口の再構成アルゴリズムの有用性を SPring-8 での偏光 X 線照射試験を通して実証した。本修論は次のような構成で議論していく。まず、2 章で本修論の基礎となる偏光と CMOS センサの原理をまとめる。次に 3 章で実際に使用する CMOS センサと読み出し系、解析フローの説明を

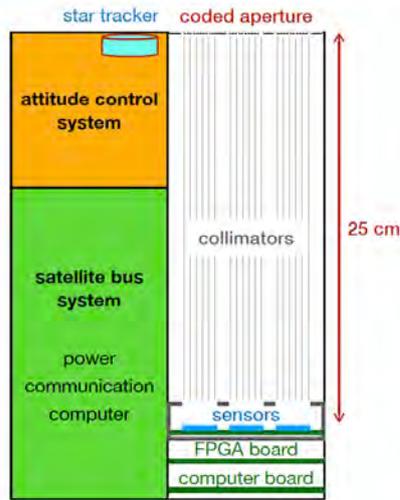


図 1.2: *cipher* で考案中の超小型衛星のモデル図。Odaka et al. (2020) より引用。

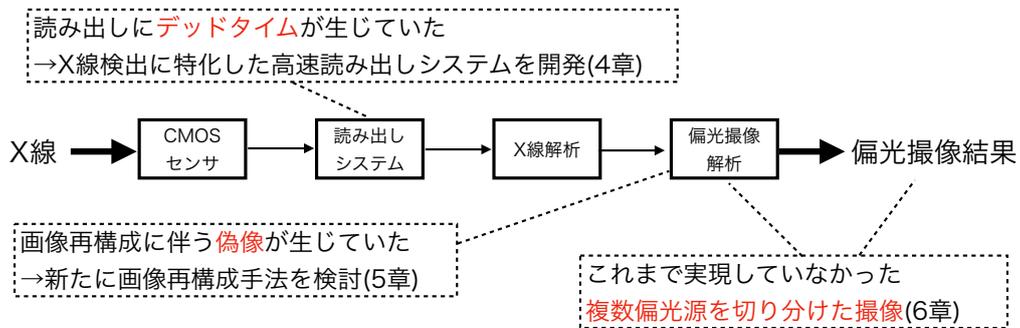


図 1.3: *cipher* のデータ処理フローの概要と本修論での改善点。

行い、無偏光 X 線に対するセンサの応答を評価する。4 章では課題であった読み出しシステムのアルゴリズム開発から FPGA による回路設計と実装までを行い、天体観測を意識したデッドタイムの短縮を目指す。さらに 5 章では撮像系を扱い、画像再構成法として従来用いられていた相関法に加えて、新たに Expectation-Maximization (EM) アルゴリズムをシミュレーションで検討し、画像再構成時に問題になるノイズの低減を目指す。そして 6 章では新たに開発した読み出しシステムと符号化開口の再構成アルゴリズムの有用性を実際の偏光 X 線照射試験を通して実証し、これまで *cipher* で実現していなかった偏光情報を含めた画像再構成を行う。

## 第2章 偏光 X 線とその検出

この章では、偏光 X 線と CMOS センサに関する原理を、*cipher* の観点から説明する。

### 2.1 偏光

この節は主に Rybicki and Lightman (1986) を参考になっている。

#### 2.1.1 偏光と偏光度

まず偏光について説明する。Maxwell 方程式の  $z$  方向に進行する電場の単色平面波解は、角周波数  $\omega$  を用いて以下のように表される。

$$\vec{E} = (E_1 \vec{e}_x + E_2 \vec{e}_y) \exp(-i\omega t) \quad (2.1)$$

ここで複素数  $E_1, E_2$  を振幅  $E_{0x}, E_{0y}$  と位相  $\phi_x, \phi_y$  を用いて  $E_1 = E_{0x} e^{i\phi_x}, E_2 = E_{0y} e^{i\phi_y}$  と表し、電場の  $x, y$  成分を実数で表すと以下のようなになる。

$$E_x = E_{0x} \cos(\omega t - \phi_x), \quad E_y = E_{0y} \cos(\omega t - \phi_y) \quad (2.2)$$

$(E_x, E_y)$  は時間によって変化するが、振幅や位相次第で  $xy$  平面上の電場ベクトルの描き方が変わってくる。この電場振動の偏り具合が偏光である。 $(E_x, E_y)$  は一般に楕円を描くため、楕円偏光と呼ばれる。

自然光はさまざまな  $E_0, \phi$  を持つ光子の重ね合わせであり、偏光成分と無偏光成分が足し合わされたものと見なすことができる。偏光度は「どれくらい偏光しているか」の指標であり、光の全強度に対する偏光成分強度の比で定義される。すなわち、偏光成分の強度  $I_{\text{pol}}$  と無偏光成分の強度  $I_{\text{unpol}}$  を用いて、偏光度  $\Pi$  は

$$\Pi = \frac{I_{\text{pol}}}{I_{\text{pol}} + I_{\text{unpol}}} \quad (2.3)$$

と表される。

## 2.1.2 ストークスパラメータ

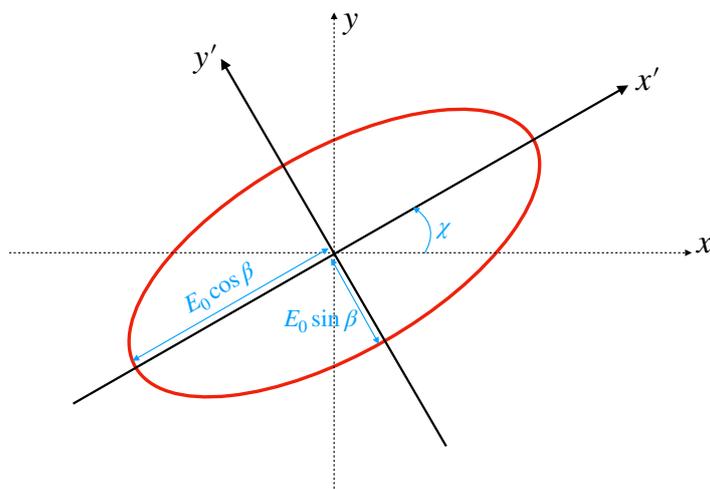


図 2.1: 電場が楕円をなす様子。

偏光成分の楕円の形状を表現したのがストークスパラメータである。図 2.1 のような楕円偏光を考える。観測系が  $(x, y)$ , 楕円の正面系が  $(x', y')$  である。 $(x', y')$  系で電場の時間変化は、 $E_0, \beta$  ( $-\frac{\pi}{2} \leq \beta < \frac{\pi}{2}$ ) をパラメータとして

$$E'_{x'} = E_0 \cos \beta \cos \omega t \quad (2.4)$$

$$E'_{y'} = -E_0 \sin \beta \sin \omega t \quad (2.5)$$

で表される。パラメータ  $\beta$  は「楕円のつぶれ具合」を示す。特殊な場合として、 $\beta = \pm \frac{\pi}{4}$  のときは円を描くので円偏光、 $\beta = 0, -\frac{\pi}{2}$  のときは直線を描くので直線偏光と呼ばれる。また、 $\beta$  の正負は右回りか左回りかに対応する。

$(x', y')$  系が  $(x, y)$  系から角度  $\chi$  傾いているとする。 $\chi$  は偏光角と呼ばれる。このとき、電場の一般形は

$$E_x = E_0 (\cos \beta \cos \chi \cos \omega t + \sin \beta \sin \chi \sin \omega t) \quad (2.6)$$

$$E_y = E_0 (\cos \beta \sin \chi \cos \omega t - \sin \beta \cos \chi \sin \omega t) \quad (2.7)$$

と表現できる。すなわち、 $E_0, \beta, \chi$  の 3 つのパラメータで偏光を表現できる。ストークスパラメータはこの 3 つの自由度を強度の次元で表現したものであり、次のように定義される。

$$I_{\text{pol}} \equiv E_0^2 \quad (2.8)$$

$$Q \equiv E_0^2 \cos 2\beta \cos 2\chi \quad (2.9)$$

$$U \equiv E_0^2 \cos 2\beta \sin 2\chi \quad (2.10)$$

$$V \equiv E_0^2 \sin 2\beta \quad (2.11)$$

すぐに分かるように、 $Q, U, V$  を用いると、偏光成分の強度  $I_{\text{pol}}$  は

$$I_{\text{pol}} = \sqrt{Q^2 + U^2 + V^2} \quad (2.12)$$

で表され、偏光度  $\Pi$  は

$$\Pi = \frac{I_{\text{pol}}}{I} = \frac{\sqrt{Q^2 + U^2 + V^2}}{I} \quad (2.13)$$

で表される。逆にストークスパラメータを用いると、

$$E_0 = \sqrt{I_{\text{pol}}} \quad (2.14)$$

$$\beta = \frac{1}{2} \arcsin \frac{V}{I_{\text{pol}}} \quad (2.15)$$

$$\chi = \frac{1}{2} \arctan \frac{U}{Q} \quad (2.16)$$

と、各パラメータを表現できる。ストークスパラメータについては6.6節で再び議論する。

## 2.2 偏光 X 線の放出過程

### 2.2.1 シンクロトロン放射

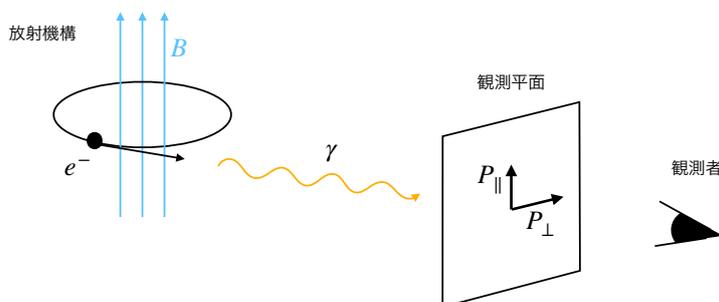


図 2.2: シンクロトロン放射の観測

相対論的な速度を持つ電子が磁場中を運動すると、磁場と垂直な方向に加速度を持ち、運動平面内に光子を放出する (図 2.2)。これがシンクロトロン放射である。放射光は電子の加速度方向に強い電場成分を持つことになるため、観測平面内では磁場に垂直な方向に強く偏光した放射光を観測することになる。観測光の単位周波数あたりの放射強度の偏光が磁場に並行な成分を  $P_{\parallel}(\omega)$ , 磁場に垂直な成分を  $P_{\perp}(\omega)$  とすると、偏光度  $\Pi(\omega)$  は

$$\Pi(\omega) = \frac{P_{\perp}(\omega) - P_{\parallel}(\omega)}{P_{\perp}(\omega) + P_{\parallel}(\omega)} = \frac{G(x)}{F(x)} \quad (2.17)$$

で表される (Rybicki and Lightman, 1986)。ただし、 $x$  は磁場方向と電子の進行方向のなす角を  $\alpha$  とし、 $\omega_c = 3\gamma^2 qB \sin \alpha / 2mc$  として  $x = \omega / \omega_c$  で表される量であり、 $F(x), G(x)$  は変形ベッセル関数  $K$  を用いて表される以下の関数である。

$$F(x) = x \int_x^{\infty} K_{\frac{5}{3}}(\xi) d\xi \quad (2.18)$$

$$G(x) = x K_{\frac{2}{3}}(x) \quad (2.19)$$

このことは、逆にシンクロトロン光の偏光を観測することができれば放射機構の磁場構造を推定できることを物語っている。

## 2.2.2 逆コンプトン散乱

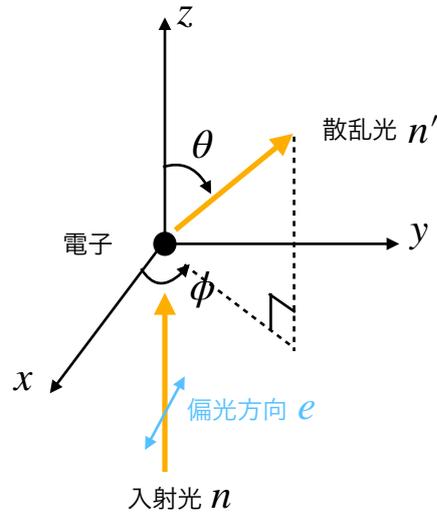


図 2.3: 逆コンプトン散乱の電子静止系での概念図

逆コンプトン散乱は、相対論的な速度で運動する電子が光子と衝突し、光子にエネルギーが受け渡されて散乱する現象である。電子静止系において、図 2.3 のように  $x$  軸に平行な偏光を持った入射光が  $z < 0$  から  $z$  軸に平行に入射し、原点にある電子に衝突して散乱するとする。この散乱の微分断面積  $d\sigma/d\Omega$  は Klein-Nishina の式 (McNamara et al., 2008) として知られており、散乱前後の光子のエネルギー比を  $\eta$  として、

$$\frac{d\sigma}{d\Omega} \propto \eta + \eta^{-1} - 2 \sin^2 \theta \cos^2 \phi \quad (2.20)$$

で表される。式 (2.20) は  $\phi = \frac{\pi}{2}$  で最大になる。すなわち、偏光方向と垂直な方向に散乱されやすいことになる。またこのとき、散乱光は散乱面と平行な方向に偏光し (Angel, 1969)、その偏光度は

$$\Pi = \frac{\sin^2 \theta}{\eta + \eta^{-1} - 2 \sin^2 \theta \cos^2 \phi} \quad (2.21)$$

で表される (Dolan, 1967)。このことは、逆コンプトン散乱からの散乱光の偏光を観測することによって、散乱体の視直径が小さくてもその幾何構造を推定できることを示している。

## 2.3 偏光 X 線の検出

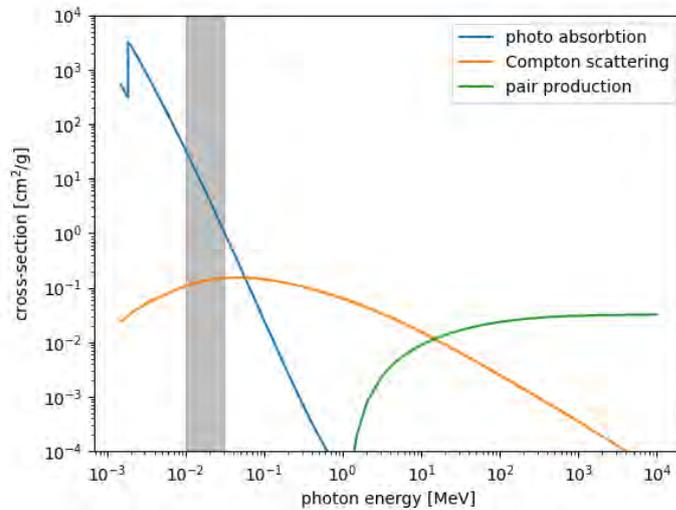


図 2.4: Si と光子の反応断面積。NIST XCOM(<https://physics.nist.gov/PhysRefData/Xcom/html/xcom1.html>) のデータベースから作成した。網掛け部は本研究のターゲットの帯域 (10-30 keV) である。

X線を検出するためには、入射光子が検出器と相互作用する必要がある。物質と光子の相互作用には主に光電効果、コンプトン散乱、電子陽電子対生成の3つがあり、入射光子のエネルギーによって支配的な反応は変わる。特に本修論で着目するSiの光子に対する微分断面積のエネルギー応答を図2.4に示す。X線の領域においては光電効果とコンプトン散乱が支配的である。以下はこの二つの反応について紹介する。

### 2.3.1 光電効果

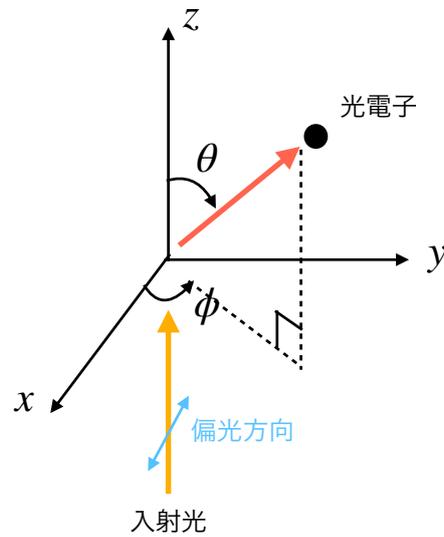


図 2.5: 光電効果の概念図

数十 keV 程度の X 線に対しては、Si 検出器との反応は光電効果が支配的である。光電効果では、光子が物質原子に束縛された電子にエネルギーを受け渡し、電子は自由電子として飛び出す。図 2.5 のように、偏光方向が  $x$  方向である光子が  $z < 0$  から  $z$  軸と平行に電子に入射する場合を考える。このとき、光電効果の微分断面積は以下のような角度依存性を持つ (Fraser et al., 1989)。

$$\frac{d\sigma}{d\Omega} \propto \frac{\sin^2 \theta \cos^2 \phi}{(1 - \beta \cos \theta)^4} \quad (2.22)$$

すなわち、入射光子の偏光角方向に光電子が飛び出しやすいことになる。この光電子の動きをトラッキングできれば、偏光を測定することが可能になる。

### 2.3.2 コンプトン散乱

入射光子エネルギーが約 100 keV を超えると、コンプトン散乱が支配的な反応になる。コンプトン散乱では光子が自由電子に衝突し、光子と電子が入射光子のエネルギーに応じた方向に散乱される。コンプトン散乱の微分断面積は、逆コンプトン散乱と同様の式 (2.20) で表される。 $\phi = \frac{\pi}{2}$  で最大になるため、光電効果と異なり偏光角と垂直な方向に放出されやすいことになる。

## 2.4 CMOS イメージセンサの原理

本修論では CMOS イメージセンサを検出器として用いる。本節では Sze (2002) を参考に、その動作原理について説明する。

## 2.4.1 半導体検出器の原理

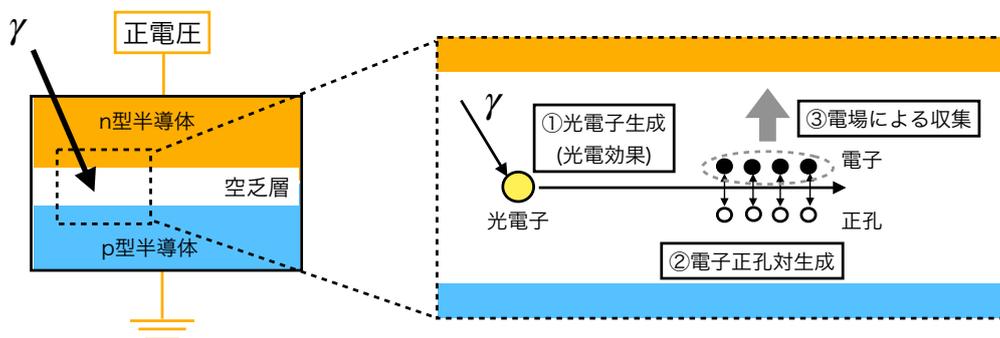


図 2.6: 半導体検出器の原理

まず CMOS に限らず、一般的な半導体検出器の原理を説明する。図 2.6 に概略図を示す。検出器として基本となる構造は pn 接合である。キャリアとして正孔を多く含んだ p 型半導体と、電子を多く含んだ n 型半導体を接合させると、接合面付近は正孔と電子が再結合し、キャリアが存在しなくなる。この領域を空乏層または空乏領域という。半導体検出器では n 型半導体に正電圧を、p 型半導体に負電圧を印加し、空乏領域を広げている。

実際の半導体検出器による X 線の検出の流れを説明する。まず X 線が空乏領域に入射し、光電効果などで光電子を生じさせる。この光電子は Si 結晶中を動き、価電子帯において原子核からの束縛が緩い他の電子を多数励起させ、光電子の運動エネルギーに比例する数の電子正孔対を生成する。この多数の電子群は電子雲と呼ばれる。この電子雲を内部電場によって収集し、その電荷量 (電流量) を測定することで到来光子のエネルギーを推定する。なおこのとき電子雲は電場によるドリフトと熱拡散という物理現象を経ており、特に微小ピクセルを持つ検出器の場合、電子雲の熱拡散が複数ピクセルにまたがることがある。また空乏層が厚いほど光子が反応しやすくなるため、検出効率は良くなる。

電荷の読み出し方法で半導体検出器の種類は分けられる。歴史的に X 線帯域の撮像には主に CCD が用いられてきたが、可視光用 CMOS イメージセンサの技術が劇的に向上したことで、CMOS の X 線帯域への応用に注目が集まっている。

## 2.4.2 MOS と MOSFET

CMOS を考える上で基礎となる MOS ダイオードの構造を図 2.7 に示す。金属 (Metal) の電極、酸化物 (Oxide) である絶縁体、半導体 (Semiconductor) がこの順に並んでいる。なお、絶縁体には  $\text{SiO}_2$  が、半導体には Si が用いられることが多く、特に本修論では半導体は Si のみについて考える。図 2.7 では p 型半導体を用いた場合を示している。p 型半導体を接地した状態で電極に負電圧を印加すると、半導体内の正孔が絶縁体の方に移動し、分布に偏りが生まれる (図 2.7a)。逆に電極に正電圧を印加すると、正孔は絶縁体から離れていき、絶縁体との境界付近に電荷の少ない空乏層が生まれる (図 2.7b)。さらに大きな

正電圧を印加すると、絶縁体との境界付近に n 型の領域が生まれる (2.7c)。これが反転層である。

MOS Field-Effect Transistor (MOSFET) は MOS 構造を利用したトランジスタである。MOSFET の構造を図 2.8 に示す。p 型の MOS の両側に n 型半導体を埋め込み、それぞれに電極を付与したものである。元々の p 型の電極をゲート電極、付与した電極をソース電極、ドレイン電極という。ソースとドレインが n 型の MOSFET を n-チャンネル MOSFET、n と p を逆転させた MOSFET を p-チャンネル MOSFET という。

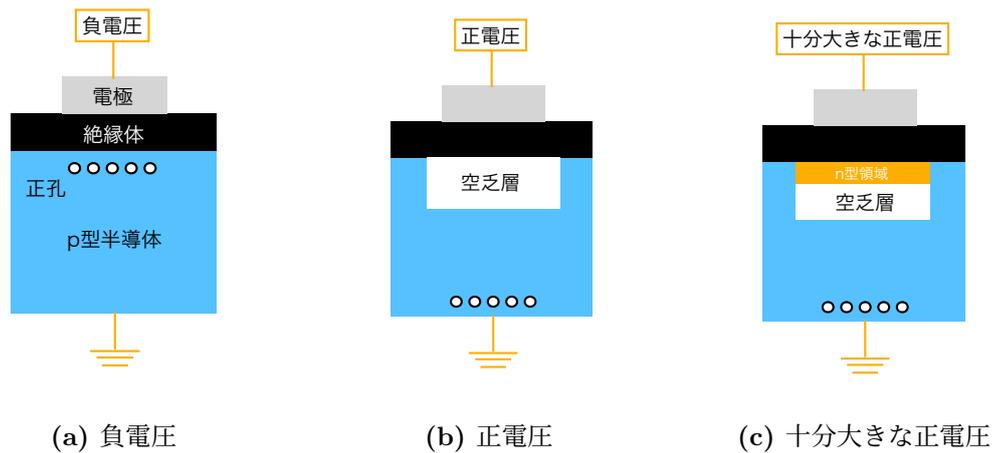


図 2.7: MOS ダイオードの構造

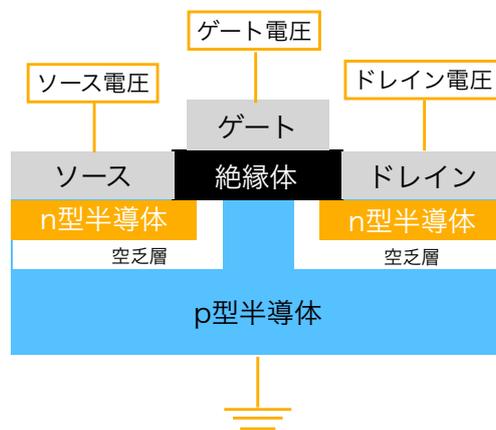


図 2.8: n-チャンネル MOSFET の構造

### 2.4.3 CMOS

Complementary MOS (CMOS) は n-チャンネル MOSFET と p-チャンネル MOSFET を相補的に組み合わせることで実現する。もっとも単純な CMOS 論理回路である CMOS インバータの構造を図 2.9 に示す。説明のため、電圧の高低を H(High) と L(Low) で表す。

CMOS インバータは入力の H と L を反転させる役割を持つ。まず入力が H のとき、n-チャンネル MOSFET 内の p 型半導体で反転層が生じ、ソースとドレインが導通する。アースと導通したため、出力は L となる。このとき、p-チャンネル MOSFET は特に動作しないことに注意する (図 2.7a と同じ状態)。逆に入力が L のときは p-チャンネル MOSFET 内でソースとドレインが導通するため、出力は H となる。リーク電流は導通した際の過渡状態でしか流れないため、消費電力は非常に低く抑えられる。これは CMOS の重要な特性の一つである。

CMOS イメージセンサは、CMOS インバータのような特定の機能を持つ回路を多数つなぎ合わせることによって成り立っている。図 2.10 に概略図を載せる。ピクセルと呼ばれる単位にフォトダイオード (PD)、アンプ、スイッチといった回路が含まれており、それぞれのピクセルで読み出された電流値をスイッチングによって収集している。各ピクセルで独立な読み出しが可能のため、X 線検出器として広く用いられている CCD 検出器と比較して、高速な読み出し速度を実現できる。これまで課題だったピクセルごとのゲインやノイズのばらつきは近年の技術の向上により克服されてきており、天文学への応用に注目が集まっている。またピクセルサイズが数  $\mu\text{m}$  程度の CMOS 検出器も登場してきており、これは光電子のトラッキングに利用できる。さらに必要な消費電力を低く抑えられることから、CubeSat に代表される超小型衛星への搭載が十分に可能である。

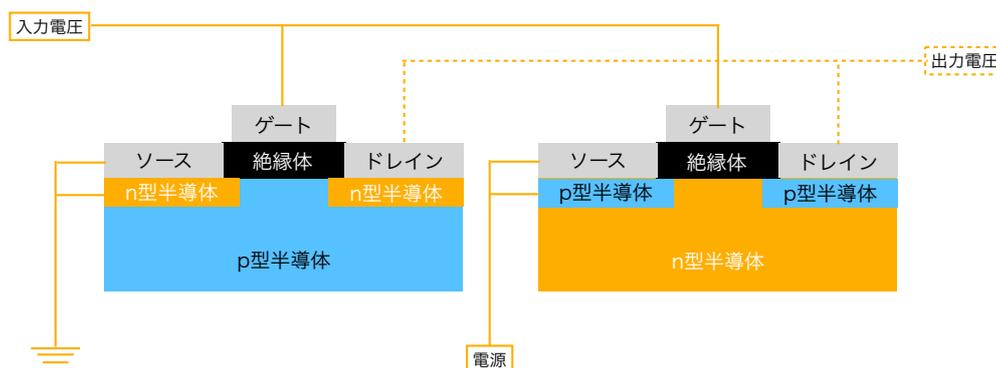


図 2.9: CMOS インバータの構造

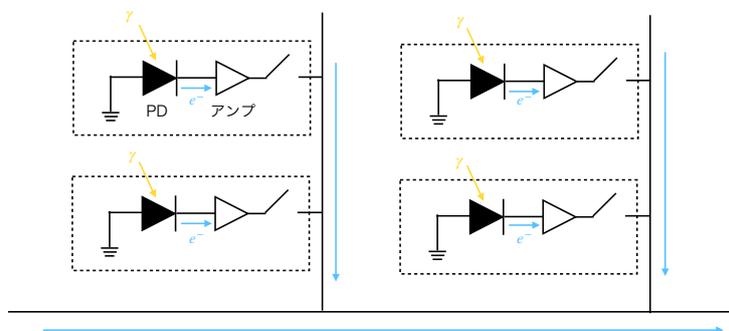


図 2.10: CMOS イメージセンサの概略図

## 2.5 2章のまとめ

偏光は光の電場ベクトルの偏り具合を示すパラメータであり、一般の偏光はストークスパラメータで表現できる。直線偏光は偏光度と偏光角の2つのパラメータで定量化できる。特に偏光 X 線を放出する物理現象にはシンクロトロン放射や逆コンプトン散乱などがあり、偏光を観測することで放射体の磁場構造や散乱体の幾何構造を推定することができる。半導体物質の Si に対して 30 keV 以下の X 線帯域では光電効果が支配的な反応であり、光電効果では偏光角方向に光電子が飛び出しやすい性質を持つ。*cipher* では微小ピクセルを持つ CMOS イメージセンサで光電子の動きをトラッキングすることによって X 線の偏光検出を試みる。

# 第3章 CMOSセンサのX線検出器としての性能評価試験

読み出しシステムの構築と撮像系の検討に臨む前に、使用する CMOS イメージセンサの X 線検出器としての特性を評価する。特に検出効率は先行研究 (畠内, 2021) では大雑把な見積もりのみとなされていた。本章では特性 X 線源として従来使われていた  $^{241}\text{Am}$  に加えて新たに  $^{109}\text{Cd}$  を使用し、CMOS センサの X 線検出器としての性能評価試験を行う。解析には 4 章で改善する前の従来の解析フローを使用する。

## 3.1 使用センサと読み出し系

### 3.1.1 CMOS センサ

本修論では GPixel 社が開発した CMOS センサである GMAX0505RF を用いる (図 3.1)。可視光用で近赤外線にも感度を持つように開発されたセンサであるが、硬 X 線帯域でも偏光感度を持ち、同社の他センサと比較して X 線検出効率が高いことが先行研究 (畠内, 2021) で確かめられている。ピクセルピッチは  $2.5\ \mu\text{m}$  で、5120 ピクセル四方の計 25M ピクセル並んでいる。Bethe-Bloch の式 (e.g., Leo 1994) より計算される 20 keV の光電子の飛程がおよそ  $5\ \mu\text{m}$  であるから、このピクセルサイズは光電子のトラッキングに適している。センサ表面には厚さ  $700\ \mu\text{m}$  の保護ガラスがついており、データ取得を行なう際はガラスによる減衰を考慮する必要がある。1 フレーム分の撮影を行うと、各ピクセルで露光時間中に収集された電荷の量が 12 bit のデジタル値として出力される。撮影をする際はゲインと露光時間を設定可能であり、ゲインは 0.5 刻みで 1 倍から 16.5 倍までの範囲で、露光時間は本読み出し系の入力クロック 50 MHz の元では約  $7\ \mu\text{s}$  刻みで約  $13\ \mu\text{s}$  から約 125 s まで指定可能である。<sup>1</sup>

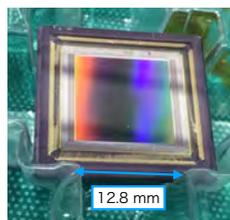


図 3.1: 本修論で使用するセンサ

<sup>1</sup>ちなみに畠内 (2021) にも露光時間が表記されているが、ソフトウェア上の実装ミスにより、実際の露光時間は表記値の 1.4 倍になっていた。本研究では修正済みである。

表 3.1: CMOS センサのメーカーによる公称値

ピクセルサイズ	2.5 $\mu\text{m}$ $\times$ 2.5 $\mu\text{m}$
ピクセル数	5120 $\times$ 5120
有効面積	12.8 mm $\times$ 12.8 mm
フレームレート	42 frames/s@ 12 bit
シャッタータイプ	グローバル
ダーク電流	1e <sup>-</sup> /pixel/s @ 30 °C
消費電力	1.1 W@ 12 bit
入力クロック周波数	40-80 MHz
データ出力方法	LVDS 18ch

### 3.1.2 読み出し系

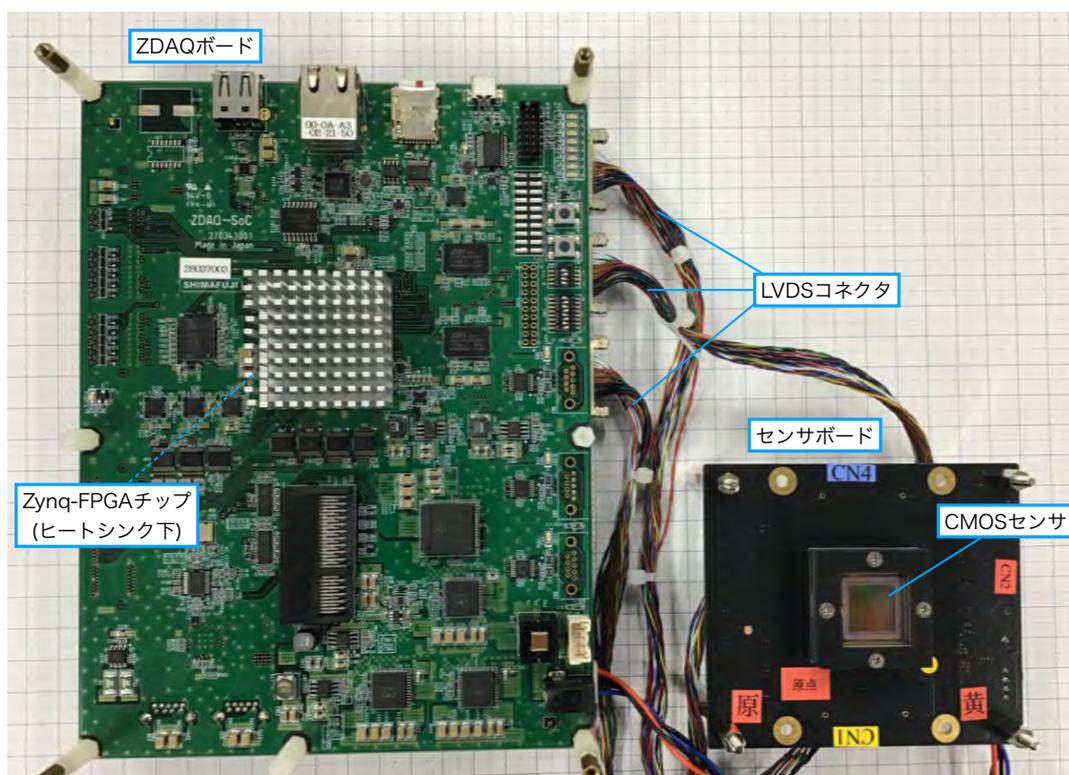


図 3.2: 読み出し系ハードウェア

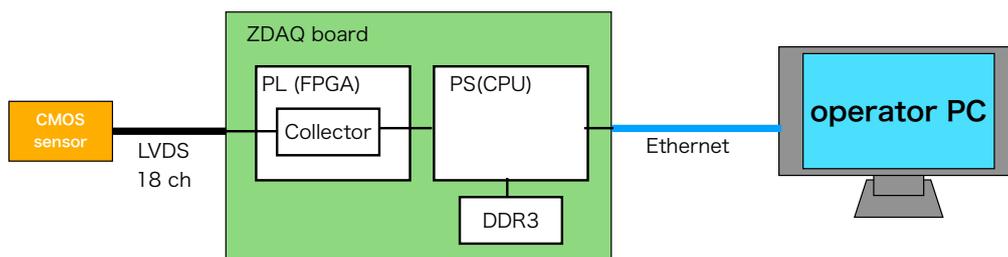


図 3.3: 読み出し系全体像

読み出し系ハードウェアの全体像を図3.2に示す。小型センサボード、拡張ボード、ZDAQ-SoC ボードの3つからなる。センサボードと拡張ボードはGPixel社製センサの衛星搭載を意識し、シマフジ電機<sup>2</sup>・大阪大学と共同開発したものである(佐久間, 2021)。またZDAQ-SoCは実際の宇宙観測ロケット実験でも採用された実績があり(Ishikawa 他, 2018)、本研究にも十分適用できると考えられる。拡張ボードは主にセンサとZDAQ-SoCボード間の入出力インターフェース調整を行なうものだが、本修論ではZDAQ-SoCボードと合わせてZDAQボードとして同一視することにする。

センサボードにはGMAX0505シリーズのセンサが設置可能であり、センサ正面方向から見て8 cm × 9 cmと超小型衛星にも搭載可能なサイズになっている。またセンサボードには表面実装型のデジタル温度センサ<sup>3</sup>が取り付けられており、センサボードの温度を±1 °Cの精度で出力可能である。ZDAQ-SoCボードにはプロセッサシステムARM-DualCore Cortex-A9(CPU, PS部ともいう)とプログラマブルロジック(FPGA, PL部ともいう)を統合したZynq-SoCを搭載しており、センサの制御を行なうことができる。実際の運用はCPU内にLinuxを立ち上げ、Linux内の専用ソフトウェアからPL部にアクセスすることで行なう(付録Bも参照)。FPGAコードは東京大学が機能と性能の仕様を策定し、シマフジ電機によって実装がなされた。センサの制御および読み出しデータのDDRメモリへの収集がなされる。読み出し用ソフトウェアは円滑な運用のため新規に開発したものであり、FPGAとのレジスタを通じた通信、DDRメモリから外部PCへのデータ転送、複数フレームの自動撮影などの役割を担っている。

実際の読み出し系の全体像を図3.3に示す。センサボードとZDAQボードはLVDSコネクタで、ZDAQボードとデータ受信PCはEthernetケーブルで接続されている。センサから読み出された25 MPixelの波高値<sup>4</sup>情報がFPGA上の収集回路によりDDRメモリへ移され、CPU内Linux上のソフトウェアでTCP通信によりデータ受信PCに転送される、という流れである。

<sup>2</sup><http://www.shimafuji.co.jp/>

<sup>3</sup>MAX6627(<https://www.mouser.jp/c/sensors/temperature-sensors/board-mount-temperature-sensors/?m=Maxim%20Integrated&series=MAX6627>)

<sup>4</sup>本修論で用いる「波高値」は、露光時間中に集められた総電荷量を指す。すなわち正確には時間波形の積分値であるが、X線天文学では慣例的に「波高値」と表現する。

### 3.2 センサ性能測定試験セットアップ

センサ性能測定試験の様子を図 3.4 に示す。可視光の混入を防ぐためセンサ全体を遮光シートで覆った上で、CMOS センサ表面から 16.2 mm 離れた位置に線源  $^{109}\text{Cd}$  または  $^{241}\text{Am}$  を設置した。各線源は表 3.2 に示す通りの特性 X 線を放出する。 $^{109}\text{Cd}$  は崩壊種が少なく、強度がデータ取得時で  $8.0 \times 10^5 \text{ Bq}$  と強いため検出効率測定に適する。一方、 $^{241}\text{Am}$  は 13-21 keV のエネルギー帯に多数の特性 X 線を持つため、エネルギー較正およびエネルギー分解能測定に適すると考える。実験は室温 (約  $25 \text{ }^\circ\text{C}$ ) で行なったが、センサの冷却系として付近から扇風機で風を送り、センサ温度を  $35 \text{ }^\circ\text{C}$  程度に保てるようにした。

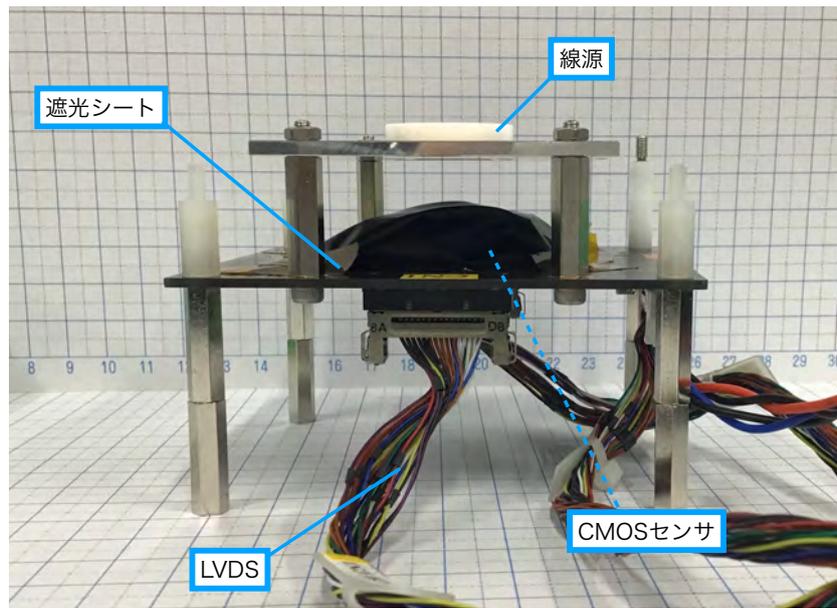


図 3.4: 性能測定実験のセットアップ

表 3.2: 使用線源の X 線帯域輝線

線源	特性 X 線	エネルギー [keV]	分岐比 (%)	磁気量子数別主要エネルギー [keV]
$^{109}\text{Cd}$	Ag $K_\alpha$	22.10	55.3	21.99 ( $K_{\alpha 2}$ ), 22.16 ( $K_{\alpha 1}$ )
$^{109}\text{Cd}$	Ag $K_\beta$	25.02	0.49	24.91 ( $K_{\beta 3}$ ), 24.94 ( $K_{\beta 1}$ ), 25.46 ( $K_{\beta 2}$ )
$^{241}\text{Am}$	Np $L_\alpha$	13.93	9.6	13.76 ( $L_{\alpha 2}$ ), 13.95 ( $L_{\alpha 1}$ )
$^{241}\text{Am}$	Np $L_\beta$	17.46	1.4	16.82 ( $L_{\beta 2}$ ), 17.06 ( $L_{\beta 4}$ ), 17.75 ( $L_{\beta 1}$ ), 17.99 ( $L_{\beta 3}$ )
$^{241}\text{Am}$	Np $L_\gamma$	21.04	0.29	20.78 ( $L_{\gamma 1}$ )

\* Table of Radioactive Isotopes(<http://nucleardata.nuclear.lu.se/toi/index.asp>) より作成。

## 3.3 検出器のダーク特性評価

### 3.3.1 ペDESTALとピクセル選定

理想的な検出器では光子0の状況では検出波高値は0になるべきだが、実際には暗電流や読み出しノイズにより、イベントの来ていないときにも検出波高値は0とならない。この「イベントの来ていないときに検出される波高値」を以降、ペDESTALと呼ぶ。検出X線を解析する際は到来光子の落とすエネルギーに加えて、このペDESTALが上乘せされていることを考慮する必要がある。一般にペDESTAL特性はピクセルごとに異なるため、本修論では各ピクセルでペDESTALの値を設定することにする。

ペDESTALの決定方法として最も単純なのは、X線が来ていないとわかっている状況で撮影を行い、その撮影時の波高値をそのままペDESTALとして採用するというものである。ただし、暗電流の量は時間とともに揺らぐため、安定した値を採用するために何フレームかの平均をとったものをペDESTALとするのが一般的である。この、X線が来ていない状況で一定時間撮影したデータのことを以降ダークデータと呼ぶ。

またペDESTALとは別に、撮影を続けているうちに検出器の特定のピクセルが壊れてしまった、という状況も考えられる。この場合、壊れたピクセルでX線イベントを検出したとしてもそのデータは信用できないと考えるのが自然である。このような状況を防ぐため、あらかじめダークデータをもとにX線検出用に使うピクセルを選定しておく作業を行なう必要がある。解析対象としたピクセルをグッドピクセル、解析対象から外したピクセルをバッドピクセルと呼ぶ。

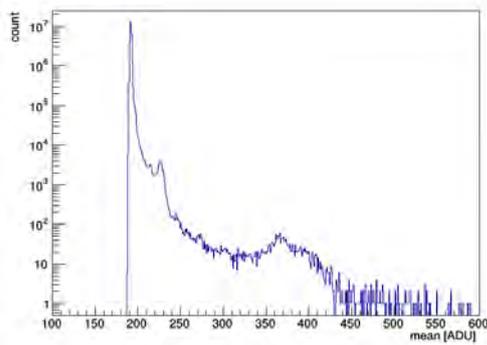
### 3.3.2 測定と結果

センサのダーク特性を評価するため、線源を設置していない状態で露光時間 600 ms, ゲイン 2.0, フレーム数 100 の連続撮影を行なった。各ピクセルの特性を理解する指標として、全フレームの平均と不偏標準偏差を出した。具体的には、 $N(= 100)$  フレーム撮影のピクセル  $l$  の  $i$  フレーム目の波高値を  $H_{i,l}$  として、

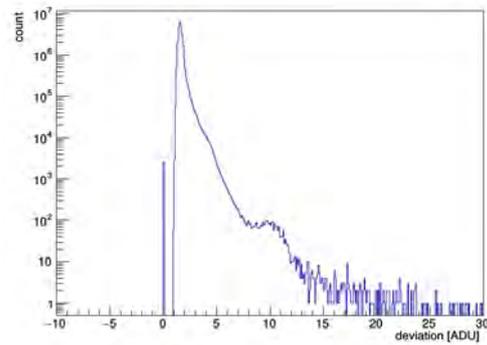
$$\text{mean}_l = \frac{1}{N} \sum_{i=0}^{N-1} H_{i,l} \quad (3.1)$$

$$\text{deviation}_l = \sqrt{\frac{1}{N-1} \sum_{i=0}^{N-1} (H_{i,l} - \text{mean}_l)^2} \quad (3.2)$$

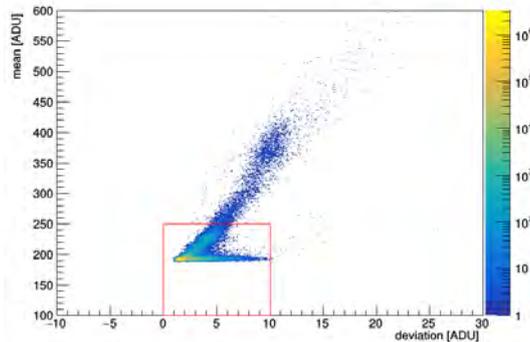
を全てのピクセルに対して計算した。計算結果の分布は図 3.5 のようになった。



(a) 波高値平均 (式 (3.1)) の分布



(b) 波高値不偏標準偏差 (式 (3.2)) の分布



(c) 縦軸に波高値平均 (式 (3.1))、横軸に不偏標準偏差 (式 (3.2)) をとった 2 次元分布。色で示した z 軸がピクセル数である。また、赤線はピクセル選定基準の範囲 (表 3.3) を示す。

図 3.5: ダークデータに対しピクセルごとに平均と不偏標準偏差を計算した分布。全点を合計すると  $5120 \times 5120$  点になる。

### 3.3.3 ピクセル選定

図 3.5 を見ると、ピクセルの多くは平均が約 200 ADU 前後、不偏標準偏差が約 2.0 ADU 前後に集まっている様子がわかる。この領域から離れているピクセルは少数だが存在し、(1) 平均と不偏標準偏差ともに高いものと (2) 平均は 200 ADU 前後だが不偏標準偏差が高いものの 2 種類が主に存在する。不偏標準偏差が大きいピクセルは X 線が入射しても検出波高値の揺らぎが大きくなってしまおうと考えられ、データの信頼性に影響を及ぼす可能性がある。また、平均が大きいピクセルは測定に重大な影響を及ぼすとは考えにくいものの、大多数のピクセルと異なる特性を持っていると考えるべきである。これらのことから、平均と不偏標準偏差の範囲を制限してグッドピクセルの選定を行なう。範囲に関しては理論的な裏付けはなく、解析者の決定するパラメータとなる。

本実験でのピクセルの選定基準を表 3.3 に示す。表 3.3 の条件で選定を行なうと解析対象外のピクセル数は 7525 で全ピクセルの 0.03 % であり、X 線解析に大きな影響は及ぼさないと考えられる。

表 3.3: グッドピクセル選定条件

平均 [ADU]	100-250
不偏標準偏差 [ADU]	0.0-10.0

## 3.4 検出器の X 線応答評価

### 3.4.1 解析フロー

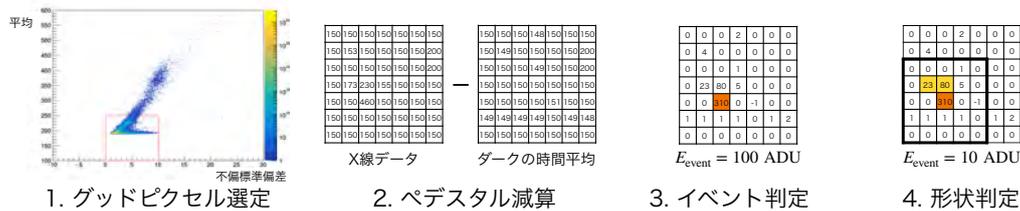


図 3.6: X 線データの解析フロー。波高値などは説明のため設定した値である。

X 線検出データの解析は、以下の流れにしたがって行なう (図 3.6)。この手法はピクセル式半導体検出器での X 線解析で一般に用いられる (e.g., Tanaka et al., 2018)。

(1) グッドピクセルの選定

3.3 節で決定したグッドピクセル以外でのデータは使わないと設定する。

(2) ペDESTALの除去

各ピクセルに対し、3.3 節で決定したペDESTAL、すなわちダークデータの波高値のフレーム平均 (式 (3.1)) を X 線取得データから引く。

(3) イベント位置の特定

ある閾値  $E_{\text{event}}$  を定める。ペDESTAL減算後、全グッドピクセルを走査し、波高値が  $E_{\text{event}}$  を超えるピクセルを抜き出す。この閾値を超えたピクセルで X 線イベントが検出されたとみなす。後の設定値と区別するため、 $E_{\text{event}}$  をイベント閾値と呼ぶ。

(4) イベント形状の判定

X 線イベントはいくつかのピクセルにまたがって広がるため、(3) で特定したピクセルそれぞれの周辺ピクセルも調べる必要がある。(3) とは別の閾値  $E_{\text{split}}$  を定める。 $E_{\text{event}}$  を超えたピクセルそれぞれに対し、周囲  $5 \times 5$  ピクセルを走査し、 $E_{\text{split}}$  を超えたピクセルの位置と波高値の情報をイベントピクセルに付与する。この周辺ピクセル情報を含めて 1 イベントとみなす。 $E_{\text{split}}$  をスプリット閾値と呼ぶ。

イベント閾値とスプリット閾値は解析フローを実行する際に決定する必要があるパラメータである。イベント閾値は「X 線が到達したと決定する境界」であり、ノイズより十分大きく、かつ到来光子エネルギーよりも十分小さい必要がある。逆にこの条件を満たし

ていればイベント閾値を多少上下させてもイベント形状、エネルギー分解能、検出効率などの解析結果に与える影響は小さい。一方でスプリット閾値はイベント形状を決定するパラメータであり、偏光解析などに影響を及ぼしうる。

この解析フローで得られるのはイベントの配列である。各イベントは次の情報を持っている。

- 時間

同条件で複数フレームを連続して撮影した場合、各イベントは何フレーム目かの情報を持っており、これはイベントの時間的な情報とみなすことができる。「連続撮影を始めてから何フレーム目か (0-indexed)」を今後、frame ID と表現する。当然ながら、frame ID が同じイベントは時間的に区別することはできない。

- 空間

空間的情報として、2次元検出器上の検出ピクセル位置が得られる。今後、検出ピクセルの  $x$  座標を  $p_x$ 、 $y$  座標を  $p_y$  で表す。ただし、 $p_x, p_y$  は整数であり、GMAX0505RF では  $0 \leq p_x < 5120$ ,  $0 \leq p_y < 5120$  である。

- エネルギー

線形性が良い場合、ペDESTAL減算後の波高値は光子が落としたエネルギーの定数倍とみなすことができる。スプリット閾値を超えた周辺ピクセル分の波高値も足し合わせると X 線イベントひとつ分のエネルギーと考えることができる。この周辺ピクセルとの和をとった波高値を今後 sum PH と表現する。単位は analog-to-digital units (ADU) である。

- イベント形状と周辺波高値分布

イベント形状は光電子の動きをトラッキングしたものとみなすことができる。また周辺波高値の分布は光電子の動きを表したものと考えることができ、偏光の情報はここに含まれる。イベント周囲でスプリット閾値を超えたピクセルの数を weight と呼ぶ。weight ごとのイベント形状の例を図 3.7 に示す。便宜上、weight が 1 のイベントを single イベント、weight が 2 のイベントを double イベント、weight が 3 のイベントを triple イベント、weight が 4 以上のイベントを extended イベントと呼ぶ。特に double イベントは光電子の動きを単純に表していると考えられるため、偏光解析によく用いられる。double イベントの中でも  $x$  方向に広がったイベントを H-type (Horizontal) イベント、 $y$  方向に広がったイベントを V-type (Vertical) イベントと呼ぶ。

以降、このイベントの配列を「イベントリスト」と表現する。また各イベントの持っている frame ID や sum PH といった情報を「カラム」と表現する。イベントリストがあれば各カラムに注目したヒストグラムを作成することによって、スペクトル (sum PH) やライトカーブ (frame ID)、プロファイル ( $p_x, p_y$ ) といった天体解析に必要な情報を得ることができる。

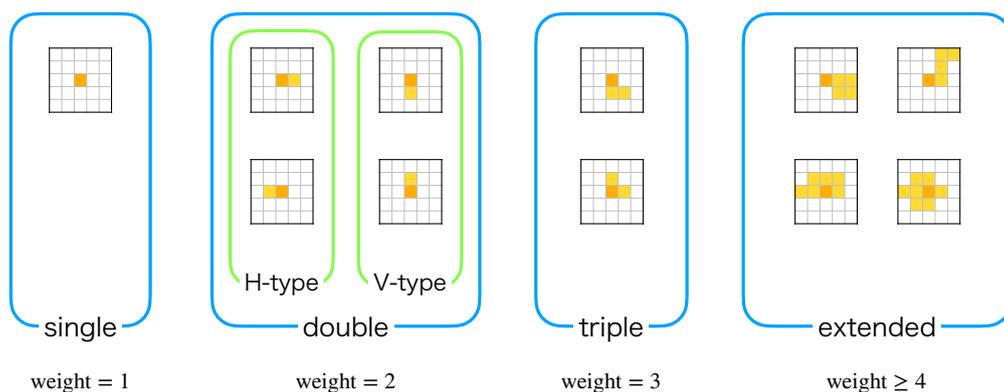


図 3.7: イベント形状の種類

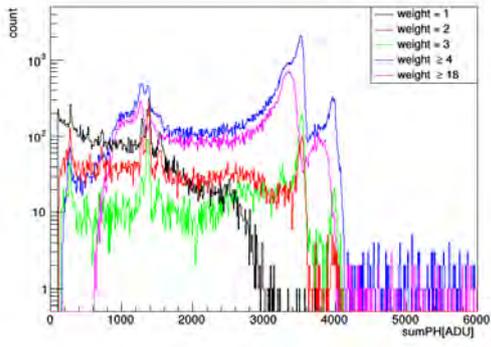
### 3.4.2 測定と結果

線源  $^{109}\text{Cd}$ ,  $^{241}\text{Am}$  をそれぞれ設置した状態で、露光時間 600 ms, ゲイン 2.0, フレーム数 1000 の連続撮影を行なった。フレーム数以外はダーク取得時と同じ設定である。グッドピクセル選定とペDESTAL減算は 3.3 節のデータを用いて行なった。解析フロー上でのイベント閾値とスプリット閾値は先行研究 (會澤, 2020; 畠内, 2021) を参考に、 $E_{\text{event}} = 100 \text{ ADU}$ ,  $E_{\text{split}} = 10 \text{ ADU}$  とした。

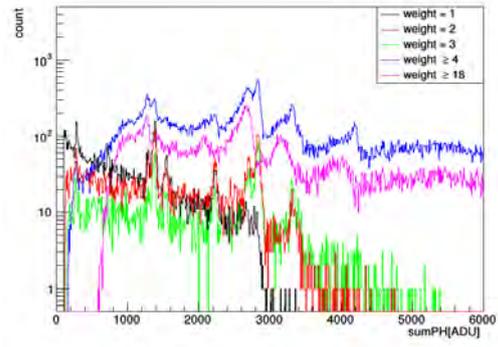
$^{109}\text{Cd}$ ,  $^{241}\text{Am}$  の検出したスペクトルを図 3.8 に、他の結果を図 3.10 に示す。図 3.8 では weight ごとにスペクトルを分けている。このうち double イベントのスペクトルから、ピークと考えられる箇所をガウシアンでフィッティングする。具体的には  $C, \sigma, E_0$  をパラメータとして、

$$f(E) = C \exp\left(-\frac{(E - E_0)^2}{2\sigma^2}\right) \quad (3.3)$$

で  $E$  の範囲を絞ってフィッティングする。厳密にはたとえば  $^{109}\text{Cd}$  の  $K_{\alpha 1}$  線と  $K_{\alpha 2}$  線は区別して 2 つのガウシアン之和になっていると考えてフィッティングするべきだが、この効果によるピーク中心のズレがエネルギー較正に大きな影響は与えないと考え、エネルギーの近いピークは単純にまとめてガウシアンでフィッティングしている。フィッティングの様子を図 3.9 に、フィッティング結果を表 3.4 に示す。なお、図 3.8 で  $\text{weight} \geq 18$  のスペクトルは  $\text{weight} = 2$  のスペクトルと比較して、ピークが低エネルギー側にずれている様子が見られる。これは広がったイベントで光電子が落としたエネルギーを集めきれない効果だと考えられるが、本修論では double イベントに特に注目するとして深くは扱わない。

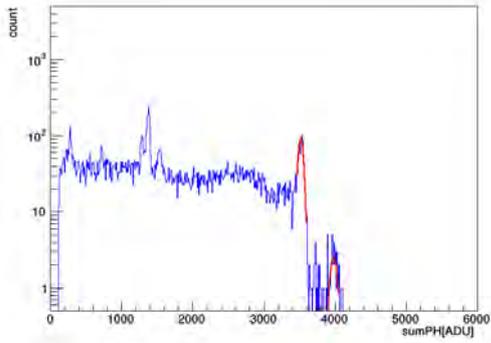


(a)  $^{109}\text{Cd}$

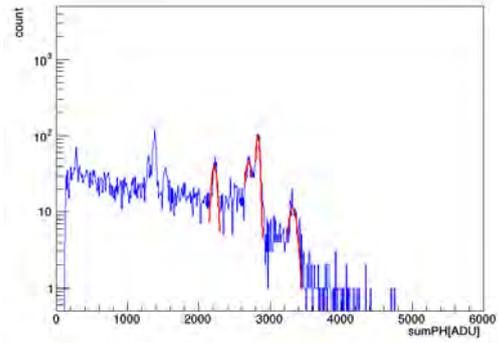


(b)  $^{241}\text{Am}$

図 3.8: weight 別の検出スペクトル



(a)  $^{109}\text{Cd}$



(b)  $^{241}\text{Am}$

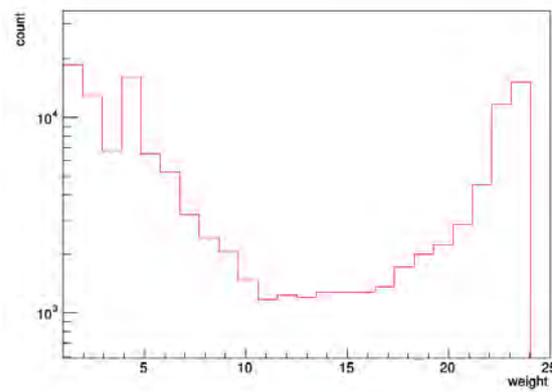
図 3.9: double イベントのガウシアンフィッティングの様子

表 3.4: double イベントスペクトルのガウシアンフィッティング範囲と結果

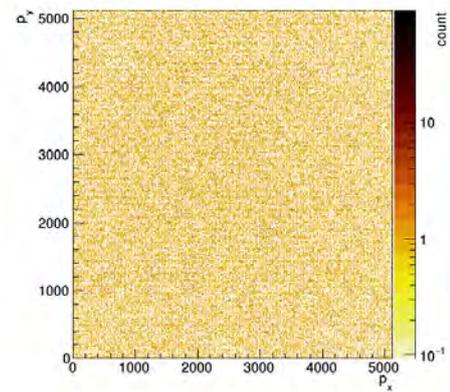
線源	下限 [ADU]	上限 [ADU]	$C^*$	$E_0$ [ADU]*	$\sigma$ [ADU]*	崩壊種**
$^{109}\text{Cd}$	3470	3600	$91 \pm 4$	$3522 \pm 2$	$35 \pm 2$	Ag $K_\alpha$
$^{109}\text{Cd}$	3900	4050	$2.6 \pm 0.7$	$(398 \pm 1) \times 10^1$	$(4 \pm 2) \times 10^1$	Ag $K_\beta$
$^{241}\text{Am}$	2150	2350	$40 \pm 3$	$2222 \pm 2$	$39 \pm 3$	Np $L_\alpha$
$^{241}\text{Am}$	2650	2750	$44 \pm 4$	$2705 \pm 5$	$48 \pm 9$	Np $L_{\beta 2}$
$^{241}\text{Am}$	2800	2900	$99 \pm 6$	$2834 \pm 2$	$26 \pm 2$	Np $L_{\beta 1}$
$^{241}\text{Am}$	3250	3450	$11 \pm 1$	$3324 \pm 6$	$57 \pm 7$	Np $L_\gamma$

\* 誤差は統計誤差に由来する  $1\sigma$  範囲である。

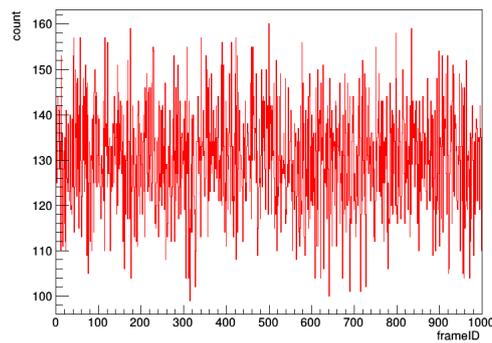
\*\* 崩壊種は  $E_0$  からの推測である。



(a) weight 分布



(b)  $(p_x, p_y)$  プロファイル



(c) ライトカーブ

図 3.10:  $^{109}\text{Cd}$  を照射したデータの解析結果

表 3.5: エネルギー較正フィッティングの結果

A	$(6.30 \pm 0.01) \times 10^{-4}$
B	$(-7 \pm 4) \times 10^{-2}$

\* 誤差は表 3.4 の誤差に由来する。

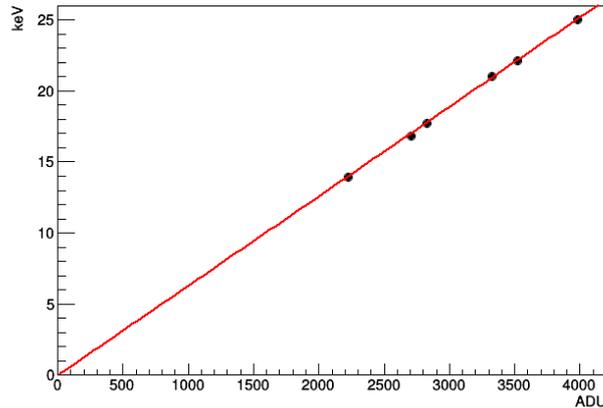


図 3.11: エネルギー較正直線。各点が実験から得られた輝線のエネルギーと平均波高値の対応点 (表 3.4) であり、統計誤差はデータ点より十分小さい。また、直線は式 (3.4) によるフィッティングの結果である。

### 3.4.3 エネルギー較正

表 3.4 の結果から、波高値 (ADU) とエネルギー (keV) を対応させることができる。各ピークのフィット結果  $E_0$  と輝線エネルギーのペアを、 $A, B$  をパラメータとする一次関数

$$(\text{エネルギー [keV]}) = A \times (\text{波高値 [ADU]}) + B \quad (3.4)$$

でフィッティングした。フィッティング結果の直線は図 3.11 に、パラメータは表 3.5 に示した。特に  $B$  の値が 0 に近く、波高値とエネルギーがほぼ比例の関係にあることが分かる。以降、較正には表 3.5 のベストフィット値を利用する。

### 3.4.4 エネルギー分解能

エネルギー分解能は検出器の性能を表す指標の一つである。一般に X 線検出器のエネルギー分解能としては、単色光を検出した際の半値全幅 (FWHM) がよく用いられる。FWHM はフィッティング結果の標準偏差  $\sigma$  から  $2\sqrt{2\log 2}\sigma$  を計算することで得られる。

表 3.4 から各輝線について FWHM を計算し、エネルギー較正を行なった結果は表 3.6 のようになった。特に  $^{241}\text{Am}$  の 17.75 keV のエネルギー分解能は  $3.2 \times 10^2$  eV であり、先行研究 (畠内, 2021) の 16 keV における GMAX0505RF のエネルギー分解能  $2.9 \times 10^2$  eV

表 3.6: エネルギー分解能

輝線種	エネルギー [keV]*	分解能 [eV]**
$^{241}\text{Am } L_{\alpha}$	13.93	$(5.0 \pm 0.8) \times 10^2$
$^{241}\text{Am } L_{\beta 2}$	17.06	$(6 \pm 2) \times 10^2$
$^{241}\text{Am } L_{\beta 1}$	17.75	$(3.2 \pm 0.7) \times 10^2$
$^{241}\text{Am } L_{\gamma}$	21.04	$(8 \pm 1) \times 10^2$
$^{109}\text{Cd } K_{\alpha}$	22.10	$(4.4 \pm 0.7) \times 10^2$
$^{109}\text{Cd } K_{\beta}$	25.02	$(6 \pm 3) \times 10^3$

\* エネルギーは理論値 (表 3.2) である。

\*\* 分解能の誤差は統計誤差に由来する。

と統計誤差の範囲で一致する。一般に半導体検出器のエネルギー分解能の  $1\sigma$  分は、

$$\delta E = W \times \sqrt{N^2 + \frac{EF}{W} + AE^2} \quad (3.5)$$

で表現される (e.g., Koyama 他, 2007)。ただし、 $W$  は検出器内の電子正孔対の平均生成エネルギー、 $N$  は RMS ノイズに対応する電子数、 $E$  は入射 X 線エネルギー、 $F$  は Fano factor、 $A$  は適当な比例定数である。平方根中の第一項は読み出しノイズや暗電流ノイズに由来する項、第二項は検出イベント数の統計的な揺らぎに由来する項、第三項は現象論的な補正項である。Si 検出器の場合  $W \sim 3.65$  eV、 $F \sim 0.12$  であり、理想的な検出器では  $N = 0$ 、 $A = 0$  となる。図 3.12 に理想的な検出器での理論限界値と実験で得られたエネルギー分解能を示す。実験データは全体として理論値よりも大きい値をとっている。このことには読み出しノイズや暗電流ノイズ、ピクセルごとのゲインのばらつきなどの複数の原因が寄与していると考えられる。とはいえ、総露光時間 600 s の撮影で  $^{241}\text{Am}$  や  $^{109}\text{Cd}$  の複数輝線を容易に分解できる程度ではあるため、X 線検出器として十分なエネルギー分解能を持っていると言える。

### 3.4.5 検出効率

検出効率は入射光子数に対する検出イベント数の割合で表される。本実験の場合、保護ガラス以外による減衰は無視し、エネルギー  $E$  の検出効率  $e(E)$  は、検出イベント数  $X(E)$ 、線源強度  $I(E)$  [Bq]、検出器の占める立体角  $\Omega$ 、保護ガラスの減衰係数  $a_g(E)$ 、総露光時間  $T$  [s] を用いて、

$$e(E) = \frac{X(E)}{\frac{\Omega}{4\pi} a_g(E) I(E) T} \quad (3.6)$$

で算出される。本来は全エネルギーに対して  $e(E)$  を算出するべきだが、エネルギーによって保護ガラスの透過率が異なるため、計算は単純ではない。そのため、ここでは強度の強い  $^{109}\text{Cd}$  の  $K_{\alpha}$  線 (22.1 keV) 付近に絞って実行的な検出効率を算出する。 $a_g(E)$  は平均自

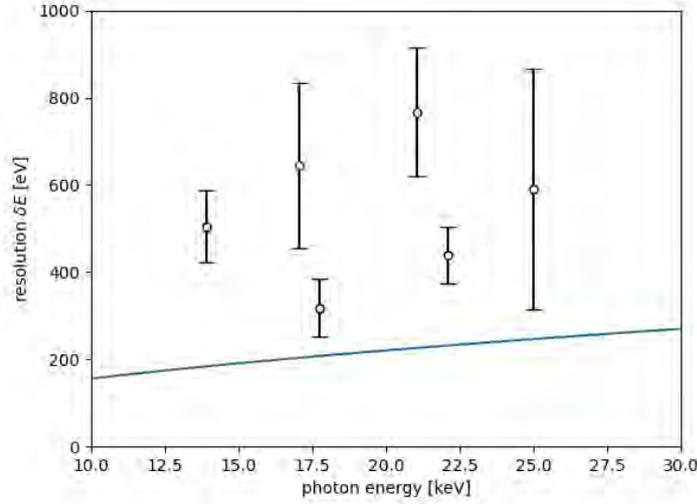


図 3.12: エネルギー分解能。プロット点は実験結果 (表 3.6)、曲線は理論限界 (式 (3.5)) である。

由行程  $\lambda_{\text{SiO}_2}(E)$  [cm], ガラス厚  $d = 0.07$  cm を用いて、

$$a_g(E) = \exp \left[ -\frac{d}{\lambda_{\text{SiO}_2}(E)} \right] \quad (3.7)$$

で計算できる。  $X(E)$  は表 3.4 の結果を用いて、  $E_0 - 2\sigma \leq E \leq E_0 + 2\sigma$  の範囲にあるイベント数を数えることで得た。このとき、weight を絞らない場合と weight が 2 の場合それぞれでイベントをカウントし、  $e(E)$  を算出した。

また、検出効率から実効的な空乏層厚  $t(E)$  [cm] を算出することができる。検出器 (Si) の平均自由行程  $\lambda_{\text{Si}}(E)$  [cm] を用いて、

$$e(E) = 1 - \exp \left[ -\frac{t(E)}{\lambda_{\text{Si}}(E)} \right] \quad (3.8)$$

$$\Leftrightarrow t(E) = -\lambda_{\text{Si}}(E) \ln [1 - e(E)]$$

で計算できる。

検出効率の計算結果は表 3.7 のようになった。特に 22.1 keV での全 weight イベントの検出効率は 0.27 %、推定される空乏層厚は  $3.4 \mu\text{m}$  という結果になった。同シリーズである GMAX0505 の空乏層厚は Asakura et al. (2019) によって約  $5 \mu\text{m}$  と報告されており、先行研究 (畠内, 2021) の結果から GMAX0505RF の検出効率はその数倍程度と見積もられている。表 3.7 の結果はそれよりも低い検出効率を出したことになるが、これは先行研究とエネルギーの違う  $^{109}\text{Cd}$  の  $K_\alpha$  輝線のみを用いたこと、連続成分を考慮に入れていないことなどが原因として考えられる。X 線天文学で広く用いられている CCD 検出器の空乏層厚は数十  $\mu\text{m}$  であり (e.g., Koyama 他, 2007)、GMAX0505RF はそれよりは検出効率が低いという結果になった。この結果が *cipher* の実現性に及ぼす影響については 6.7 節で後述する。

表 3.7:  $^{109}\text{Cd}$  の 22.1keV 輝線から得られた検出効率と実行的な空乏層厚

	イベント数	検出効率 (%) <sup>*</sup>	空乏層厚 [ $\mu\text{m}$ ] <sup>*</sup>
全 weight	21293	$0.27 \pm 0.03$	$3.4 \pm 0.4$
double	782	$0.010 \pm 0.001$	$0.12 \pm 0.01$

<sup>\*</sup> 誤差は線源から検出器までの距離が 1 mm 前後するとしたときの系統誤差である。統計誤差は十分小さい。

## 3.5 温度応答

これまで解析手法と得られた結果を説明してきたが、紹介した手法は完全ではない。ダーク取得のデータをペDESTALとするのは、ペDESTALの時間変化が安定していることを前提としている。一般に暗電流ノイズは温度が高いほど大きくなるため、X線データ取得時に大きな温度変化があった場合、ペDESTALが不正確ということになる。本節ではセンサ周辺の温度を変化させたときに検出結果にどのような影響が出るのかを簡単に確認する。

### 3.5.1 簡易温度変化試験

温度を変化させた時のデータを取得した。基本的なセットアップは3.2節と同様で、 $^{109}\text{Cd}$ 線源を設置した状態でゲイン2.0、露光時間600 ms、フレーム数1000の連続撮影を行なった。唯一異なるのは、冷却用の扇風機の扱いである。データ取得開始と同時に冷却を停止し、frame IDが500になるところで冷却を再開した。センサボード搭載の温度計で測定していた温度の時間変化が図3.13である。データ取得開始からセンサボード自身の発熱により温度は上昇し、最高で45°Cに達した。冷却を再開してからは温度は下がっていき、frame IDが約800になるところで36°Cで安定状態に入った。

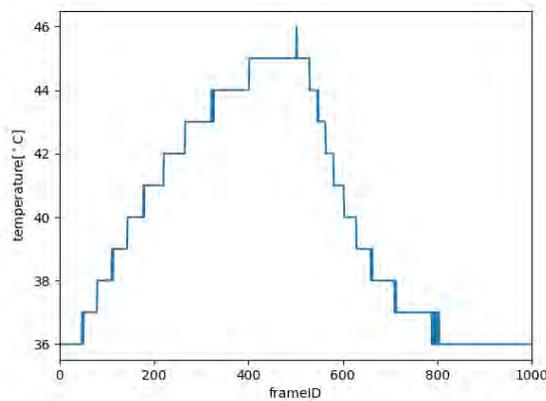


図 3.13: 温度変化試験での温度の時間変化。センサボード搭載温度計に基づく。

### 3.5.2 温度変化データの特徴

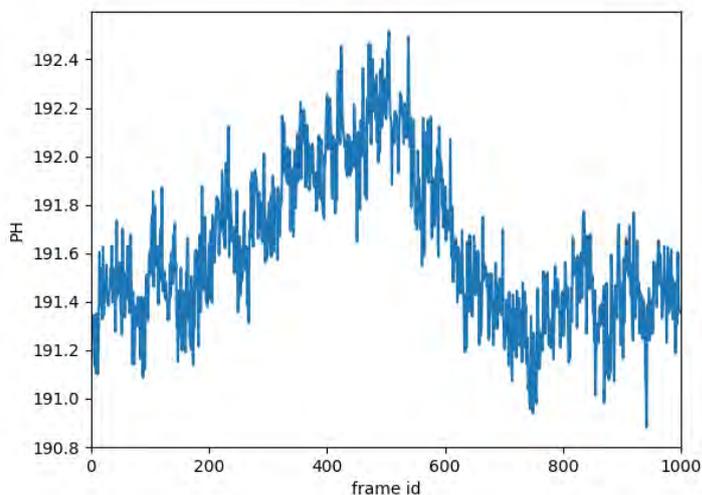
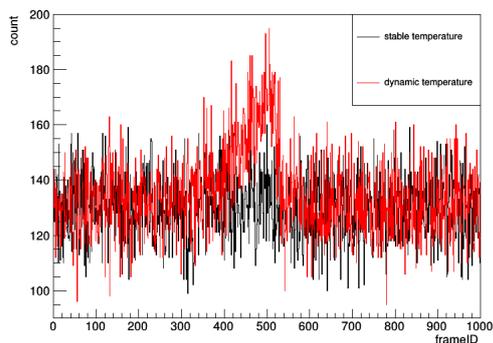
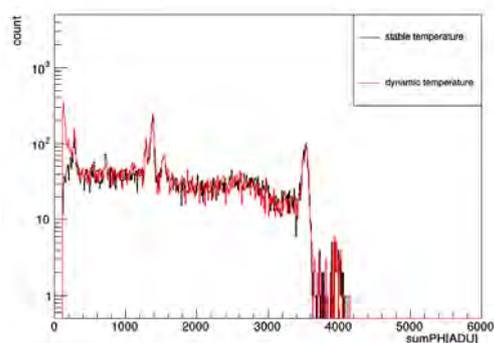


図 3.14: 温度変化試験での、生波高値の全ピクセル平均の時間変化



(a) ライトカーブ



(b) double イベントスペクトル

図 3.15: 温度変化時取得  $^{109}\text{Cd}$  データの解析結果。赤線が温度変化時のデータ、黒線が 3.4 節のデータである。

まず、ペDESTALを引く前の波高値のデータから、フレームごとに波高値の全ピクセル平均をとったものを時間変化として図 3.14 に示す。温度の上昇に伴い、わずかではあるが波高値が上下する様子が見られる。またライトカーブと double イベントスペクトルを図 3.15 に示す。比較のため、3.4 節のデータを重ねて描画している。

まず時間変化に着目すると、frame ID が 400 から 500 までの区間でイベント数が上昇する様子が見られる。これは温度が高くなっていた区間に一致する。またスペクトルから、温度変化時のデータは約 200 ADU 以下でイベント数が多くなっている様子が分かる。これは温度が上昇した際に実際のペDESTALが上昇し、 $E_{\text{event}}$  に達するピクセルが増えたためだと考えられる。

実際の偏光解析に影響を及ぼすかをこれらだけで判断するのは難しいが、時間による温度変化がスペクトルなどに影響を与える様子を確認できた。次章ではこの問題を解決するような解析手法を考えていく。

### 3.6 3章のまとめと次章への展望

特性 X 線源を用いて、CMOS イメージセンサ GMAX0505RF の X 線検出器としての性能評価試験を行なった。解析はグッドピクセル判定、ペDESTAL 除去、 $E_{\text{event}}$  によるイベント位置判定、 $E_{\text{split}}$  による形状判定という流れで行なう。解析の結果、エネルギー分解能は 17.75 keV のエネルギーで  $3.2 \times 10^2$  eV と十分な性能を持っている一方で、検出効率は 22.1 keV で 0.27 % であることが分かった。また、ペDESTAL が検出器周辺の温度変化によって応答する様子確かめた。次章では温度変化に耐性のある解析アルゴリズムを考察し、観測効率を改善できるような読み出しシステムの開発を行なう。

# 第4章 FPGAを用いた高速読み出しシステム開発

*cipher*の一貫したデータ処理フローを確立するに当たって、特に大きな課題であったのがX線検出に特化した実用的な読み出しシステムの構築である。従来の読み出しシステムでは連続撮影時に1.0 s程度のデッドタイムが生まれていた。信号が微弱で時間変動を捉える必要がある天体観測ではこのデッドタイムは致命的である。また3.4.5小節で議論した通り、現状の検出器では検出される光子統計量が少ないため、実効的な観測効率の観点からもデッドタイムを短縮する必要がある。本章ではデッドタイムの短縮を目指した読み出し系の開発を行なう。なお本章では、表4.1のように演算を定義する。

表 4.1: 4章の演算定義

表記	意味
$a \% m$	整数 $a$ を整数 $m$ で割ったあまり
$\text{floor}(x)$	実数 $x$ の整数部分

## 4.1 背景

従来の読み出し系(図4.1)を時間の観点で見直す。このシステムの課題として、読み出しの時間がかかってしまっていることがある。従来の読み出し系の簡単なタイミングチャートを図4.2に示す。フレーム1枚分の撮影(今後、特に注釈がなければ「撮影」はフレーム1枚分のこととする)をすると、 $2 \text{ bytes} \times 5120 \times 5120 = 52 \text{ MB}$ のデータが回路上を流れる事になる。撮影を開始してからセンサからDDRメモリへの転送が完了するまでのデータ収集の時間(capture)は露光時間と同程度であるが、DDRメモリからEthernetを

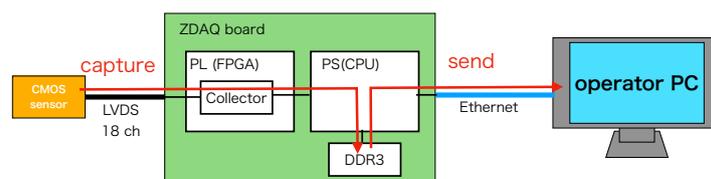


図 4.1: 従来の読み出し系。

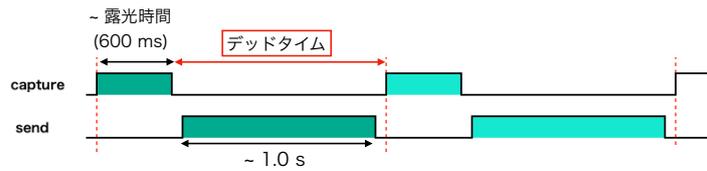


図 4.2: 従来の読み出し系のタイミングチャート。capture は撮影回路の動作を示し、send は解析用 PC へのデータ転送を示す。

通して解析用 PC に転送するまでの時間 (send) はおよそ 1.0 s かかっていた。この速度はセンサ本体の最速フレームレート 24 ms/frame よりも大幅に遅い上、連続撮影時にデッドタイムが生まれてしまい、観測効率の低下にもつながる。この、読み出しのデッドタイムが長いという課題を解決するのがこの章の目的である。

課題解決の基本的な発想は、Ethernet で転送するデータ量を削減することにある。センサから読み出された全ピクセル分の波高値データ全てが重要なわけではなく、X線の解析に必要なのは 3.4 節で示した通りイベント閾値およびスプリット閾値を超えたピクセルのみである。イベントの広がりを見逃して大雑把にイベント数を必要なピクセル数と思うと、 $10^4$  photon/s ほどの検出量<sup>1</sup>でも転送データ量は従来の  $10^4 / (5120 \times 5120) \approx 1 / (3 \times 10^3)$  ほどに抑えられる見込みがある。データ量を削減する代わりにデータ転送レートを上げるといふ解決法も考えられるが、衛星への搭載を考えるとデータ取得系の処理レートを上げても地上局への転送ができない。そのような観点からも、データ削減は有効な手段と言える。以上のことから、ZDAQ ボードからのデータ送出前に必要なピクセルの選別を行なうという方針で以下の節を進める。データ削減を実現させるにあたって必要なのは、(1) ペデスタルを引いた上での必要ピクセル抽出、(2) ZDAQ ボード上での実装の実現、の 2 つである。

## 4.2 動的ペデスタル

### 4.2.1 概要

まず、ペデスタルの処理について考える。説明のため、フレーム番号  $i$  でピクセル番号  $l$  の生の波高値を  $H_{i,l}$ 、ピクセル番号  $l$  のペデスタルを  $P_l$  とする。ただし、ピクセル番号は  $(p_x, p_y)$  のペアと 1 対 1 対応する数値であり、 $x$  方向の全ピクセル数  $N_x$  を利用して

$$l(p_x, p_y) = N_x p_y + p_x \Leftrightarrow \begin{cases} p_x(l) = l \% N_x \\ p_y(l) = \text{floor}(l / N_x) \end{cases} \quad (4.1)$$

<sup>1</sup> フレームあたりに得るイベントが多すぎる場合、1 フレーム内の時間に同じピクセルに 2 つ以上のイベントが重なって検出されてしまう恐れがある。これは「パイルアップ」と呼ばれ、パイルアップを起こさないような観測をするのが理想である。なお、実際の Crab 観測では 3.4.5 小節で議論した通りカウントレートが非常に低いため、特にパイルアップの問題はない。

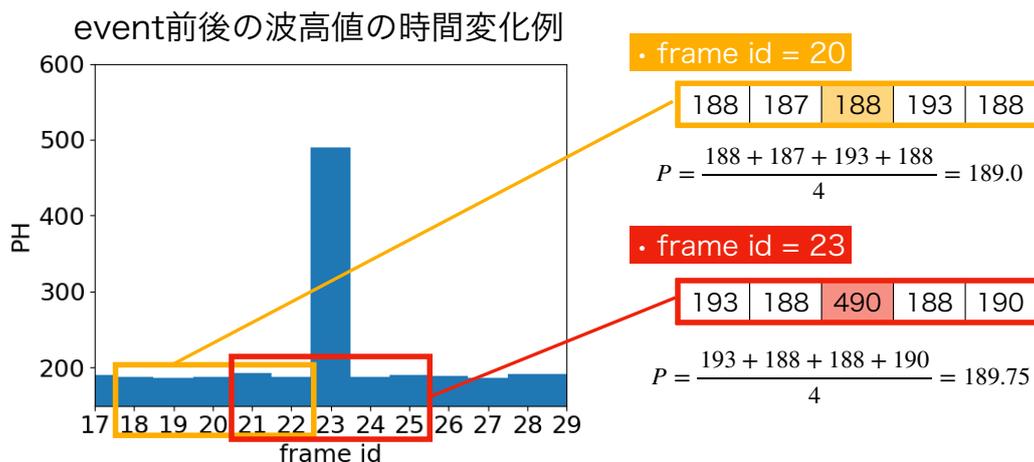


図 4.3: あるピクセルの X 線イベント到来前後の波高値の時間変化例。横軸はフレーム、縦軸は ADU 値である。黄枠と赤枠内は前後 2 フレームを含めた ADU 値である。

と関係付けられる。3.3 節で紹介したように  $N$  フレームのダーク取得の平均波高値をペデスタルとするのは

$$P_l = \frac{1}{N} \sum_{i=0}^{N-1} H_{i,l} \quad (4.2)$$

とすることに相当する。自然な発想として、3.3 節のように 100 フレームほどダーク取得を行い、その結果から式 (4.2) を計算し、結果を ZDAQ ボード内メモリに保存しておくという方法が考えられる。後述の方法と区別するため、この方法を静的ペデスタル (static pedestal) と呼ぶ。だが、3.5 節で述べた通り、ペデスタルは時間経過に伴い温度変化などの影響を受けて挙動が変化する恐れがある。そのため、ある程度の時間変化に耐えられる手法が要求される。

そこで、ペデスタルを X 線データ取得中のフレーム数とともに動的に変化させる方法を考える。データ取得の前後数フレームでは各ピクセル波高値の挙動は安定していると仮定する。このとき、注目するフレームの前後数フレームの波高値の平均値をペデスタルとする。厳密には、考慮する過去のフレーム数を  $n_p$ 、未来のフレーム数を  $n_f$  としたとき、 $i$  フレーム目のピクセル番号  $l$  のペデスタル  $P_{i,l}$  を、

$$P_{i,l} = \frac{1}{n_p + n_f} \left( \sum_{j=i-n_p}^{i-1} H_{j,l} + \sum_{j=i+1}^{i+n_f} H_{j,l} \right) \quad (4.3)$$

とする。参考までに、ペデスタル計算の具体例を図 4.3 に示す。ちなみに、もしあるピクセル  $l$  の到来イベントが  $i$  フレーム目の他に過去  $n_p$  フレームか未来  $n_f$  フレームの中にあつたとすると、 $P_{i,l}$  は本来のペデスタルよりも高く評価されることになる。その場合は  $i$  フレーム目の到来イベントがイベントとしてみなされなくなる恐れがあるが、ピクセルあたりの統計量の少ない X 線観測ではひとまず無視できる。以降、このペデスタル決定方法を動的ペデスタル (dynamic pedestal) と呼ぶ。

## 4.2.2 線源データへの適用

動的ペDESTALの発想そのものはすでに取得したデータにも適用可能である。ここでは3.4節で取得した $^{109}\text{Cd}$ と $^{241}\text{Am}$ 線源照射時のデータに適用し、動的ペDESTALに正当性があることを確認する。性能比較のため、 $n_p = 2, n_f = 2$ の場合と $n_p = 4, n_f = 4$ の場合について調べた。double イベントのみのスペクトルは図4.4のようになり、 $^{109}\text{Cd}$ の $K_\alpha$ 線を用いたエネルギー分解能、検出効率を表4.2のようになった。動的ペDESTAL適用前と適用後でいずれも性能に大きな違いはないことが分かる。また特に $n_p$ と $n_f$ に性能が大きく依存していないことも分かった。

また、3.5節の温度変化データに動的ペDESTALを適用した結果を図4.5に示す。静的ペDESTAL適用時に見られたframe IDが400から500までの間イベント数が増加する現象や、スペクトルの約200 ADU以下にピークができる現象は、動的ペDESTALを導入することで解消している。この結果は、動的ペDESTALが温度変化にある程度の耐性を持つ手法であることを示唆している。これは今回のセットアップでの過去 $n_p$ フレームと未来 $n_f$ フレームの間の時間が温度変化のタイムスケールに比べて十分短いためと考えられ、想定通りの結果である。

これらの結果から、動的ペDESTALを新たな読み出し系の基盤アルゴリズムとして採用することにする。

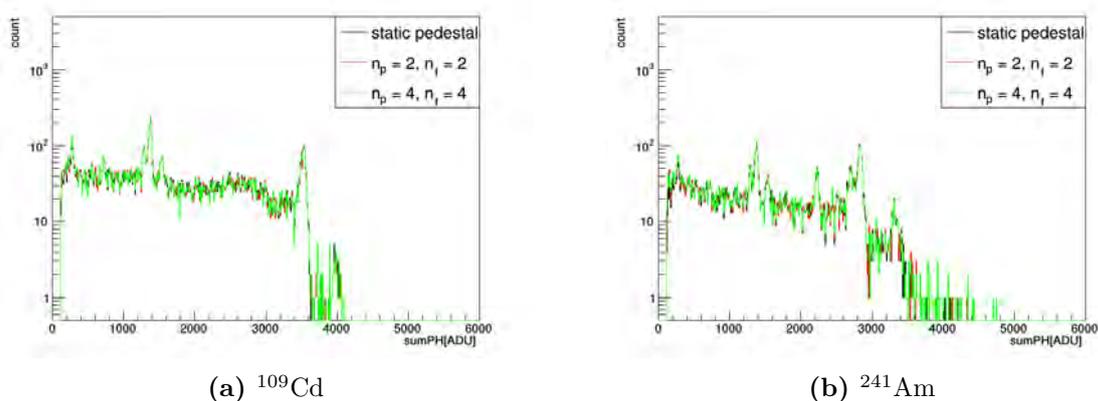
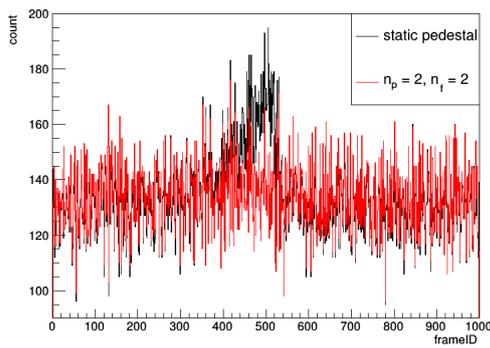


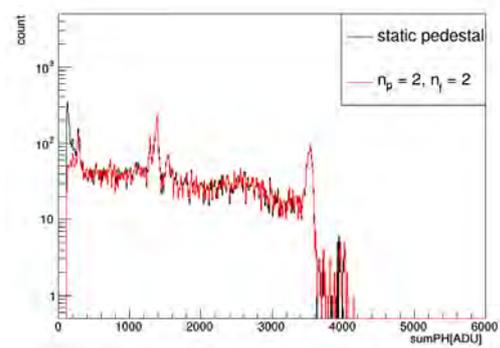
図 4.4: 動的ペDESTALのスペクトル比較。weight が 2 のイベントのみ抜き出している。

表 4.2:  $^{109}\text{Cd}$  の  $K_\alpha$  線を用いた動的ペDESTAL性能評価

	エネルギー分解能 [eV]	検出効率 (%)
静的ペDESTAL (3.4 節)	$(4.4 \pm 0.7) \times 10^2$	$0.27 \pm 0.03$
$n_p = 2, n_f = 2$	$(4.2 \pm 0.6) \times 10^2$	$0.26 \pm 0.03$
$n_p = 4, n_f = 4$	$(4.3 \pm 0.6) \times 10^2$	$0.26 \pm 0.03$



(a) ライトカーブ



(b) double イベントスペクトル

図 4.5: 温度変化時取得  $^{109}\text{Cd}$  データに動的ペデスタルを適用した結果。

## 4.3 データ削減アルゴリズム

前節の動的ペデスタルのアイデアを、この節では ZDAQ ボード内に実装できるようにチューニングしていく。GMAX0505RF 用にカスタマイズされている部分もあるが、基本的に他のセンサにも応用できるロジックである。

### 4.3.1 アルゴリズム内容

まず前提として、データ削減は撮影とは独立に、撮影と並列して動作させることを考える。このこと自体は FPGA で撮影回路と独立に回路を組むことで容易に実現が可能である。並列化することで、シングルスレッド処理からデッドタイムを大幅に減らすことが期待できる。ただこの場合、撮影中のデータが収納されるメモリ領域とデータ削減回路で使用するメモリ領域が重ならないように注意する必要がある。撮影中にデータ削減回路で計算する対象フレームを、数フレーム前で撮影したデータにすることによって、メモリ領域の重複なく並列な動作が可能になる。

ペデスタルの過去分のフレーム数を  $n_p$ 、未来分のフレーム数を  $n_f$  とする。DDR3 メモリの割り当てを以下のように定める。まず、撮影 1 フレーム 25 M ピクセルそのままの波高値が収納される領域が  $n_a \equiv n_p + n_f + 2$  枚分ある。これらの各領域を  $A_i$  ( $0 \leq i < n_a$ ) とする。これらには撮影中の 1 フレーム分に加えて、解析対象の 1 フレーム分とその前後  $n_p + n_f$  フレーム分を考慮している。また、 $A_i$  とは別にペデスタル情報記録用の領域を 1 枚分確保し、これを  $B$  とする。さらに、送信対象のピクセル番号と波高値を収納する領域を用意し、それぞれ  $C_0, C_1$  とする。全部合わせて必要なのは  $n_a + 3$  フレーム分のメモリ領域となり、これは  $n_p = 2, n_f = 2$  とすれば現状の ZDAQ ボード搭載の DDR3 メモリに十分収まり、現実的である。

撮影時の大まかな流れを説明する。撮影は撮影回路を動作させることで行なうが、収納されるメモリ領域を撮影フレーム番号によって変えていく。具体的には、 $i$  フレーム目の撮影では領域  $A_{i \% n_a}$  にセンサから読み出されたデータを収納するようにする。1 フレーム撮影中に、メモリ内のデータを使って 1 フレーム分の転送データを  $C_0, C_1$  に抽出する処

理を行なう。このとき、抽出処理対象のメモリ領域は  $A_{(i-n_f-1)\%n_a}$  である。 $C_0, C_1$  に抽出されたデータは送信対象として、外部PCへ転送される。 $C_0, C_1$  内の対象データの容量が元の1フレーム分の容量よりも小さいことが、データ削減の根拠になっている。

1フレーム分のデータ削減はさらに (a)add, (b)extract, (c)subtract の3つの操作に分けられる。 $i$ フレーム目 ( $i \geq n_a - 1$ ) の撮影を始めたとき、領域  $B$  には  $A_{i\%n_a}$  と  $A_{(i-1)\%n_a}$  以外の撮影データの和が入っている。正確には、

$$\overrightarrow{M(B)} = \sum_{\substack{0 \leq j < n_a \\ j \notin \{i\%n_a, (i-1)\%n_a\}}} \overrightarrow{M(A_j)} \quad (4.4)$$

である。ここで、メモリ領域  $X$  内のデータ配列を  $\overrightarrow{M(X)}$  と表現している。3ステップについて詳細を以下で説明する。

(a) add

領域  $B$  に前サイクルで撮影したデータを足す。式で表せば、

$$\overrightarrow{M(B)} \leftarrow \overrightarrow{M(B)} + \overrightarrow{M(A_{(i-1)\%n_a})} \quad (4.5)$$

である。この結果、領域  $B$  は

$$\overrightarrow{M(B)} = \sum_{\substack{0 \leq j < n_a \\ j \notin \{i\%n_a\}}} \overrightarrow{M(A_j)} \quad (4.6)$$

となっている。

(b) extract

データ削減対象のフレームから設定閾値を超えるピクセルの情報を取り出す。 $k = (i - n_f - 1)\%n_a$  として、 $A_k$  がデータ削減対象の領域である。領域  $B$  には領域  $A_k$  と  $A_k$  の過去  $n_p$  フレーム、 $A_k$  の未来  $n_f$  フレームの撮影データの和が入っている。領域  $A_k$  を除いた  $n_p + n_f$  フレームの平均をペDESTAL  $\vec{P}$  とすると、

$$\vec{P} = \frac{\overrightarrow{M(B)} - \overrightarrow{M(A_k)}}{n_p + n_f} \quad (4.7)$$

である。閾値を  $\Theta$  として  $A_k$  のピクセル番号  $l$  のピクセルが抽出される条件は、

$$M(A_k)_l - P_l \geq \Theta \quad (4.8)$$

となる。実際の出力は条件を満たしたピクセルの波高値とピクセル番号であり、 $n_c$  を全ピクセル数として

$$\begin{aligned} \overrightarrow{M(C_0)} &\leftarrow \{ M(A_k)_l - P_l \mid 0 \leq l < n_c, M(A_k)_l - P_l \geq \Theta \} \\ \overrightarrow{M(C_1)} &\leftarrow \{ l \mid 0 \leq l < n_c, M(A_k)_l - P_l \geq \Theta \} \end{aligned} \quad (4.9)$$

となる。

(c) subtract

領域  $B$  から次サイクルで撮影する領域のデータを引く。すなわち、

$$\overrightarrow{M(B)} \leftarrow \overrightarrow{M(B)} - \overrightarrow{M(A_{(i+1)\%n_a})} \quad (4.10)$$

とする。この結果、領域  $B$  は

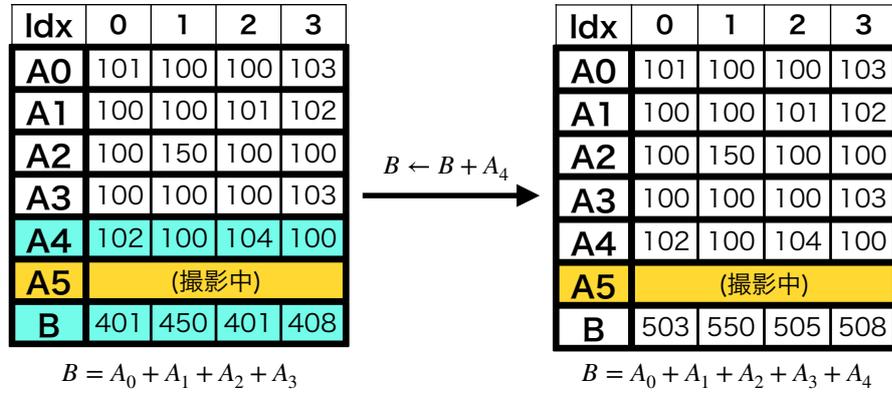
$$\overrightarrow{M(B)} = \sum_{\substack{0 \leq j < n_a \\ j \notin \{(i+1)\%n_a, i\%n_a\}}} \overrightarrow{M(A_j)} \quad (4.11)$$

となっている。

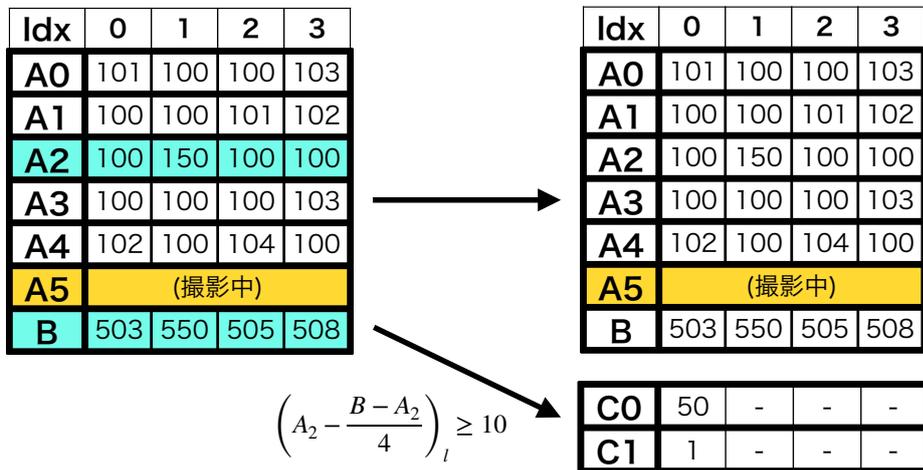
add, extract, subtract の 3 ステップを終えたとき、領域  $B$  には  $A_{(i+1)\%n_a}$  と  $A_{i\%n_a}$  以外の撮影データの和が入っており、次サイクル  $i \leftarrow i + 1$  の開始時点の状態になっている。また、 $i < n_a - 1$  のときは add のみ行い、extract と subtract は実行しない。

### 4.3.2 動作例

#### (a) add



#### (b) extraction



#### (c) subtraction

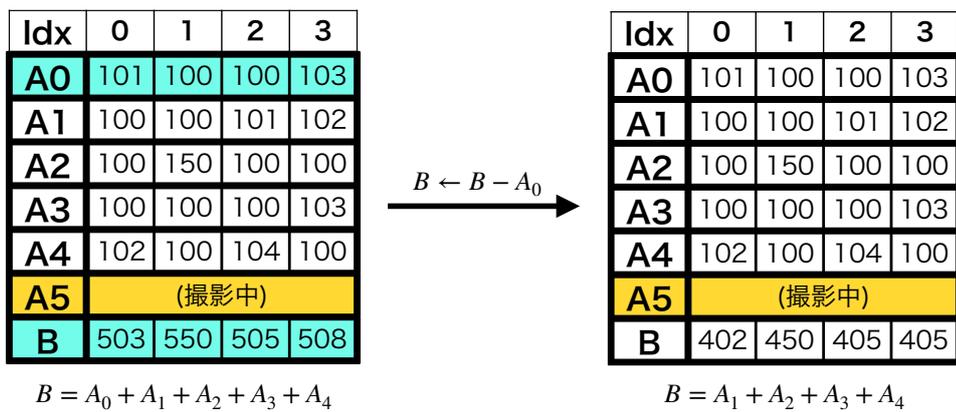


図 4.6: データ削減アルゴリズムの動作例。

1 フレーム撮影中のデータ削減の動作例を図 4.6 に示す。ただし、 $n_p = 2, n_f = 2, i = 5, n_c = 4, \Theta = 10$  の場合を再現している。説明の簡略化のため、メモリ領域名  $X$  とそのデータ配列  $\overline{M(X)}$  を同一視する。また、表中の波高値はわかりやすさのためこちらで設定した数値であり、実際の実験データとは関係がない。 $A_5$  は撮影で使用中のため、データ削減回路では触れない。この設定の元で抽出処理対象のメモリ領域は  $(i - n_f - 1) \% n_{na} = 2$  から  $A_2$  である。

まず add を実行する。add を開始する前は、 $B$  には前フレーム撮影領域の  $A_4$  以外の 4 枚分の和が入っている。式 (4.5) はここでは  $B$  に  $A_4$  を足すことに相当し、add 実行後は  $B$  には抽出処理対象である  $A_2$  と過去 2 フレーム ( $A_0, A_1$ ) と未来 2 フレーム ( $A_3, A_4$ ) の和が入っていることになる。

次に  $A_2$  を抽出処理対象とする extract を実行する。式 (4.8) はこの例では、

$$\left( A_2 - \frac{B - A_2}{4} \right)_l \geq 10 \quad (4.12)$$

に相当する。実際に計算するとこの条件を満たすのは  $l = 1$  のみであり、 $C_0$  にはペDESTアル減算後波高値である 50 が、 $C_1$  にはピクセル番号である 1 が先頭に詰められて収納される。仮に全ピクセルで閾値超過条件を満たしたとすると、 $C_0$  と  $C_1$  には全ピクセル分の波高値とピクセル番号が入ることになる。この間、 $B$  に変化はない。

最後に subtract を実行する。式 (4.10) はここでは  $B$  から  $A_0$  を引くことに相当する。次の撮影時には  $A_0$  が使われるため、次の撮影が始まる前にここまで完了しておく必要がある。subtract が終了したとき、 $B$  には  $A_0$  と  $A_5$  以外の和が入っていることになる。この時点で  $B$  の状態は  $i = 6$  の開始時点と同等になっており、一貫性を持っていることが確認できる。

例を挙げると分かりやすいように、本アルゴリズムで書き込みを行なうメモリ領域は  $C_0, C_1, B$  のみであり、 $A_i (0 \leq i < 6)$  に対しては読み込みしか行わない。撮影データ本体に触れずに実行することで安全にデータ操作を行える点が、本アルゴリズムの単純ながら優れた点である。

### 4.3.3 タイミングチャート

撮影中の各処理の時間変化例を図 4.7 に示す。まず、撮影回路は露光時間とセンサから DDR メモリに転送されるまでの時間の和であり、後者は現状では約 40 ms である。撮影をしている間にデータ削減回路が動作し、add, extract, subtract の 3 ステップが進む。また、extract で領域  $C_0, C_1$  に入れられたデータは外部へ転送する必要があるが、これは extract が終了し次第開始する。外部転送対象のピクセル数が  $10^4$  程度であれば、単純計算で転送時間は  $2.0 \times (10^4 / 5120^2) = 0.8$  ms 程度と見積もることができ、撮影時間より十分短く抑えることが見込まれる。次のフレームサイクルに移るのは、撮影、データ削減、外部転送が全て完了してからである。

デッドタイムを短縮するのに重要になってくるのはデータ削減ロジックの処理時間となる。説明したデータ削減処理はソフトウェアでも実行可能であるが、実際にソフトウェアで実装すると add だけでも 5.7 s 程度と大幅に時間がかかってしまう。ソフトウェアによ

る実装の詳細は付録 B.2 に示した。このことから、データ削減には CPU よりも高速な処理が要求されることが分かる。

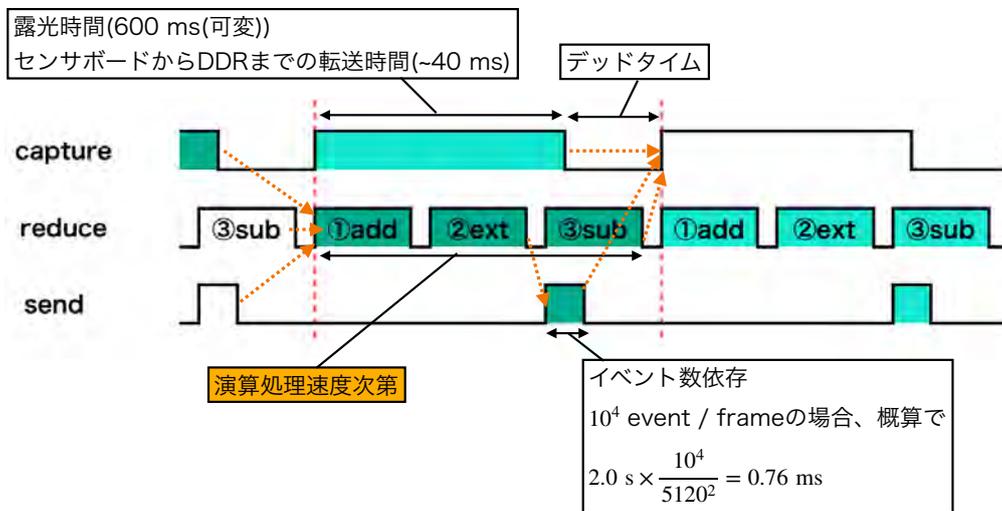


図 4.7: データ削減アルゴリズムに基づくタイミングチャート。capture は撮影回路の動作、reduce はデータ削減回路の動作、send は解析用 PC への転送を示す。また、橙の矢印に関しては、矢印先端の動作は矢印根本の動作が完了してから行われることを示している。

## 4.4 FPGA 上の回路設計と実装

データ削減の高速処理を実現するため、ZDAQ ボード上の FPGA にデータ削減アルゴリズムを実装した。この節では詳細を説明する。

### 4.4.1 FPGA

Field Programmable Gate Array (FPGA) は論理仕様をプログラム可能な論理デバイスである。開発者のプログラムした回路設計に合わせて、論理ブロックと呼ばれる素子の配線を組み替えることができる。近年の高集積化と性能向上に伴い、電子機器だけでなくビッグデータ処理や株式市場の高頻度取引など様々な分野への FPGA の応用に注目が集まっている (Putnam et al., 2014; Lockwood et al., 2012)。ソフトウェアと比べて FPGA を用いる最大のメリットは、単純な処理を高速に実行できることにある。クロックごとに回路内で並列に信号処理をすることが可能なため、回路設計次第で CPU よりも高い処理パフォーマンスを期待できる。

本節で用いる FPGA に関する用語の簡単な解説を以下に述べる。

- VHDL

VHDL はハードウェア記述言語 (Hardware Description Language) のひとつであり、実現したい回路設計をテキストベースで記述できる。入出力の信号線を宣言した上で、配線の接続や信号同士の演算を所定の文法に従って記述することになる。信号

の演算には and, or などの論理演算や加減乗除の他に if 文, for 文なども対応している。一般に知られる高級プログラミング言語と大きく異なる点は、回路記述部はスクリプトの上から実行するためのレシピではなく、論理素子や配線の空間的な繋ぎ方を示しているという点にある。そのため、信号処理の時間的な順序が重要な回路を組む際はクロックによる制御が必須である。実際には入力クロックごとに演算を行なうプロセス文を記述するのが一般的であり、たとえば後述のステートマシンなどはプロセス文内部に記述することになる。

- IP

FPGA 開発において、intellectual Property (IP) は企業などによって開発された再利用可能な回路部品を指す。規模の大きな回路の開発を効率的に進めるために、よく利用する機能の回路はブロック化して再利用ができるようになっている。IP を利用することで、開発者は抽象度の高い回路設計に集中することができる。

- AXI-Stream

Advanced eXtensible Interface (AXI) はハンドシェイクなデータ転送方式である。ここでは AXI の簡易版である AXI-Stream について説明する。図 4.8 に送信方式の概要を示す。送信側をマスタと呼び、受信側をスレーブと呼ぶ。AXI の中では多種類の信号線がマスタとスレーブを繋いでおり、クロックごとに決められた信号をやりとりする。その中でも特に重要なのは ready 信号と valid 信号である。valid 信号はマスタからスレーブへ送られる、送信しようとしているデータが有効であることを示す信号である。それに対し、ready 信号はスレーブからマスタへ送られる、受信の準備ができていることを示す信号である。クロックが変化した際に、valid 信号と ready 信号が同時にアサートされている場合、送受信が成立したものとみなされる。送受信が成立したところで、送信側は送信対象を次のアドレスに移動させる。このやりとりをクロック変化に合わせて繰り返すことで、受信側へ大量のデータを転送することが可能になる。最後のアドレスに達したとき、送信側は last 信号を立てる。last 信号が立っているときにデータの送受信が成立すると、一連のデータ転送は終了する。図 4.9 に送受信のタイミングチャート例を示す。1 クロックあたりに送受信するデータ量は設定が可能であるが、32bit から 1024bit 程度が一般的である。

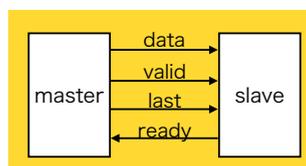


図 4.8: AXI-Stream の概要。

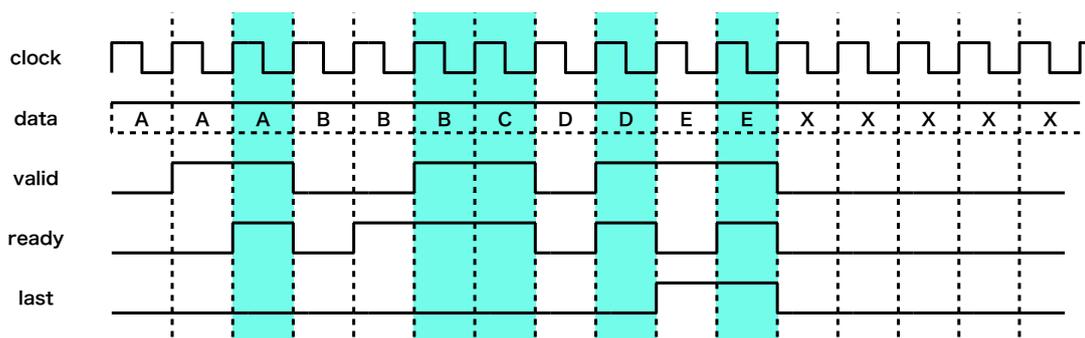


図 4.9: AXI-Stream での送受信のタイミングチャート例。valid と ready が同時にアサートされているとき、送信が成立する。この例では色つきのタイミングで送信が成立し、送信されるデータは「ABCDE」となる。

#### 4.4.2 回路全体とデータ削減 IP の設計

回路全体の設計の概要は図 4.10 のようになる。DDR メモリには 512 MB 分のメモリが割り当てられており、4.3 節の必要なメモリ領域を含む。FPGA は大きくデータ削減用の回路と撮影読み出し用回路の 2 つに分けられる。4.3 節で解説した通り、撮影中にアクセスするメモリ領域とデータ加工時にアクセスするメモリ領域とが重ならなければ、データ削減回路と撮影読み出し回路は完全に独立に動作させることができる。この工夫によってデータ削減を撮影と並列に進めることができ、デッドタイムを減らすことが可能になる。撮影読み出し回路はすでに実装されているものを流用し、本開発ではデータ削減回路を新規に作成した。

データ削減回路は中心となるデータ削減 IP と、DDR メモリとの入出力を実現する Direct Memory Access (DMA) IP (Xilinx, Inc., 2019) からなる。DMA には Xilinx 社提供のソフト IP コアを利用した。DMA を用いることで、DDR メモリ内の空間的に広がったデータを AXI-Stream 上のシーケンシャルアクセスに置き換えることが可能になる。データ削減 IP は 4.3 節のアルゴリズムを実現する VHDL コードを記述することで作成した。データ削減 IP には式 (4.5), (4.9), (4.10) それぞれ専用の回路を内部に備え、設定用レジスタによって 3 つのモードを切り替えられるようにする。IP には図 4.11 のように 2 入力 2 出力のインターフェース  $In_0, In_1, Out_0, Out_1$  を設け、その内部には後述するリングバッファと出力ステートマシンというロジックを実装した。入出力のメモリ領域  $InMem_0, InMem_1, OutMem_0, OutMem_1$ , 閾値  $\theta$ , ペDESTALフレーム数  $n_p, n_f$  をソフトウェアで指定するとして、IP に要求される機能を改めて挙げると以下のようなになる。

(a) add

$$\overrightarrow{M(OutMem_0)} \leftarrow \overrightarrow{M(InMem_0)} + \overrightarrow{M(InMem_1)} \quad (4.13)$$

(b) extract

$$\vec{P} = \frac{\overrightarrow{M(\text{InMem}_1)} - \overrightarrow{M(\text{InMem}_0)}}{n_p + n_f} \quad (4.14)$$

として、

$$\begin{aligned} \overrightarrow{M(\text{OutMem}_0)} &\leftarrow \{ M(\text{InMem}_0)_l - P_l \mid 0 \leq l < n_c, M(\text{InMem}_0)_l - P_l \geq \Theta \} \\ \overrightarrow{M(\text{OutMem}_1)} &\leftarrow \{ l \mid 0 \leq l < n_c, M(\text{InMem}_0)_l - P_l \geq \Theta \} \end{aligned} \quad (4.15)$$

(c) subtract

$$\overrightarrow{M(\text{OutMem}_0)} \leftarrow \overrightarrow{M(\text{InMem}_0)} - \overrightarrow{M(\text{InMem}_1)} \quad (4.16)$$

FPGA の制御は CPU 上のソフトウェアから行なう (付録 B も参照)。図 4.10 には明確に記していないが、各 IP には制御と状態確認のためのレジスタが複数用意されている。たとえば DMA であれば、DDR メモリアドレスの指定、データ読み込みの開始、読み込みのステータス確認、などにそれぞれ 32bit のレジスタが用意されている。各レジスタは OS 内に対応仮想アドレスを持っており、CPU 上のソフトウェアからアクセスすることで FPGA の操作が可能になる。また、CPU から DDR メモリに直接アクセスすることも可能であり、従来設計ではメモリ領域  $A_i$  から外部への転送を行っていた。本開発では必要なデータだけ集めたメモリ領域  $C_0, C_1$  から外部への転送を行なう。

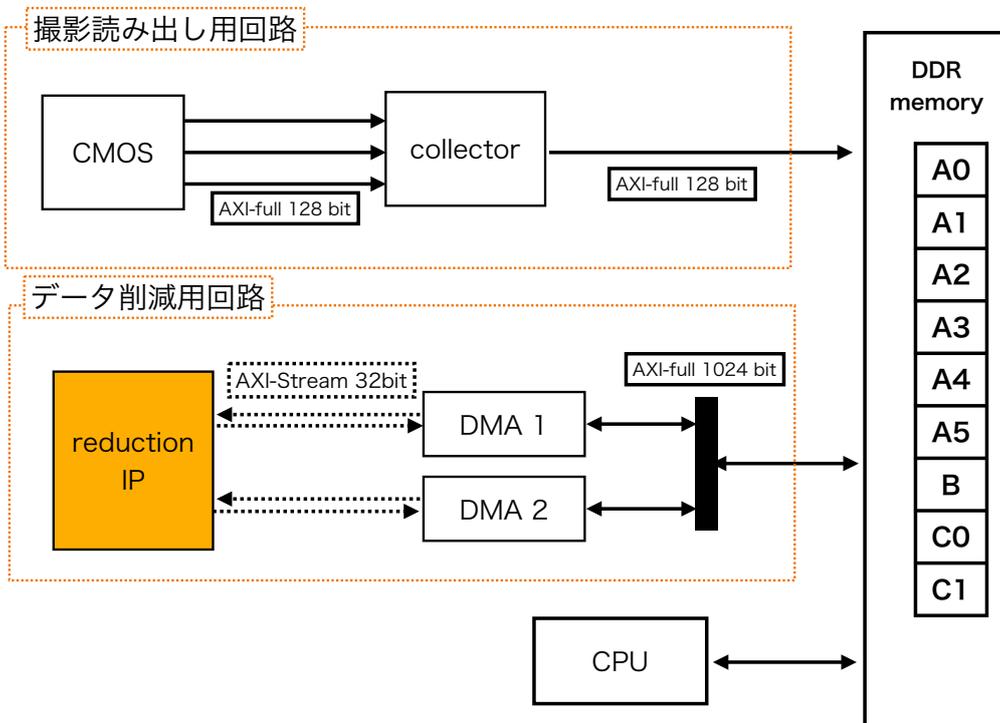


図 4.10: FPGA 上に実装した回路の概要。

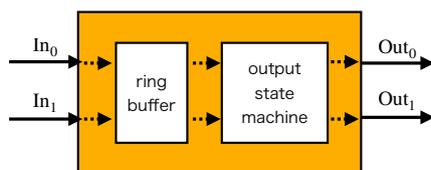


図 4.11: データ削減 IP の概要。

### 4.4.3 リングバッファ

データ削減 IP の入出力は AXI-Stream のプロトコルに基づき、クロックごとに送受信の成立不成立が決められ、送受信が成立すればメモリアドレスが進むようになっている。IP に接続された 2 つの DMA が DDR メモリからデータを読み出すタイミングは必ずしも同時ではなく、加算などの 2 入力の演算を行なうためには適切にアドレスを合わせるロジックが必要となる。IP の 2 つの入力  $In_0$ ,  $In_1$  に対して  $In_0$  のみ送受信が成立し、 $In_1$  の送受信が進まないままである場合、 $In_0$  の送信データは次のアドレスのものに置き換わるため、 $In_1$  の送受信が成立するまで  $In_0$  のデータを記録しておかなければ演算ができない。すなわち、データ削減 IP の中にデータを貯蓄しておく簡易メモリが必要となる。簡易メモリは Block Random Access Memory (BRAM) として FPGA 内に組み込まれているが、サイズは全体で数十 kB 分と限られており、1 フレーム分の 52MB 全てを保存しておくことはできない。そこで、一定サイズのバッファを配列として用意し、動的に読み込み、計算、書き出しを実行するロジックを実装した。

バッファの動作としてはシンプルである。 $In_0$ ,  $In_1$  に対応するバッファ  $buffer_0$ ,  $buffer_1$  と読み込み用ポインタ  $R_0$ ,  $R_1$ 、書き出し用ポインタ  $W$  を用意する。各ポインタ  $R_0$ ,  $R_1$ ,  $W$  の示す位置を  $p_{R_0}$ ,  $p_{R_1}$ ,  $p_W$  と表記する。初期状態では  $p_{R_0}$ ,  $p_{R_1}$ ,  $p_W$  は全て同じにする。まず各  $x = 0, 1$  に対して、入力  $In_x$  に有効なデータが来たら、 $buffer_x$  内の位置  $p_{R_x}$  にデータを収納し、 $R_x$  をインクリメントする。次に書き出しは、 $R_0$ ,  $R_1$  がともに  $W$  より進んでいる場合のみ有効にし、 $buffer_0$ ,  $buffer_1$  内の位置  $p_W$  のデータ内容を元に add, subtract, extract いずれかの演算を行い、出力データとする。また、各ポインタが確保した配列の最後にある状態でインクリメントする場合、ポインタを配列の先頭に移動させる。このようにすることで少量のメモリでも一時的な記憶装置としての役割を十分に果たすことができる。ポインタが配列内で輪を描くような動作をすることから、この仕組みは一般にリングバッファと呼ばれる。リングバッファでは各ポインタの位置からデータが溜まった量を判定することができ、バッファの深さを  $d$  として  $(p_{R_x} - p_W) \% d$  が保存中のデータ量である。データ量が空の場合は書き出しを進めることはできず、逆にバッファ満杯の状態では読み込みを進めることはできない。

add の動作例を図 4.12 に示す。(a) は初期状態で、 $In_0$  に有効なデータが入ってきており、 $R_0$  の位置に来たデータ (1) を入れ、 $R_0$  をインクリメントする。1 クロック後 (b) では入力  $In_0$ ,  $In_1$  ともに読み込まれる一方、この時点では  $buffer_1$  が空であるため書き出しは成立せず、出力は無効である。さらに 1 クロック後 (c) になってようやく  $buffer_0$ ,  $buffer_1$  ともにデータのストックができたため、 $W$  の位置のデータ 2 つを足し合わせたデータ (2)

が出力され、Wがインクリメントされる。Wのインクリメント後(d)もR<sub>0</sub>, R<sub>1</sub>が進んでいてストックがあるため、この場合でも書き出しが有効となる。

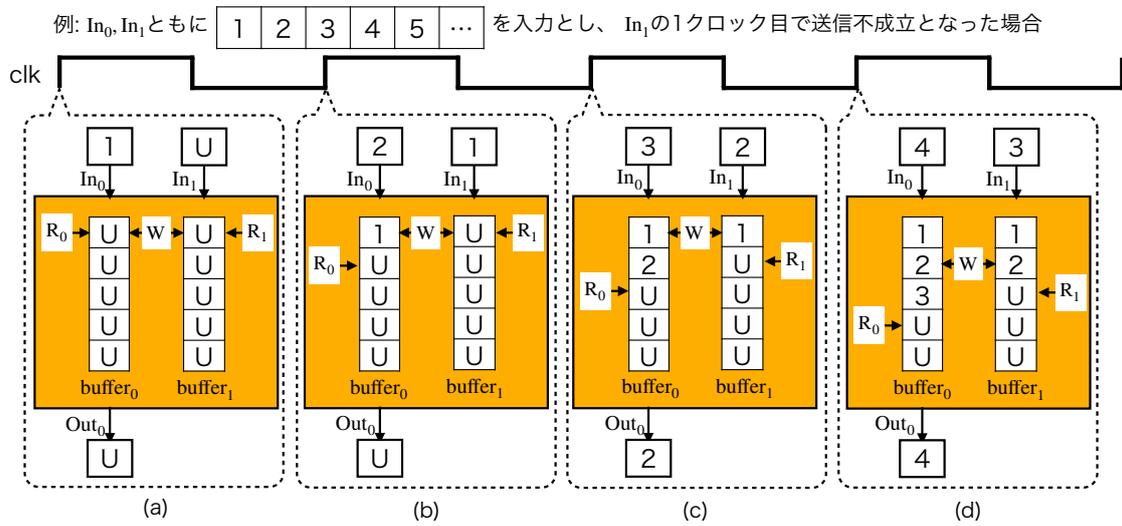


図 4.12: 加算回路でのリングバッファ動作例。R<sub>0</sub>, R<sub>1</sub>, W はそれぞれ入力 In<sub>0</sub>, In<sub>1</sub> の読み込み用ポインタと書き出し用ポインタを示す。また、U は未定義 (undefined) なデータであることを示す。

#### 4.4.4 ステートマシン

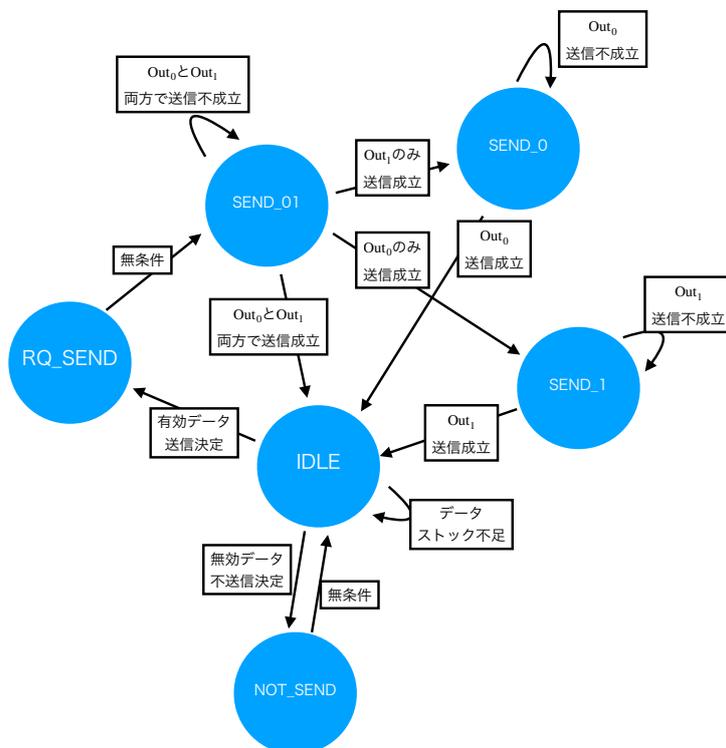


図 4.13: extract 出力データ用のステートマシン。1クロックごとに状態が移り変わっていく。

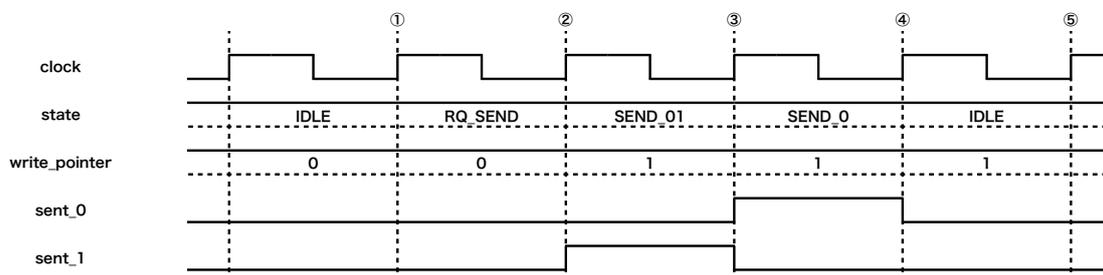
表 4.3: extract の出力ステートマシンでとりうる状態

状態名	内容	次に進む状態への判定条件
IDLE	基本状態。	データが削減対象か 送信対象か
RQ_SEND	valid を立てる準備をする状態。	無条件
SEND_01	Out <sub>0</sub> , Out <sub>1</sub> 両方で 送出しようとしている状態。	送出が成立したか
SEND_0	Out <sub>0</sub> のみ送出しようとしている状態。	送出が成立したか
SEND_1	Out <sub>1</sub> のみ送出しようとしている状態。	送出が成立したか
NOT_SEND	送出はせず、ポインタ W を 進めるための状態。	無条件

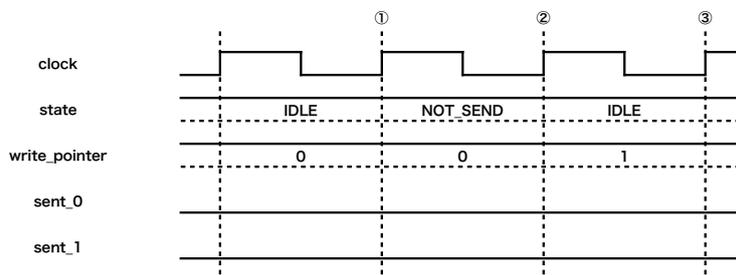
FPGA 開発では、クロックに同期して各シグナルを変化させていく。簡単な回路の場合には各シグナルのクロックごとの遷移をプロセス文で記述するだけで十分だが、高度な回路を記述しようとする、シグナル同士の結果を互いに参照しあうなど論理が複雑になっ

てくる。この状態では記述ミス元になりやすいだけでなく、複数の信号が互いの変化を待ち続けるデッドロックなどの原因にもなりうる。そこでシグナルとは別に、各クロックごとに「現在どの状態にいるのか」を状態 (state) として管理しておくことで論理が整理されやすい。クロックが変わるごとに条件によって状態は遷移することになる。このような遷移の仕方をプロセス文で記述したものはステートマシンと呼ばれる。

特に論理が複雑な extract の出力ステートマシンを例として挙げる。とりうる状態は表 4.3 に、状態遷移の全体像は図 4.13 に示した。IP の各出力  $Out_0$ ,  $Out_1$  にデータを送出するかを判断する役割を持っている。状態には基本状態の IDLE と、送出せずポインタを進めるだけの状態 NOT\_SEND と、送出タイミングに関わる他の状態がある。まず基本状態の IDLE で、リングバッファに送出データのストックがあるか、あるとすれば次の送出データがデータ削減アルゴリズムに照らして送信対象かを判断する。送信することにした場合はポインタ  $W$  のインクリメントを行い、出力  $Out_0$ ,  $Out_1$  の valid 信号を立てる。 $Out_0$ ,  $Out_1$  それぞれで送受信成立が成立するまでは SEND\_01, SEND\_0, SEND\_1 のいずれかの状態で待つことになり、両方の送信が完了すれば IDLE に状態移動する。また、データが削減対象である場合は状態は NOT\_SEND に移動し、送信はせずポインタ  $W$  のインクリメントのみ行なう。こうすることでリングバッファの見かけ上は送信完了したことになり、タイミングに一貫性を持ったまま IP の動作を続けることができる。例として、(a) 転送対象のときと (b) 削減対象のときそれぞれの状態遷移のタイミングチャートを図 4.14 に示す。いずれの場合も write\_pointer が進んでいることに注意する。



(a) データが転送対象のとき。RQ\_SEND の状態で  $Out_0$ ,  $Out_1$  の valid 信号が立てられるが、実際に送受信が成立するのは  $Out_0$  は③、 $Out_1$  は②のタイミングとした例である。



(b) データが削減対象のとき。送受信は行われず、状態は NOT\_SEND を経由するのみである。

図 4.14: 状態遷移時のタイミングチャート例。write\_pointer は書き出しポインタ  $W$  であり、sent\_0, sent\_1 は出力の送受信が成立したときにアサートされる信号である (表 A.1)。

#### 4.4.5 VHDLによる実装

実際のFPGAの開発は統合開発環境Xilinx Vivado Design Suite 2020.2<sup>2</sup>上で行なった。主な開発内容はデータ削減IPのVHDLによる実装である。実装の際は小林(2016);長谷川(2015)などを参考にしたが、基本的には筆者がアルゴリズムをVHDLで記述した。リングバッファ、ステートマシン、その他の詳細な実装は付録Aにまとめた。ここではデータ削減IPの入出力信号の一部を表4.4にまとめるのみにとどめる。データ削減IPの入力クロックは125 MHzとした。

表 4.4: データ削減IPの入力信号と出力信号

信号名	入出力	bit 数	意味/役割
extraction_mode	in	1	extract かそうでないか
sum_mode	in	1	add か subtract か
contain_event_frame	in	1	ペDESTALデータにイベントフレームが含まれているか
threshold	in	16	extract 実行時に用いられる閾値
frame_numbers	in	4	extract 実行時に用いられる、ペDESTALフレーム数
counter	out	27	どのピクセルまで計算が完了したかのカウンタ

### 4.5 動作確認テストと性能

テストとして、3章と同じように線源からX線を照射し、実装したFPGAを実際に動作させた。本節では詳細を説明する。

#### 4.5.1 データ削減モードでのデータ取得

データ取得は基本的に3.4節と同じセットアップで行なった。バッドピクセル選定のためのダーク取得を100フレーム行なったのち、<sup>109</sup>Cd線源を設置した状態で露光時間600 ms、ゲイン2.0、フレーム数1000の連続撮影を行なった。唯一異なるのは、 $n_p = 2$ ,  $n_f = 2$ ,  $\Theta = 10$  ADUとしたデータ削減モードでデータ取得を行なった点である。なお、動的ペDESTALの原理上、 $A_0$ から $A_4$ に撮影データが埋まるまではPCにデータは送信されない。そのため、送信されるフレーム数は実際の撮影数よりも $n_p + n_f + 1 = 5$ フレーム少ないことになる。実際に要求されるのは解析に利用できるデータであるため、データ削減モードでは設定フレーム数は送信フレーム数の方を指すことにする。

<sup>2</sup>ダウンロードページ:<https://japan.xilinx.com/support/download/index.html/content/xilinx/ja/downloadNav/vivado-design-tools/2020-2.html>

また、読み出し性能を測定するため、 $^{109}\text{Cd}$ を設置して $\Theta = 10$  ADUで露光時間を変えた試験と、 $^{109}\text{Cd}$ を設置せず露光時間を600 msに保ったまま $\Theta$ を変えた試験を行なった。前者はデータ削減回路と撮影回路の動作時間を、後者はデータ削減回路と外部転送の動作時間を比較する目的がある。表4.5の通りにパラメータを変えて50フレームずつデータ削減モードで撮影を行った。このとき、フレームごとの撮影開始前にLinuxの時刻計測機能を利用して取得時刻をms単位で出力した。

表 4.5: 読み出し性能測定試験でのパラメータ

露光時間 [ms]	10, 100, 300, 400, 450, 500, 550, 600, 700, 800, 1000
閾値 $\Theta$ [ADU]	1, 3, 5, 8, 10, 12, 15, 17, 20, 22, 25, 30

## 4.5.2 解析手法

データ削減モードで得られるデータはペDESTALを差し引いた波高値とそのピクセル番号である。そのため、3.4節の解析でペDESTALを引く作業が不要になる。このことを踏まえて、データ削減モードで取得したデータの解析は以下のような流れで行われる。

- (1) グッドピクセルの選定
- (2) データ読み込み  
0で初期化した仮想フレームを用意し、各ピクセル番号に対応するピクセルに波高値を入れる。
- (3)  $E_{\text{event}}$ によるイベント位置特定
- (4)  $E_{\text{split}}$ によるイベント形状判定

また、フレームあたりのデッドタイムはフレーム間のデータ取得時間差からフレームあたり露光時間を引き、その50フレーム分の平均を求めることで算出した。 $i$ フレーム目の取得開始時刻 [ms] を  $t_i$  として、

$$\text{デッドタイム} = \frac{1}{49} \sum_{i=0}^{49} (t_{i+1} - t_i - \text{露光時間}) \quad (4.17)$$

で求められる。ただし、露光時間は設定値 [ms] である。

## 4.5.3 検出性能

データ削減モードで取得した $^{109}\text{Cd}$ データのスペクトルを図4.15に、フィッティングで得られた検出性能を表4.6に示す。3.4節の結果(表3.6,3.7)と比較して、エネルギー分解能、検出効率ともに大きな違いがないことがわかる。

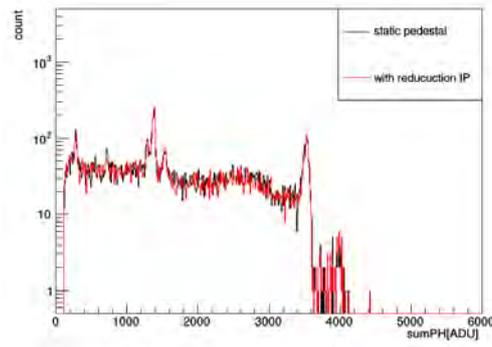


図 4.15: データ削減モードでの  $^{109}\text{Cd}$  データ取得時の weight2 スペクトル。

表 4.6:  $^{109}\text{Cd}$  の  $K_{\alpha}$  線を用いたデータ削減モードの性能評価

	エネルギー分解能 [eV]*	検出イベント数**	検出効率 (%)***
静的ペDESTAL (3.4 節)	$(4.4 \pm 0.7) \times 10^2$	21293	$0.27 \pm 0.03$
データ削減モードでの取得	$(4.2 \pm 0.6) \times 10^2$	21624	$0.26 \pm 0.02$

\* エネルギー分解能の誤差は統計誤差に由来する。

\*\* 検出イベント数および検出効率は全 weight の全吸収ピーク成分から算出している。

\*\*\* 検出効率の誤差は線源から検出器までの距離が 1 mm 前後するときの系統誤差である。統計誤差は十分小さい。

#### 4.5.4 読み出し性能

まず、フレームあたりの転送データサイズの露光時間依存性の結果を図 4.16a に示す。露光時間が伸びると到来イベント数も増えるため、データサイズもそれに比例して増えることになる。この考察は図 4.16a の結果と一致する。露光時間が 1000 ms の場合でもデータサイズは 7.6 kB であり、全ピクセル分のデータを送っていた従来 (52 MB) の  $1.5 \times 10^{-4}$  倍とデータ量を抑えることに成功している。

デッドタイムの露光時間依存性の結果を図 4.16b に示す。データ削減回路の動作時間を基準として、露光時間が短い場合はデータ削減回路動作時間がボトルネックに、露光時間が長いときは撮影に伴うデッドタイムがボトルネックになると仮定する。前者では撮影サイクル時間が露光時間に依らなくなるため、デッドタイムは露光時間とともに線形で減少していくと予想でき、後者ではデッドタイムは露光時間に依存しないと予想できる。そこで、 $a, b, x_0$  をパラメータとして、露光時間依存性のデータを

$$f(x) = \begin{cases} a(x - x_0) + b & (x < x_0) \\ b & (x \geq x_0) \end{cases} \quad (4.18)$$

でフィッティングした。フィッティング結果を表 4.7 に示す。

また、閾値 $\Theta$ を変えたときのフレームあたりデータサイズの変化を図4.17aに示す。 $\Theta$ が小さくなるほどダーク由来のノイズが入り込んでいる様子が分かる。さらに、このデータを元に描いたフレームあたりデッドタイムのデータサイズによる変化を図4.17bに示す。ただし、 $a, b, x_0$ をパラメータとして、

$$f(x) = \begin{cases} a(x - x_0) + b & (x > x_0) \\ b & (x \leq x_0) \end{cases} \quad (4.19)$$

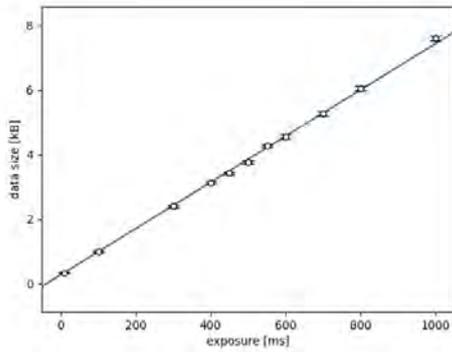
でフィッティングした結果を重ねている。表4.8にフィッティング結果パラメータを示す。この結果は、データサイズが多くなると外部転送時の処理が重くなり、そちらがデッドタイムのボトルネックになることを示している。なお外部転送がボトルネックになるのはフレームあたり転送データが1.2 MBを超えたときであり、フレームサイズが52 MBであることを考えると、有効統計量の少ない天体観測ではほとんど無視できるレベルだということが分かる。

露光時間とデータサイズに依存するとはいえ、最高性能でフレームあたりのデッドタイムを38 msに抑えることに成功した。改善前のデッドタイムが1.0 s程度であったことを考えれば、これは劇的な進歩である。観測効率 $E$ として、

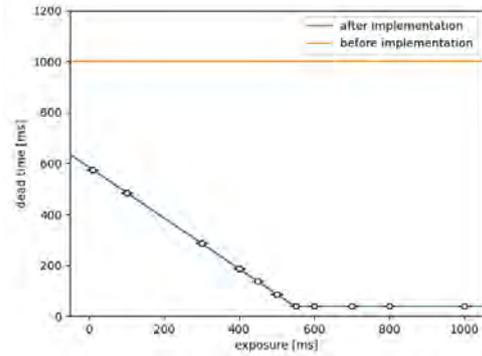
$$E = \frac{\text{実撮影時間}}{\text{実働時間}} = \frac{\text{実撮影時間}}{\text{実撮影時間} + \text{デッドタイム}} \quad (4.20)$$

という量を考える。たとえば露光時間1.0 sで連続撮影を続けるとすると、観測効率 $E$ はデータ削減実装前は0.50、実装後は0.96と、約2倍になっている。CubeSatの寿命を1年(=  $3.2 \times 10^7$  s)とすると、 $3.0 \times 10^7$  sを実観測に使えるようになったことになる。

表4.7と図4.16bの結果は、露光時間が548 msよりも短い撮影ではデータ削減回路の動作がネックになっており、露光時間が548 msを超える撮影ではデータ削減回路と無関係な処理がネックになっていることを示している。言い換えれば、 $x_0$ はデータ削減処理を高速化することで0に近づけていくことができると期待される。一方で、 $b$ は撮影読み出し回路の動作に依存すると予想され、データ削減のアイディアのみでは改善が難しい。なお、以上の考察は $^{109}\text{Cd}$ 線源のカウントレートでの動作に基づいており、より検出レートの高い場合での動作は6.2.2小節で確認する。

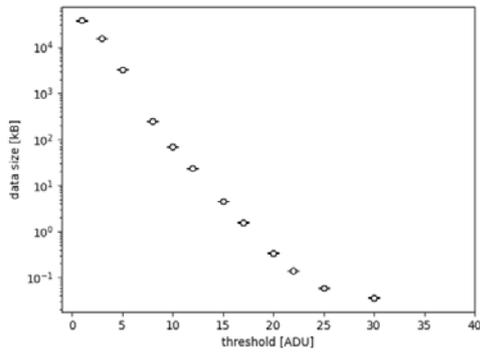


(a) フレームあたりデータサイズの露光時間依存性。点が実験値であり、誤差は統計誤差に由来する。青線は一次関数によるフィッティング結果である。

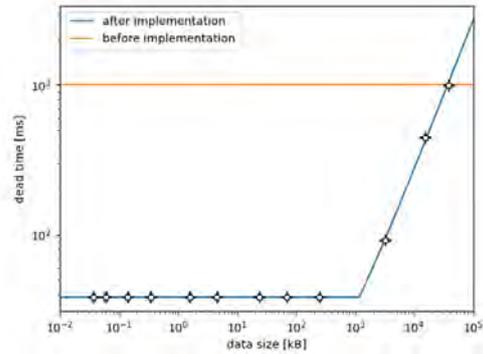


(b) フレームあたりデッドタイムの露光時間依存性。点が実験値であり、誤差は統計誤差に由来する。青線は式 (4.19) によるフィッティング結果であり、橙線は実装前の性能である。

図 4.16: 露光時間依存性試験の結果



(a) フレームあたりデータサイズの閾値依存性。誤差は統計誤差に由来する。



(b) フレームあたりデッドタイムのデータサイズ依存性。点が実験値であり、誤差は統計誤差に由来する。青線は式 (4.19) によるフィッティング結果であり、橙線は実装前の性能である。

図 4.17: 閾値依存性試験の結果

表 4.7: 露光時間依存性フィッティングパラメータ

$a$	$-0.999 \pm 0.002$
$b$ [ms]	$38.3 \pm 0.2$
$x_0$ [ms]	$548 \pm 1$

\* フィッティング関数は式 (4.18) である。

表 4.8: データサイズ依存性フィッティングパラメータ

$a$ [ms/kB]	$0.027 \pm 0.002$
$b$ [ms]	$38.3 \pm 0.2$
$x_0$ [kB]	$(1.2 \pm 0.5) \times 10^3$

\* フィッティング関数は式 (4.19) である。

## 4.6 4章のまとめ

従来の読み出しシステムはそのデータ量の多さから、連続撮影時にデッドタイムが生まれてしまっていた。この問題を解決するため、ペDESTALを動的に変化させるアルゴリズムを採用し、ZDAQ ボード内の FPGA 上に回路として実装した。 $^{109}\text{Cd}$  の特性 X 線照射による動作確認テストの結果、X 線検出性能を大きく変えることなく、フレームあたりの読み出しのデッドタイムを露光時間 600 ms 従来の 1.0 s 程度から 38 ms に抑えることに成功し、観測効率を従来の約 2 倍にすることが可能になった。

## 第5章 符号化開口のノイズ低減を目指した画像再構成法の検討

*cipher* では撮像系に符号化開口を採用する。これまでの研究で、CubeSat サイズで符号化開口による撮像が可能であることやランダムパターンを並行して用いるとノイズが減少することは確かめられてきたが (Kasuga et al., 2020)、広がった光源の画像再構成に伴うノイズが信号の 50 % 程度のレベルまで出ることがあるという課題があった (図 5.1)。このままでは観測の精度が阻害され、6 章で行う偏光を含めた撮像に影響を及ぼす恐れがある。本章では、マスクパターンの選定法と新たな再構成手法を検討することで、再構成時のノイズの低減を目指す。

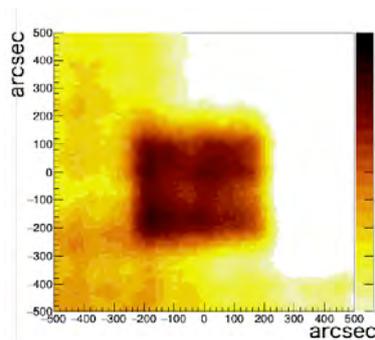


図 5.1: 先行研究での広がった天体を模した再構成画像。ランダムパターンで符号化された検出イベントを相関法により再構成している。特に  $\theta_x < 0$ ,  $\theta_y < 0$  の領域にノイズが多く出ている様子が分かる。畠内 (2021) の付録 D より引用。

### 5.1 符号化開口イメージング

本節は、Caroli et al. (1987); 春日 (2019) を参考に行っている。

### 5.1.1 概要

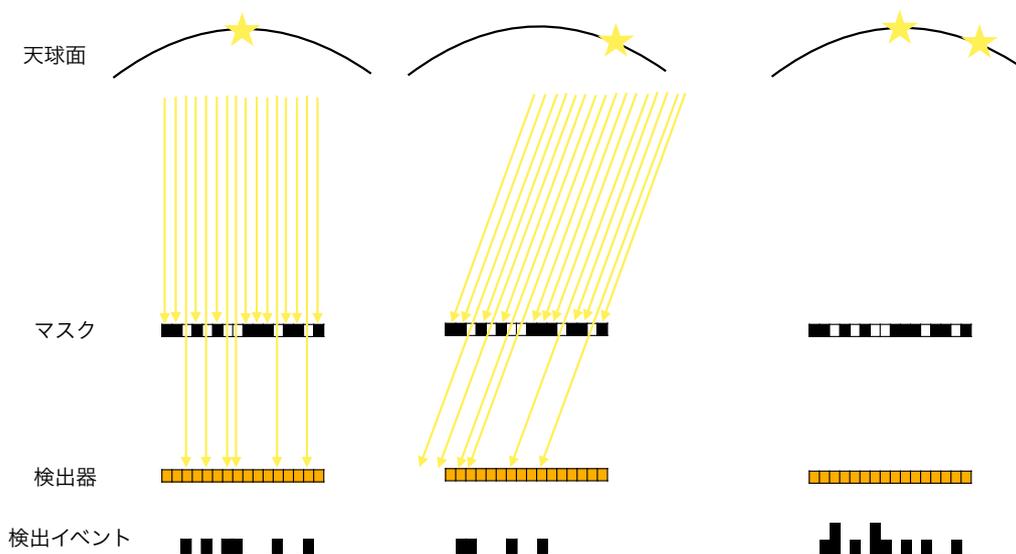


図 5.2: Encode される様子

符号化開口は、光が検出器に入射する前に符号化開口マスクと呼ばれるフィルターを通すことで、どの方向から光がやってきたかの情報を検出器面上の検出イベントの空間分布として記録する撮像手法である。図 5.2 に概要図を示す。本修論では遠距離撮像のみを扱うこととし、光源の位置情報は到来光子の入射角で表されるとする。符号化開口マスクには決まったパターンに従って穴が開けられており、到来光子は穴の空いている箇所だけ通過し、検出器上で検出される仕組みになっている。この過程は符号化 (Encoding) と呼ばれる。到来光子の入射方向によって検出の許される箇所が異なるため、逆に検出イベントの分布から天球面のどこからきたのかを推測することができる。この過程は再構成 (Decoding) と呼ばれる。

符号化開口の重要な性質のひとつに、線形性が挙げられる。すなわち、天球面上に複数の光源があったとき、検出イベント分布は各天球面要素からの検出イベントの和になっていると考えることができる。この性質は画像再構成時に重要になってくる。また、符号化開口では天体からの光子を直接検出器で検出することができ、ミラーを用いた場合のミラーによる偏光観測への影響を考慮する必要がないという利点がある。

符号化開口を利用するにあたって、重要になってくるのは (1) 再構成法の選定、(2) マスクパターンの選定、の二つである。本章では (1) として相関法と EM アルゴリズムを検討し、(2) として相関法の性能を引き出すようなパターンの選定を行なう。

### 5.1.2 撮像性能

撮像の性能を示す指標として、視野と角度分解能が挙げられる。遠距離撮像の符号化開口において、視野は「どれだけの角度までならマスクを通った光子が検出器に入りうる

か」を考えればよい。1次元方向の視野と角度分解能を考える。検出器からマスクまでの距離 [ $\mu\text{m}$ ] を  $d$ 、マスクの要素間ピッチ [ $\mu\text{m}$ ] を  $\Delta_m$ 、マスクの要素数を  $N_m$  とすると、視野は

$$\text{Field of View [rad]} \approx \arctan \frac{\Delta_m N_m}{d} \quad (5.1)$$

とみなすことができる。また、角度分解能は「ある検出器ピクセルがあるマスク要素を見上げたときに捉えられる視野」と考えると、検出器のピクセルピッチを  $\Delta_p$  として、

$$\text{Angle Resolution [rad]} \approx \arctan \frac{\sqrt{\Delta_p^2 + \Delta_m^2}}{d} \quad (5.2)$$

で計算される (Vetter et al., 2006)。

## 5.2 モンテカルロシミュレーションによる調査

### 5.2.1 セットアップ

本格的な再構成の議論に入る前に、検証用のシミュレータのセットアップの説明をする。シミュレーションにはシミュレーションライブラリ Geant4 (Agostinelli et al., 2003; Allison et al., 2006, 2016) を利用したモンテカルロシミュレーションソフトウェア ComptonSoft (Odaka 他, 2010) を用いた。基本的に図 5.2 のように、符号化開口マスクとセンサを設置した系に光子を入射させることを入射角度を変えながら繰り返す。実際のセンサとマスクの状況を再現するように、表 5.1 のパラメータに従ってセットアップを行なった。各パラメータは 6 章のビームライン試験を再現するように設定している。シミュレーションの結果として得られるのはイベントリスト (3.4 節) であり、これを使えば実際の実験データと同様の解析ができる。また、本章でのシミュレーション実行と再構成の計算環境を表 5.2 に示す。

### 5.2.2 データセット

シミュレーションではマスクパターンを変えて検証するが、パターン以外は基本的に同じ条件でデータセットを取得する。1つのデータセットは視野内に二次元に広がった光源を模したのからなり、これは点光源の集合である。各点光源データは照射方向を  $(\theta_x, \theta_y)$  として、平行光を光子数  $1 \times 10^5$  個入射することで得られる。この照射を  $-210'' \leq \theta_x \leq 210''$ ,  $-210'' \leq \theta_y \leq 210''$  の範囲で  $\theta_x, \theta_y$  それぞれ  $15''$  間隔で変えて繰り返す。得られたイベントリストを足し合わせれば、 $420''$  四方の正方形光源を模したデータが得られる。これは 6 章で後述するビーム照射試験の「スキャン」に相当する。以降では、このスキャンデータをそのまま光源としたものを「二次元面光源」、 $\theta_x = 0''$ ,  $\theta_y = 0''$  の一点のみを抜き出したものを「点光源」と表現する。

表 5.1: シミュレーションパラメータ

記号	意味	値
$N_{mx}, N_{my}$	マスク要素数	$64 \times 64$
$\Delta_{mx}, \Delta_{my}$	マスクピッチ	$35 \mu\text{m} \times 35 \mu\text{m}$
$N_{px}, N_{py}$	検出器ピクセル数	$896 \times 896$
$\Delta_{px}, \Delta_{py}$	検出器ピクセルピッチ	$2.5 \mu\text{m} \times 2.5 \mu\text{m}$
$d$	検出器からマスクまでの距離	25 cm
-	ビーム半径	2 mm
-	光源位置	検出器上空 26 cm
-	照射エネルギー	16 keV
$E_{\text{event}}$	イベント閾値	1.0 keV
$E_{\text{split}}$	スプリット閾値	50 eV
-	空乏層内電子拡散定数	$0.3 \mu\text{m}$

表 5.2: 5章での計算環境

モデル	MacPro7,1
CPU	Intel(R) Xeon(R) W-3275M CPU @ 2.50GHz
メモリ	384 GB
OS	MacOS 11.6

### 5.2.3 撮像ノイズ指標

再構成の性能を示す指標 $\nu$ として、本修論では光源領域の強度平均に対するバックグラウンド領域の不偏標準偏差の比を用いる。式で表せば、離散化した天球面分布を $S_{\theta_x, \theta_y}$ 、光源領域、バックグラウンド領域の天球面座標集合をそれぞれ $U_s, U_b$ として、

$$\nu = \frac{\sqrt{\frac{1}{|U_b|-1} \sum_{(\theta_x, \theta_y) \in U_b} \left( S_{\theta_x, \theta_y} - \frac{1}{|U_b|} \sum_{(\theta'_x, \theta'_y) \in U_b} S_{\theta'_x, \theta'_y} \right)^2}}{\frac{1}{|U_s|} \sum_{(\theta_x, \theta_y) \in U_s} S_{\theta_x, \theta_y}} \quad (5.3)$$

である。 $\nu$ が0に近いほど、ノイズレベルが低いと考えることができる。

二次元面光源の $U_b, U_s$ は境界から余裕を持った領域として、

$$U_s = \{(\theta_x, \theta_y) | (-200'' \leq \theta_x \leq 200'') \wedge (-200'' \leq \theta_y \leq 200'')\}$$

$$U_b = \left\{ (\theta_x, \theta_y) \left| \begin{array}{l} (-500'' \leq \theta_x \leq -220'') \vee (220'' \leq \theta_x \leq 500'') \\ \vee (-500'' \leq \theta_y \leq -220'') \vee (220'' \leq \theta_y \leq 500'') \end{array} \right. \right\} \quad (5.4)$$

とする。

## 5.3 画像再構成法: 相関法

### 5.3.1 バックプロジェクションによる再構成

本小節の説明は春日 (2019) を参考に行っている。マスクパターンの穴の開きが2次元の配列で表されるとする。すなわち、

$$A_{m_x, m_y} = \begin{cases} 1 & (\text{要素 } (m_x, m_y) \text{ が穴}) \\ 0 & (\text{要素 } (m_x, m_y) \text{ がマスク}) \end{cases} \quad (5.5)$$

でマスクパターンを表す。簡単のためマスク要素のピッチと検出器ピクセルのピッチが一致しているとする。このとき、検出器上ピクセル  $(p_x, p_y)$  で検出されるイベント数  $D_{p_x, p_y}$  は、

$$D_{p_x, p_y} = \sum_{(i_x, i_y) \in \text{FoV}} A_{p_x+i_x, p_y+i_y} S_{i_x, i_y} + B_{p_x, p_y} \quad (5.6)$$

で表される。ただし、 $B_{p_x, p_y}$  はバックグラウンド、 $S_{i_x, i_y}$  は方向  $(i_x, i_y)$  の光量である。この式は、「 $(p_x, p_y)$  で検出されるイベント数は視野内のあらゆる方向からの光のうち、マスクを通ったものの足し合わせである」ことを意味している。画像再構成に要求される課題は、既知のマスクパターン  $A_{m_x, m_y}$  と検出イベント分布  $D_{p_x, p_y}$  から天球面光量分布  $S_{i_x, i_y}$  を推測することにある。

相関法では、 $S_{i_x, i_y}$  をバックプロジェクションにより推測する。すなわち、あるパターン  $G_{m_x, m_y}$  があるとして、検出分布  $D_{p_x, p_y}$  から、

$$\tilde{S}_{i_x, i_y} = \sum_{p_x, p_y} G_{p_x+i_x, p_y+i_y} D_{p_x, p_y} \quad (5.7)$$

を計算し、 $\tilde{S}_{i_x, i_y}$  を得られる再構成画像とする。 $\tilde{S}_{i_x, i_y}$  を  $S_{i_x, i_y}$  にどれだけ近づけられるかが再構成の精度を決定する。式 (5.6), (5.7) から  $\tilde{S}_{i_x, i_y}$  と  $S_{i_x, i_y}$  の関係式は、

$$\begin{aligned} \tilde{S}_{i_x, i_y} &= \sum_{j_x, j_y} \sum_{p_x, p_y} G_{p_x+i_x, p_y+i_y} (A_{p_x+j_x, p_y+j_y} S_{j_x, j_y} + B_{p_x, p_y}) \\ &= \sum_{j_x, j_y} (G \star A)_{j_x-i_x, j_y-i_y} S_{j_x, j_y} + \sum_{j_x, j_y} \sum_{p_x, p_y} G_{p_x+i_x, p_y+i_y} B_{p_x, p_y} \end{aligned} \quad (5.8)$$

となる。ただし、 $G$  と  $A$  の畳み込み演算  $G \star A$  を

$$(G \star A)_{n_x, n_y} = \sum_{m_x, m_y} G_{m_x, m_y} A_{m_x+n_x, m_y+n_y} \quad (5.9)$$

とした。式 (5.8) の第一項に着目すると、理想的には  $G \star A$  が

$$(G \star A)_{n_x, n_y} = \begin{cases} 1 & ((n_x, n_y) = (0, 0)) \\ 0 & ((n_x, n_y) \neq (0, 0)) \end{cases} \quad (5.10)$$

といったデルタ関数を形成するとよい。

プロジェクションパターン  $G_{m_x, m_y}$  として考えられる最も単純なものは  $A_{m_x, m_y}$  をそのまま採用することである。これは「各検出イベントを実際に光が来た方向を含めあらゆる光源方向全てに投影する」ということに相当するが、実際にはこれだけではうまく再構成できない。なぜなら、実際には各検出イベントについて光が来ていない方向にも想定光を投影してしまい、偽像を形成してしまうからである。相関法では偽像を打ち消す手段として、 $A_{m_x, m_y}$  の代わりにマスク部分に負の重み  $-c$  ( $c > 0$ ) をつけた

$$G_{m_x, m_y} = \begin{cases} 1 & (\text{要素 } (m_x, m_y) \text{ が穴}) \\ -c & (\text{要素 } (m_x, m_y) \text{ がマスク}) \end{cases} \quad (5.11)$$

をプロジェクションパターンに採用する。 $c$  はパターンの開口率  $f$  によって決まり、

$$c = \frac{f}{1-f} \quad (5.12)$$

である。この  $G$  と  $A$  の畳み込み  $G \star A$  を相互相関と呼ぶ。図 5.3 にバックプロジェクションの様子を示す。

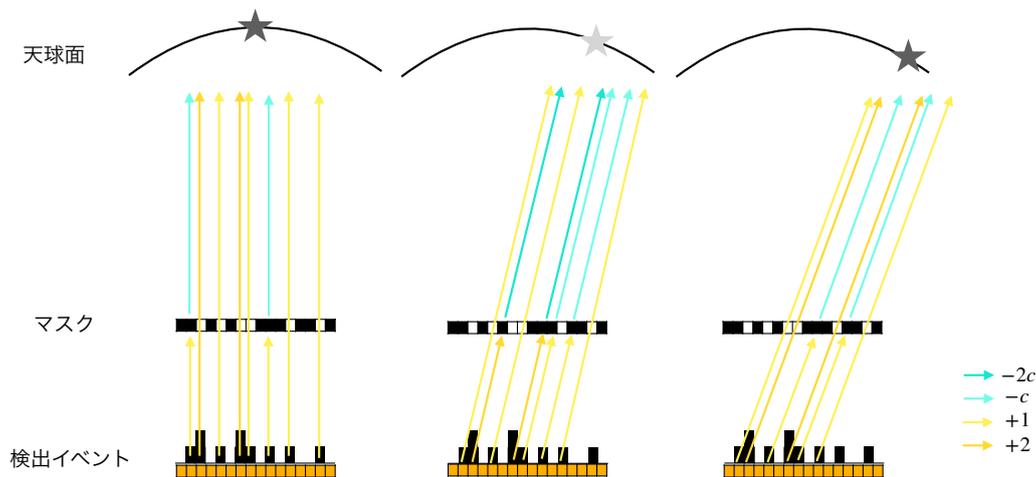


図 5.3: 相関法でバックプロジェクションする様子

### 5.3.2 実装

実際の計算機上での画像再構成の流れは、ピクセルの大きさなども考慮して以下のようになる。

- (1) 再構成分布  $\tilde{S}$  をいたるところ 0 で初期化する。
- (2) 各検出器ピクセル  $(p_x, p_y)$  と各天球面座標  $(\theta_x, \theta_y)$  に注目する。

- (3) ピクセル  $(p_x, p_y)$  から  $(\theta_x, \theta_y)$  の方向を見上げたときのマスク要素座標  $(m_x, m_y)$  を計算する。これは、マスク座標と検出器座標で原点が一致しているとして、

$$m_x = \text{Round} \left( \frac{\Delta_{px} p_x + d \tan \theta_x}{\Delta_{mx}} \right) \quad (5.13)$$

$$m_y = \text{Round} \left( \frac{\Delta_{py} p_y + d \tan \theta_y}{\Delta_{my}} \right) \quad (5.14)$$

で計算できる。ただし、 $\text{Round}(x)$  は実数  $x$  の整数への四捨五入である。

- (4)  $(m_x, m_y)$  がマスク座標系内にあるか、すなわち  $G_{m_x, m_y}$  が定義されているか確認する。なければこの  $(p_x, p_y)$  と  $(\theta_x, \theta_y)$  の組み合わせでのステップは終わる。
- (5) 値  $G_{m_x, m_y} D_{p_x, p_y}$  を  $\tilde{S}_{\theta_x, \theta_y}$  に加える。

### 5.3.3 並行ランダムパターン

相関法では  $G \star A$  がデルタ関数となるようなパターンの選定が必要になってくる。定性的には、パターンがなるべく周期性を持たないようにするのが理想である。周期性が弱い中で開口率を 50 % 程度にすることが可能なパターンには、ランダムパターン、M-sequence(MacWilliams and Sloane, 1976), URA(Fenimore and Cannon, 1978) などが知られているが、本修論ではランダムパターンのみを考えることにする。

ランダムパターンはその名の通り、全マスク要素の中から開ける穴をランダムに選んで開けていく、という作業を目標の開口率になるまで繰り返すことで得られる。URA に比べてランダムパターンはパターンの選び方次第で偽像を少なく抑えることができ、バックグラウンドを低減する効果が期待できる。また、複数のランダムパターンで並行して Encoding と Decoding を行なった後、各パターンの再構成画像を足し合わせると偽像が減少することが先行研究で明らかになっている (Kasuga et al., 2020)。本研究でも並行イメージングを採用することにする。

図 5.4 に並行ランダムパターンの一例を示す。64 × 64 要素のランダムパターンが 8 種類並んでいる。

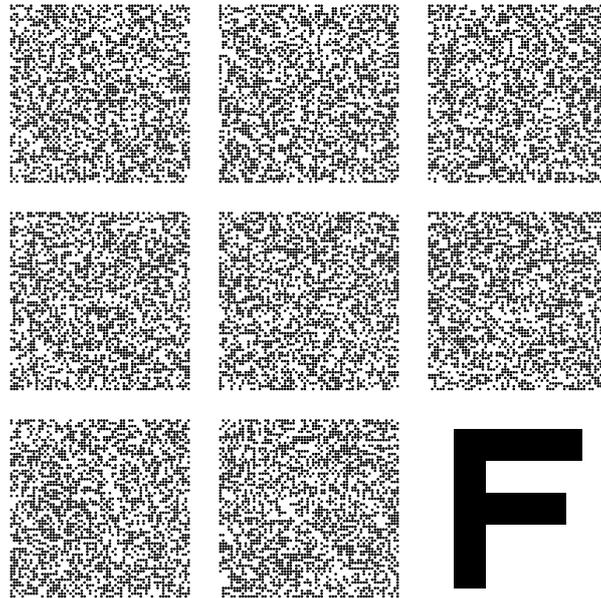


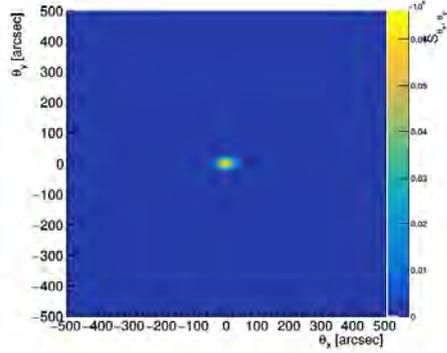
図 5.4: ランダムパターン。拡大すると黒い穴がランダムに開けられている様子が分かる。穴は実際には直径  $35 \mu\text{m}$  で、パターン一つあたり  $64 \times 64$  要素並んでいる。最終的にこれが6章のビームライン実験で使用された。右下の“F”はマスク方向の確認用に空けてある。

### 5.3.4 ランダムパターン選定: 大局的な考察

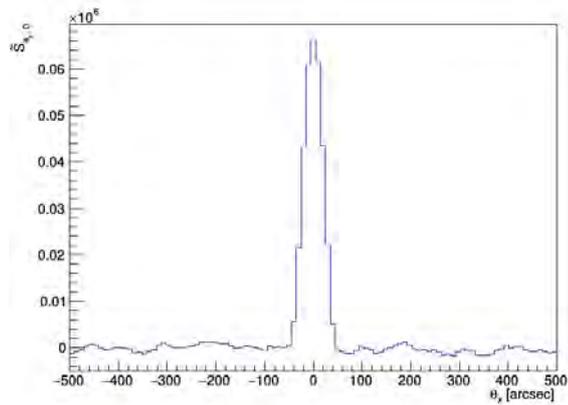
図5.4のパターンを用いてシミュレーションを行い、点光源と二次元面光源それぞれで相関法で再構成した画像を図5.5に示す。再構成時の天球面座標は  $-500'' \leq \theta_x \leq 500''$ ,  $-500'' \leq \theta_y \leq 500''$  範囲内で、 $\theta_x, \theta_y$  ともにピッチは  $10''$  である。概ね再構成が成功していると考えられることもできるが、特に二次元面光源では  $\theta_x < 0, \theta_y > 0$  の領域に集中的に偽像が生まれており、偽像が本来の信号と同レベルにまで達している。撮像ノイズ  $\nu$  (式(5.3)) は  $0.95$  である。

この現象がなぜ起こるのか大局的に考察する。極端な例として、図5.6のように穴の開き方に偏りがある場合を考える。正面点光源を符号化すると、穴の多く空いている領域を光は通過しやすく、これは検出器上のイベント分布にも反映される。この偏りはバックプロジェクションの際にも影響を受け、穴の多く空いた方向に投影されやすいと考えることができる。

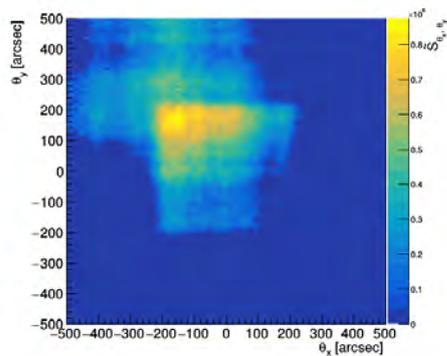
式を用いて簡単な考察を行なう。たとえば1次元マスクパターンの領域を2分割し、それぞれの領域を  $a, b$ 、開口率を  $f_a, f_b$  とする。正面から光量  $I$  の光が来たとする、 $a, b$  に対応する検出器領域にはそれぞれ  $f_a I, f_b I$  の光量の光が到達する。このとき相関法によるバックプロジェクションを行なうと、検出器領域  $a$  からマスク領域  $b$  を通って射影され



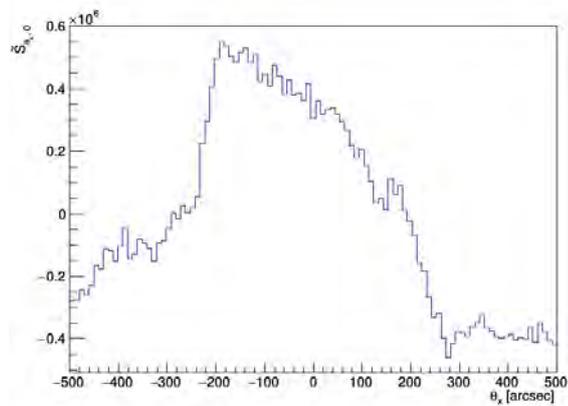
(a) 2次元画像 (点光源)



(b)  $\theta_y = 0$  による断面 (点光源)



(c) 2次元画像 (二次元面光源)。 $\nu = 0.95$  である。



(d)  $\theta_y = 0$  による断面 (二次元面光源)

図 5.5: 図 5.4 のパターンを用いたシミュレーションの相関法による再構成画像。8 パターン分の合計を示している。ただし見やすさのため、2次元画像のバックプロジェクションの結果が負であるところは0に切り上げてある。

る期待光量は、 $f_a I \times (f_b - c(1 - f_b)) = ((1 + c)f_b - c) f_a I$  と見積もられる。この光量が0になるのは  $(1 + c)f_b - c = 0$ 、すなわち  $f_b$  が全体の開口率  $f$  と一致するときである (式 (5.12) を参照)。一般には各領域で開口率が均等ではないため、正面以外にノンゼロの光量が射影される。これが偽像となって表れていると考えられる。

実際に図 5.4 の領域ごとの開口率を示したものを図 5.7 に示す。偽像が多く出ていた  $\theta_x < 0, \theta_y > 0$  の領域で他よりも開口率が高いことがわかる。このことから、パターン全体だけでなく領域を区切ったときにも開口率が均等となるようなパターンが理想的であるという仮説が生まれる。

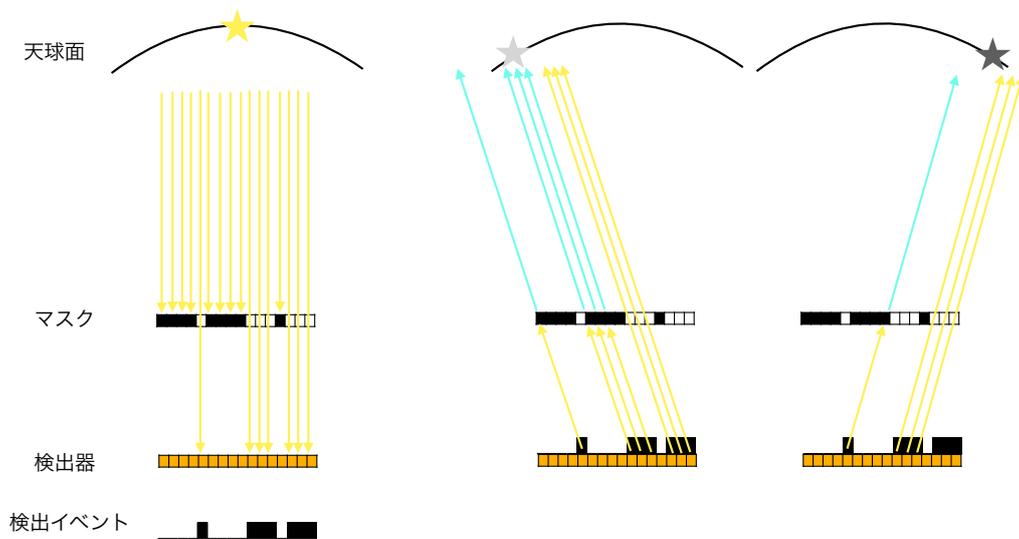


図 5.6: マスクパターンに偏りがあるときの Encoding と Decoding の様子

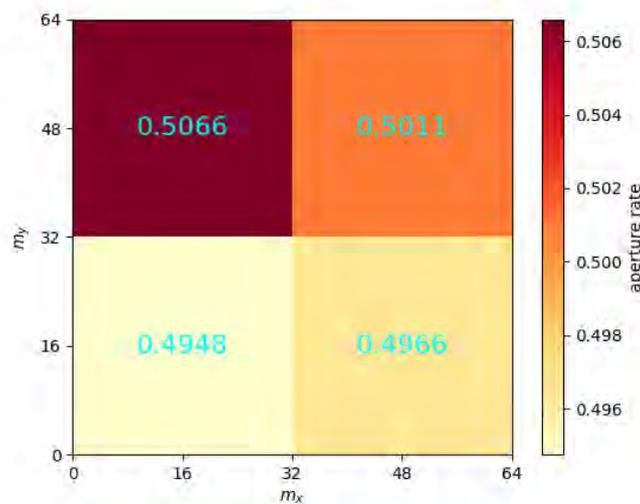


図 5.7: 図 5.4 を  $2 \times 2$  の領域に区切ったときの領域ごとの開口率

### 5.3.5 ランダムパターン選定: 一様束縛条件の導入

領域ごとの開口率が一定となるように、ランダムパターンに束縛条件を導入することを考える。ここでは、後に並行パターンごとの再構成画像を足し合わせることを前提として、複数パターンで足し合わせた開口率が一定となるように設計する。パターン番号  $k$  を導入し、穴の開き具合 (式 5.5) を  $A_{k,m_x,m_y}$  で表す。このとき、パターン数を  $K$ 、開口率を

$f$ として、

$$\sum_{k=0}^{K-1} A_{k,m_x,m_y} = Kf \quad (\forall m_x, m_y) \quad (5.15)$$

であれば条件を満たす。ただし、 $K$ と $f$ は $Kf$ が整数となるように選ぶものとする。ここで、各パターン要素 $(m_x, m_y)$ で $k$ の分布として $A_{k,m_x,m_y}$ を考える。式(5.15)を満たすような $(A_{0,m_x,m_y}, \dots, A_{K-1,m_x,m_y})$ の組は $K$ 個から $Kf$ 個選ぶ組み合わせと等価であり、 ${}_K C_{Kf}$ 通りある。この ${}_K C_{Kf}$ 通りを各 $(m_x, m_y)$ に均等に割り振ることで、各パターンでの周期性を打ち消すランダム性と $f$ 程度の開口率を保ったまま、式(5.15)の束縛条件を満たすことができる。

前節のセットアップに合わせて具体的にパターンの選び方を考える。説明上、ある要素 $(m_x, m_y)$ に着目したパターン番号ごとの穴の開き方の組 $(A_{0,m_x,m_y}, \dots, A_{K-1,m_x,m_y})$ を「ホールセット」と呼ぶ。今は $K = 8, f = 0.5$ であるから、ホールセットは ${}_8 C_4 = 70$ 通り存在する。各パターンの要素数は $64 \times 64 = 4096$ 個であり、各ホールセットに4096要素を均等に割り振ることになる。 $4096 = 70 \times 58 + 36$ から、58要素ずつをホールセット70通りに、さらに余分な36要素を異なるホールセットに割り振る。以下のようにパターンセット $A$ を決定する。

- (1) 番号付けされた4096個の要素が入った箱を用意する。
- (2) 相補的なホールセットのペアを重複なしで $36/2 = 18$ 組ランダムに選ぶ。ここで相補的なホールセットとは、あるホールセットに対して01を逆転させたものをいう。合計で異なる36通りのホールセットが選ばれる。
- (3) 70通りのホールセットに対して、そのホールセットが(2)で選ばれていた場合59要素を、そうでない場合は58要素を、(1)の箱からランダムに取り出して対応グループにすることを繰り返す。
- (4) これで各ホールセットに対して所属する要素が決まり、各要素に対してホールセットに従って穴を開けていく。

図5.8にこのパターンセット決定法の概略図を示す。この手順に従うと、各パターンに関しては $f = 0.5$ の開口率が保たれたまま、各ホールセットに対しても8つのうち4つだけが開口していることになる。(1)で相補的なパターンを用いたが、これは各パターンで開口率 $f = 0.5$ を保ったままという制約を満たすために「余り」をなるべく均等に選ぶことにしたためである。そのためこのパターン選定方法は他の開口率の場合にはそのまま適用はできないが、パターンごとの開口率を気にしなければ、「余り」の分はホールセットを重複なしで適当に選んで補うことで他の開口率にも応用可能である。

### 5.3.6 ランダムパターン選定: 相互相関からの最適パターンの選出

すでに述べた通り(式(5.10))、相関法ではパターン $A$ は相互相関 $G \star A$ がデルタ関数をなすのが理想である。そこで、 $G \star A$ のピーク以外での要素の二乗和を、パターンの理想

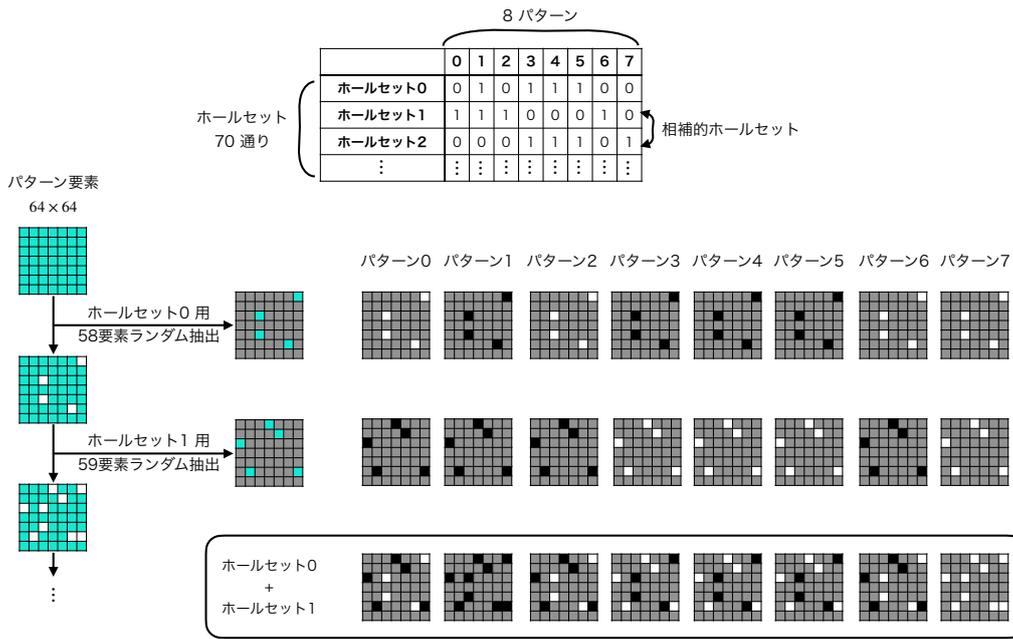


図 5.8: 一様束縛条件を入れたランダムパターンセットの作成図。分かりやすさのため、パターン要素数は実際の  $64 \times 64$  より少なくしてある。まず「余り補正」の相補的ホールセットを選択する (図上)。次にパターン要素の箱 (図中左) からランダムに要素を 58 個抽出し、ホールセットに従って各パターンに穴を開ける (図中右)。最初に選んだ相補的ホールセットの場合は 59 個抽出する。この作業を全要素を使い切る 70 ホールセット分まで繰り返す。

値からのずれを示すパターンスコアとして導入する。式で表すと、

$$\text{Pattern Score} = \sum_{k=0}^{K-1} \sum_{(n_x, n_y) \neq (0,0)} \left\{ (G \star A)_{k, n_x, n_y} \right\}^2 \quad (5.16)$$

である。

前節の一様束縛条件の効果を確認するため、一様束縛条件を導入した場合と導入していない場合について、式 (5.16) によるパターンスコアを比較した。ただし、いずれもランダム性によって性能は前後すると考えられるので、それぞれ 1000 種類の seed によるパターンセットを用意し、パターンスコアを計算した。一様束縛条件がない場合も一様束縛がある場合と条件を合わせるため、開口率 50 % のパターン 8 枚を 1 セットとして扱って計算した。計算結果のヒストグラムを図 5.9 に示す。また、統計情報を表 5.3 に示す。平均値、最小値ともに一様束縛条件を導入した方が良いスコアを出している。このことから、一様束縛条件の有用性を確認できた。

### 5.3.7 最適パターンでの画像再構成

一様束縛条件を導入した中で、最もパターンスコアが低かったものを最適パターンセットとして、実際にシミュレーションによる符号化と相関法による再構成をする。表 5.3 の

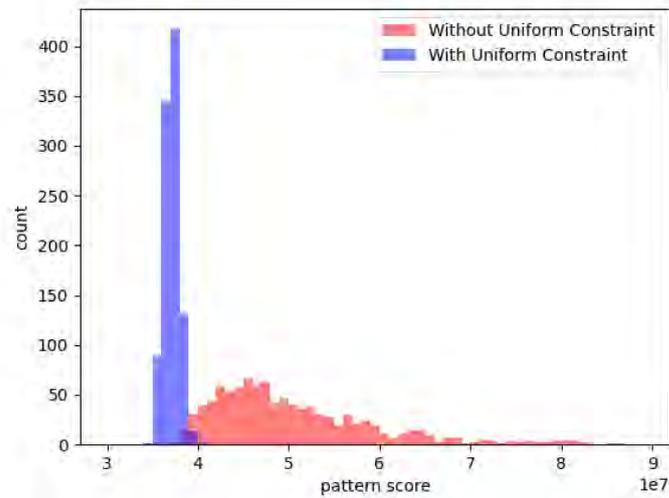


図 5.9: パターンスコア (式 5.16) の比較。赤が一樣束縛条件なし、青が一樣束縛条件ありのパターンである。

表 5.3: ランダムパターン性能比較

	パターンスコア 平均値	パターンスコア 最小値
一樣束縛条件なし	$5.08 \times 10^7$	$3.58 \times 10^7$
一樣束縛条件あり	$3.71 \times 10^7$	$3.48 \times 10^7$

\* 式 (5.16) で計算される値

一樣束縛条件ありのパターンスコア最小値を出したパターンを図 5.10 に示す。また、最適化していないパターン (図 5.4) と最適パターンでの相互相関の分布 (式 (5.9)) を図 5.11 に示す。相互相関の分布に偏りがある図 5.4 のパターンと比較して、一樣束縛条件を導入した最適パターンは分布が均等になっている様子が分かる。

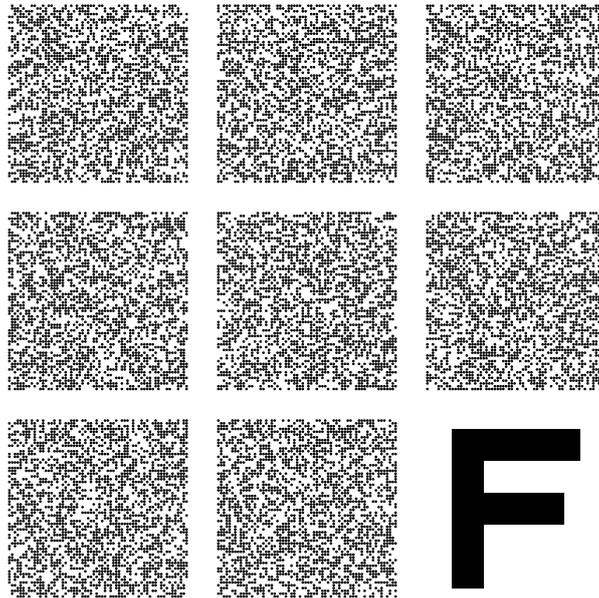
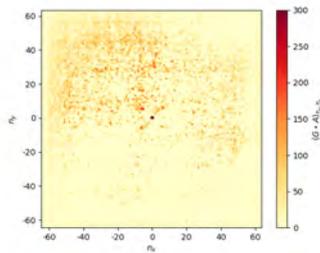
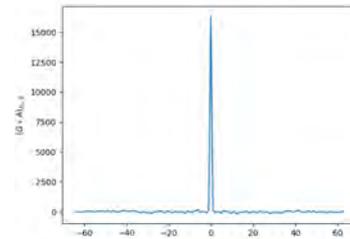


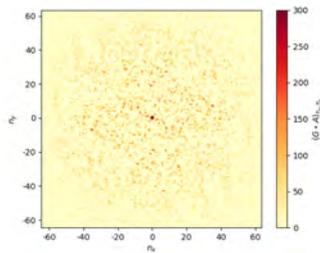
図 5.10: 一様束縛条件を導入したランダムパターンセット。拡大すると、各要素位置について8パターン中穴が空いているのは4つであることが確認できる。



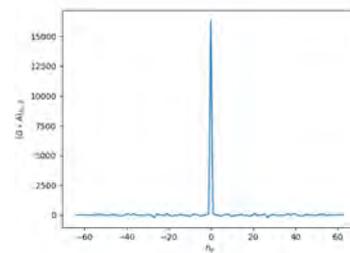
(a) 最適化をしていないパターンの  $(n_x, n_y)$  上二次元分布



(b) 最適化をしていないパターンの  $n_x = 0$  での断面



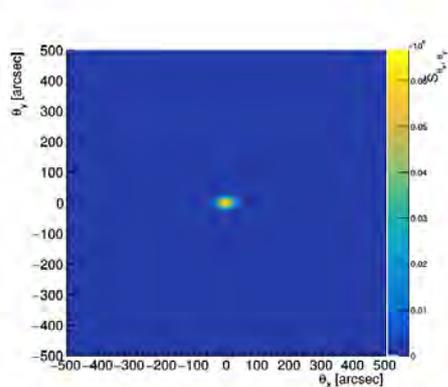
(c) 最適パターンの  $(n_x, n_y)$  上二次元分布



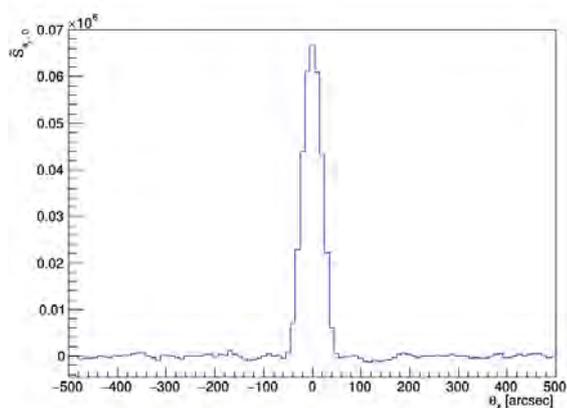
(d) 最適パターンの  $n_x = 0$  での断面

図 5.11: 相互相関の分布。ただし、二次元分布の方はノイズ分布の見やすさのためカラーバーの上限を 300 で切っている。

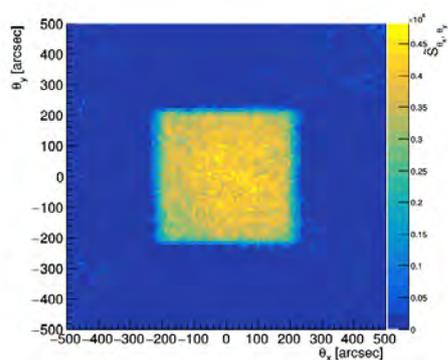
最適パターンで二次元スキャンのシミュレーションを行なったのち、相関法で再構成を行なった。再構成時の天球面座標は  $-500'' \leq \theta_x \leq 500''$ ,  $-500'' \leq \theta_y \leq 500''$  範囲内で、 $\theta_x, \theta_y$  ともにピッチは  $10''$  である。再構成後の画像を図 5.12 に示す。図 5.5 の結果と比較して、領域によって偽像が出やすくなるということはなくなり、二次元面光源の撮像ノイズ  $\nu$  (式 (5.3)) は 0.097 と最適化をしていないパターンの 0.10 倍程度に抑えられている。



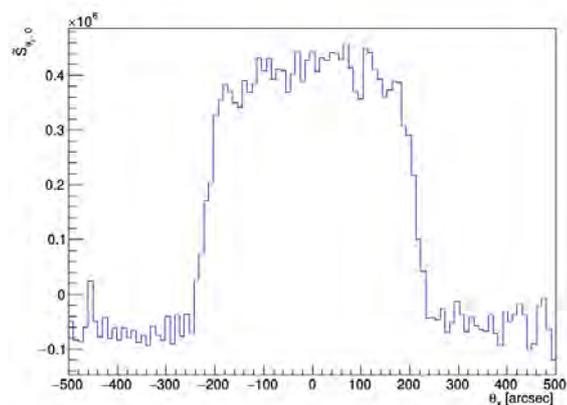
(a) 2次元画像 (点光源)



(b)  $\theta_y = 0$  による断面 (点光源)



(c) 2次元画像 (二次元面光源)。 $\nu = 0.097$  である。



(d)  $\theta_y = 0$  による断面 (二次元面光源)

図 5.12: 図 5.10 のパターンを用いたシミュレーションの相関法による再構成画像。8 パターン分の合計を示している。ただし見やすさのため、2次元画像のバックプロジェクションの結果が負であるところは0に切り上げてある。

## 5.4 画像再構成法: EM アルゴリズム

本節では画像再構成法として新たに Expectation-Maximization (EM) アルゴリズムを検討する。EM アルゴリズムに適用するための符号化開口イメージング用のモデルを新たに作成し、マスクパターンが理想的でない場合でも偽像の少ない画像が得られるかを調査する。

### 5.4.1 観測データへの最尤法適用

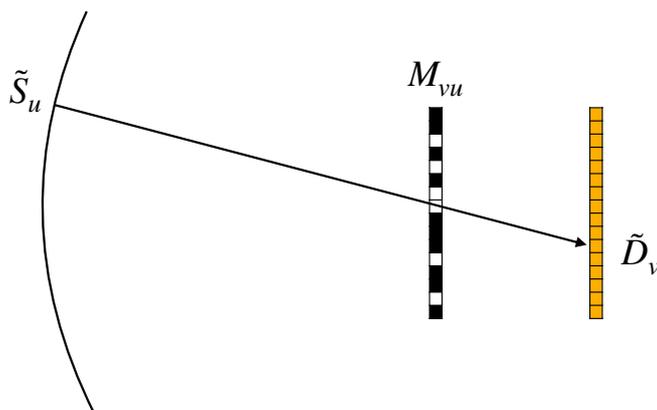


図 5.13: 5.4 節の変数定義。

符号化開口に限らず、一般的な光子観測過程を想定する。本小節の説明は Ikeda et al. (2014) を参考に行っている。離散化された天球面座標の集合を  $U$ , 検出器座標の集合を  $V$  とし、仮想的な天球面の光子分布を  $\tilde{S}$ , 仮想的な検出器上の検出イベント分布を  $\tilde{D}$  で表すことにする (図 5.13)。ただし、 $\tilde{S}, \tilde{D}$  はそれぞれ規格化されている。すなわち、

$$\sum_{u \in U} \tilde{S}_u = 1 \quad (5.17)$$

$$\sum_{v \in V} \tilde{D}_v = 1 \quad (5.18)$$

である。また、「ある天球面座標  $u$  から到来する光子が、各検出器座標  $v$  にどれだけ降り注ぎやすいか」の事後確率分布を  $M_{vu}$  とする。 $M_{vu}$  は各  $u$  について規格化されており、

$$\sum_{v \in V} M_{vu} = 1 \quad (\forall u \in U) \quad (5.19)$$

である。ある天球面分布  $\tilde{S}$  を仮定したとき、検出イベント分布  $\tilde{D}$  は行列  $M$  を  $\tilde{S}$  に掛けたものと思えることができる。成分をあらわに書けば、

$$\tilde{D}_v = \sum_u M_{vu} \tilde{S}_u \quad (5.20)$$

である。さらに、検出器座標  $v$  で検出されたイベント数を  $D_v$ 、 $D_v$  を規格化した検出イベント分布を  $\delta_v$  とする。このとき、

$$\sum_{v \in V} \delta_v = 1 \quad (5.21)$$

である。

画像再構成のゴールは、観測データ  $\delta_v$  を実現するような天球面分布  $\tilde{S}$  を推定することにある。ここでは相関法での定義 (5.3) と異なり  $\tilde{S}$  は規格化されているが、天球面上の相

対値のみに関心があるとして、本節では規格化された状態をそのまま画像とみなす。各検出イベントがポアソン分布に従っていると仮定する。EM アルゴリズムでは、最尤推定を繰り返し行うことでパラメータを決定していく。仮に決定したパラメータからデータの期待値を計算し (Expectation)、期待値と測定値の比較から尤度を上げる方向にパラメータを更新する (Maximization) ことを交互に繰り返す。具体的には、 $l$  ステップ目での天球面分布を  $\tilde{S}^{(l)}$  として、

$$\tilde{D}_v^{(l)} = \sum_u M_{vu} \tilde{S}_u^{(l)} \quad (\forall v) \quad (5.22)$$

$$\tilde{S}_u^{(l+1)} = \sum_v \delta_v \frac{M_{vu} \tilde{S}_u^{(l)}}{\tilde{D}_v^{(l)}} \quad (\forall u) \quad (5.23)$$

と  $l+1$  ステップ目の天球面分布  $\tilde{S}^{(l+1)}$  を更新する (Shepp and Vardi, 1982)。式 (5.22) を E-step、式 (5.23) を M-step と呼ぶ。まず、E-step で仮決定中の天球面分布  $\tilde{S}_u^{(l)}$  をもとに検出イベント分布  $\tilde{D}_v^{(l)}$  を推定している。その後 M-step で、 $M_{vu} \tilde{S}_u^{(l)} / \tilde{D}_v^{(l)}$  の分布上での各観測データ  $\delta_v$  の期待値を天球面に射影し直したものが次の  $\tilde{S}^{(l+1)}$  と考えることができる。

EM アルゴリズムではステップの実行後の尤度は実行前の尤度から必ず上昇することが保証されている (Dempster et al., 1977) ため、E-step と M-step を繰り返せば  $\tilde{S}$  は次第に求める解に近づいていく。現実的には  $\tilde{D}$  のステップ間の差が十分に狭くなったところで計算を終了する。どれくらい収束したかを確認する指標としては適当なノルムが用いられるが、ここではユークリッドノルムを採用する。式で表すと以下である。

$$\Delta \tilde{D}^{(l)} = \sqrt{\sum_v \left( \tilde{D}_v^{(l+1)} - \tilde{D}_v^{(l)} \right)^2} \quad (5.24)$$

## 5.4.2 符号化開口への適用

EM アルゴリズムを適用するため、符号化開口イメージングの場合でのモデルを考える。このとき、 $U, V, M_{vu}$  を事前に決定する必要がある。本研究において  $V$  は検出器上のピクセル  $(p_x, p_y)$  の集合と対応し、 $U$  は角度分解能程度のピッチを持つ視野角の集合と対応する。 $M_{vu}$  は符号化開口マスクの特性を表現すると考えることができる。符号化開口マスクと検出器面の距離  $[\mu\text{m}]$  を  $d$ 、検出器ピクセルピッチ  $[\mu\text{m}]$  を  $(\Delta_{px}, \Delta_{py})$ 、マスクパターンの要素間ピッチ  $[\mu\text{m}]$  を  $(\Delta_{mx}, \Delta_{my})$ 、マスク要素数を  $(N_{mx}, N_{my})$  とする。また、検出器座標とマスク座標で原点が一致しているとする。ある検出器座標  $v = (p_x, p_y)$  からある天球面座標  $u = (\theta_x, \theta_y)$  を見上げたとき、対応するマスク要素  $(m_x, m_y)$  は

$$m_x(u, v) = \text{Round} \left( \frac{\Delta_{px} p_x + d \tan \theta_x}{\Delta_{mx}} \right) \quad (5.25)$$

$$m_y(u, v) = \text{Round} \left( \frac{\Delta_{py} p_y + d \tan \theta_y}{\Delta_{my}} \right) \quad (5.26)$$

で計算できる。ただし、 $\text{Round}(x)$  は実数  $x$  の整数への四捨五入である。符号化開口マスク上では、パターン要素  $(m_x, m_y)$  が穴であれば光子が通過でき、逆に塞がっているもし

くはマスク範囲外のときは光子は止められると考えることができる。式で表せばパターン要素  $(m_x, m_y)$  の通過率  $\text{Pass}(m_x, m_y)$  を、

$$\text{Pass}(m_x, m_y) = \begin{cases} 1 & (0 \leq m_x < N_{m_x} \text{ and } 0 \leq m_y < N_{m_y} \text{ and } A_{m_x, m_y} = 1) \\ \tau & (\text{otherwise}) \end{cases} \quad (5.27)$$

とすることができる。ただし、 $\tau$  はマスク部分の光子透過率 ( $0 \leq \tau \leq 1$ ) であり、理想的には  $\tau = 0$  である。以降、特に断りがなければ  $\tau = 0$  とする。 $M_{vu}$  はマスクパターンの通過率を各  $u$  について規格化したものと考えることができるため、式 (5.25), (5.26), (5.27) から各  $u, v$  に対応する  $M_{vu}$  を次のように決めることができる。

$$M_{vu} = \frac{\text{Pass}(m_x(u, v), m_y(u, v))}{\sum_{v'} \text{Pass}(m_x(u, v'), m_y(u, v'))} \quad (5.28)$$

本来は検出器の検出効率も考慮すべきだが、ピクセルごとの検出効率の差は十分小さいと考え、本研究では考慮しない。

また、並行パターンを採用する場合、パターン番号を  $k$  として、 $v = (k, p_x, p_y)$  で検出器座標を表し、マスクの状況は  $A_{k, m_x, m_y}$  で表す。式 (5.25), (5.26), (5.27) それぞれに対して、 $v$  と  $A$  を置き換えれば自然に並行パターンに拡張できる。これは天球面分布セットアップはそのままに、単にパターンの要素数を増やしたものと考えられることもできる。

### 5.4.3 実装

実際の画像再構成の流れは以下のようになる。

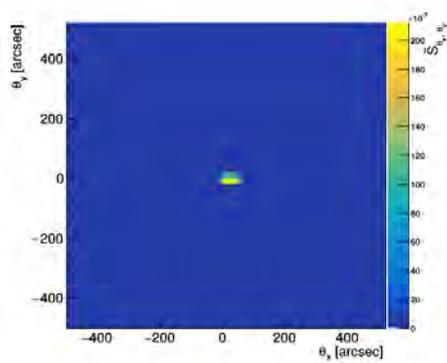
- (1)  $\tilde{S}_u, \tilde{D}_v$  の初期化  
 $\tilde{S}_u, \tilde{D}_v$  はともに一様分布で初期化する。
- (2)  $M_{vu}$  の構成  
 マスクの情報  $A$  を元に、式 (5.25), (5.26), (5.27), (5.28) に従って行列  $M_{vu}$  を作成する。
- (3) 逐次ステップ  
 実験データ  $\delta_v$  を元に、式 (5.22) と式 (5.23) の実行を交互に繰り返す。このとき、各ステップで  $\tilde{S}^{(l)}$  の値と  $\Delta \tilde{D}^{(l)}$  を記録しておく。
- (4) 適当なステップで繰り返しを終了する。本節では最適な終了条件の判断のため、1000 ステップまで繰り返しを行なう。

### 5.4.4 シミュレーションデータの再構成

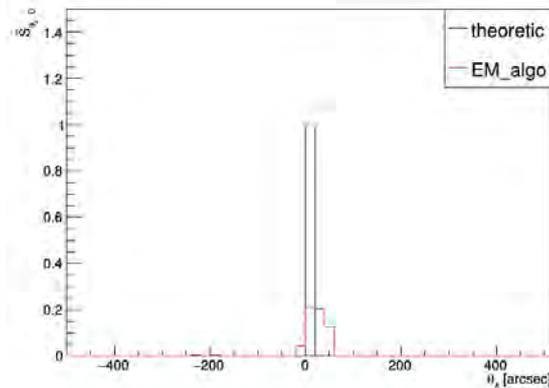
シミュレーションで得たデータから、EM アルゴリズムを用いて画像再構成を行なう。EM アルゴリズムの相関法に対する効果を調べるため、相関法で最適化したランダムパターン (図 5.10) ではなく、あえて最適化されていないランダムパターン (図 5.4) のデータ

を用いる。天球面座標  $U$  は  $-500'' \leq \theta_x \leq 500''$ ,  $-500'' \leq \theta_y \leq 500''$  の範囲で  $20''$  ごとにとる。

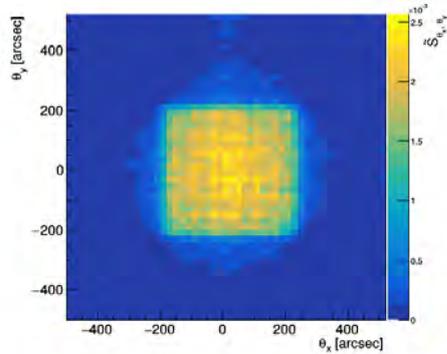
まず、ステップ数 999 の際に得られた再構成画像を図 5.14 に示す。相関法による再構成 (図 5.5) と比較して、再構成時の偽像が少なくなっている様子がわかる。二次元面光源画像の撮像ノイズ  $\nu$  (式 (5.3)) は 0.11 であり、一様束縛条件を導入したランダムパターンセットの相関法による画像再構成 (図 5.12) の  $\nu = 0.097$  には及ばないものの、同じマスクパターンでの相関法による画像再構成 (図 5.5) の  $\nu = 0.95$  の 0.12 倍と大幅に改善している。このことから、マスクパターンが最適でないときでも EM アルゴリズムを用いれば相関法より少ない偽像で画像再構成が可能であることが分かった。



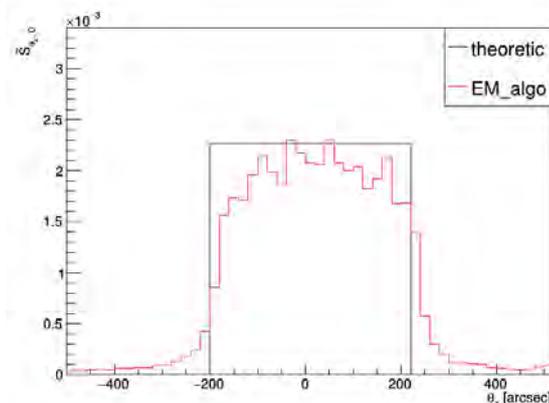
(a) 2次元画像 (点光源)



(b)  $\theta_y = 0$  による断面 (点光源)



(c) 2次元画像 (二次元面光源)。 $\nu = 0.11$  である。



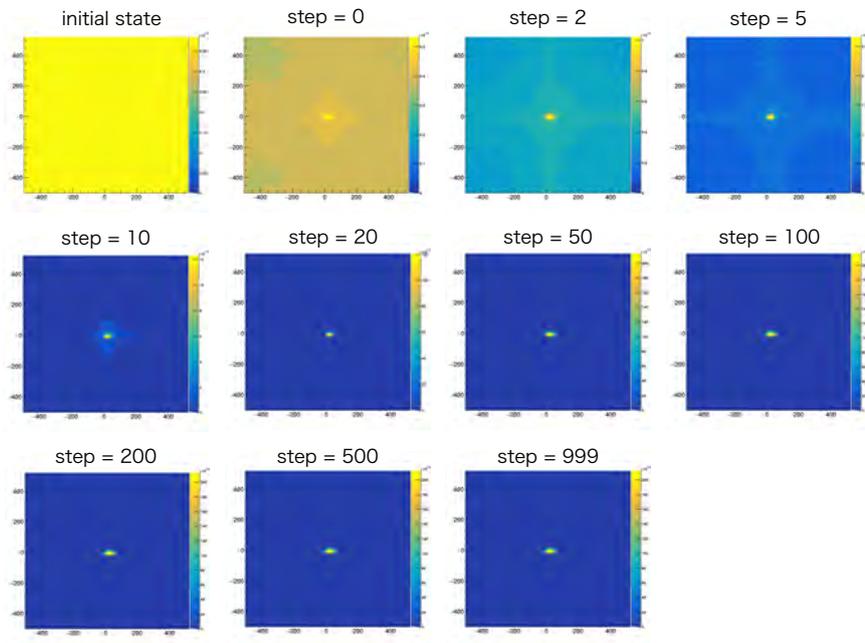
(d)  $\theta_y = 0$  による断面 (二次元面光源)

図 5.14: EM アルゴリズムによる再構成画像。いずれも 999 ステップ目を示している。

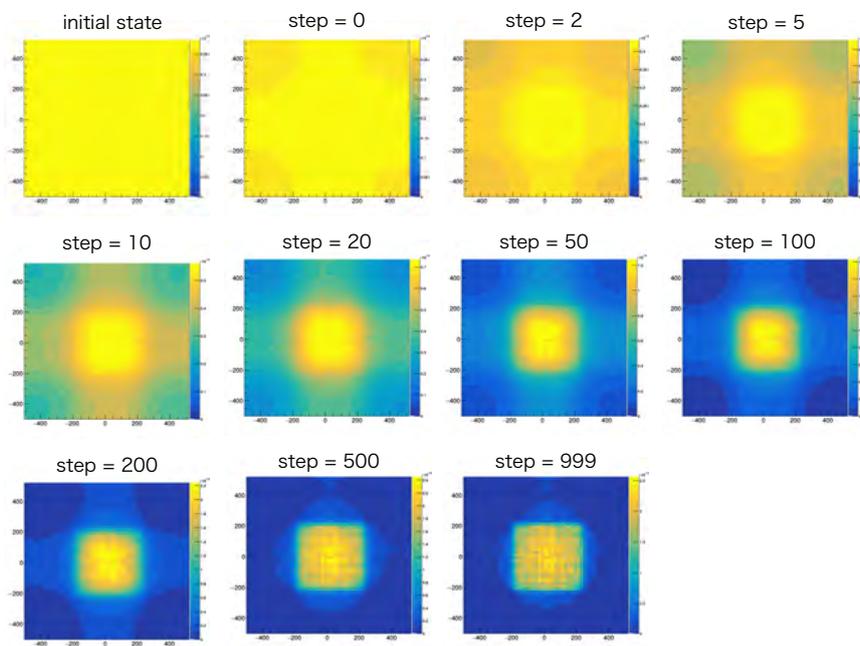
また、二次元面光源シミュレーションデータの再構成各ステップでの  $\tilde{S}$  を図 5.15 に、収束指標  $\Delta\tilde{D}$  の変化を図 5.16 に示す。ステップが進むにつれて  $\Delta\tilde{D}$  が小さくなり、 $\tilde{S}$  が想定される二次元面光源に近づいていく様子がわかる。さらにステップを追加すれば再構成画像はより鮮明になっていくと予想できる。

EM アルゴリズムのデメリットを一つ挙げるとするならば、その計算量の多さである。実際、本節のセットアップでは  $|U| = 51 \times 51 = 2601$ 、 $|V| = 896 \times 896 \times 8 = 6422528$  で

ある。1ステップの E-step, M-step それぞれで  $1.7 \times 10^{10}$  回の四則演算を行なう必要があり、1000ステップ繰り返せば  $3.4 \times 10^{13}$  もの計算量になる。本節の計算環境は表 5.2 に示した通り一般的なコンピュータと比較して潤沢な環境であるが、図 5.14 の画像を得るのに約 3 日の時間を要している。一方で、収束指標 (図 5.16) と画像 (図 5.15) のステップ変化を改めて見てみると、ステップが 500 を超えたあたりから変化が緩やかになっていく様子がわかる。そこで、以降 EM アルゴリズムを利用する際は、計算時間と収束具合のバランスが取れるような境界  $\Delta \tilde{D}_{\text{finish}}$  を設定し、 $\Delta \tilde{D}$  が  $\Delta \tilde{D}_{\text{finish}}$  に到達したら計算を終了するようにする。図 5.16 と図 5.15 から収束指標の変化が鈍くなっている、かつ再構成画像が十分に光源を表現できていると考えられる収束指標としては、 $\Delta \tilde{D}_{\text{finish}} = 1 \times 10^{-8}$  が適当であると考えられる。

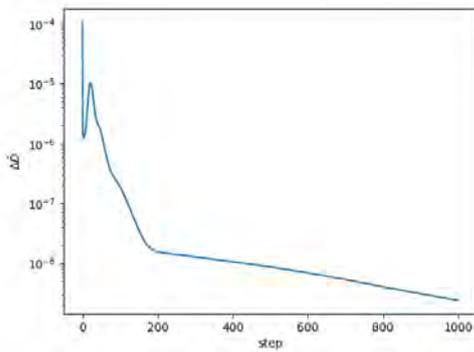


(a) 点光源

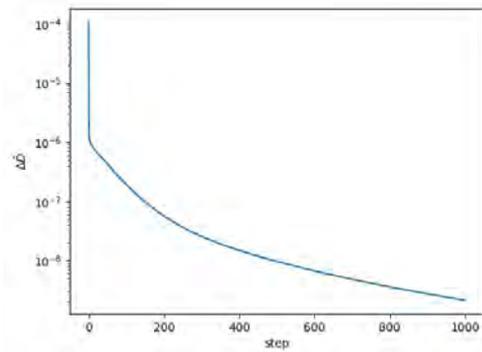


(b) 二次元面光源

図 5.15: シミュレーションデータの EM アルゴリズムによる再構成ステップでの天球面分布  $\tilde{S}$ 。



(a) 点光源



(b) 二次元面光源

図 5.16:  $\Delta\tilde{D}$  のステップ変化

## 5.5 5章のまとめ

撮像時の偽像を減らすため、符号化開口イメージングの画像再構成法として、相関法とEMアルゴリズムを検討した。モンテカルロシミュレーションで生成したデータを再構成し、撮像ノイズを測定した。結果を改めて表5.4にまとめる。相関法は単純なバックプロジェクションであり、偽像の量はパターンの決め方に大きく左右される。本研究ではランダムパターンを複数足し合わせた時に領域ごとの開口率が一定となるような束縛条件を導入し、最適化をしていないパターンの0.10倍程度に偽像が減少する様子を確認した。EMアルゴリズムは最尤推定を逐次処理で繰り返すことで天球面分布を求める手法であり、本研究で符号化開口イメージング用のモデルを新たに作成した。EMアルゴリズムによる画像再構成の結果、最適化をしていないマスクパターンでも偽像を相関法の0.12倍程度に抑えられることが分かった。

表 5.4: 5章で調査したパターンと再構成手法による、二次元面光源の再構成画像と撮像ノイズのまとめ

パターン	手法	画像	撮像ノイズ $\nu$ (式(5.3))
最適化なし	相関法	図 5.5	0.95
最適化あり	相関法	図 5.12	0.097
最適化なし	EM アルゴリズム	図 5.14	0.11

## 第6章 ビームライン実験による偏光撮像実証

撮像偏光計の性能を評価するには、偏光角と偏光度が既知で統計量が十分な平行光の入射が不可欠である。そこで我々は2021年11月に大型放射光施設 SPring-8 内ビームライン BL20B2 にて、偏光撮像実証のためのビーム実験を行なった。この章ではビーム照射実験の詳細と4章, 5章の方法によるデータ解析に関して述べる。また *cipher* でこれまで実現していなかった偏光情報を含めた撮像を実践する。

### 6.1 セットアップ

ビームラインハッチ内の写真を図6.1に示す。照射ビーム、光学系、読み出しデータについて説明する。

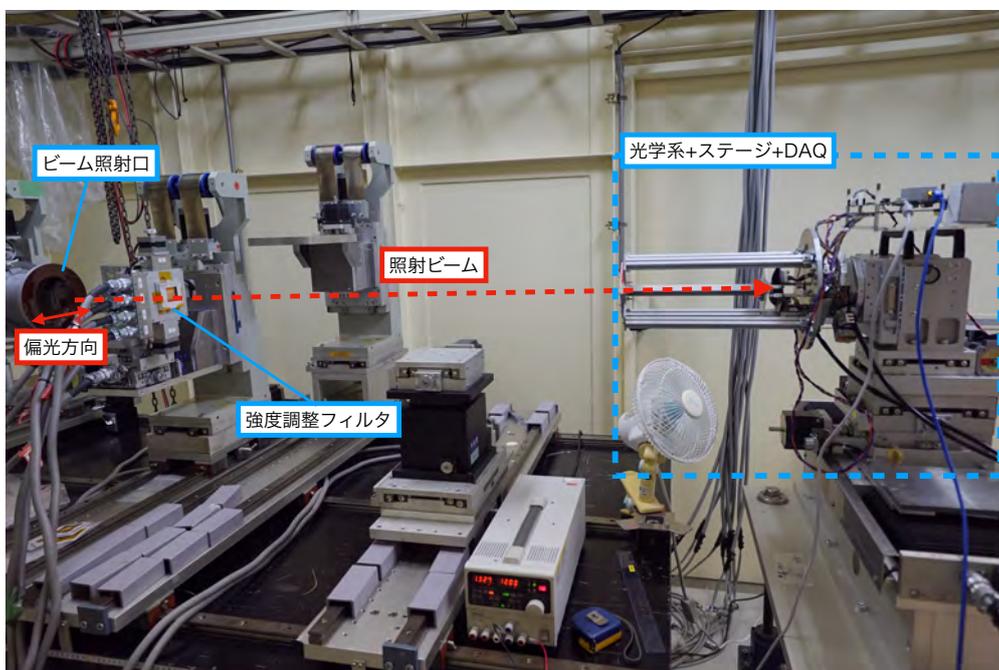


図 6.1: ビームラインハッチ内の写真。

### 6.1.1 照射ビーム

ビームには水平方向にほぼ100%偏光した10-30 keVの単色X線を利用できる。ビーム断面は約10.5 mm角の正方形であり、スリット操作によって各辺を最小約10  $\mu\text{m}$ 程度まで狭めることができる。ビームの強度調整は、厚さ10  $\mu\text{m}$ から640  $\mu\text{m}$ のAlまたはCuのフィルターを組み合わせることで減衰させることで行なった。

### 6.1.2 光学系

光学系の全体像を図6.2に示す。符号化開口マスク、コリメータ、CMOSセンサの一式が回転ステージに固定されている。光学系に向けて照射されたビームは、まず符号化開口マスクを通して符号化される。符号化されたビームは、領域分割用のコリメータを通してからCMOSセンサに当たる。CMOSセンサはLVDSコネクタを通してZDAQボードに接続されており、センサで撮影されたデータはZDAQボードに転送される。

符号化開口マスクはSUS304<sup>1</sup>できており、厚さは0.1 mmである。マスクの要素数、マスク要素間ピッチ、マスク-センサ間距離は5.2節で行なったシミュレーションと同様であり(表5.1)、理論的な角度分解能は29''である。マスクパターンには5.3節で議論した相関法最適パターン(図5.10)になるように設計をしていたが、最終的に特に最適化されていないランダムパターン(図5.4)で実験を行なうことになった。<sup>2</sup>なお、NIST XCOMから計算されるこのマスクの光子透過率は16 keVで2.5%、22 keVで22%である。

光学系を接続するステージには、神津精機の6軸マニピュレータMPS-SSSD-B010<sup>3</sup>を使用した。ステージに専用のコントローラを接続することにより、Ethernetを介したTCP通信で各軸の制御が可能である。各軸の位置はパルスと呼ばれる整数値で指定され、軸と目標パルス値を指定した操作命令をステージコントローラに送ると目標パルス上で止まるまで一定速度で動作するようになっている。ステージの各軸の動作パラメータは表6.1の通りである。

表 6.1: 6 軸ステージ動作パラメータ

軸	最小稼働ピッチ (1パルスあたりの動作量)	最大稼働範囲
$x$	0.5 $\mu\text{m}$	$\pm 5$ mm
$y$	0.5 $\mu\text{m}$	$\pm 5$ mm
$z$	0.5 $\mu\text{m}$	$\pm 5$ mm
$\theta_x$	1.358''	$\pm 3^\circ$
$\theta_y$	1.008''	$\pm 3^\circ$
$\theta_z$	12.96''	$\pm 180^\circ$

<sup>1</sup>FeにCr,Niなどを混ぜたステンレス

<sup>2</sup>最適パターンは製作・納品が間に合わなかった。

<sup>3</sup><https://www.kohzu.co.jp/products/compact6-axismanipulator/mps/>

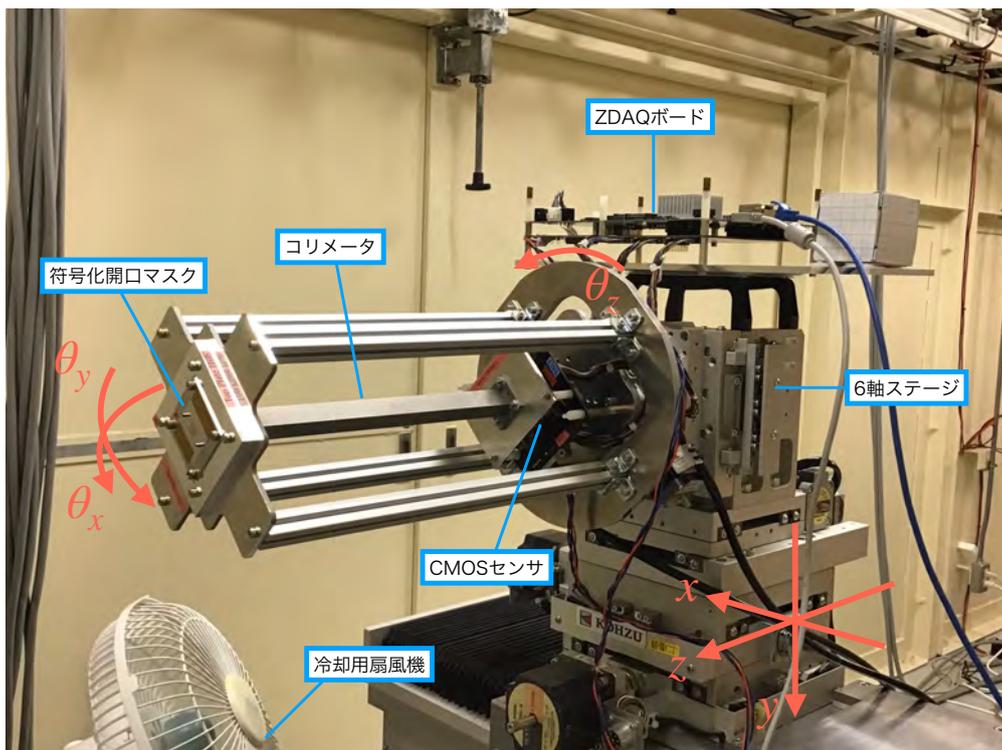


図 6.2: 光学系セットアップ。ステージは  $\theta_z = -45^\circ$  の状態にある。赤字で記した軸はステージ座標系である。

### 6.1.3 データフロー

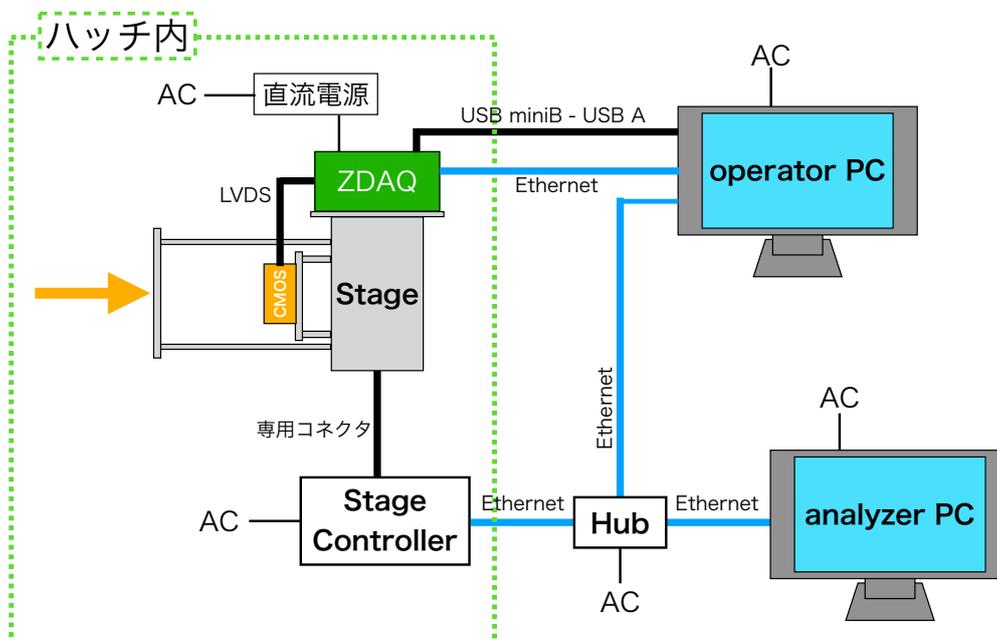


図 6.3: SPring-8 実験の電気通信系統図

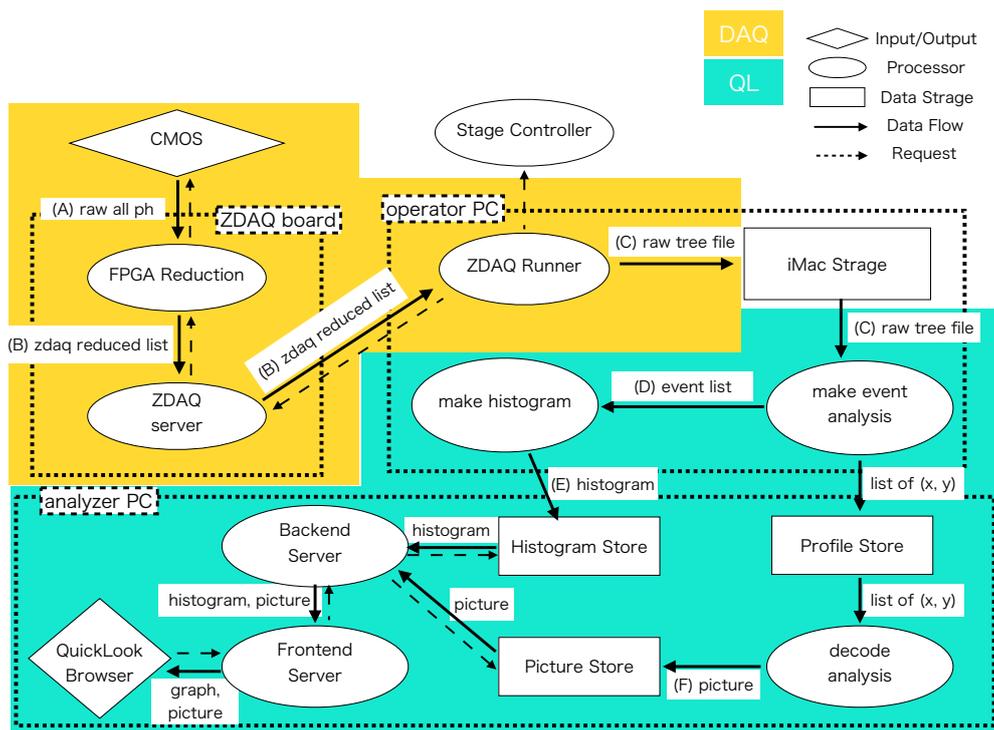


図 6.4: SPring-8 実験データフロー。左上の CMOS センサから左下の QL ブラウザ表示までの流れを示している。

電気・通信システムの全体は図 6.3 に示した。撮影はオペレータ PC と ZDAQ ボードの間で行われる。オペレータ PC が撮影指令を ZDAQ ボードに送り、ZDAQ ボード上のソフトウェアが Ethernet を介して受けた指令を解釈して撮影し、取得データがオペレータ PC へ転送されてストレージ上に保存される。また、ステージ操作はオペレータ PC からステージコントローラへ動作指令を送ることで行われる。データ取得とステージ操作はいずれもオペレータ PC 上からできるように、TCP 通信ベースの Ruby ソフトウェアを作成した。あらかじめ動作計画を設定しておくことで、データ取得とステージ操作を交互に自動で繰り返すことが可能である。

データフローは図 6.4 に示した。大きくデータ読み出し部分と解析部分に分けられる。実験自体はデータ読み出し部分だけで完結するが、取得中のデータに問題がないかをリアルタイムで確認するために作成したクイックルック (QL) 用のソフトウェアを動作させている。図 6.4 内それぞれのデータ形式の詳細は表 6.2 の通りである。

データ取得中のクイックルックソフトウェアの画面を図 6.5 に示す。Vue.js<sup>4</sup>をベースとしたソフトウェアであり、png 画像とヒストグラム形式データの描画に対応している。データ取得と同時に進行する解析の進捗をデータベースから逐一取得することで、画像再構成がうまくいっているか、スペクトルや温度変化に異常がないかなどをリアルタイムでチェックすることが可能である。

<sup>4</sup>Web アプリケーション開発に広く用いられている JavaScript フレームワーク。ドキュメントページ:<https://jp.vuejs.org/index.html>

表 6.2: SPring-8 実験でのデータフロー内のデータ形式のリスト

データ形式名	説明
(A) raw all ph	センサから直接読み出されるデータ。 25M ピクセル分の波高値がピクセル番号順に並んでいる。
(B) zdaq reduced list	FPGA 内データ削減回路を通した後の出力。 式 (4.9) の $\overline{M(C_0)}, \overline{M(C_1)}$ に相当する。ピクセル番号とペDESTAL減算後波高値それぞれのリストが、バイナリ形式で ZDAQ ボードからオペレータ PC へ転送される。
(C) raw tree file	ZDAQ runner が zdaq reduced list を解釈して $p_x, p_y$ , 波高値のリストを作成し、ストレージ内に保存した ROOT ファイル。 ファイルは1フレームごとに作成される。 基本的にこのファイルが生データとして扱われる。
(D) event list (3.4 節)	4.5.2 節の解析に従って生成されたイベントのリスト。 各イベントが $p_x, p_y$ , sum PH, weight, frame ID などの情報を持っている。
(E) histogram	各カラムに関してイベントリストから作成したヒストグラム。 例えば sum PH からヒストグラムを作成するとスペクトルとみなすことができる。
(F) picture	png 形式の画像。 $(p_x, p_y)$ 分布など生データの容量が多い場合、画像に変換された状態でやり取りされる。

### 6.1.4 アライメント

限られた観測時間を有効に利用して精度の良いデータを得るには、アライメントを手際よく正確に行なうことが不可欠である。理想的なセットアップではアライメントは必要とされないが、現実的には数カ所あるネジの固定具合などによって、ビームと光学系で座標系にずれが生じる。この場合、ステージを操作することでまずビーム座標系の原点を探るといった作業が必要になる。今回の実験で位置合わせを必要とするのは  $x$  軸、 $y$  軸、 $\theta_x$  軸、 $\theta_y$  軸の4軸である (図 6.2)。 $x$  軸と  $y$  軸はビームと垂直方向の平行移動であり、ビーム中心をセンサ中心に合わせ、マスクパターンをセンサ内に収めるために必要である。 $\theta_x$  軸と  $\theta_y$  軸は視野内を見回す「首振り」に相当し、視野の正面を探るために必要である。

実際のアライメントの手順を説明する。ステージ操作とセンサでの撮影制御はいずれもオペレータ PC 上のソフトウェアで制御可能なので、ステージを動かし、数フレーム撮影し、ステージを動かし、... といった具合にステージ動作と撮影を交互に素早く行なうことができる。今後、この一連のオペレーションをスキャンと表現し、スキャン中の特定のステージ位置のことをスキャンポイントと表現する。

まず、 $x$  軸については、以下の手順で行なった。 $y$  軸も同様である。

- (1) ビーム幅を  $500 \mu\text{m}$  角程度に絞る。これは到来ビームがマスクパターン間を跨いで

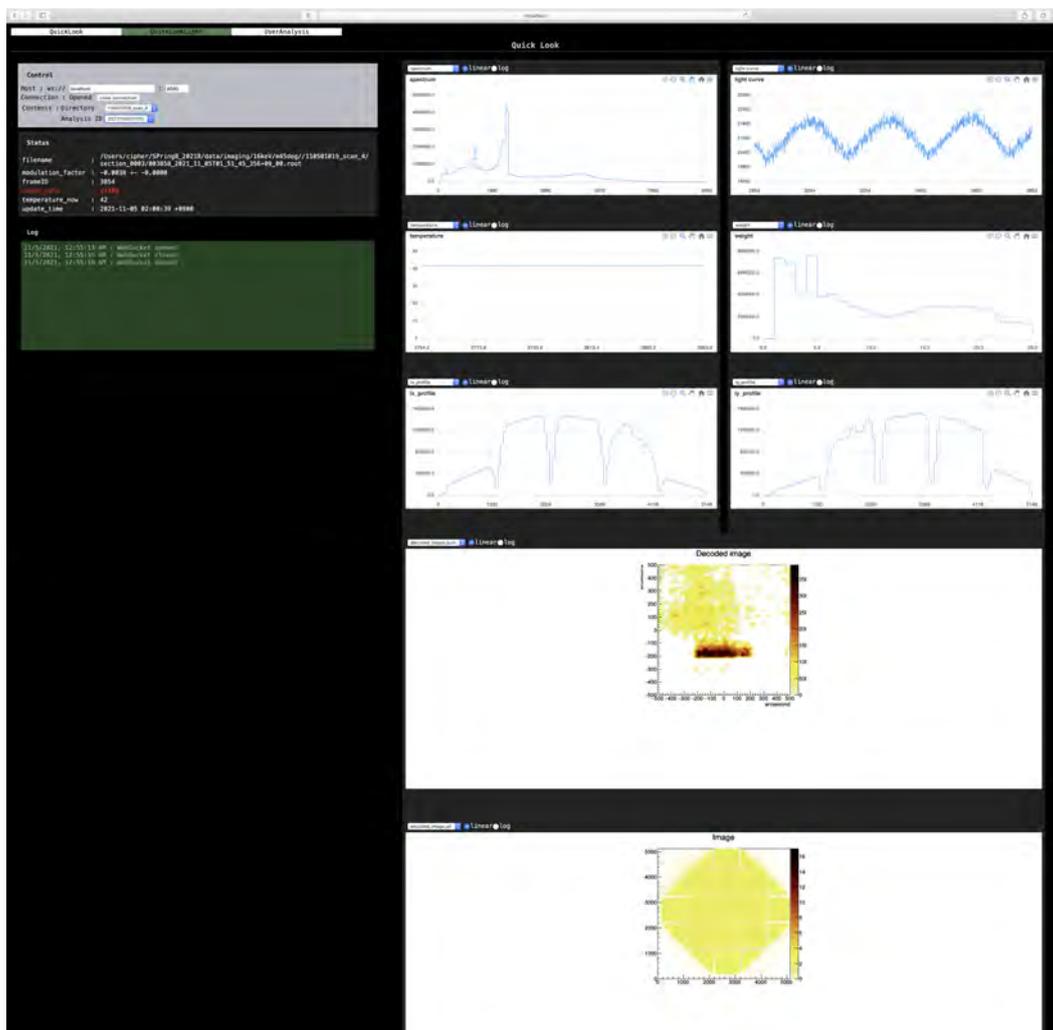


図 6.5: クイックルックソフトの画面。大きく分けて左側がコントロールエリア、右側が描画エリアである。描画エリアで表示しているのは上から順にスペクトル、イベント数時間変化、温度時間変化、weight 分布、検出器面上プロファイル、再構成後画像、再構成前画像である。

重心位置がビーム中心とずれるのを防ぐためである。

- (2) スキャンポイントごとの撮影フレーム数を一致させた上で  $x$  軸方向にスキャンを行なう。
- (3) 撮影データを 4.5.2 小節の通りにイベント解析する。
- (4) イベント解析後、スキャンポイントごとにイベントの  $p_x$  の重心を求める。
- (5)  $p_x$  重心がセンサ中心に最も近いスキャンポイントを、求める原点とする。

また、 $\theta_x$  軸については、以下の手順で行なった。ビーム正面から  $\theta_x$  が傾いているとき、コリメータによって検出イベント数が減少することを利用している。 $\theta_y$  軸も同様である。

- (1) スキャンポイントごとの撮影フレーム数を一致させた上で  $\theta_x$  軸方向にスキャンを行なう。

- (2) 撮影データを 4.5.2 小節の通りにイベント解析する。
- (3) イベント解析後、スキャンポイントごとにイベント数を求める。
- (4) イベント数が最も多いスキャンポイントを、求める原点とする。

例として、 $x$  軸、 $\theta_y$  軸それぞれのスキャン中の各スキャンポイントでの撮影イメージの変化を図 6.6 に示した。なお、平行移動と首振りが完全に独立ではないことと、原点から遠いスキャンポイントは原点特定には不要であることから、実際には  $x$  軸  $y$  軸スキャンと  $\theta_x$  軸  $\theta_y$  軸スキャンを交互に繰り返して、スキャン間隔を徐々に狭めていって精度をあげていくという方法をとった。撮像の角度分解能が  $29''$  であることから、最終的なスキャン間隔は分解能よりも十分細かい  $\Delta\theta_x = 14''$ ,  $\Delta\theta_y = 10''$ ,  $\Delta x = 5 \mu\text{m}$ ,  $\Delta y = 5 \mu\text{m}$  程度とした。

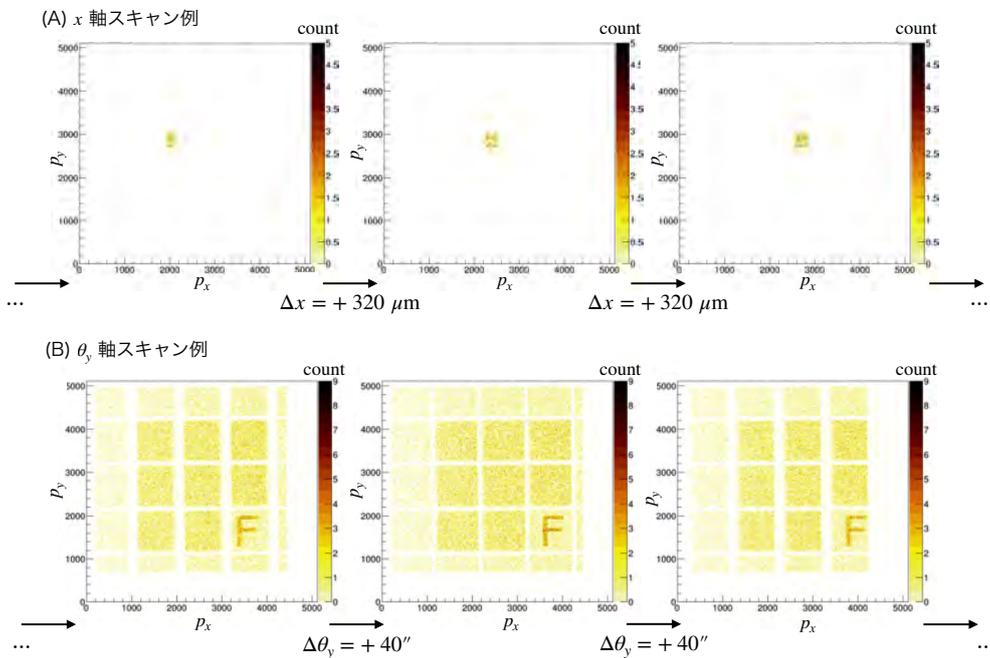


図 6.6: アラインメントスキャン例。(A) $x$ が増加するに従って、イベントイメージの中心の  $x$  座標も増加していく。(B) $\theta_y$ が最適などころではパターンが切れることなく撮像されており、それよりずれるとパターン間の低レート領域がコリメータによって広がっているように見える。

## 6.2 データ取得

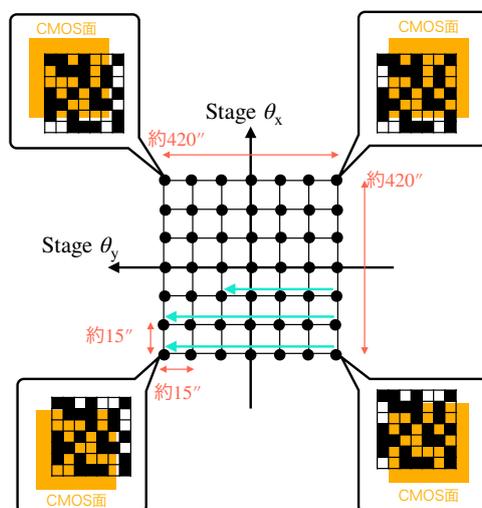


図 6.7: 16 keV 撮像スキャンのイメージ図。ビーム光源から光学系を見下ろす方向で描いている。スキャンポイントは実際には  $29 \times 29$  点である。

### 6.2.1 データセット

取得データのメインは広がった天体を想定した撮像スキャンである。平行光である入射ビームに対し、ステージの  $\theta_x$ ,  $\theta_y$  を変化させることで視野内に広がった光源を模した像を実現させる。具体的なイメージを図 6.7 に示した。アラインメントが完了したときの  $\theta_x$ ,  $\theta_y$  を原点として、 $-210'' \leq \theta_x \leq 210''$ 、 $-210'' \leq \theta_y \leq 210''$  内の範囲で  $\Delta\theta_x = 15''$ 、 $\Delta\theta_y = 15''$  を間隔とした格子点  $29 \times 29$  点のスキャンポイントを設定する。各スキャンポイントに対して、フレームあたり露光時間 600 ms、ゲイン 2.0、データ削減閾値  $\Theta = 20$  ADU、フレーム数 10 のデータ削減モードによる撮影を行なう。前述のデータ取得ソフトウェアを使用してステージ動作と撮影を交互に繰り返すことで、全スキャンポイントの撮影データを取得する。この一連の流れをここでは撮像スキャンと表現する。なお、露光時間とゲインは 3 章と合わせてあり、データ削減閾値  $\Theta$  はデータ容量節約のため十分ノイズの入り込まない値として設定した (図 4.17a を参照)。

表 6.3 に取得したデータセットを示す。まずビームエネルギーを 16 keV に設定し、ステージの  $\theta_z$  を  $0^\circ, 90^\circ, 45^\circ, -45^\circ$  と移動させてそれぞれ撮像スキャンを 4 回ずつ行なった。次にビームエネルギーを 22 keV に設定し、 $\theta_z$  を  $0^\circ, 90^\circ, 45^\circ, -45^\circ$  と移動させて、それぞれスキャンポイントあたり 8 フレームの撮像スキャンを 1 回ずつ行なった。ビームエネルギーが 22 keV の場合は符号化開口マスクの透過率から十分な撮像性能を出せないと考えられるため、16 keV のデータを主なスキャンデータとして扱う。16 keV のスキャンを 4 回に分けて行なっているのは実験の時間配分とオペレーションの手間を総合的に判断してのことであり、実際の解析時にはスキャン 4 回分は足し合わせて一つのデータセットとする。なお、各撮像スキャンの前にグッドピクセル選定 (3.3 節) のために raw フレームモー

ドで 100 フレームの dark 取得を行なった。 $\theta_z$  を変えてデータを取得するのは、重ね合わせで任意の偏光角データを再現するためである (6.6 節で後述)。

また、撮像スキャンとは別にビームエネルギー 16 keV で  $\theta_x, \theta_y$  をそれぞれ約 160' の範囲で 21 点とる 1 次元スキャンを行なった。この粗いスキャンには理論視野限界での撮像を試みる目的と、コリメータの遮蔽性能を評価する目的があるが、本修論では特に扱わない。

表 6.3: SPring-8 実験取得データセット

照射 エネルギー	ステージ 回転角 $\theta_z$	スキャン ポイント あたり フレーム数	スキャン 回数	スキャンポイント
16 keV	0°	10	4	420'' × 420'' 内の 29 × 29 点
16 keV	90°	10	4	420'' × 420'' 内の 29 × 29 点
16 keV	45°	10	4	420'' × 420'' 内の 29 × 29 点
16 keV	-45°	10	4	420'' × 420'' 内の 29 × 29 点
22 keV	0°	8	1	420'' × 420'' 内の 29 × 29 点
22 keV	90°	8	1	420'' × 420'' 内の 29 × 29 点
22 keV	45°	8	1	420'' × 420'' 内の 29 × 29 点
22 keV	-45°	8	1	420'' × 420'' 内の 29 × 29 点
16 keV	0°	10	1	$\theta_x$ の 160'' 内の 21 点
16 keV	0°	10	1	$\theta_y$ の 160'' 内の 21 点

## 6.2.2 時間効率

基本的にビーム照射時はデータ削減モードでデータ取得を行なうため、データ取得の時間効率は 4.5 節の通りになる。しかし本実験は 4.5 節で行なった特性 X 線源を用いたテストとは異なり、検出イベント数がおよそ 200 倍の  $2 \times 10^5$  /frame 程度になるように調整されている。イベント数が多いと転送データ量も多くなるため、読み出しに時間がかかる可能性がある。だが  $\Theta = 20$  とした本実験でのデータ取得速度は線源取得時とほぼ変わらなかった。たとえば照射エネルギー 16 keV の 1 スキャン分は総露光時間が  $0.6 \text{ s/frame} \times (10 \times 29 \times 29) \text{ frame} = 5046 \text{ s} = 84 \text{ min}$  で、実際に  $\theta_z = 0^\circ$  のスキャン一回目でかかった時間は 91 min であった。計算上の限界値は  $84 \times (638 \text{ ms}/600 \text{ ms}) = 89 \text{ min}$  であるから、読み出しのほぼ最高性能を引き出している。最高性能と差異が生じているのは検出データ量が増えた効果や、解析を同時に動作させていることによる operator PC の負担が原因として考えられる。実際のデータ量はフレームあたり平均で  $7.6 \times 10^2 \text{ kB}$  であり、4.5 節の結果 (図 4.17b) で外部転送がボトルネックになる境目付近であることが分かる。

## 6.3 検出器性能評価

### 6.3.1 イベント解析

まず、取得データを4.5.2節のイベント解析フローに従って解析する。ここでイベント閾値とスプリット閾値はそれぞれ、 $E_{\text{event}} = 100$  ADU,  $E_{\text{split}} = 20.1$  ADUとした。 $E_{\text{split}}$ は先行研究(畠内, 2021)の8.5 ADUよりも高い値に設定している。一般に $E_{\text{split}}$ が高くなると偏光解析に必要なdoubleイベントやさらに広がったイベントの統計量が減少する恐れがあるが、本実験は読み出しの高速化とアラインメント効率化の恩恵を受け、先行研究の約4倍の統計量を確保できていることと、本実験は撮像性能を確かめる目的が大きいいため、スプリット閾値を高くしても結果に大きな影響はないと判断した。 $E_{\text{split}} = 20.1$  ADUとしたのはデータ削減モードでの設定閾値が $\Theta = 20$  ADUであり、それよりも高い中なるべく低くしたためである。

照射エネルギー16 keV, 22 keVの $\theta_x = 0^\circ$ での1スキャン分データのスペクトルを図6.8に、weight分布を図6.9に示す。スペクトルは照射エネルギー16 keVでは2540 ADU付近に、照射エネルギー22 keVでは3500 ADU付近にピークを持っており、それ以上のエネルギー帯にカットオフ構造を持っている。またweight分布を比較すると、照射エネルギー22 keVの方がweightの高いイベントの割合が多くなっている。これは入射光子が高エネルギーであるとき、空乏層内の電子雲がより広範囲に広がる効果だと考えられる。

また、weight2のイベントを用いて、スペクトルのビーム成分をガウシアン(式(3.3))でフィッティングした結果を表6.4に示す。エネルギーを校正して(図3.11)算出したFWHMも示している。誤差が大きいですが、3.4節で示した線源測定による分解能(表3.6)と比較して、大きな違いはないことが分かる。

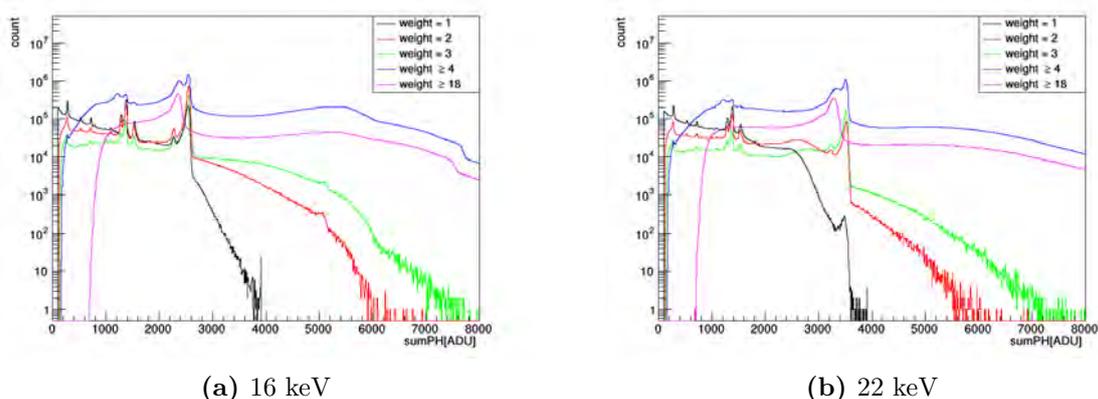


図 6.8:  $\theta_x = 0^\circ$ での1スキャン分のスペクトル

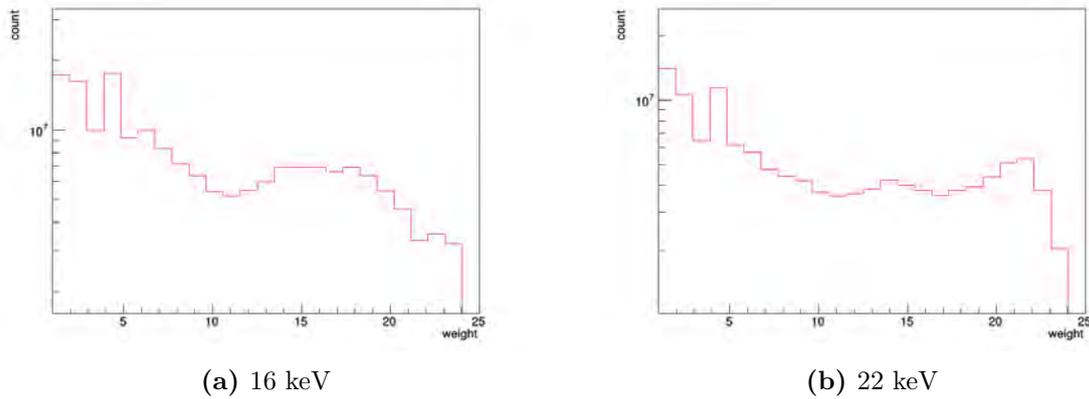


図 6.9:  $\theta_x = 0^\circ$  での 1 スキャン分の weight 分布

表 6.4: double イベントスペクトルのガウシアンフィッティング結果とエネルギー分解能

照射 エネルギー	$C^*$	$E_0$ [ADU]*	$\sigma$ [ADU]*	FWHM * [eV]
16 keV	$(6917 \pm 5) \times 10^2$	$2548.44 \pm 0.01$	$25.50 \pm 0.01$	$(3.0 \pm 0.4) \times 10^2$
22 keV	$(786 \pm 1) \times 10^2$	$3508.88 \pm 0.07$	$36.8 \pm 0.1$	$(4.7 \pm 0.4) \times 10^2$

\* 誤差は統計誤差に由来する  $1\sigma$  範囲である。

## 6.4 偏光応答評価

### 6.4.1 モジュレーションファクターの算出

本修論での偏光解析には、光電子放出角の推定が容易な double イベント、特に H-type イベントと V-type イベントのみを用いるものとする (図 3.7)。このとき、検出器の偏光応答性能を示すモジュレーションファクターを以下の手順で算出する。この手法は先行研究 (島内, 2021; 會澤, 2020) に基づいている。

#### (1) H-type イベントと V-type イベントの抽出

各ステージ回転角  $\theta_z$  について、エネルギーをビーム由来と考えられるものに絞ったのち、イベント形状から H-type, V-type のイベント数  $n_H(\theta_z), n_V(\theta_z)$  をそれぞれカウントする。エネルギーは double イベントのスペクトルをガウシアンでフィッティングしたのち、ピークから  $\pm 2\sigma$  の範囲を抽出範囲とした。

#### (2) 検出器の光電子放出角異方性の補正

理想的には  $n_H(\theta_z), n_V(\theta_z)$  を偏光応答データとみなせるが、実際には無偏光でも検出器の特性により H-type イベント、V-type イベントの出やすさに偏りが生じてしま

うため、補正をする必要がある。そこで、無偏光に対する H-type イベント、V-type イベントのイベント数  $n_H, n_V$  に対して、 $f_H, f_V$  を、

$$f_H = \frac{n_H}{n_H + n_V} \quad (6.1)$$

$$f_V = \frac{n_V}{n_H + n_V} \quad (6.2)$$

で求め、

$$C_H = \frac{f_H + f_V}{2f_H} \quad (6.3)$$

$$C_V = \frac{f_H + f_V}{2f_V} \quad (6.4)$$

を補正係数とする。 $f_H, f_V$  は「無偏光で単位放出角あたりに検出される相対的なイベント量」と考えることができ、その逆数を定数倍したものが  $C_H, C_V$  である。本研究では擬似的な  $f_H, f_V$  を、 $\theta_z = 0^\circ, 90^\circ, 45^\circ, -45^\circ$  それぞれの H-type イベント、V-type イベントの割合を足し合わせることで求めた。式で表すと、

$$f_H = \sum_{\theta_z} \frac{n_H(\theta_z)}{n_H(\theta_z) + n_V(\theta_z)} \quad (6.5)$$

$$f_V = \sum_{\theta_z} \frac{n_V(\theta_z)}{n_H(\theta_z) + n_V(\theta_z)} \quad (6.6)$$

である。 $C_H, C_V$  を用いて、各  $\theta_z$  について補正をかけた値

$$n'_H(\theta_z) = C_H n_H(\theta_z) \quad (6.7)$$

$$n'_V(\theta_z) = C_V n_V(\theta_z) \quad (6.8)$$

$$(6.9)$$

を補正済みのイベント数として、以降の偏光情報として扱う。

### (3) 三角関数によるフィッティング

各ステージ回転角  $\theta_z$  について、H-type イベント数の割合  $t(\theta_z)$  を以下のように求める。

$$t(\theta_z) = \frac{n'_H(\theta_z)}{n'_H(\theta_z) + n'_V(\theta_z)} \quad (6.10)$$

このとき、検出データが  $\theta_z$  によらない無偏光のデータと  $\theta_z$  に対して正弦波をなす偏光データの和だと考え、データセット  $(\theta_z, t(\theta_z))$  を、 $A, B, C$  をパラメータとした三角関数

$$P(\theta) = A \cos(2(\theta - B)) + C \quad (6.11)$$

でフィッティングする。このフィッティング結果パラメータから、100%偏光に対するモジュレーションファクター (MF) は照射ビームの偏光度を  $f$  として

$$\text{MF} = \frac{1}{f} \frac{A}{C} \quad (6.12)$$

で算出できる。MF が大きいほど検出偏光成分の割合が多く、検出器の偏光応答性は高いことになる。

照射エネルギー 16 keV, 22 keV それぞれの場合について、スキャンデータを用いて上記の手順で偏光解析を行なった。図 6.10 にそれぞれのモジュレーションカーブを示す。また、フィッティングのパラメータとモジュレーションファクターを表 6.5 に示す。

先行研究(畠内, 2021)では、100%偏光光に対して GMAX0505RF が 16 keV で  $0.108 \pm 0.002$ 、20 keV で  $0.158 \pm 0.003$  の MF を出すと報告されている。先行研究の結果は表 6.5 の結果と概ね一致する。先行研究との MF の差異には、 $\theta_z$  を 17 点取得した先行研究とは異なり、本実験でとった  $\theta_z$  は 4 点のみであること、解析方法が異なることなどが原因として考えられる。

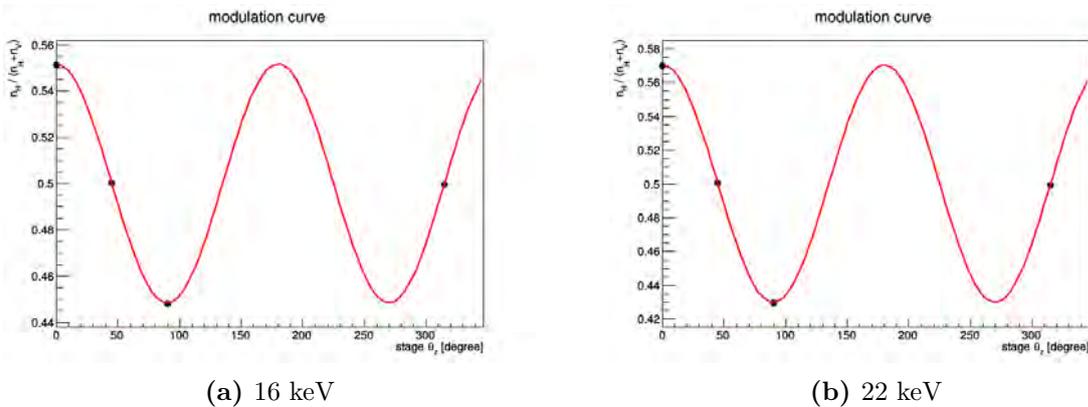


図 6.10: モジュレーションカーブ。H-type, V-type イベントのイベント数の比をもとに描いている。黒点は実験値を示しており、統計誤差は各点の大きさより十分小さい。

表 6.5: モジュレーションカーブのフィッティング結果と 100%偏光に対するモジュレーションファクター

照射 エネルギー	A	B [°]	C	MF
16 keV	$(5.164 \pm 0.008) \times 10^{-2}$	$0.21 \pm 0.05$	$0.49995 \pm 0.00006$	$0.1033 \pm 0.0002$
22 keV	$(7.02 \pm 0.04) \times 10^{-2}$	$0.2 \pm 0.2$	$0.5000 \pm 0.0003$	$0.1404 \pm 0.0008$

## 6.5 撮像性能評価

まず、偏光を考えず撮像そのものが実現できているか確かめる。なお、以降の天球面座標  $(\theta_x, \theta_y)$  はステージ座標の  $(\theta_x, \theta_y)$  とは異なり、検出器の  $(p_x, p_y)$  と対応した座標系であることに注意する(図 6.2, 6.7 も参照)。

### 6.5.1 データ抽出手順

まず、スキャンのイベントリストから撮像に必要なデータを抽出する。今回は偏光を含めた撮像を行なうことを見据え、以下の手順で符号化済みの検出データを作成した。

(1) イベントセレクション

ビーム由来と考えられる中で、偏光解析に使用できるイベントを選ぶ。具体的にはエネルギーとイベント形状でセレクションをかける。エネルギーはdouble イベントの全吸収ピークをガウシアンでフィッティングしたのち、ピークから  $\pm 2\sigma$  の範囲を抽出範囲とした。その中で H-type イベントと V-type イベントを取り出し、再構成対象のイベントリストとする。

(2) 検出器領域の切り分け

検出器上に8パターン分の影が検出されているが、撮像のためには各パターンの原点  $(g_x, g_y)$  を決定する必要がある。これにはコリメータによる影の仕切りを用いる。各パターンについておおよその原点範囲  $g_{x\min} \leq g'_x \leq g_{x\max}, g_{y\min} \leq g'_y \leq g_{y\max}$  を定めておき、正面撮像データに対してイベント数が最も多くなるような原点の決め方を選択する。式で表すと、

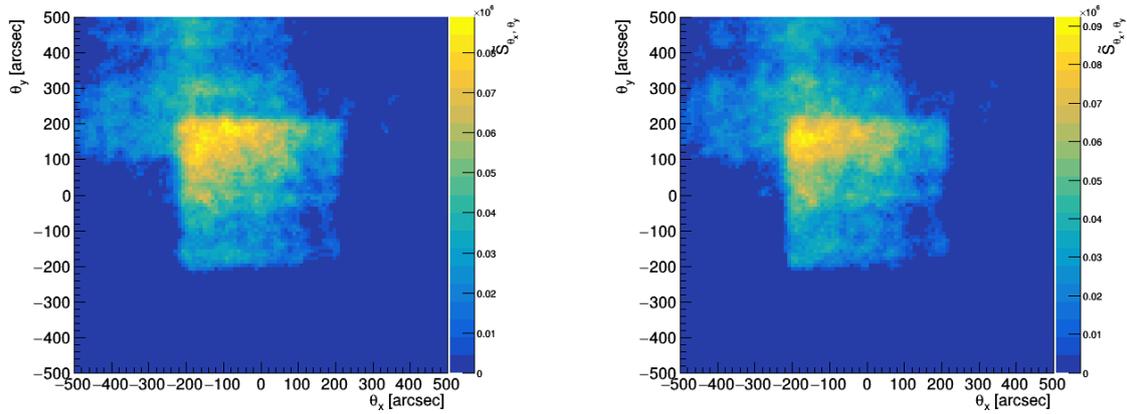
$$(g_x, g_y) = \underset{\substack{g_{x\min} \leq g'_x \leq g_{x\max}, \\ g_{y\min} \leq g'_y \leq g_{y\max}}}{\operatorname{argmax}} \left[ \sum_{p_x=0}^{N_{px}-1} \sum_{p_y=0}^{N_{py}-1} D_{g'_x+p_x, g'_y+p_y} \right] \quad (6.13)$$

となる。

### 6.5.2 相関法によるスキャンデータ再構成

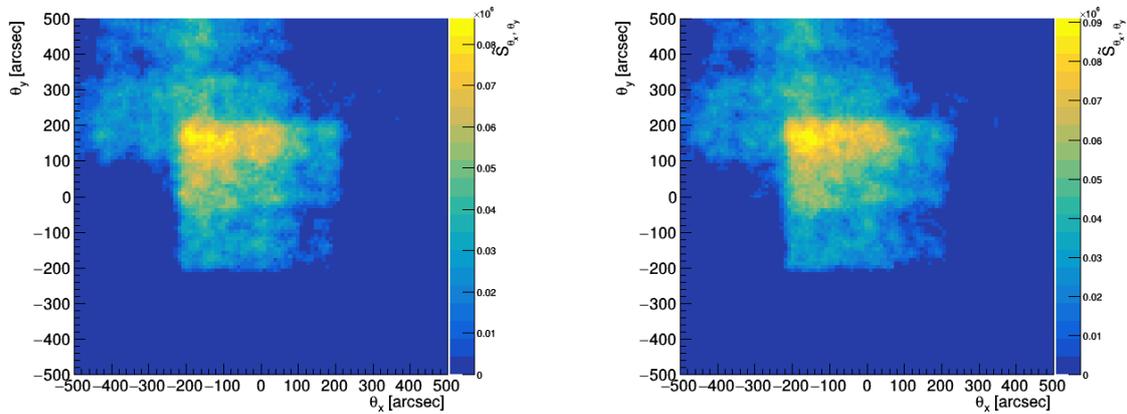
相関法による再構成を試みる。天球面座標は  $-500'' \leq \theta_x \leq 500''$ ,  $-500'' \leq \theta_y \leq 500''$  の範囲で  $10''$  ピッチでとる。この設定で、16 keV のスキャンデータの相関法による再構成画像は図 6.11 のようになった。また、それぞれの撮像ノイズ (式 5.3) を表 6.6 に示している。

シミュレーションデータの結果 (図 5.5) と同様に、パターン由来のノイズが出てしまっている様子が分かる。この状態では領域ごとに強度が異なることにもなってしまうため、次節で行なう偏光を含めた撮像に影響が出てしまう。なおシミュレーションでの結果  $\nu = 0.95$  よりもノイズが低く抑えられているのは、光子統計量が実験の方が多いたことが原因として考えられる。



(a)  $\theta_z = 0^\circ$

(b)  $\theta_z = 90^\circ$



(c)  $\theta_z = 45^\circ$

(d)  $\theta_z = -45^\circ$

図 6.11: 16 keV スキャンデータの相関法による再構成画像。見やすさのため、バックプロジェクションの結果が負であるところは0に切り上げてある。

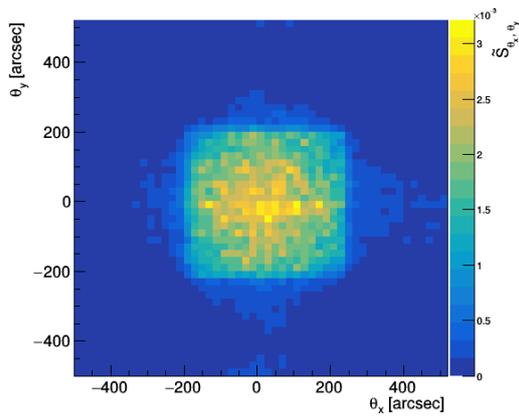
### 6.5.3 EM アルゴリズムによるスキャンデータ再構成

同じデータを、EM アルゴリズムによって再構成する。再構成のやり方は 5.4 節で行なったものと同様であり、天球面座標は  $-500'' \leq \theta_x \leq 500''$ ,  $-500'' \leq \theta_y \leq 500''$  の範囲で  $20''$  ピッチでとり、ステップは  $\Delta \tilde{D} \leq 10^{-8}$  に達したところで終了する。16 keV のスキャンデータの EM アルゴリズムによる再構成画像は図 6.12 のようになった。また、撮像ノイズ (式 5.3)  $\nu$  と終了ステップを表 6.6 に示している。

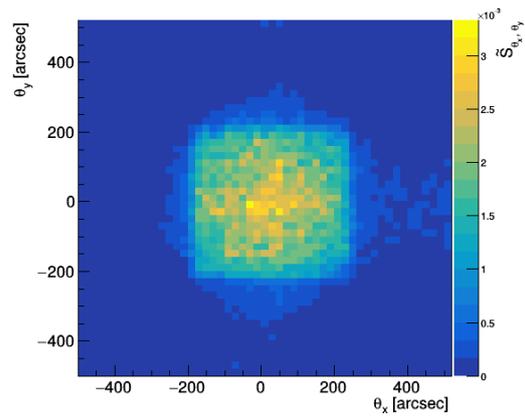
相関法による再構成と比較して、16 keV ではうまく再構成画像が得られている。実際に  $\nu$  は相関法の  $1/7$  程度に抑えられている。このことから、以降の解析では EM アルゴリズムを再構成手法として採用する。

また、22 keV のスキャンデータも再構成を行うが、前述の通り本実験の符号化開口マスクの構造では、計算上 22 keV の光子は 22 % 透過してしまう。そこで、式 (5.27) にお

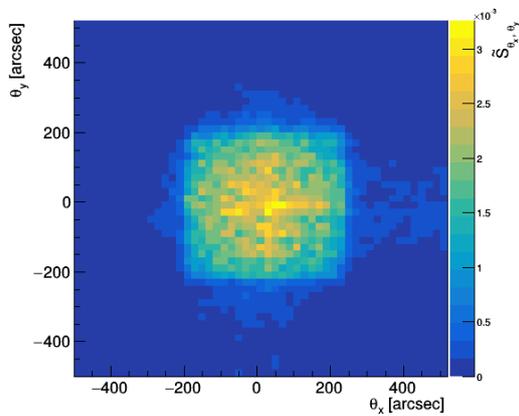
いて  $\tau = 0.22$  として再構成を行なう。このように変更した EM アルゴリズムで再構成して得た 22 keV の  $\theta_z = 0^\circ$  スキャンの画像を図 6.13 に示す。比較として、 $\tau = 0$  として再構成した画像も示している。点光源 (図 6.13a) は再構成できてはいるが、偽像が少し目立つ。またスキャン画像 (図 6.13b) は薄く輪郭が見えてはいるが、16 keV のときと比較すると想定光源内にまばらに光が散らばっている。 $\tau = 0$  とした再構成画像 (図 6.13c) では光源の形状が想定の子正方形と大きく異なっており、 $\tau$  を光子透過率で設定することの有意性が見られる。



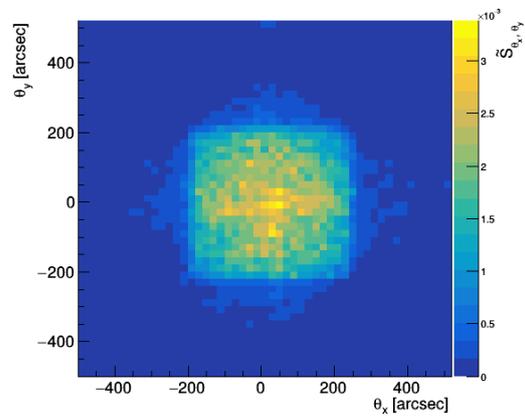
(a)  $\theta_z = 0^\circ$



(b)  $\theta_z = 90^\circ$



(c)  $\theta_z = 45^\circ$

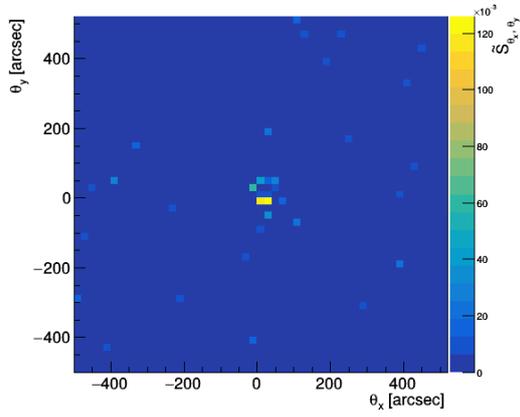


(d)  $\theta_z = -45^\circ$

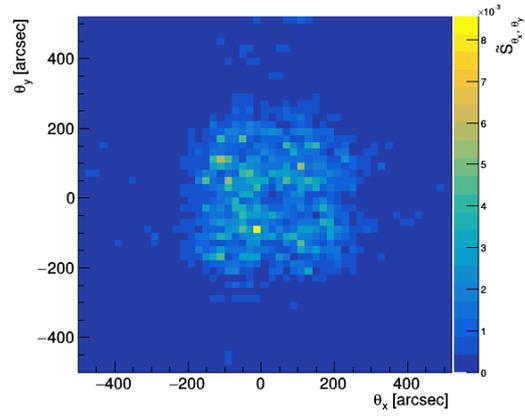
図 6.12: 16 keV スキャンの EM アルゴリズムによる再構成画像。  $\Delta \tilde{D} < 10^{-8}$  となったステップでの画像を示している。

表 6.6: 16 keV スキャンデータの再構成の終了ステップとノイズ

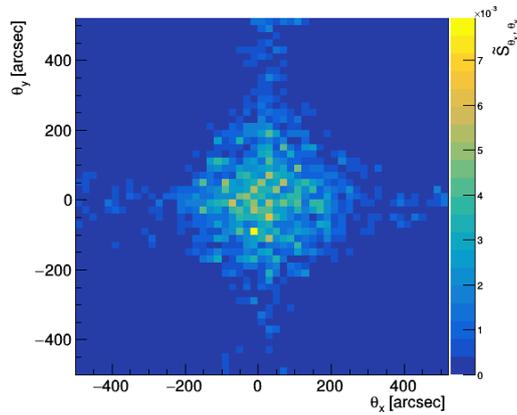
再構成法	$\theta_z$	終了ステップ	$\nu$
相関法	$0^\circ$		0.70
相関法	$90^\circ$		0.73
相関法	$45^\circ$		0.73
相関法	$-45^\circ$		0.69
EM	$0^\circ$	501	0.099
EM	$90^\circ$	498	0.099
EM	$45^\circ$	501	0.10
EM	$-45^\circ$	496	0.098



(a)  $\theta_z = 0^\circ$ , 点光源,  $\tau = 0.22$



(b)  $\theta_z = 0^\circ$ , 二次元面光源,  $\tau = 0.22$



(c)  $\theta_z = 0^\circ$ , 二次元面光源,  $\tau = 0$

図 6.13: 22 keV スキャンの EM アルゴリズムによる再構成画像。  $\Delta\hat{D} < 10^{-8}$  となったステップでの画像を示している。

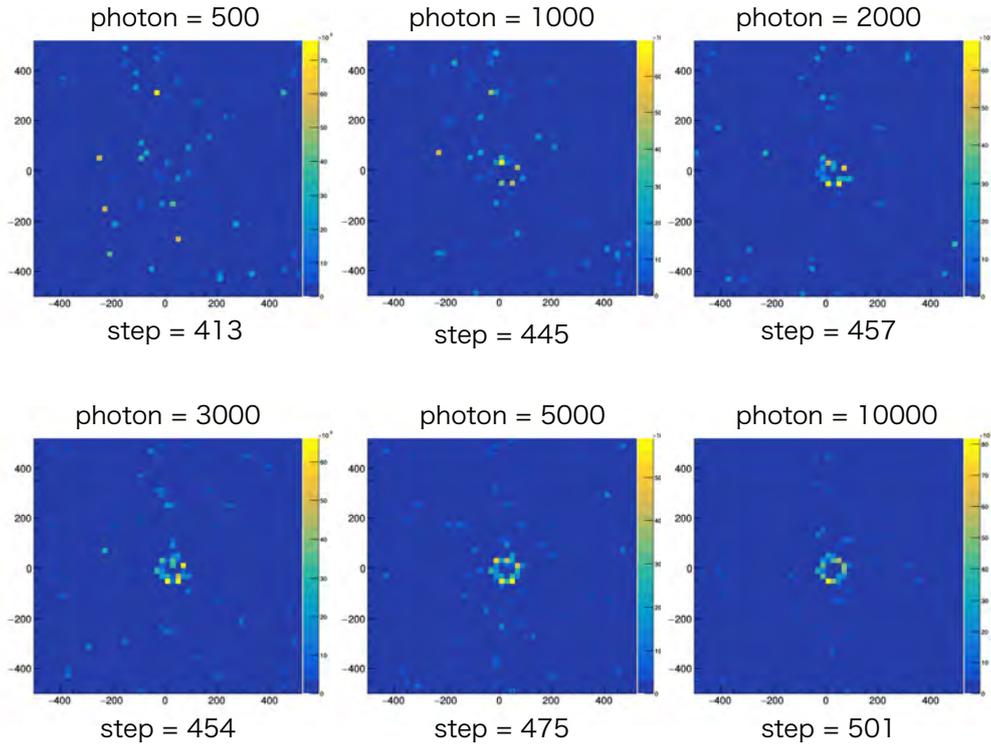


図 6.14: 円環の EM アルゴリズムによる再構成画像。 $\Delta\tilde{D} \leq 10^{-8}$  となるステップでの画像を示している。

#### 6.5.4 光子統計量が少ない場合の円環再構成

光子統計量が少ないときに再構成が可能か調査する。Crab(図 1.1) の観測を意識して、半径  $45''$  の円環領域の光源を再構成する。具体的には、 $\theta_z = 0^\circ$  スキャンのうち、スキャンポイントが

$$U_{\text{Signal}} = \left\{ (\theta_x, \theta_y) \mid 30'' \leq \sqrt{\theta_x^2 + \theta_y^2} \leq 60'' \right\} \quad (6.14)$$

の領域にあるものをピックアップする<sup>5</sup>。円環領域内スキャンポイントのイベントで 6.5.1 小節のセレクションを通過した中から、ランダムにイベントを選んで抽出し、光子統計の少ないデータセットを作成した。抽出するイベント数を変えてそれぞれ画像再構成を行い、画像再構成に必要な光子統計量を調査した。

各イベント数での EM アルゴリズムによる円環再構成画像は図 6.14 のように、再構成画像ノイズ  $\nu$ (式 5.3) は図 6.15 のようになった。ただし、ノイズ算出時の光源領域とバックグラウンド領域は、

$$\begin{aligned} U_s &= \{(\theta_x, \theta_y) \mid 10'' \leq \sqrt{\theta_x^2 + \theta_y^2} \leq 80''\} \\ U_b &= \{(\theta_x, \theta_y) \mid 130'' \leq \sqrt{\theta_x^2 + \theta_y^2}\} \end{aligned} \quad (6.15)$$

としている。 $U_s$  に実際の光源領域  $U_{\text{Signal}}$  と差異があるのは、角度分解能に由来する広がりやアライメントによる光源原点のずれを十分含むようにしたためである。

<sup>5</sup>正確には三次元極座標の  $\theta$  で判断するべきだが、今は微小角なので単純に  $\theta_x, \theta_y$  平面の原点からの距離としてピックアップしている。

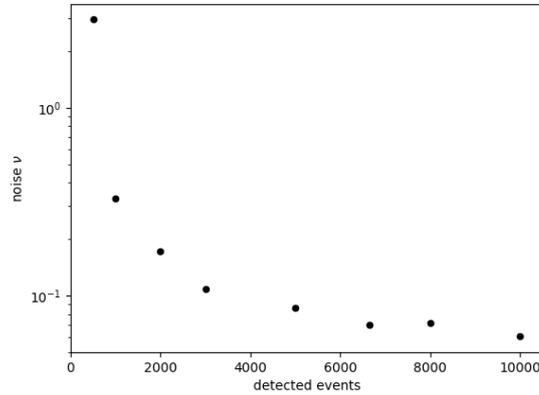


図 6.15: 再構成ノイズの光子統計量による依存性

全体として概ね円環再構成に成功してはいるが、イベント数が 3000 を下回ると偽像が多くなって円環が崩れてしまい、 $\nu$  も急激に大きくなる。図 6.15 と図 6.14 から、再構成可能な光子統計数の最低ラインは 3000 程度だと考えることが出来る。なお、光源の占める視野面積が広いほど必要光子統計量は多くなるため、実際にはこの数倍程度のイベント数が必要になると考えられる。また、6.6 節で後述する偏光を含めた撮像にはさらに数倍程度のイベント数が必要だと考えられる。この結果については 6.7 節で再び議論する。

## 6.6 偏光撮像性能評価

本節では得られたデータを使って、*cipher* の強みである偏光情報を含めた画像再構成を実証する。現段階では偏光撮像を実現できるかを検証するのが重要と考え、光子統計量などはここでは考慮しない。

### 6.6.1 重ね合わせによる偏光の表現

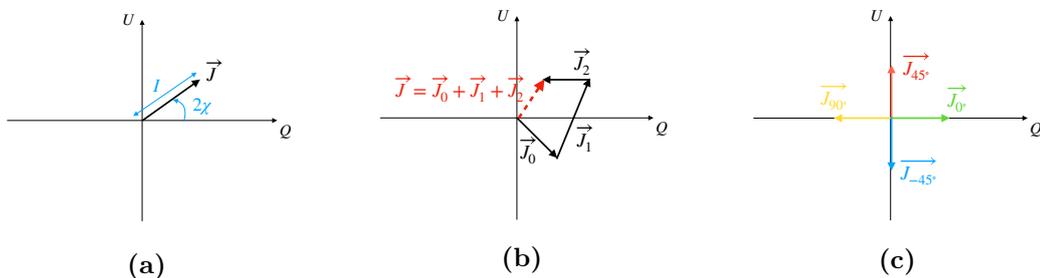


図 6.16:  $Q$ - $U$  空間での偏光の重ね合わせ

撮像に偏光の情報を組み込んでいく前に、取得データセットが 2.1 節のストークスパラメータとどう対応するか確認する。まず、本実験では直線偏光のみを考えているので

$\beta = 0$ と考えるが良い。このとき、偏光角  $\chi$  に対して、

$$Q = I_{\text{pol}} \cos 2\chi \quad (6.16)$$

$$U = I_{\text{pol}} \sin 2\chi \quad (6.17)$$

である。直線偏光の重ね合わせは、図 6.16 のように  $Q, U$  を座標軸とする 2 次元平面を考えるとわかりやすい。光子 1 つは 100% 偏光していると考え、100% 偏光光を要素光とした重ね合わせを考える。強度  $I$ 、偏光角  $\chi$  の 100% 偏光光は  $Q-U$  空間では動径  $I$ 、偏角  $2\chi$  の極座標で表されるベクトル  $\vec{J}$  である (図 6.16a)。複数の要素光が重ね合わさったとき、重ね合わせ光はそれぞれの要素光のベクトルの和になる (図 6.16b)。式で表すと、 $K$  種の 100% 偏光光  $\vec{J}_0, \vec{J}_1, \dots, \vec{J}_{K-1}$  を重ね合わせた光の偏光ベクトル  $\vec{J}$  は、

$$\vec{J} = \sum_{k=0}^{K-1} \vec{J}_k \quad (6.18)$$

で表され、重ね合わせ光の全体の強度  $I$  は

$$I = \sum_{k=0}^{K-1} |\vec{J}_k| \quad (6.19)$$

と全要素光の強度の単純な足し合わせとなる。ここから重ね合わせ光の偏光度  $\Pi$  は

$$\Pi = \frac{|\vec{J}|}{I} = \frac{|\sum_{k=0}^{K-1} \vec{J}_k|}{\sum_{k=0}^{K-1} |\vec{J}_k|} \quad (6.20)$$

で得ることができる。特に  $K$  種の光がそれぞれ異なる偏光角  $\phi$  からなる強度  $I_\phi$  の 100% 偏光光の場合、重ね合わせ光の  $I, Q, U$  をあらわに書くと、

$$I = \sum_{\phi} I_\phi, \quad Q = \sum_{\phi} I_\phi \cos 2\phi, \quad U = \sum_{\phi} I_\phi \sin 2\phi \quad (6.21)$$

である。ここから重ね合わせ光の偏光度  $\Pi$  と偏光角  $\chi$  は

$$\Pi = \frac{\sqrt{Q^2 + U^2}}{I}, \quad \chi = \frac{1}{2} \arctan \frac{U}{Q} \quad (6.22)$$

で計算できる。さらに特別な場合として、光源として  $\phi = 0^\circ$  と  $\phi = 90^\circ$  の光のみを考慮する場合、 $I, Q, U$  は

$$I = I_{0^\circ} + I_{90^\circ}, \quad Q = I_{0^\circ} - I_{90^\circ}, \quad U = 0 \quad (6.23)$$

となり、 $\Pi$  と  $\chi$  は

$$\Pi = \frac{|I_{0^\circ} - I_{90^\circ}|}{I_{0^\circ} + I_{90^\circ}} \quad (6.24)$$

$$\chi = \begin{cases} 0 & (I_{0^\circ} - I_{90^\circ} > 0) \\ \frac{\pi}{2} & (I_{0^\circ} - I_{90^\circ} < 0) \end{cases} \quad (6.25)$$

で表される。

本実験で取得したのは  $\chi = 0^\circ, 90^\circ, 45^\circ, -45^\circ$  のデータであり、これらを  $Q-U$  空間上で表現すると図 6.16c のように  $Q, U$  それぞれの座標軸に沿ったベクトルになる。これらを適切な量ずつ足し合わせることで、 $Q-U$  空間上の領域を強度の許す範囲で全て表現することができる。たとえば  $\chi = 30^\circ$ , 強度  $I$  のデータを表現するなら、 $\chi = 0^\circ$  のデータを  $\frac{1}{2}I$  の量だけ、 $\chi = 45^\circ$  のデータを  $\frac{\sqrt{3}}{2}I$  の量だけ抽出し、足し合わせればよい。この場合の偏光度は  $\frac{I}{\frac{1}{2}I + \frac{\sqrt{3}}{2}I} = \sqrt{3} - 1$  となる。

## 6.6.2 複数偏光源データの作成

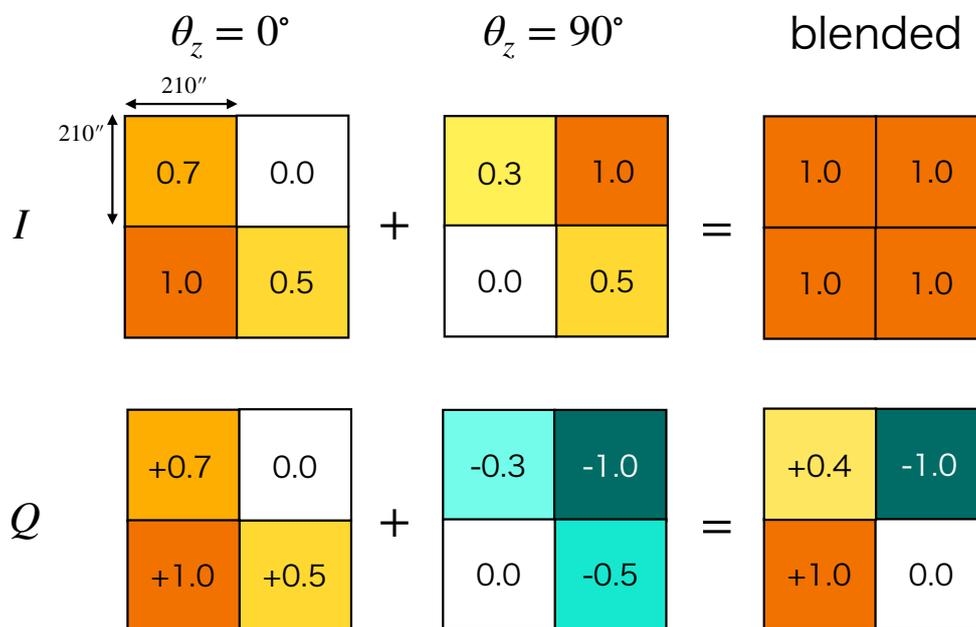


図 6.17: 混ぜ合わせによる複数偏光源データ作成の様子。値は相対値である。

天球面内で偏光度の違う領域を切り分けて撮像できるかを検証するため、 $\theta_z = 0^\circ$  のデータと  $\theta_z = 90^\circ$  のデータを混ぜ合わせる。二次元面光源を 4 つの領域に分割し、表 6.7 のように各領域での光量を割り当てる。分割はデータ取得時のステージ座標をもとに行い、光量の調整は各スキャンポイントでのフレーム数 (16 keV スキャンでは 10) を制限することで行なう。たとえば、 $\theta_x > 0$ ,  $\theta_y < 0$  では  $\theta_z = 0^\circ$  のデータは各スキャンポイントでフレーム数が 5 となるように抽出する。本来このとき入射ビーム強度の違いを考慮してイベント数を調整する必要があるが、データの加工を最小限に抑えるためにビーム強度補正は再構成時に考慮することにする。

この混ぜ合わせを  $I, Q$  の加算として捉えると図 6.17 のようになる。 $I$  の合計としては一様な二次元面光源であるが、領域によって  $Q$  が異なる様子が見える。データの混ぜ合わせには結果的に検出データを利用することになるが、実際に行っているのは「光源の足し合わせ」であることに注意する。 $Q$  の違いを検出できるかが偏光撮像を含めた画像再構成の課題の一つである。

表 6.7: 混ぜ合わせの光量配分

領域	$I_{0^\circ}^*$	$I_{90^\circ}^*$
$\theta_x < 0, \theta_y < 0$	1.0	0.0
$\theta_x > 0, \theta_y > 0$	0.0	1.0
$\theta_x > 0, \theta_y < 0$	0.5	0.5
$\theta_x < 0, \theta_y > 0$	0.7	0.3

\* 光量は相対値である。

### 6.6.3 EM アルゴリズムの偏光撮像への拡張

偏光を含めた撮像を行なうため、位置のみを考えていた 5.4 節のアルゴリズムを拡張する。ここまで検出器座標  $v \in V$  はパターン番号  $k$  と検出器ピクセル  $(p_x, p_y)$  による  $(k, p_x, p_y)$  の組で、天球面座標  $u \in U$  は 2 次元視野角  $(\theta_x, \theta_y)$  の組で記述してきた。再構成に偏光を加えるには、検出器座標には離散化した検出光電子放出角  $\phi_d$  を、天球面座標には離散化した偏光角  $\phi_s$  を付与し、 $v = (k, p_x, p_y, \phi_d)$ ,  $u = (\theta_x, \theta_y, \phi_s)$  で表現する。この設定の元で、 $M_{vu}$  は各  $\theta_x, \theta_y, k, p_x, p_y$  の組に対してはこれまでと同様に式 (5.27) の通りに「見上げた方向に穴が空いているか」を判断する。穴が空いている箇所それぞれで、天球面  $\tilde{S}$  の偏光角  $\phi_s$  から検出器  $\tilde{D}$  の検出光電子放出角  $\phi_d$  への対応を決定する。 $\phi_s$  と  $\phi_d$  の対応には現実的な検出器の偏光検出特性を考慮する必要がある。検出器の偏光応答は 6.4 節で議論したモジュレーションカーブで記述できる。検出器の 100% 偏光に対するモジュレーションファクター (表 6.5) を  $m$  とし、規格化前の  $M_{vu}$  の成分  $M'_{vu}$  をあらわに書くと、

$$M'_{vu} = \text{Pass}(m_x(u, v), m_y(u, v)) \sum_{\phi_s} [1 + m \cos 2(\phi_s - \phi_d)] \quad (6.26)$$

となる。特に  $\phi_s = 0^\circ, 90^\circ$ ,  $\phi_d = 0^\circ$  (H-type),  $90^\circ$  (V-type) のみ考慮する場合、穴が空いている箇所の  $(0^\circ, 90^\circ)$  から (H-type, V-type) への変換行列は

$$\begin{pmatrix} \text{H-type イベント数} \\ \text{V-type イベント数} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 + m & 1 - m \\ 1 - m & 1 + m \end{pmatrix} \begin{pmatrix} 0^\circ \text{成分} \\ 90^\circ \text{成分} \end{pmatrix} \quad (6.27)$$

として表現される。 $M_{vu}$  は以下のように  $M'_{vu}$  を各  $u$  について規格化することで得られる。

$$M_{vu} = \frac{M'_{vu}}{\sum_{v'} M'_{v'u}} \quad (6.28)$$

入射ビーム強度のスキャン別、パターン別の補正と検出器の放出角特性の補正は検出イベント  $D_v$  に重みをかけることで行なう。具体的には、 $\theta_z$  のスキャンで、パターン  $k$  の領域でのピーク成分の H-type イベントと V-type イベントの合計検出イベント数が  $w_{\theta_z, k}$  の

とき、

$$D'_{k,p_x,p_y,H} = \frac{C_H}{w_{\theta_z,k}} D_{k,p_x,p_y,H} \quad (6.29)$$

$$D'_{k,p_x,p_y,V} = \frac{C_V}{w_{\theta_z,k}} D_{k,p_x,p_y,V} \quad (6.30)$$

を補正済みの検出イベントとして扱い、規格化した

$$\delta_v = \frac{D'_v}{\sum_{v'} D'_{v'}} \quad (6.31)$$

を検出イベント分布とする。

$M_{vu}$  と  $\delta_v$  を拡張する以外は、元のアルゴリズムと同様である。 $\tilde{S}$  と  $\tilde{D}$  を規格化した一様分布で初期化し、E-step(式(5.22))とM-step(式(5.23))を交互に繰り返す。結果として得られるのは偏光角  $\phi_s$  で分割した天球面分布  $\tilde{S}_{\theta_x,\theta_y,\phi_s}$  であり、ここから重ね合わせ光の偏光度と偏光角も得ることができる。特に  $\phi_s = 0^\circ, 90^\circ$  のみを考える場合、

$$\tilde{I}_{\theta_x,\theta_y} = \tilde{S}_{\theta_x,\theta_y,0^\circ} + \tilde{S}_{\theta_x,\theta_y,90^\circ} \quad (6.32)$$

$$\tilde{Q}_{\theta_x,\theta_y} = \tilde{S}_{\theta_x,\theta_y,0^\circ} - \tilde{S}_{\theta_x,\theta_y,90^\circ} \quad (6.33)$$

$$(6.34)$$

とでき、偏光角情報を含めた  $Q$  方向の偏光度  $\Pi^{(Q)} \equiv \frac{Q}{I}$  は

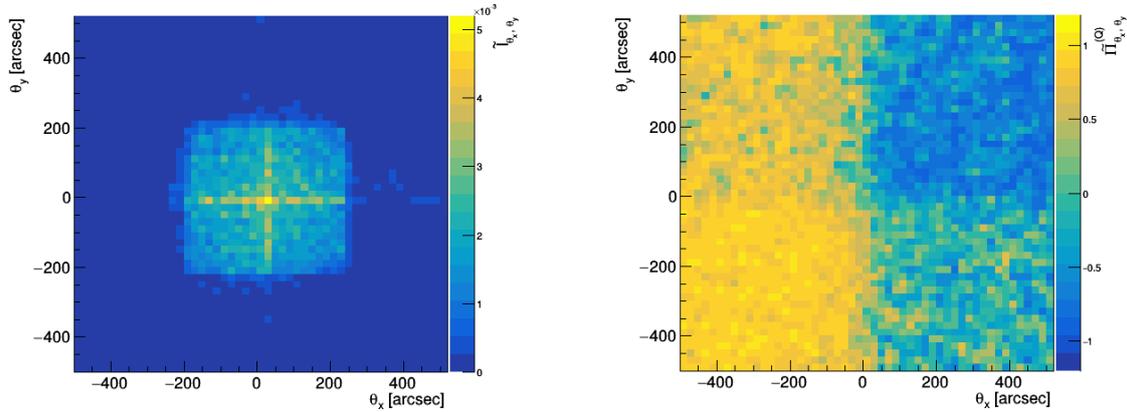
$$\tilde{\Pi}^{(Q)}_{\theta_x,\theta_y} = \frac{\tilde{S}_{\theta_x,\theta_y,0^\circ} - \tilde{S}_{\theta_x,\theta_y,90^\circ}}{\tilde{S}_{\theta_x,\theta_y,0^\circ} + \tilde{S}_{\theta_x,\theta_y,90^\circ}} \quad (6.35)$$

で推測できる。

#### 6.6.4 複数偏光源データの再構成

照射エネルギー 16 keV で 4 スキャン分の  $\theta_z = 0^\circ$ ,  $\theta_z = 90^\circ$  のデータを 6.6.2 小節に従ってブレンドしたデータを、偏光ありの EM アルゴリズムで再構成した結果を図 6.18 に示す。また、領域ごとの  $Q$  方向偏光度の平均値を表 6.8 に示す。

理論値との差異はあるが、領域ごとに有意に異なる偏光度を出すことに成功した。これは *cipher* での偏光撮像の実現を意味する。 $\tilde{\Pi}^{(Q)}$  の系統誤差としてはパターンごとの検出器座標の原点のずれや、 $\theta_z = 0^\circ$  と  $\theta_z = 90^\circ$  の天球面原点のずれなどが考えられる。実際、図 6.18a では  $\theta_x = 0''$ ,  $\theta_y = 0''$  に沿って推定光が強くなっているが、これは  $\theta_z = 0^\circ$  のデータと  $\theta_z = 90^\circ$  のデータを重ね合わせる際に天球面の原点を厳密に合わせられていない効果だと考えられる。



(a) 強度  $I$  の分布

(b)  $Q$  方向偏光度  $\Pi^{(Q)}$  の分布

図 6.18: 複数偏光源データ (図 6.17) の拡張 EM アルゴリズムによる再構成画像。999 ステップ目を示している。

表 6.8: 偏光撮像の結果

領域	実験からの推定値 $\tilde{\Pi}^{(Q)*}$	理論値 $\Pi^{(Q)}$
$-200'' \leq \theta_x \leq -50''$ , $-200'' \leq \theta_y \leq -50''$	$+0.846 \pm 0.002$	$+1.000$
$50'' \leq \theta_x \leq 200''$ , $50'' \leq \theta_y \leq -200''$	$-0.654 \pm 0.002$	$-1.000$
$50'' \leq \theta_x \leq 200''$ , $-200'' \leq \theta_y \leq -50''$	$-0.051 \pm 0.001$	$0.000$
$-200'' \leq \theta_x \leq -50''$ , $50'' \leq \theta_y \leq 200''$	$+0.532 \pm 0.002$	$+0.400$

\* 誤差は統計誤差に由来する。

### 6.6.5 任意の偏光角での画像再構成

ここまでで示した偏光光は  $Q$ - $U$  空間 (図 6.16c) 上では  $Q$  軸に沿った一次元的な空間にある。 $Q$  軸から離れて任意の偏光角データを再現するには  $\theta_z = 45^\circ, -45^\circ$  のデータを混ぜ合わせれば良い。ただ、このデータを正確に再構成するためには検出器上の光電子放出角の自由度も増やす必要がある。たとえば  $\phi_d = 45^\circ, -45^\circ$  を追加できれば式 (6.26) の通りに行列を構築することで再構成ができると予想されるが、double イベントのうち斜め方向のイベント数は少ないため、信頼性に欠ける。そのため、より weight の大きい triple や extended イベントを利用することが必要になってくるが、広がったイベントの光電子放出角を正確に推定するのは難しい。畠内 (2021) ではニューラルネットワークを用いた機械学習によって weight が 3 以上のイベントの光電子放出角の推定に成功している。しかし、教師データを生成するためのシミュレーションをデータに合うように最適化する必要がある、現時点ではこの実験に用いたセンサの応答を考慮したモデルの作成に至っていない。

い。これらのことから、 $Q$  方向以外の偏光角を含めた偏光撮像は今後の展望とし、本修論では扱わないものとする。

## 6.7 光子統計量からの *cipher* の実現性の考察

最後に、光子統計量の観点から *cipher* の実現性を考察する。光量の多い天体として、Crab Nebula を 1 年間観測するとする。Crab の硬 X 線帯域のスペクトルは典型的なシンクロトロン放射の power law で記述でき、その  $E_0$  から  $E_1$  のエネルギー範囲のフラックス [keV/cm<sup>2</sup>/s] は  $\Gamma = 2.1$ ,  $N = 9.7$  として

$$\text{flux} = \int_{E_0}^{E_1} N E^{1-\Gamma} dE \quad (6.36)$$

で良く表される (Kirsch et al., 2005)。検出器の検出効率が  $e(E)$  であるとき、検出されるイベントレート [/cm<sup>2</sup>/s] は

$$\text{Event Rate} = \int_{E_0}^{E_1} N E^{-\Gamma} e(E) dE \quad (6.37)$$

で求められる。 $E_0 = 10$  keV,  $E_1 = 30$  keV とし、検出器 1 つあたり  $1.28 \times 1.28$  cm<sup>2</sup> 上で 1 年間 (=  $3.1 \times 10^7$  s) 観測するとする。また検出器の特性として、3.4.5 小節で求めた表 3.7 の空乏層厚が全エネルギー帯でも成り立つと仮定する。このとき、検出されるイベント数は double イベントで  $9.7 \times 10^3$  個、全 weight で  $2.7 \times 10^5$  個と計算される。実際に撮像を行う場合、さらにマスクの開口率や検出器上の有効撮像ピクセルの制限を考慮する必要がある。本実験に合わせてマスク開口率が 50%、有効ピクセルが  $896 \times 896 \times 8$  であったとすると、撮像に使えるイベント数は double イベントで  $1.2 \times 10^3$  個、全 weight で  $3.4 \times 10^4$  個となる。

6.5.4 小節では、Crab スケールで広がった円環を再構成するのに必要なのは 3000 イベント程度と見積もった。偏光を含めた撮像にはこの数倍程度のイベントが必要だと考えられる。これは weight の大きいイベントを解析対象にし、センサを複数枚搭載することで達成可能である。ただし、上述の見積もり値を実現するにはセンサのエネルギー応答や weight の大きいイベントの偏光応答を十分に調査しておく必要がある (おわりにも参照)。

## 6.8 6章のまとめ

新たな読み出しシステムを用いて、ビームラインからの 100% 直線偏光光を GMAX0505RF に照射する試験を行なった。光学系をステージの電氣的な制御で細かく動かしながらデータ取得を行なうことで、420" 角に広がった 2 次元天体を模したデータセットを得ることができた。解析の結果、偏光感度が先行研究と同等であることを再確認した上で、EM アルゴリズムによる画像再構成に成功した。また Crab を模した円環光源の画像再構成を光子統計量を変えて行い、再構成に必要な光子統計量を 3000 イベント程度と見積もった。さらに異なる偏光角データを重ね合わせて領域ごとに偏光度の異なるデータセットを作成し、新たに考案した偏光を考慮した再構成手法を適用することで、天球面領域を切り分けて偏光度を出すことが可能であることを示した。

## 第7章 結論

本研究では、微小ピクセルCMOSセンサと符号化開口マスクを用いた撮像偏光計において、これまで未確立だったセンサ信号の読み出しから偏光撮像の解析までの一貫したデータ処理フローの構築に取り組んだ。特に重要であった、天体観測を意識したデータ読み出しシステムの開発、ノイズ低減を目指した画像再構成手法の検討、および偏光情報を含めた画像再構成手法の考案と実験データへの適用を行い、一貫したデータ処理フローを確立させた。以下に具体的な成果をまとめる。

- 特性X線源を用いて、GMAX0505RFのX線検出性能評価を行なった。検証の結果、エネルギー分解能は17.75 keVの輝線で $3.2 \times 10^2$  eVと十分な性能を持つ一方で、22.1 keVでの検出効率は0.27%、実効的な空乏層厚は $3.4 \mu\text{m}$ であることが分かった。また、ペDESTALの温度による変動を評価し、それが解析に与える影響を見積もった。
- 読み出しシステムの高速化に取り組んだ。温度変化に強いと考えられる動的にペDESTALを減算するアルゴリズムを採用し、イベントに由来しないデータを捨てるロジックをZDAQボード内のFPGAに実装した。X線源を用いた動作確認の結果、従来1.0 sほどあったフレームあたりのデッドタイムが、新たな読み出しシステムでは露光時間600 msで38 msまで抑えられていることを確認できた。
- 撮像ノイズ低減を目指して、符号化開口の再構成手法を相関法とEMアルゴリズムの双方で検討した。相関法ではランダムパターンセットに開口率が一樣になるような束縛条件を導入することで、最適化をしていないパターンセットの0.10倍程度の偽像の再構成画像を得られることが分かった。また逐次的に最尤推定を行なうEMアルゴリズムに適用できる偏光情報を含めた符号化開口イメージングのモデルを構築した。このモデルを用いた結果として、相関法で偽像が多かったパターンセットでも0.12倍程度の偽像の再構成画像を得られることが分かった。
- 新たな読み出しシステムを用いた撮像偏光計にSPring-8ビームラインの100%偏光X線を照射し、ステージ操作によって2次元天球面内に広がった光源を模したデータセットを取得した。その後の解析により、GMAX0505RFが偏光感度を持つことを再確認し、EMアルゴリズムによる画像再構成を成功させた。また、Crabを模した円環光源の画像再構成を行い、画像再構成に必要な光子統計量が3000イベント程度であることが分かった。さらに、場所ごとに偏光度の異なるデータを作成し、EMアルゴリズムの拡張によって天球面領域を切り分けて偏光度を出す再構成が可能であることを示した。

## おわりに: *cipher* の今後

冒頭で述べた通り、*cipher* プロジェクトでは東京大学馬場研究室の学生が中心となって研究と開発が進められてきた。その斬新なアイディアは「宇宙硬 X 線偏光撮像の開拓者 *cipher*」<sup>1</sup>として高く評価された。本格的な始動から約3年の月日が流れたが、実際に衛星として運用できる段階までには解決すべき課題が未だ多く残されている。本修論の終わりに、筆者の考える *cipher* が今後解決すべき課題を以下に述べる。これらの課題を克服し、*cipher* がまだ見ぬ宇宙の姿を明らかにできるような日が来るのを筆者は期待している。

- 光子統計量

筆者が2年間頭を悩ませ続けた問題である。読み出しシステムが改善されたとはいえまだ十分な統計量とはいえない。センサの空乏層がより厚ければ検出効率は上昇すると考えられるが、偏光感度を持つような微小ピクセルを持ちつつ空乏層がさらに厚いセンサは、現時点で筆者の知る限りまだ登場していない。今後の技術の発展で出てくる可能性はあるが、一般に空乏層の厚さは時間分解能・エネルギー分解能の良さとのトレードオフであり、商業的に可視光を中心として展開する CMOS センサ市場においてそのような方向に開発が進むかは分からない。<sup>2</sup>

有効面積を増やすためにセンサを多数並べるとするのは一つの案である。その場合、読み出しボードから複数センサを同時に制御する必要があり、更なる読み出しシステムの開発が求められる。

- センサ特性のさらなる評価

本修論では偏光解析と画像再構成には 16 keV の H-type イベントと V-type イベントのみを用いた。だが、実際には weight の大きいイベントも偏光感度を持っていると考えられ、これらの光電子放出角が推定できれば有効統計量は大きく増える。またエネルギー帯域をピーク付近に限定せず、広範囲なエネルギー帯を使用できれば有効統計量は格段に上昇する。そのためにはセンサのエネルギーごとの応答を理解しておく必要があり、スペクトル構造を説明するための適切な物理モデルの検討も必要になってくる。また 3.5 節で温度応答に触れたが、実際の宇宙空間での運用を踏まえたより正確な温度応答を知る必要もある。

- 軌道上バックグラウンド評価

実際に衛星を運用する場合、衛星軌道上の高エネルギーの宇宙線や大気粒子、地磁

---

<sup>1</sup>2019 年に開催された第 27 回衛星設計コンテストにおいてコンセプトを発表したときの題目。文部科学大臣賞並びにアイデア大賞を受賞している。

<sup>2</sup>「微小ピクセル」は可視光でも高解像度の撮影として威力を発揮するが、一般に X 線とは比較にならないほどの光量を観測する可視光では空乏層を厚くする意義は薄い。

気に束縛された荷電粒子などに衛星が晒されることになる。これらバックグラウンド粒子をシールドによって防ぐか、解析で天体由来イベントと区別できるようにするなど、対策が必要とされる。バックグラウンド粒子のデータは SPENVIS<sup>3</sup>などにあり、これを入力としたモンテカルロシミュレーションによってある程度のバックグラウンドレベルの見積もりを立てることはできる。

- サイエンス面の意義

本修論ではターゲットとして明るい X 線天体の Crab を挙げた。シンクロトロン光の偏光を観測することで磁場構造を明らかにすることが期待できるが、具体的に *cipher* でどのような構造がわかる可能性があるのかなどの検討はなされていない。これもモンテカルロシミュレーションによって、もしくは本修論で取得したビーム実験のデータを用いてある程度の見積もりを立てることはできる。

- 衛星設計

本格的な開発に当たっては、衛星全体の構造を検討する必要がある。バックグラウンド対策にシールドを設けて、かつ読み出しボード、センサ、光学系、電源系、姿勢制御系、などの各要素を、サイエンスが達成可能でかつ CubeSat のサイズに収納できるように設計しておく必要がある。

---

<sup>3</sup><https://www.spennis.oma.be/>

# 謝辞

世界的な疫病の流行は私の大学院生活にも少なからぬ影響を与えました。そのような中でもこの修士論文を完成させられたのは、多くの方々の支えがあってこそです。

まず指導教員である馬場彩 准教授は、正反対と言えるほど興味の対象がずれていた私の意思を尊重し、私に研究の意義を与えてくださいました。また、ふさぎ込みがちな私にかけ続けてくださった励ましの言葉がなかったら、この修士論文は存在していなかったでしょう。小高裕和 助教には研究テーマのアイデアから実験の細部まで非常に多岐に渡る的確なアドバイスをいただきました。多角的な視点からのご指摘は研究を進める上で非常に参考になりました。

外部の方々にもお世話になりました。まず東京大学 Kavli IPMU の高橋忠幸 教授をはじめとする高橋研究室の方々には、SPring-8 実験でデータ取得の鍵となる6軸ステージを貸与していただいたほか、実験の手伝いもしていただきました。SPring-8 実験では上杉健太郎主席研究員、星野真人主幹研究員にビームラインを使用させていただきただけでなく、ビームのエネルギー調整やスリット操作についても教えていただきました。また素粒子物理国際研究センターの神谷好郎 助教は同じ CMOS センサを使用する共同研究者として、実験道具の貸借や研究手法の相談に乗っていただきました。

研究室のメンバーにも大きく支えられました。博士2年の丹波翼さんには SPring-8 実験の進行役を引き受けていただいた他、ソフトウェア関係を中心に非常に多くの相談に乗っていただきました。PD の谷本敦さんには学会発表などで丁寧な添削をしていただきました。博士1年の高嶋聡さんは忙しい中 SPring-8 実験に参加して下さり、自分が休んでいる間のデータ取得を手伝っていただきました。修士1年の市橋正裕さんは SPring-8 実験で私が手の回らなかった部分のソフトウェア開発を手伝ってくれました。同じく修士1年の岩田季也さんは SPring-8 実験のメカ開発に尽力してくれた他、人の少ない研究室内での私のくだらない雑談に付き合ってくれました。同期の南木宙斗くんは、ともすると気が滅入ってしまうような時分において、ともに研究活動に臨んだ仲間であります。また言うまでもなく、この修士論文は *cipher* の先駆者である博士3年の春日知明さんをはじめ、卒業生の會澤優輝さん、畠内康輔さんの努力の上に成り立っています。そして、我々が日々の研究に専念できるのは秘書の沓間はづきさんと事務の河野久仁子さんのサポートがあってこそです。

最後に、研究漬けで辛いときに気分転換に付き合ってくれた高校時代の友人たち、そして気の弱い私を心身ともに支えてくれた家族に感謝の意を表して謝辞といたします。

令和4年1月6日  
渡邊 泰平

## 付録A データ削減IPの実装

具体的なデータ削減IPのVHDLによる実装例を示す。なお説明の単純化のため、実際の実装とは変数名や処理の一部を変更している。ここで用いる、表4.3と表4.4に載せていない信号のリストを表A.1に示す。

表 A.1: 信号名定義

信号名	型	説明
clk	std_logic	入力クロック
aresetn	std_logic	入力リセット信号
read_pointer_0	integer range 0 to 7	読み込みポインタ R <sub>0</sub> (R <sub>1</sub> も同様)
write_pointer	integer range 0 to 7	書き出しポインタ W
read_0	std_logic	buffer <sub>0</sub> で読み込みが行われたか (buffer <sub>1</sub> も同様)
state	表 4.3	出力の状態
sent_0	std_logic	出力 Out <sub>0</sub> で送信が成立したか (Out <sub>1</sub> も同様)
valid_0	std_logic	出力 Out <sub>0</sub> の valid 信号 (Out <sub>1</sub> も同様)
ready_0	std_logic	出力 Out <sub>0</sub> の ready 信号 (Out <sub>1</sub> も同様)

まず、リングバッファそのものは std\_logic\_vector の配列 (コード A.1) で表現することができる。ここでは深さを 8 とする。

コード A.1: リングバッファ

```
1 type BUF_TYPE is array (0 to 7) of std_logic_vector(31 downto 0);
2 signal buffer_0 : BUF_TYPE;
3 signal buffer_1 : BUF_TYPE;
```

リングバッファを指すポインタの動作はたとえばコード A.2 のようにプロセス文で記述できる。

コード A.2: ポインタの動作例

```
1 process (clk)
2 begin
3   if (rising_edge (clk)) then
4     if (aresetn = '0') then
5       read_pointer_0 <= 0;
6     else
7       if (read_A = '1') then
```

```

8      if (read_pointer_0 = 7) then
9          read_pointer_0 <= 0;
10     else
11         read_pointer_0 <= read_pointer_0 + 1;
12     end if;
13 end if;
14 end if;
15 end if;
16 end process;

```

出力の送受信が成立したかの判断は、コード A.3 に示す通り valid と ready がともにアサートされているかどうかで判断する。

#### コード A.3: 出力の判断

```

1 sent_0 <= valid_0 and ready_0;
2 sent_1 <= valid_1 and valid_1;

```

出力のステートマシンはコード A.4 のようになる。

#### コード A.4: ステートマシン

```

1 process (clk)
2     variable next_write_pointer : integer range 0 to 7;
3     variable pedestals_data : std_logic_vector(31 downto 0);
4     begin
5         if (rising_edge(clk)) then
6             if (aresetn = '0') then
7                 state <= IDLE;
8             else
9                 next_write_pointer := calc_next_pointer(write_pointer);
10                case (state) is
11                    when IDLE =>
12                        if (have_buffer_stock(read_pointer_0, read_pointer_1,
13                            write_pointer) = '1') then
14                            if (extraction_mode = '0') then
15                                state <= RQSEND;
16                            else
17                                if (contain_event_frame = '1') then
18                                    pedestals_data := std_logic_vector(unsigned(buffer_1(
19                                        next_write_pointer)) - unsigned(buffer_0(
20                                        next_write_pointer)));
21                                else
22                                    pedestals_data := buffer_1(next_write_pointer);
23                                end if;
24                                if ((is_event(buffer_0(next_write_pointer), pedestals_data,
25                                    event_threshold, frame_numbers) = '1') then
26                                    state <= RQSEND;
27                                else
28                                    state <= NOTSEND;
29                                end if;
30                            end if;
31                        else
32                            state <= IDLE;
33                        end if;
34                    end case;
35                end if;
36            end if;
37        end if;
38    end process;

```

```

31     when RQ_SEND =>
32         state <= SEND_01;
33
34     when SEND_01 =>
35         if (sent_0 = '1' and sent_1 = '1') then
36             state <= IDLE;
37         elsif (sent_0 = '1') then
38             state <= SEND_1;
39         elsif (sent_1 = '1') then
40             state <= SEND_0;
41         end if;
42
43     when SEND_0 =>
44         if (sent_0 = '1') then
45             state <= IDLE;
46         end if;
47
48     when SEND_1 =>
49         if (sent_1 = '1') then
50             state <= IDLE;
51         end if;
52
53     when NOT_SEND =>
54         state <= IDLE;
55
56     when others =>
57         state <= IDLE;
58     end case;
59 end if;
60 end if;
61 end process;

```

ここで、選別時の関数は式 4.8 にしたがって、コード A.5 のように実装できる。AXIS のデータ長が 32 bit で、その中に 16 bit の波高値データが 2 ピクセル分あるため、そのどちらかが条件を満たせば送信対象にする実装になっている。

#### コード A.5: 選別関数

```

1  function is_event( in_data : std_logic_vector(31 downto 0);
2                    pedestals_data : std_logic_vector(31 downto 0);
3                    event_threshold : std_logic_vector(15 downto 0);
4                    frame_numbers : std_logic_vector(3 downto 0)
5  ) return std_logic is
6  begin
7      if (unsigned(in_data(15 downto 0)) >= unsigned(event_threshold) +
8         unsigned(pedestals_data(15 downto 0)) / unsigned(frame_numbers)
9         or unsigned(in_data(31 downto 16)) >= unsigned(event_threshold) +
10        unsigned(pedestals_data(31 downto 16)) / unsigned(frame_numbers)) then
11          return '1';
12      else
13          return '0';
14      end if;
15  end;

```

# 付録B ZDAQボード上読み出しソフトウェア

## B.1 ソフトウェアの実装概要

ZDAQ ボード上のソフトウェアはホスト PC との通信と FPGA 操作の役割を担っており、C 言語と C++ で実装されている。FPGA 内のレジスタや DDR メモリは Linux 内に仮想メモリとしてアドレスマッピングされており、設定したアドレスを C 言語の `mmap` で変数に割り当てることでソフトウェア内からの FPGA 側へのアクセスが可能になる。たとえば DDR メモリへのポインタを `mapDDR` という変数で扱う場合はコード B.1 のようにする。

コード B.1: メモリマッピング

```
1 void *mapDDR;
2
3 void mapping() {
4     char *dev="/dev/mem";
5     int fd = open( dev, ORDWR | O_SYNC);
6     mapDDR = mmap(0, 0x1F400000, PROT_READ|PROT_WRITE, MAP_SHARED, fd, 0
7     x20000000);
8 }
```

DDR メモリをフレームサイズで分割し、領域 0 と領域 1 内の値を足した結果を領域 2 に出力するという処理 (式 4.13) はコード B.2 のように実装できる。

コード B.2: DDR メモリデータの CPU による演算

```
1 const int FRAME_SIZE = 5120*5120*2;
2
3 void add_DDRmemory_by_CPU() {
4     int in_0 = 0;
5     int in_1 = 1;
6     int out_0 = 2;
7     for (int j=0; j<5120; j++){
8         for (int i=0; i<5120/2; i++) {
9             int addr = 4*(i + j*5120/2);
10            *(volatile unsigned long *) (mapDDR + FRAME_SIZE*out_0 + addr) = *(
11            volatile unsigned long *) (mapDDR + FRAME_SIZE*in_0 + addr) + *(
12            volatile unsigned long *) (mapDDR + FRAME_SIZE*in_0 + addr);
13        }
14    }
15 }
```

データ削減 IP に接続した 2 つの DMA を動作させて DDR メモリからデータを流すには、マッピングした変数を mapDMA1, mapDMA2 として、コード B.3 のように実行する。レジスタのアドレスや役割はドキュメント (Xilinx, Inc., 2019) を参照されたい。

コード B.3: DDR メモリデータの FPGA による演算

```

1 void add_DDRmemory_by_FPGA() {
2     int in_0 = 0;
3     int in_1 = 1;
4     int out_0 = 2;
5
6     *(volatile unsigned long *) (mapDMA1 + 0x0000) |= 0x01;
7     *(volatile unsigned long *) (mapDMA1 + 0x0018) = 0x20000000 + in_0 *
8     FRAME_SIZE;
9     *(volatile unsigned long *) (mapDMA1 + 0x0028) = FRAME_SIZE;
10
11    *(volatile unsigned long *) (mapDMA2 + 0x0000) |= 0x01;
12    *(volatile unsigned long *) (mapDMA2 + 0x0018) = 0x20000000 + in_1 *
13    FRAME_SIZE;
14    *(volatile unsigned long *) (mapDMA2 + 0x0028) = FRAME_SIZE;
15
16    *(volatile unsigned long *) (mapDMA1 + 0x0030) |= 0x01;
17    *(volatile unsigned long *) (mapDMA1 + 0x0048) = 0x20000000 + out_0 *
18    FRAME_SIZE;
19    *(volatile unsigned long *) (mapDMA1 + 0x0058) = FRAME_SIZE;
20
21    // wait until DMA jobs are done
22    while (((*(volatile unsigned long *) (mapDMA1 + 0x0004) & 0x02) == 0 ||
23            (*(volatile unsigned long *) (mapDMA2 + 0x0004) & 0x02) == 0 || (*(
24            volatile unsigned long *) (mapDMA1 + 0x0034) & 0x02) == 0)) {}
25 }

```

実装したコードは Ubuntu20.04 上でクロスコンパイルし、生成したバイナリファイルを Zynq 上の Linux 内にコピーすることで実行可能になる。

## B.2 CPU と FPGA の処理速度比較

コード B.4: CPU 確保メモリデータの CPU による演算

```

1 unsigned long cpu_memory_0[5120*5120/2], cpu_memory_1[5120*5120/2],
2   cpu_memory_2[5120*5120/2];
3 void add_CPUmemory_by_CPU() {
4     for (int j=0; j<5120; j++){
5         for (int i=0; i<5120/2; i++) {
6             int ind = (i + j*5120/2);
7             cpu_memory_0[ind] = cpu_memory_1[ind] + cpu_memory_2[ind];
8         }
9     }
10 }

```

処理方法を変えて、add の処理 (式 4.13) 速度を比較した。DDR メモリデータの CPU による処理 (コード B.2)、CPU 確保メモリの CPU による処理 (コード B.4)、DDR メモリ

データの FPGA による処理 (コード B.3) をそれぞれ ZDAQ ボード内で実行し、処理前後の時刻を Linux の時刻計測機能を利用して ms 単位で計測した。

時間計測の結果を表 B.1 に示す。まず DDR メモリデータの CPU による処理は 5.7 s と大幅に時間がかかっており、これでは図 4.7 のタイミングチャートに示すような速度でデータ削減を行なうことはできない。次に、CPU のみによる処理は 899 ms と DDR メモリを介する処理に比べて高速である。これは DDR メモリと CPU との通信に時間がかかっているためだと考えられる。そして FPGA を利用した DDR メモリ処理は前述の 2 手法よりもさらに高速な 194 ms であり、FPGA 処理の速さを確かめられる結果となった。

表 B.1: add 処理速度比較

	処理速度 [ms]
DDR メモリデータの CPU による処理 (コード B.2)	5720
CPU 確保メモリの CPU による処理 (コード B.4)	899
DDR メモリデータの FPGA による処理 (コード B.3)	194

## 参考文献

- Abarr, Q., Awaki, H., Baring, M. G., et al. (2021) “XL-Calibur - a second-generation balloon-borne hard X-ray polarimetry mission,” *Astroparticle Physics*, Vol. 126, p. 102529, March, DOI: 10.1016/j.astropartphys.2020.102529.
- Agostinelli, S., Allison, J., Amako, K., et al. (2003) 「Geant4—a simulation toolkit」, 『Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment』, 第506卷, 第3号, 250–303頁, DOI: [https://doi.org/10.1016/S0168-9002\(03\)01368-8](https://doi.org/10.1016/S0168-9002(03)01368-8).
- Allison, J., Amako, K., Apostolakis, J., et al. (2006) “Geant4 developments and applications,” *IEEE Transactions on Nuclear Science*, Vol. 53, No. 1, pp. 270–278, DOI: 10.1109/TNS.2006.869826.
- Allison, J., Amako, K., Apostolakis, J., et al. (2016) “Recent developments in Geant4,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Vol. 835, pp. 186–225, DOI: <https://doi.org/10.1016/j.nima.2016.06.125>.
- Angel, J. R. P. (1969) “Polarization of Thermal X-Ray Sources,” *apj*, Vol. 158, p. 219, October, DOI: 10.1086/150185.
- Asakura, K., Hayashida, K., Hanasaka, T., et al. (2019) “X-ray imaging polarimetry with a 2.5- $\mu\text{m}$  pixel CMOS sensor for visible light at room temperature,” *Journal of Astronomical Telescopes, Instruments, and Systems*, Vol. 5, p. 035002, July, DOI: 10.1117/1.JATIS.5.3.035002.
- Caroli, E., Stephen, J. B., Di Cocco, G., et al. (1987) “Coded Aperture Imaging in X-Ray and Gamma-Ray Astronomy,” *SSR*, Vol. 45, No. 3-4, pp. 349–403, September, DOI: 10.1007/BF00171998.
- Chauvin, M., Friis, M., Jackson, M., et al. (2017) “Calibration and performance studies of the balloon-borne hard X-ray polarimeter PoGO+,” *Nuclear Instruments and Methods in Physics Research A*, Vol. 859, pp. 125–133, July, DOI: 10.1016/j.nima.2017.03.027.
- Chauvin, M., Florén, H. G., Friis, M., et al. (2018) “The PoGO+ view on Crab off-pulse hard X-ray polarization,” *mnras*, Vol. 477, No. 1, pp. L45–L49, June, DOI: 10.1093/mnrasl/sly027.

- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977) “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society: Series B (Methodological)*, Vol. 39, No. 1, pp. 1–22.
- Dolan, J. F. (1967) “The polarization of celestial x-rays,” *Space Science Reviews*, Vol. 6, No. 5, pp. 579–600.
- Fenimore, E. E., & Cannon, T. M. (1978) “Coded aperture imaging with uniformly redundant arrays,” *Appl. Opt.*, Vol. 17, No. 3, pp. 337–347, Feb, DOI: 10.1364/AO.17.000337.
- Fraser, G., Lees, J., & Pearson, J. (1989) “Measurement of vectorial effects in the X-ray and UV photoemission from CsI: A novel polarimeter for soft X-ray astronomy,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Vol. 284, No. 2, pp. 483–508, DOI: [https://doi.org/10.1016/0168-9002\(89\)90321-5](https://doi.org/10.1016/0168-9002(89)90321-5).
- Hitomi Collaboration (2018) “Detection of polarized gamma-ray emission from the Crab nebula with the Hitomi Soft Gamma-ray Detector,” *pasj*, Vol. 70, No. 6, p. 113, December, DOI: 10.1093/pasj/psy118.
- Ikeda, S., Odaka, H., Uemura, M., et al. (2014) “Bin mode estimation methods for Compton camera imaging,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, Vol. 760, pp. 46–56, DOI: <https://doi.org/10.1016/j.nima.2014.05.081>.
- Ishikawa, Shin · Takahashi, Tadayuki · Watanabe, Shin, 他 (2018) 「High-speed X-ray imaging spectroscopy system with Zynq SoC for solar observations」, 『Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment』, 第 912 卷, 191–194 頁, DOI: <https://doi.org/10.1016/j.nima.2017.11.033>, New Developments In Photodetection 2017.
- Kasuga, T., Odaka, H., Hatauchi, K., et al. (2020) “Artifact-less coded aperture imaging in the x-ray band with multiple different random patterns,” *Journal of Astronomical Telescopes, Instruments, and Systems*, Vol. 6, p. 035002, July, DOI: 10.1117/1.JATIS.6.3.035002.
- Kirsch, M. G., Briel, U. G., Burrows, D., et al. (2005) “Crab: the standard x-ray candle with all (modern) x-ray satellites,” in Siegmund, O. H. W. ed. *UV, X-Ray, and Gamma-Ray Space Instrumentation for Astronomy XIV*, Vol. 5898, pp. 22 – 33, International Society for Optics and Photonics: SPIE, DOI: 10.1117/12.616893.
- Koyama, Katsuji · Tsunemi, Hiroshi · Dotani, Tadayasu, 他 (2007) 「X-Ray Imaging Spectrometer (XIS) on Board Suzaku」, 『Publications of the Astronomical Society of Japan』, 第 59 卷, 第 sp1 号, S23–S33 頁, 01 月, DOI: 10.1093/pasj/59.sp1.S23.

- Leo, W. R. (1994) *”Techniques for Nuclear and Particle Physics Experiments: A How-to Approach” Second Revised Edition*: Springer.
- Lockwood, J. W., Gupte, A., Mehta, N., et al. (2012) “A Low-Latency Library in FPGA Hardware for High-Frequency Trading (HFT),” in *2012 IEEE 20th Annual Symposium on High-Performance Interconnects*, pp. 9–16, DOI: 10.1109/HOTI.2012.15.
- MacWilliams, F., & Sloane, N. (1976) “Pseudo-random sequences and arrays,” *Proceedings of the IEEE*, Vol. 64, No. 12, pp. 1715–1729, DOI: 10.1109/PROC.1976.10411.
- McNamara, A. L., Kuncic, Z., & Wu, K. (2008) “X-ray polarization signatures of Compton scattering in magnetic cataclysmic variables,” *Monthly Notices of the Royal Astronomical Society*, Vol. 386, No. 4, pp. 2167–2172, 04, DOI: 10.1111/j.1365-2966.2008.13174.x.
- Odaka, Hirokazu • Sugimoto, Soichiro • Ishikawa, Shin, 他 (2010) 「Development of an integrated response generator for Si/CdTe semiconductor Compton cameras」, 『Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment』, 第 624 卷, 第 2 号, 303–309 頁, DOI: <https://doi.org/10.1016/j.nima.2009.11.052>, New Developments in Radiation Detectors.
- Odaka, H., Kasuga, T., Hatauchi, K., et al. (2020) “Concept of a CubeSat-based hard x-ray imaging polarimeter: cipher,” in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Vol. 11444 of Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, p. 114445V, December, DOI: 10.1117/12.2560615.
- Putnam, A., Caulfield, A. M., Chung, E. S., et al. (2014) “A reconfigurable fabric for accelerating large-scale datacenter services,” in *2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA)*, pp. 13–24, DOI: 10.1109/ISCA.2014.6853195.
- Rybicki, G. B., & Lightman, A. P. (1986) *Radiative processes in astrophysics*: John Wiley & Sons.
- Shepp, L. A., & Vardi, Y. (1982) “Maximum Likelihood Reconstruction for Emission Tomography,” *IEEE Transactions on Medical Imaging*, Vol. 1, No. 2, pp. 113–122, DOI: 10.1109/TMI.1982.4307558.
- Sze, S. M. J. (2002) *”Semiconductor Devices: Physics and Technology” 2nd Edition*,: Wiley & Sons, Inc.
- Tanaka, T., Uchida, H., Nakajima, H., et al. (2018) “Soft X-ray Imager aboard Hitomi (ASTRO-H),” *Journal of Astronomical Telescopes, Instruments, and Systems*, Vol. 4, p. 011211, January, DOI: 10.1117/1.JATIS.4.1.011211.

- Vetter, K., Mihailescu, L., Nelson, K., et al. (2006) “Gamma-ray imaging methods,” Technical report, Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States).
- Weisskopf, M. C., Ramsey, B., O’Dell, S., et al. (2016) “The Imaging X-ray Polarimetry Explorer (IXPE),” in den Herder, J.-W. A., Takahashi, T. & Bautz, M. eds. *Space Telescopes and Instrumentation 2016: Ultraviolet to Gamma Ray*, Vol. 9905, pp. 356 – 365, International Society for Optics and Photonics: SPIE, DOI: 10.1117/12.2235240.
- Xilinx, Inc. (2019) “AXI DMA Controller,” [https://japan.xilinx.com/products/intellectual-property/axi\\_dma.html](https://japan.xilinx.com/products/intellectual-property/axi_dma.html), (Accessed on 12/09/2021).
- 黒田正博 (2020) 『統計学 One Point 18 EM アルゴリズム』, 共立出版.
- 佐久間翔太郎 (2021) 「MIXIM 用 CMOS 素子のデータ処理システム開発と荷電粒子バックグラウンド・放射線劣化の研究」, 修士論文, 大阪大学大学院理学研究科.
- 春日知明 (2019) 「宇宙 X 線偏光観測を目指した符号化開口イメージング手法の設計と評価」, 修士論文, 東京大学大学院理学系研究科.
- 小林優 (2016) 『FPGA プログラミング大全 Xilinx 編』, 秀和システム.
- 長谷川裕恭 (2015) 『改訂・VHDL によるハードウェア設計入門』, CQ 出版.
- 天野英晴 (編) (2016) 『FPGA の原理と構成』, オーム社.
- 畠内康輔 (2021) 「CMOS イメージセンサを用いた超小型 X 線偏光計の開発」, 修士論文, 東京大学大学院理学系研究科.
- 會澤優輝 (2020) 「ブラックホール降着系の理解を目指した CMOS センサ型 X 線偏光計の性能評価」, 修士論文, 東京大学大学院理学系研究科.