突発天体検知のための MITSuME望遠鏡用 自動解析パイプラインの GPUを用いた高速化

東京工業大学河合研究室

庭野　聖史

共同研究者

村田 勝寛,伊藤 亮介,橘 優太朗(東京工業大学) 下川辺 隆史(東京大学)
河合 誠之,谷津 陽一,森田 浩太郎,大枝 幹,飯田 康太,白石 一輝,安達 稜
(東京工業大学)

目次

1. イントロダクション
2. 目的
3. 方法
4. 時間計測
5. 新スクリプト構築
6. 性能評価
7. まとめ

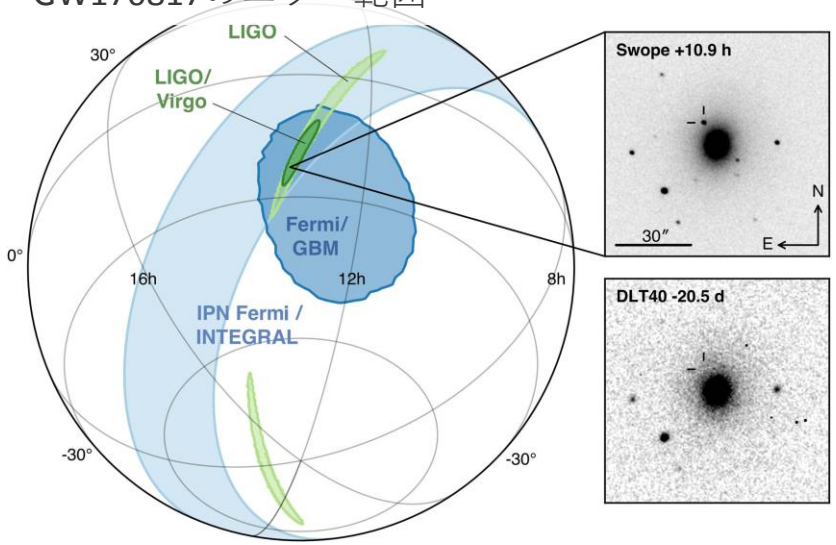
1. イントロダクション
突発天体の即時観測の意義

- ガンマ線バースト、重力波等は何時何処で発生するか分からない上に短時間変動する

⇛直ちに詳細な位置決定をすること重要

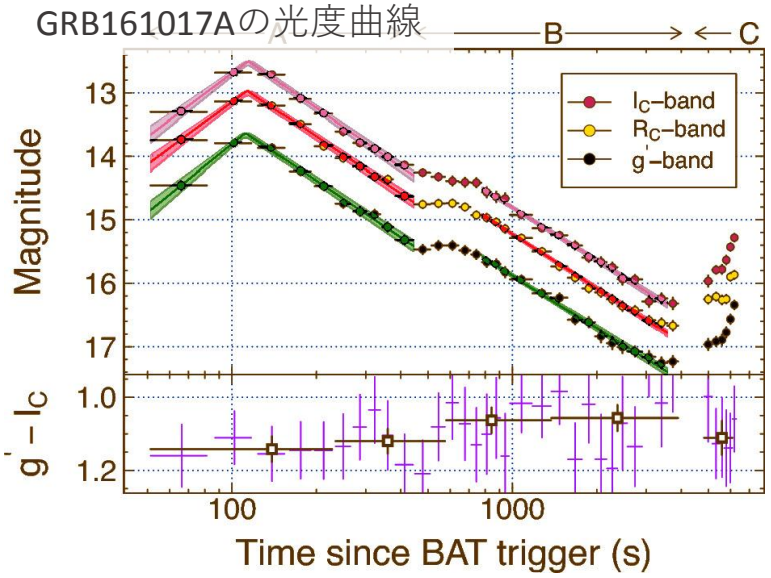
⇛即時観測、解析が必要

GW170817のエラー範囲



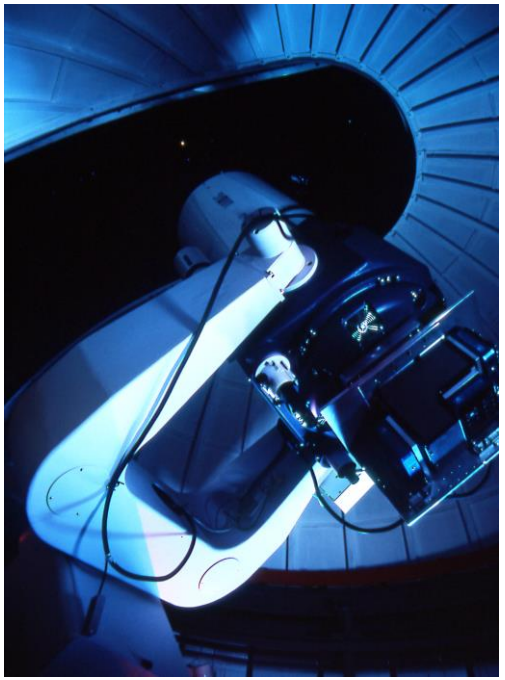
LIGO Scientific Collaboration, et,al. 2017

GRB161017Aの光度曲線



Tachibana, et,al. 2018

1. イントロダクション
MITSuME



- 東工大河合研が運用する観測システム
  - 所在は北杜市(山梨県)、浅口市(岡山県)
  - g’,Rc,Icの3色を同時撮影
- 突発天体の追観測が主目的であり、速報の受信から観測、解析までが自動化されている。

画像提供: 河合研 谷津助教

1. イントロダクション
画像一次処理

- 生のCCD画像には星の光以外の要素が含まれており、これを取り除かなければならない

$$\mathrm{Signal} + \mathrm{Sky} = \frac{\mathrm{Count} - \mathrm{Bias} - \mathrm{Dark}}{\mathrm{Flat}}$$

- 高いS/Nが要求される場合、重ね合わせも行う

$$(\mathrm{Signal} + \mathrm{Sky})_{\mathrm{combine}} = \frac{1}{N}\sum_{i}^{N}(\mathrm{Signal} + \mathrm{Sky})_i$$

2. 目的

- MITSuMEの自動解析パイプライン、特に一次処理の段階で時間が掛かっている。

⇛ 高速化し、一刻も早い突発天体の発見に繋げたい

3. 方法

① 時間計測

現行パイプラインの実行時間を計測し、効率的な高速化の方法を探る

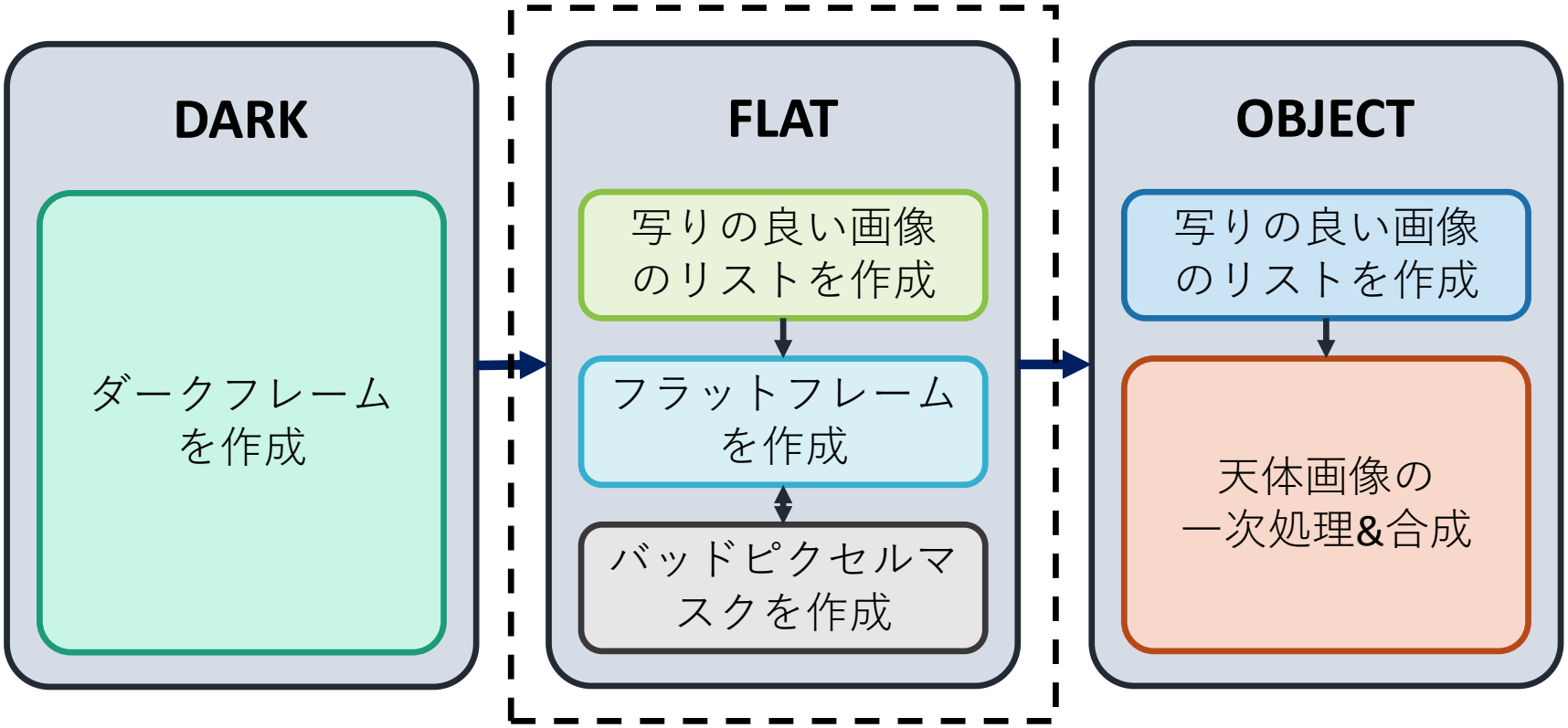
② 新スクリプト構築

③ 性能評価

高速化しつつ、現行パイプラインと同じ結果を出力できているか確かめる

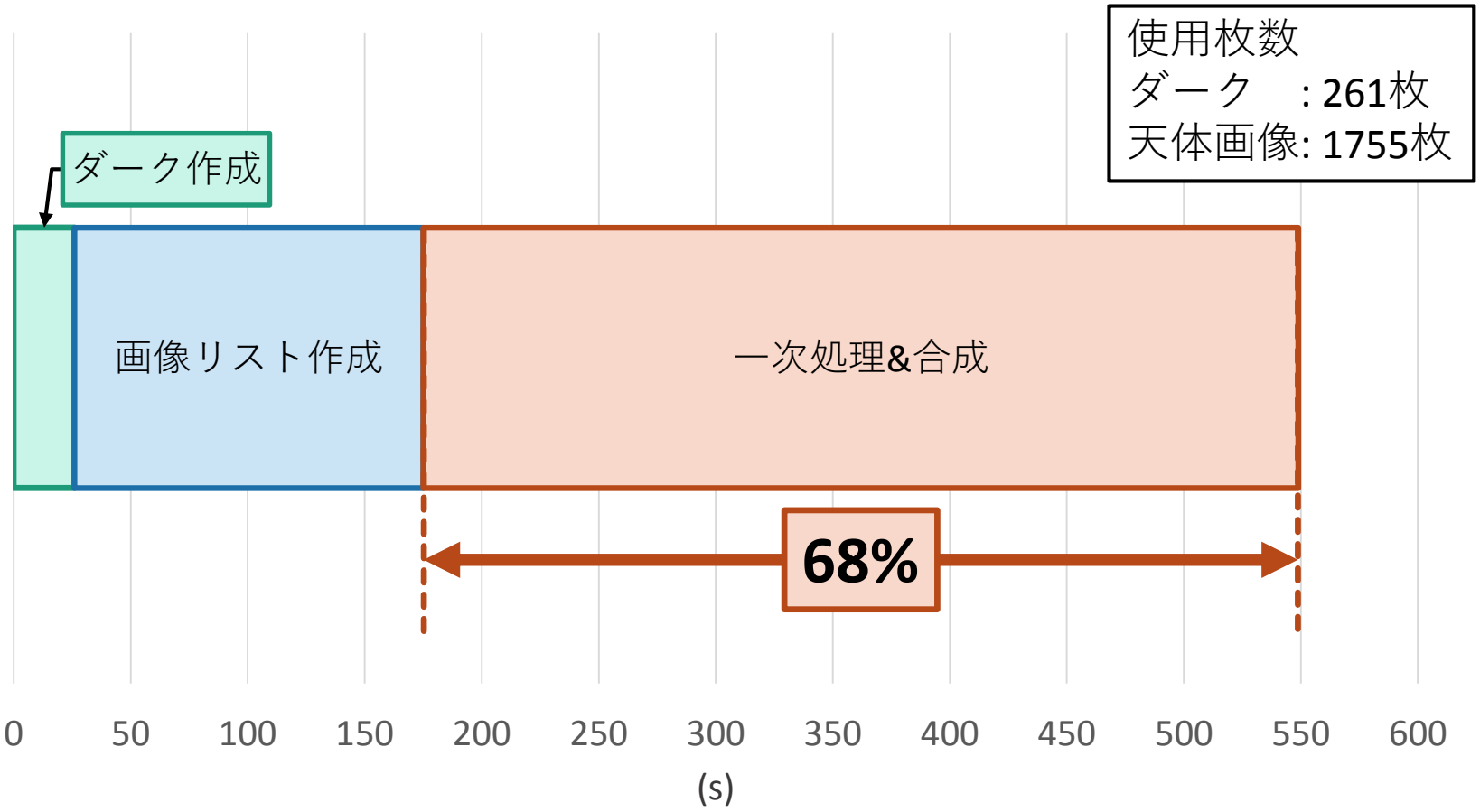
4. 時間計測

4. 時間計測
パイプラインの流れ



※フラットは毎回作成しなくとも良い

4. 時間計測
計測結果

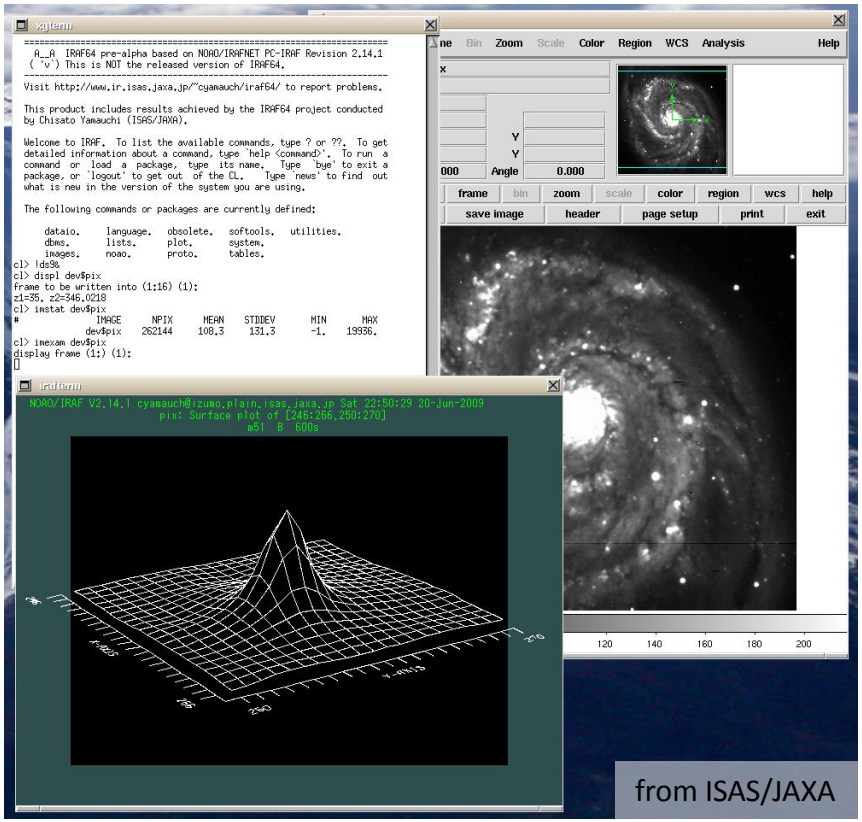

使用枚数
ダーク　: 261枚
天体画像: 1755枚
ダーク作成
画像リスト作成
一次処理&合成
68%
0
50
100
150
200
250
300
350
400
450
500
550
600
(s)

4. 時間計測
―次処理スクリプト

―次処理スクリプトはPyRAFを介してIRAFを利用するPythonスクリプト

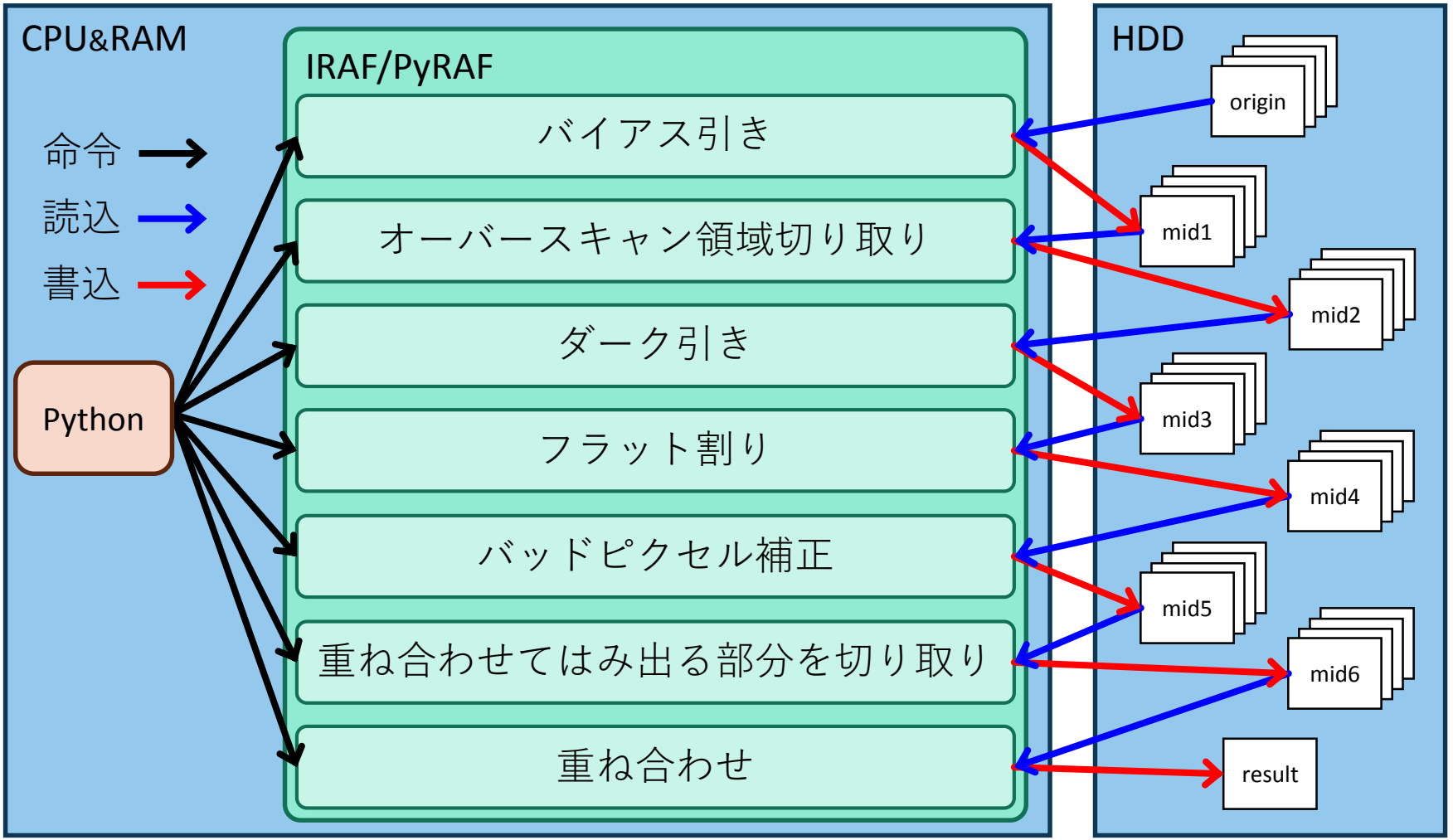
IRAF

(Image Redcution and Analysis Facility)
:アメリカ国立天文台の
開発した天文学用ソフトウェア群



from ISAS/JAXA

4. 時間計測
一次処理スクリプトの内容



4. 時間計測
対応策

- **中間ファイルを作成しない**
  - ファイルI/Oが律速している可能性があるため
  - IRAFを使用しない。
- **処理の並列化を行う**
  - 単純で並列化が容易な処理が多いため

(例 : 全ピクセルから一律にバイアスを引く、etc.)

  - 画像データの処理をGPU上で行うようにする

5. 新スクリプト構築

5. 新スクリプト構築
GPU

GPU (Graphics Processing Unit) :

主に画像処理を目的とした演算装置

- CPUに比べて**単純な処理しかできない**代わりに、**並列処理能力が格段に高い**ことが特徴。
- 画像処理以外の目的で使用されることも増えている。(GPGPU)

	クロック	コア数	メモリ帯域幅	価格
CPU (Intel Core i9-9900K)	3.6 GHz	8	41.6 GB/s	約7万円
GPU (NVIDIA GeForce GTX 1070Ti)	1.6 GHz	2432	256 GB/s	約6万円

5. 新スクリプト構築
使用ソフトウェア

CuPy

Astropy :

天文学用Pythonライブラリ

CuPy :

CUDAを利用するためのPythonライブラリ

NumPyと同様のAPIを持つ

※CUDA : NVIDIAの提供するGPGPU用フレームワーク

```
In [1]: import numpy as np

In [2]: A = np.array([[1,2],[3,4]])

In [3]: B = np.array([[5,6],[7,8]])

In [4]: np.dot(A,B)
Out[4]:
array([[19, 22],
       [43, 50]])
```

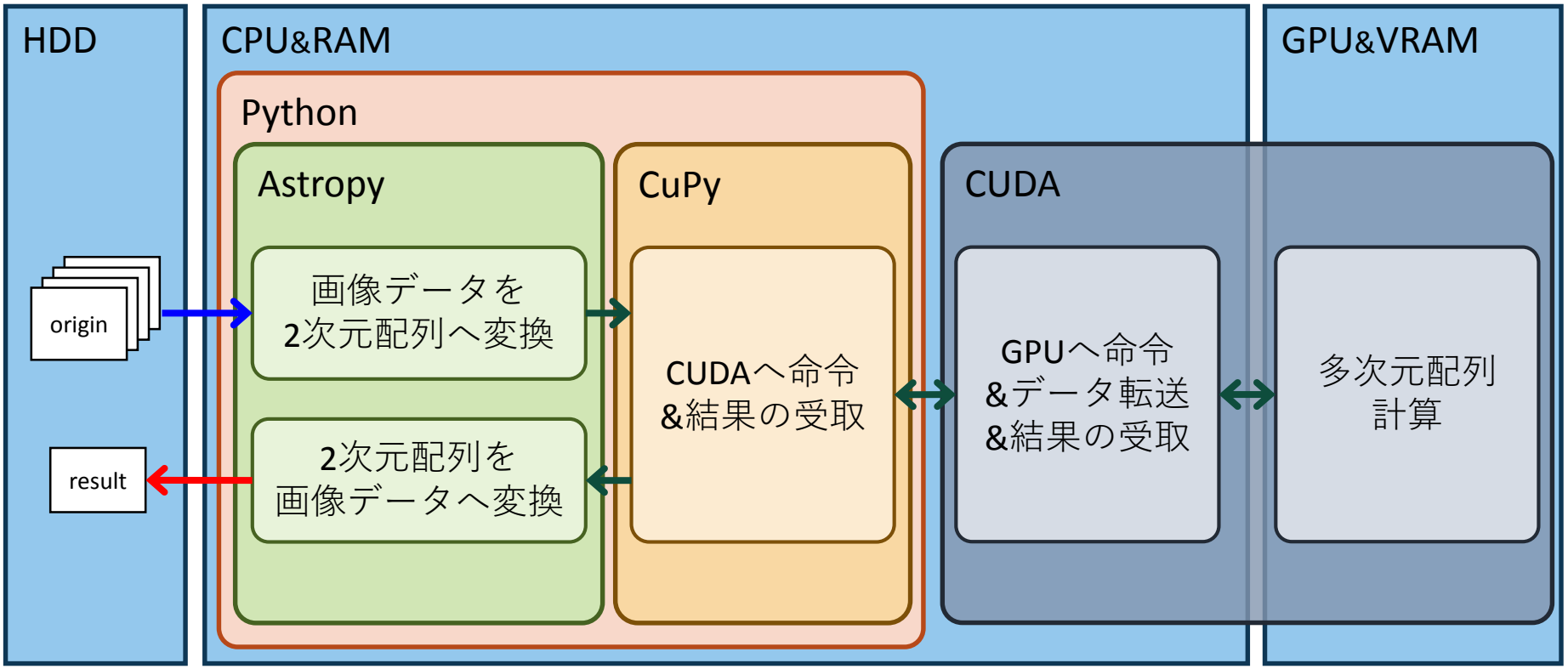
```
In [1]: import cupy as cp

In [2]: A = cp.array([[1,2],[3,4]])

In [3]: B = cp.array([[5,6],[7,8]])

In [4]: cp.dot(A,B)
Out[4]:
array([[19, 22],
       [43, 50]])
```

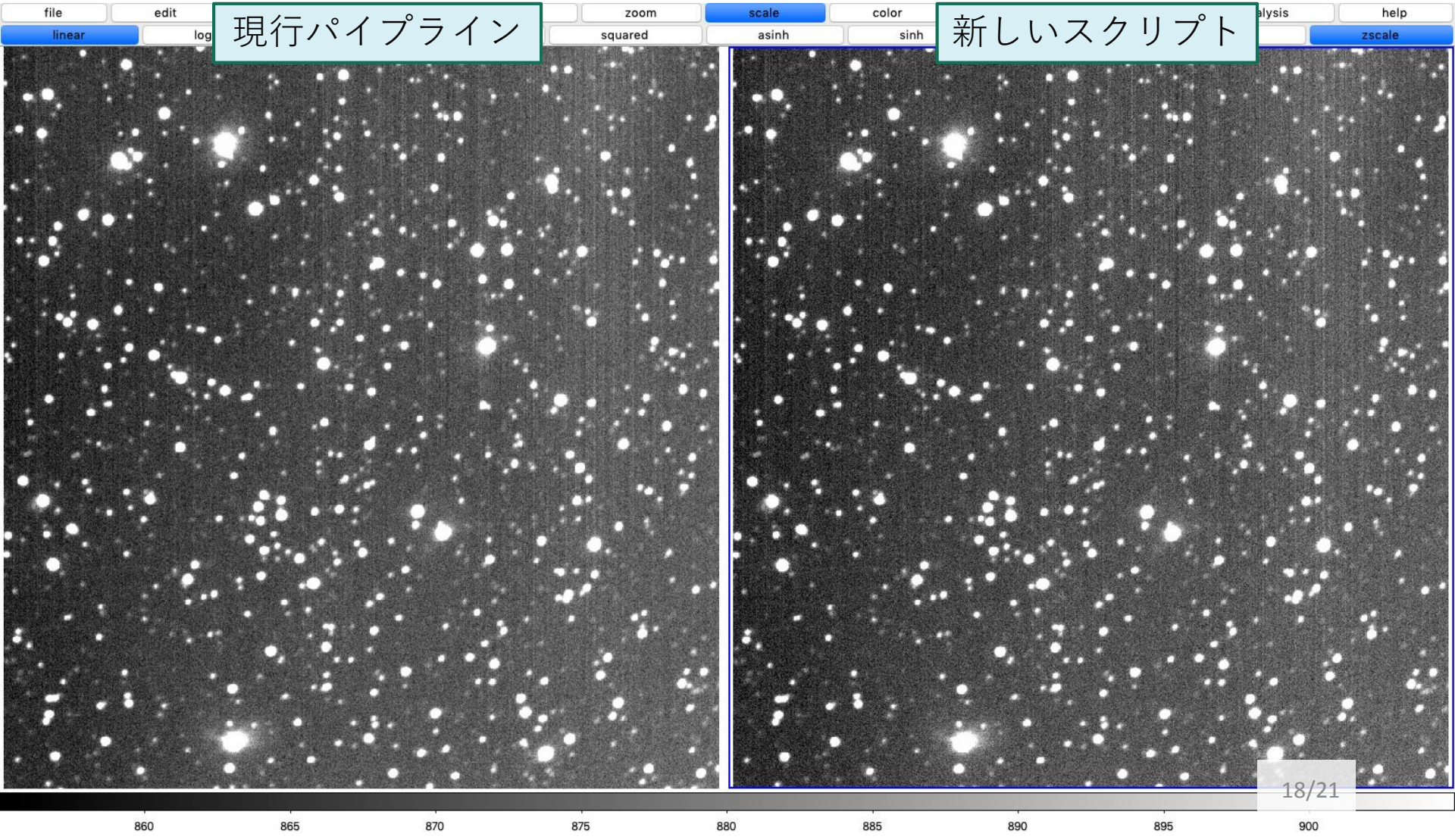
5. 新スクリプト構築
概要



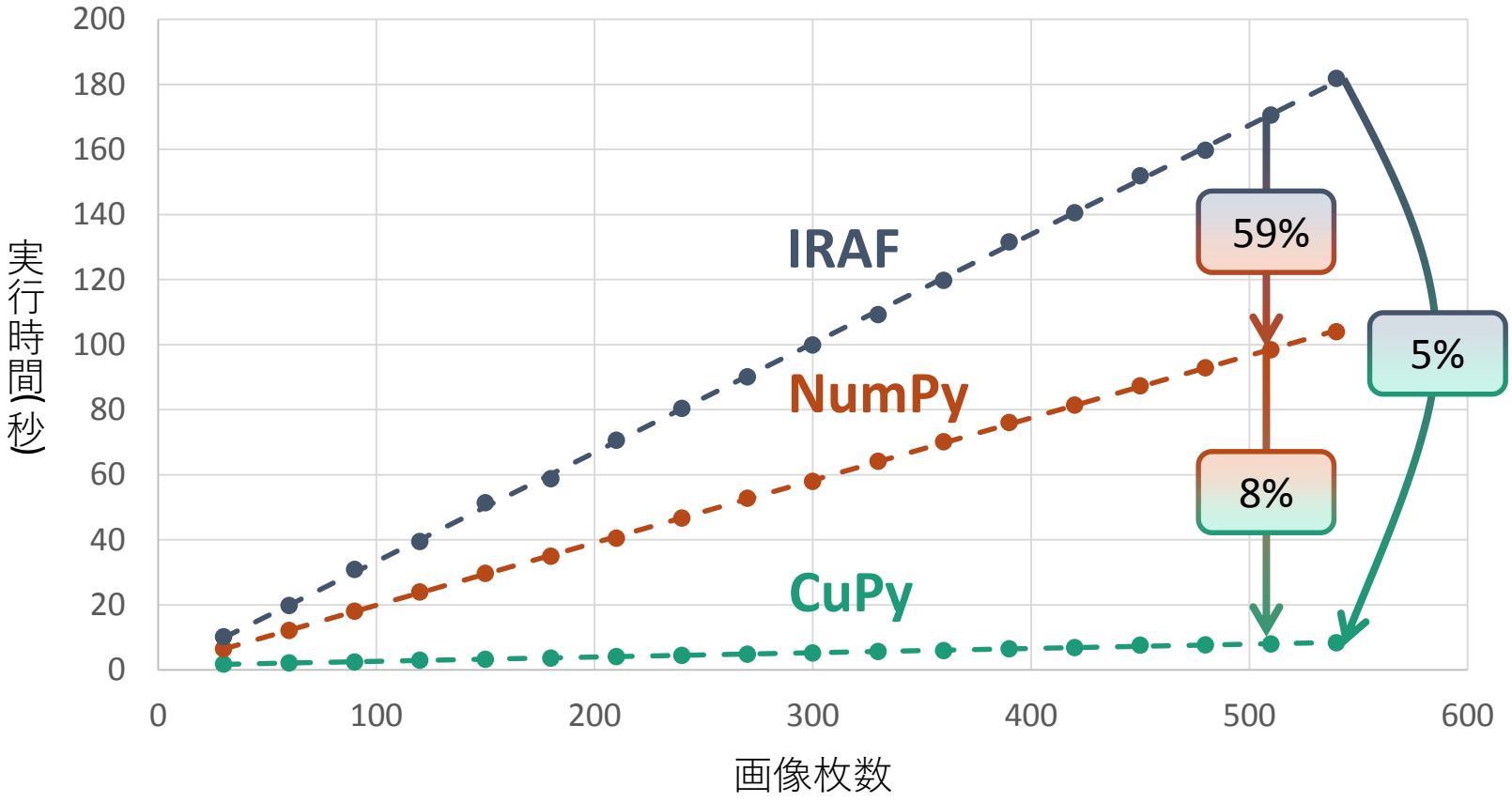
6. 性能評価

6. 性能評価
出力の比較

差 (新−旧)
$\Delta\mathrm{FWHM} \approx -0.1$ [pixel]
$\Delta\mathrm{Mag} \approx \pm 0.01$



6. 性能評価
実行時間の比較



6. 性能評価
高速化の恩恵

対象天体が、1枚では検知不可能なほど暗く、短時間の変動をしている場合

⇛様々な画像の組み合わせで合成して、よりS/Nの高くなる組み合わせを探るより外ない

2つの望遠鏡で60秒露光で3バンド撮影し、1バンドあたり30枚程度重ねる場合

旧 ⇛ 処理が撮像に追いつかない

新 ⇛ 撮像の度に10回以上処理を実行可能

7. まとめ

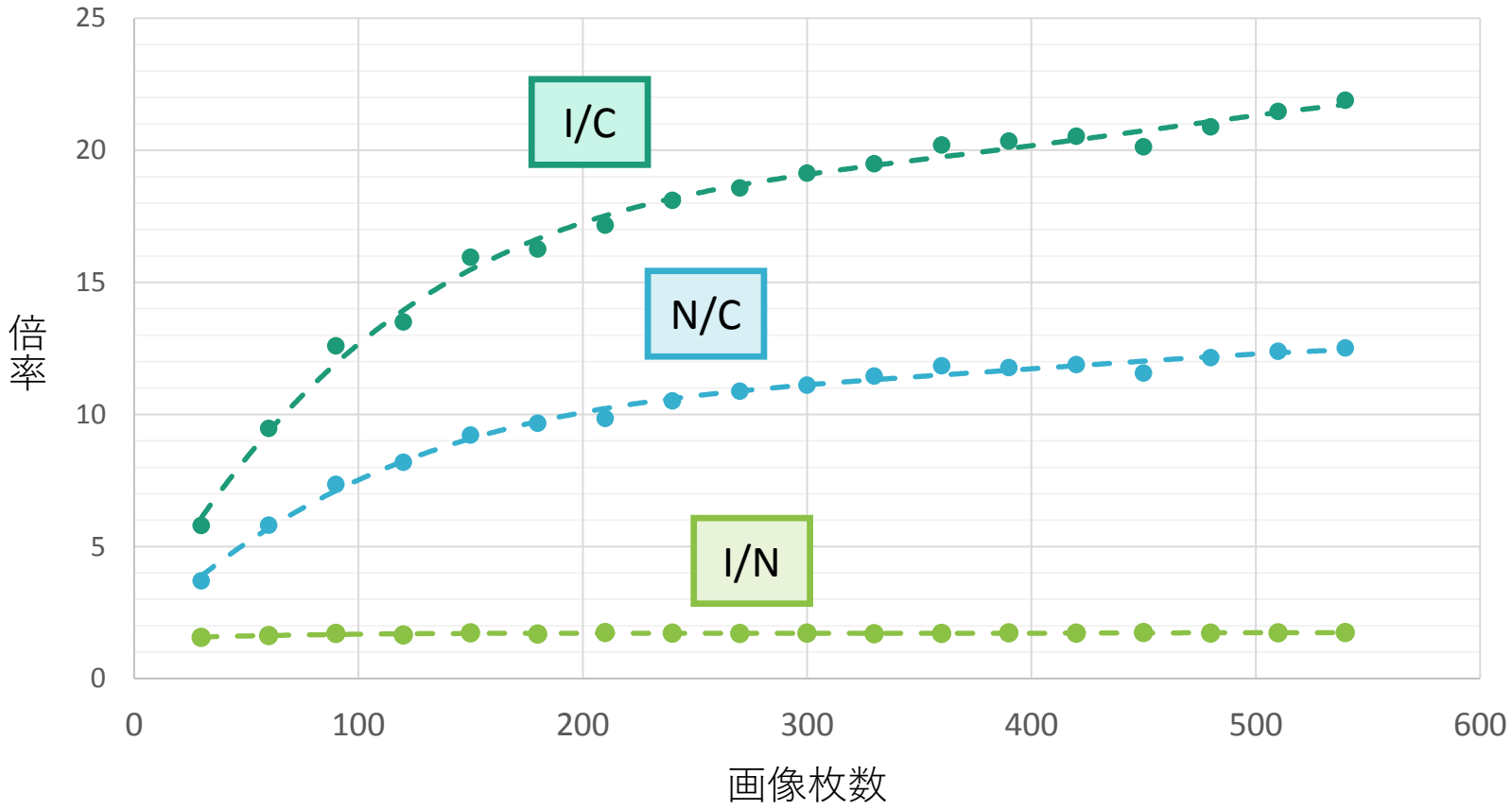
- MITSuME用自動解析パイプラインの実行時間を調べ、ファイルI/Oに時間がかかっていることを示唆する結果を得た。

- ファイルI/Oの省略に加え、GPUを取り入れることで、旧版と同様の処理の精度を保ちつつ、10倍以上の高速化を実現した。

付録

付録

I : IRAF版
N : NumPy版
C : CuPy版



付録

- 使用した計算機：crasan (河合研の解析用ワークステーション)

CPU: Intel Core i7-7700 (core:4 , thread:8 , clock:3.6GHz)

DRAM: DDR4 8GB×4

ストレージ(システム): WD20EFRX 2TB

OS: Ubuntu 16.04.2 LTS 64bit

- 使用したデータ：山梨の18/5/14の画像(全2016枚)

object	枚数(全体)	枚数(good)
DARK	261	
1ES0647+250	108	105
GRB180514	864	645
OJ49	135	132
MAXI J1820+070	648	303

付録

- 使用した計算機 : qubeley (河合研のGPGPU用ワークステーション)

CPU : Intel Xeon E5-1650 v4 (core:6 , thread:12 , clock:3.6GHz)

DRAM : 8GB×8

ストレージ

boot : Intel SSDSC2BB12 120GB

data : Seagate ST2000VN0001 2TB

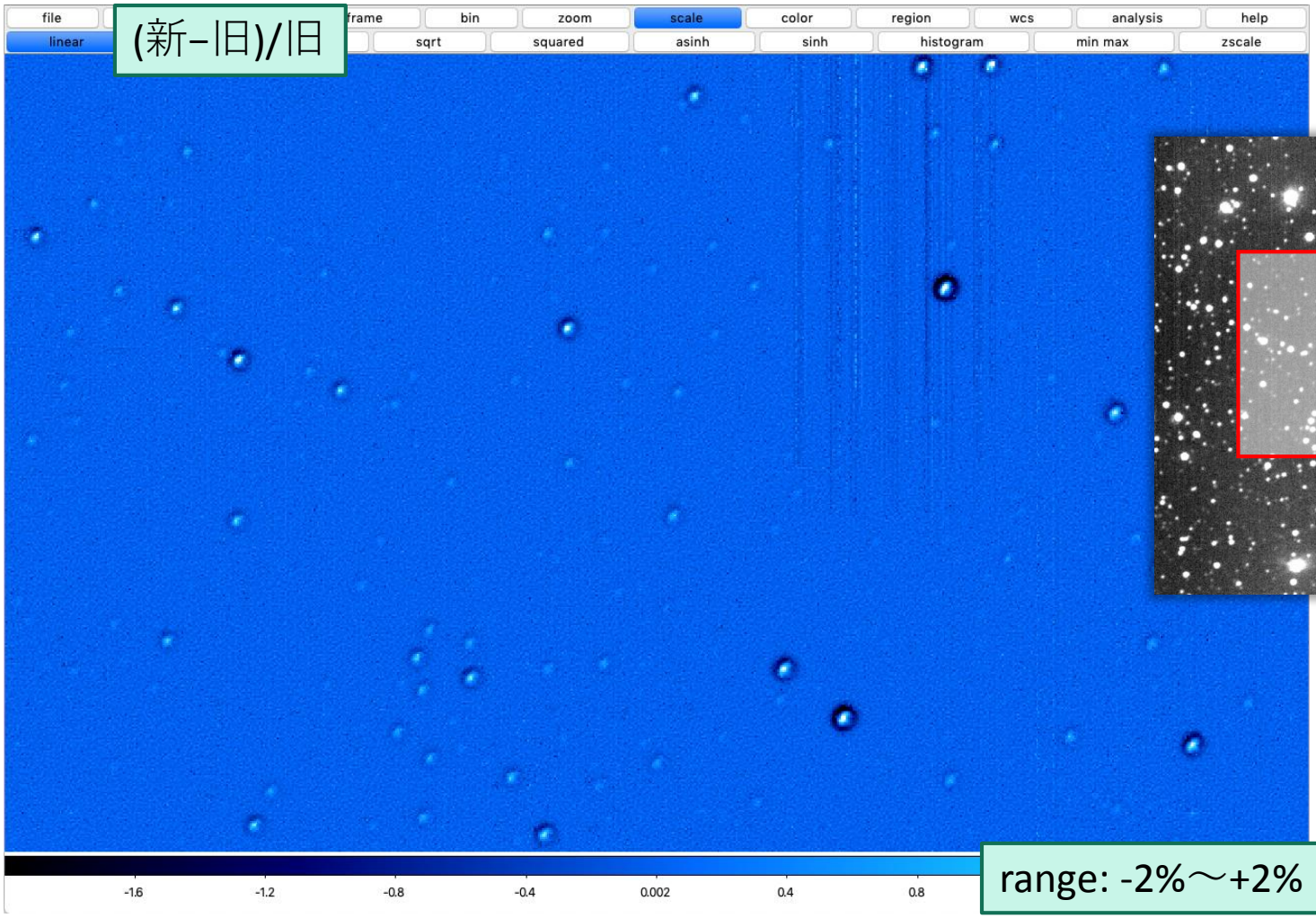
グラフィック

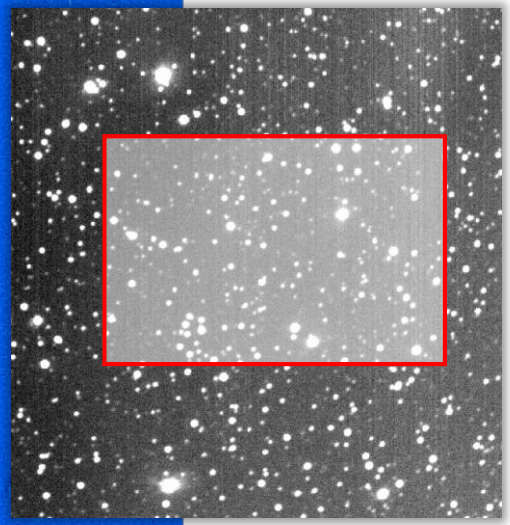
GPU : NVIDIA TITAN X

VRAM : GDDR5X 12GB

OS: Ubuntu 14.04.5 LTS 64bit　CUDA:v8.0.61　Chainer:v1.17.0

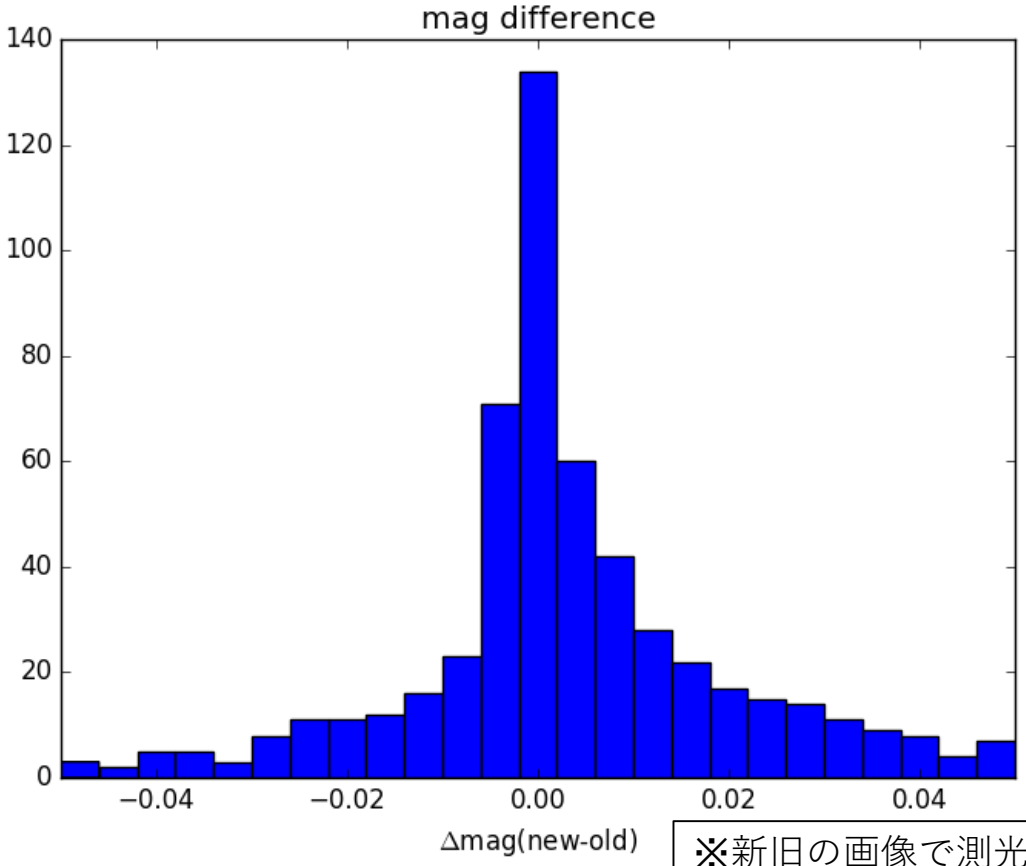
6. 性能評価
出力の差分 (1ES0647+250)





range: -2%～+2%

6. 性能評価
等級の比較 (1ES0647+250)



※新旧の画像で測光をして
同じ星で等級の差を求めて
ヒストグラムにしたもの