

# 「starflattener2」仕様書

## 1. 背景

光害のある街中でも星の写真を撮りたい。そんな気持ちで starflattener を作って早 5 年が経過しました。その間に AI ブームが到来し、今や星の写真の加工にも AI が使われるようになり、宇宙望遠鏡で撮影したような美しい写真が雑誌を飾るようにもなりました。SNS でも連日、美しい星の写真を見ることができます。画像処理技術も高度化して専用ツールに組み込まれ、技術を手にした人なら誰でもトップレベルの星の写真に仕上げることができます。画像処理講座や指南書の紹介も頻繁に目にするようになりました。相変わらず曇りや雨が多い日本ではなく、世界各地の晴天率の高い地域に設置されたりリモート観測所を使って、寝ている間に星の写真を撮影できるようにもなりました。

一方で、ハレー彗星の写真が撮りたくて始めた星の写真は、私の中には趣味としてしっかり定着していて、時々くる天体イベントの度に「撮りたいな」「晴れるといいな」と思いを寄せています。幸運にも撮影機会と晴天に恵まれた際には、できるだけキレイな写真に仕上げたいと思う気持ちは変わりません。撮影した写真を加工して、年に 1 回 PC やスマホの壁紙を更新するのが楽しみになっています。撮影時の記憶が残るように少し風景を織り交ぜてみたり…。昔ながらのスナップ写真のような、星の写真もそんな存在になれるような気がします。

starflattener2 には風景と星空の両方をバランスよくキレイにするために、いくつかの機能を追加しました。トップレベルの機能はありませんが、効果的な設定を厳選して実装しました。ソースコードは使い勝手の良い python に変更してオープンソースにしました。腕に覚えのある方は、ご自身で改造してキレイな星の写真に磨きをかけてください。私自身もいくつか派生版を作って、用途に応じて使い分けています。CUDA が使える PC 用など…。もちろん、python 知らねえという人のために exe ファイルも用意してあります。

いつもキレイな星の写真と共に居られるよう、お役に立てると幸いです。

## 2. 「starflattener2」の概要

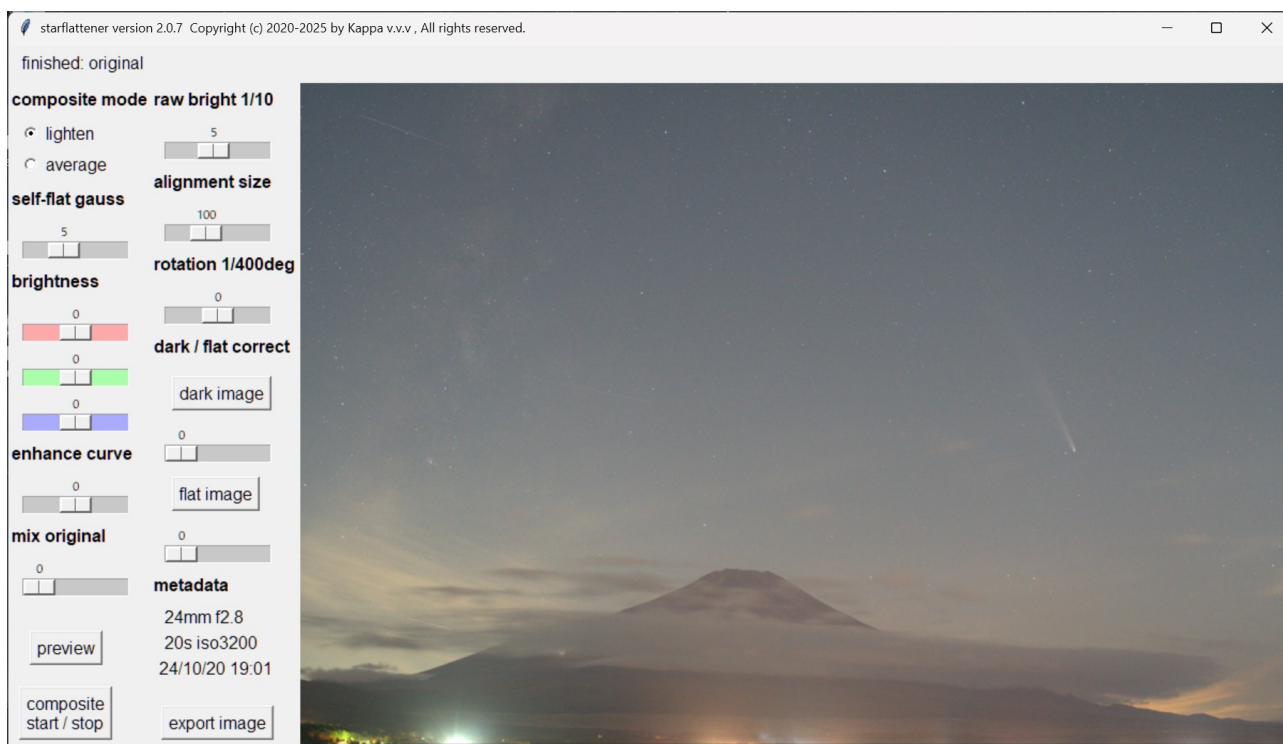
「starflattener2」は、星の写真をキレイに加工処理するためのソフトです。私の好みは、どちらかと言うと星空がメインで風景が写りこんだとしても少しの写真です。気持的には星野写真なのですが、最近の分類では星景写真に入るのかも知れません。画角は広角から中望遠くらいが多く、近所の風景や木などがどうしても写りこんでしまいます。低空ほど空が明るいので星の写りが悪いのですが、出来るだけフラットに真っ暗な背景の星空にしたいと思っています。そんな星の写真をキレイに加工するにはどうするか...を、ツールにしてみました。

「starflattener2」の機能を紹介します。目玉は何とんでも GUI 搭載です。デスクトップなどにショートカットを作成し、画像をドラッグ&ドロップすれば GUI が起動します。python 環境を構築すれば Mac でも動作しますが、ドラッグ&ドロップは Windows だけの特権のようです。他にも次の機能があります。

- (1) コンポジット機能 : 複数の写真を重ねてノイズ低減します。  
比較明合成と加算平均に対応しています。
- (2) アライメント機能 : 複数の写真の星や月の位置を合わせます。
- (3) セルフフラット機能 : 背景の明るさを均等にします。
- (4) RGB 調整 : 各色の明るさを調整します。
- (5) トーンカーブ調整 : 背景を暗くしてコントラストを上げます。
- (6) 元画像の合成 : 処理済み画像と元画像を加算平均します。
- (7) RAW 現像処理 : RAW 現像後に画像処理をします。
- (8) 画像回転機能 : 複数画像を回転させながらコンポジットします。
- (8) ダーク減算機能 : ダーク減算でピクセルノイズを低減します。
- (9) 光学フラット補正 : 光学フラット補正で周辺減光を低減します。

次章では、「starflattener2」起動のセットアップ方法と、その操作方法について説明します。

### 3. 「starflattener2」起動のセットアップ

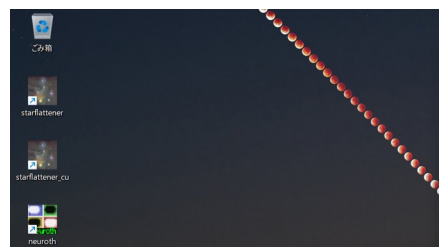


「starflattener2」から GUI 起動をサポートしました。画面を見ながら細かな調整ができるので色調を追い込みやすくなりました。

#### 3.1. GUI 起動のセットアップ

ここでは Windows でのセットアップの一例をご紹介します。様々なセットアップ方法があるとは思いますが、私が一番使いやすいと思っている方法です。

- (1) starflattener.exe のショートカットをデスクトップに作成します。  
アイコンを摘んでデスクトップまで持っていき、alt キーを押しながら離します。
- (2) ショートカットの名前をクリックして、好みの名前に変更します。
- (3) 処理したい画像ファイルを束ねてショートカットの上にドラッグ&ドロップします。10～15 秒くらいすると GUI 画面が起動します。



starflattener2 は python で作成していて、exe ファイルは pyinstaller で作成しています。pyinstaller で作成した exe ファイルは、python 環境を展開してから python ソースコードを実行するので、起動までに 10~15 秒の待ち時間があります。これに耐えられない人やソースコードを修正して使用したい人は、お使いの python から starflattener.py を直接起動するようなバッチファイルを作成して、同様のショートカットを作成することも可能です。

--- starflattener\_custom.bat --- の例

```
@echo off  
C:\Users\kappa\miniforge3\python starflattener.py %*
```

## 3.2. コマンドライン起動のセットアップ

以前の starflattener と同じですが、実行パスの通ったフォルダに exe ファイルを置いて、コマンドプロンプトなどから起動します。

(1) 実行パスの通ったフォルダが無いときは作成します。私は、Documents¥bin を好んで使っています。starflattener.exe を置きます。

(2) Windows の設定画面の検索から「環境変数を編集」を起動し、ユーザー環境変数欄の「Path」変数を編集して、元々記載されている内容は変更せずに、%USERPROFILE%¥Documents¥bin を追加します。

(3) コマンドプロンプトを開いて、starflattener.exe を起動します。

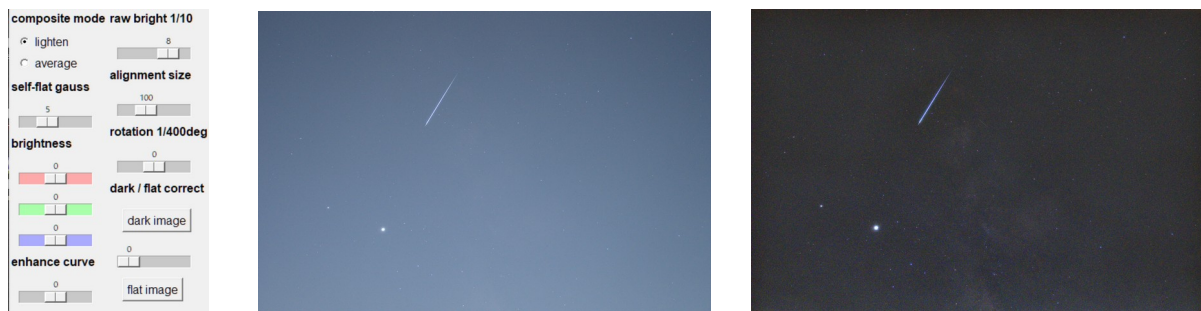
ただし、-vsize 以外の引数を何も指定せずに、画像ファイルだけを指定して起動した場合は GUI モードで起動するようになっています。-vsize は mp4 動画サイズを指定するオプションなので、GUI モードでも使えるようにしました。

次の章では、いくつかのシーンに応じた starflattener の便利な使い方をご紹介します。ご自身の画像ファイルを用意して、チュートリアル的にお試しください。

## 4. シーン別の使用方法（例）

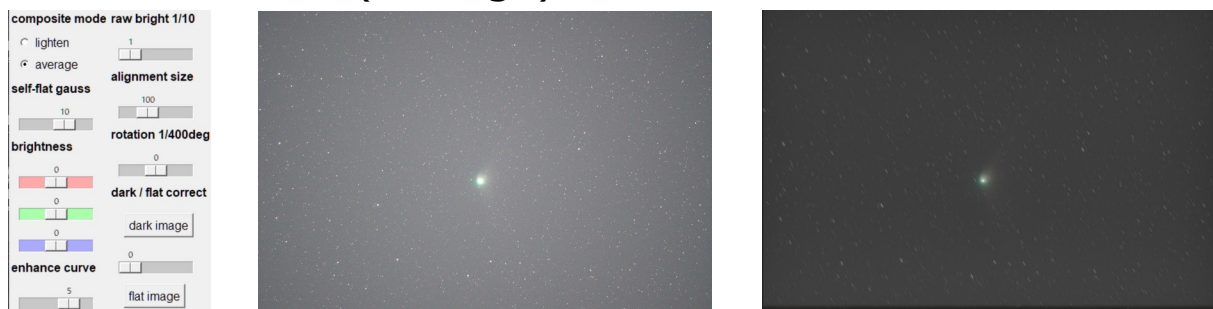
### 4.1. 星の写真を1枚だけキレイにしたい。

たくさん撮ったうちの1枚に流星が写っていたり、スマホや SeeStar で撮った写真を仕上げたりなど、最近はこの手のシーンが増えてきた気がします。1枚処理の際には、比較明合成(lighten)モードにするとノイズ低減処理がかかってキレイになります。RAW ファイルを入力した際には raw bright でコントラスト調整ができます。必要に応じて他のパラメータを調整し、preview ボタンで画像を確認します。最後に export ボタンで画像を保存します。



### 4.2. ガイド撮影した彗星をキレイにしたい。

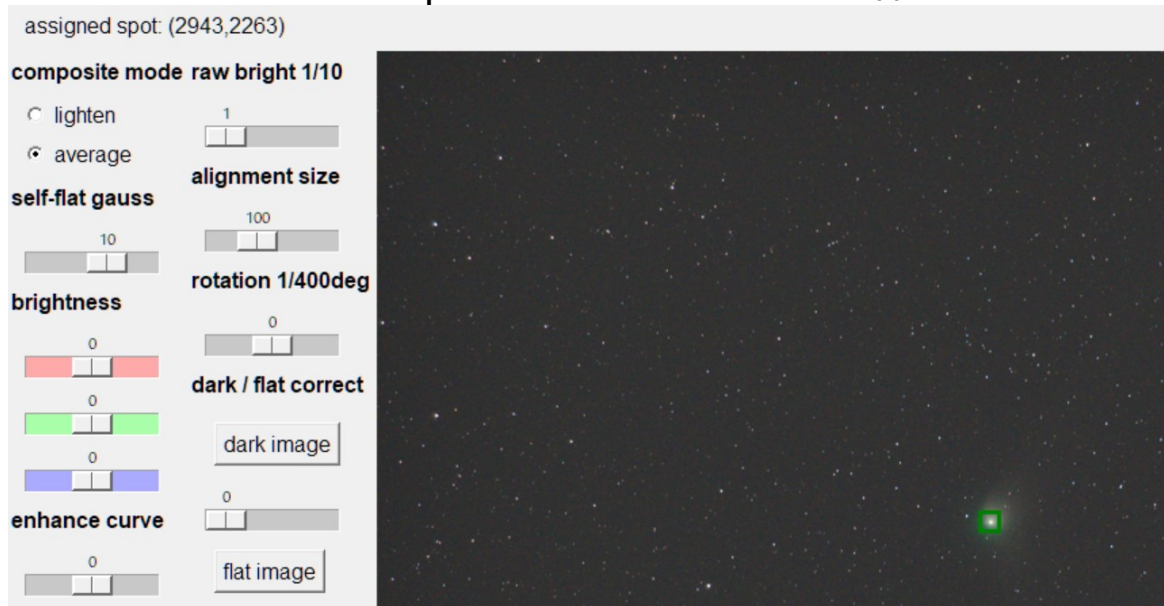
2024 年に接近した紫金山アトラス彗星は、肉眼彗星となって多くの人が長く伸びた尾を眺めました。そこまで明るくなければ、これまでも数年に一度は写真に尾が写る彗星が接近しています。彗星は恒星と違う速度で移動するので、彗星でアライメントするメトカーフコンポジットをします。ノイズを低減して尾を抽出するために加算平均(average)モードを使用します。



彗星の尾は淡いので 10 枚以上のコンポジットがお勧めです。raw bright でコントラストを下げて明るいコマと淡い尾を共存させ、self-flat 処理サイズを 10 前後に大きくします。一度 preview で画面を更新した後、画面上のコマをクリックしてアライメント領域(緑の枠)を彗星のコマに合わせます。この時

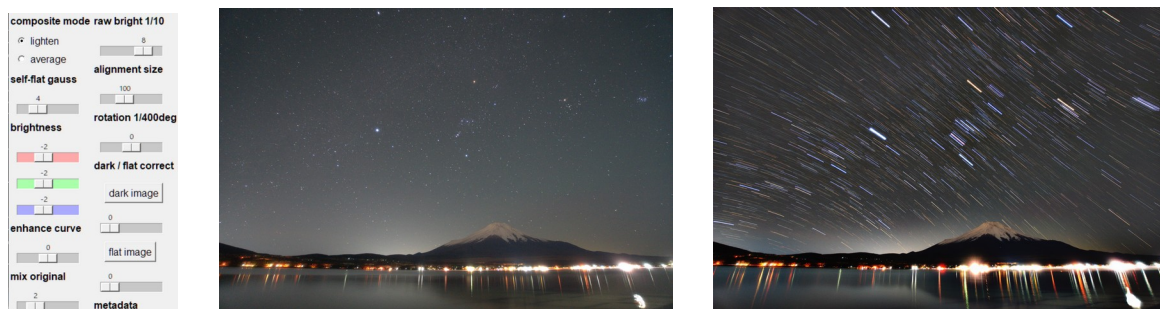


alignment size で領域サイズを変更できます。アライメント領域が定まったら composite ボタンを押すとコンポジットが始まります。コンポジットの度に画面が更新されます。途中で中止したいときは再度 composite ボタンを押します。コンポジットが終了したら、export ボタンを押して画像を保存します。



#### 4.3. 明るい景色と星空を共存させてキレイにしたい。

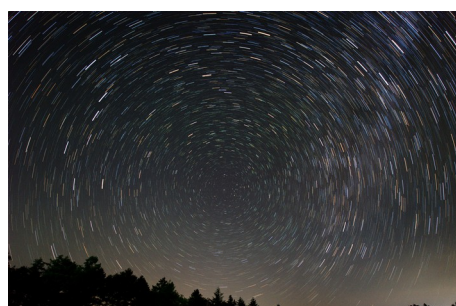
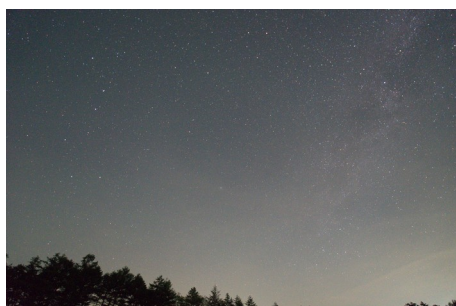
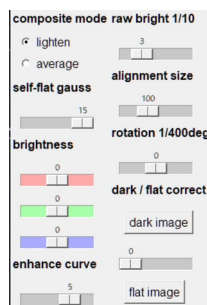
景色の明るさによってコツがあるのでご紹介します。まずは星空よりも景色が明るい場合です。景色を暗く、星空を明るく処理するために、self-flat 処理サイズを 10 以下に設定します。景色と星空の境界のグラデーション(オーラ)が目立たないように調整します。mix original で元画像を少し混ぜても目立たなくなる効果があります。RAW ファイル入力の場合は、raw bright を少し上げると、元画像を混ぜた時に濃淡が強調されます。



#### 4.4. 暗い景色と星空を共存させてキレイにしたい。

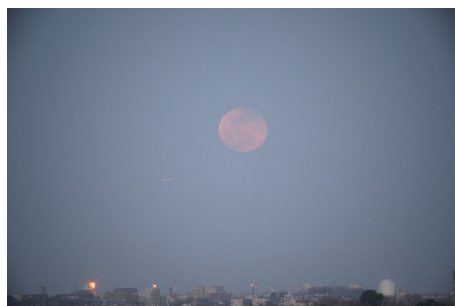
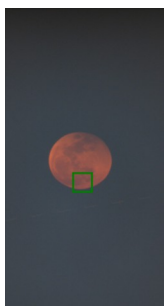
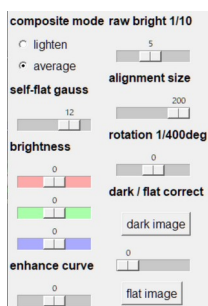
つぎに星空より景色が暗い場合です。星空を暗く処理するのですが、景色を明るくしてしまうとノイズが目立ってキレイに仕上がりにません。self-flat 処理サイ

ズを 11 以上にすると、比較的暗い部分を処理対象外にします。暗い景色が明るくなるのを防止できますが、星空の暗い部分にも影響するので全体的にフラット処理が弱くなり光害が残ります。RAW ファイル入力の場合は raw bright でコントラストを下げると、光害の傾斜が目立たなくなります。もし星の輝きが弱いと感じたら、enhance curve を大きくすると多少改善します。



#### 4.5. 月と景色を共存させてキレイにしたい。

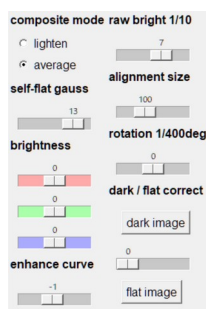
月は周囲に比べて明るいので、self-flat サイズを大きくして月の周りだけ空が暗くなるのを防ぎます。11 以上にすると弱い処理になるのですが、11～15 の間は滑らかに強度が変化します。今回の場合は 12 で全体の雰囲気が良くなりました。alignment size を調整して、box の中に月の海と縁が両方入るようにすると、アライメントでうまく重なります。



#### 4.6. 低空がガスっていた写真をキレイにしたい。

複数枚数を撮影していて低空のガスが動いている場合は、加算平均 (average) モードでガスを目立たなくすることができます。低空の場合は景色が写り込んでいることが多いので、self-flat サイズを 11～15 の間で雰囲気の良い設定を見つけます。RAW 入力の場合は raw bright で星空のコントラストのグラデーションを調整します。調整でなんとかなることがあるので、撮影条件が悪くても諦めずに複数枚撮影しておきましょう…。

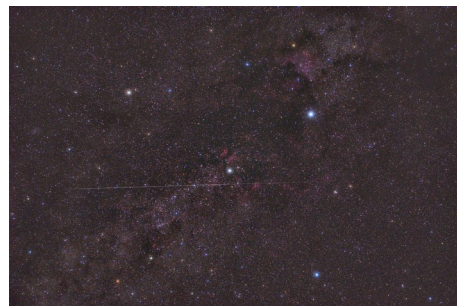
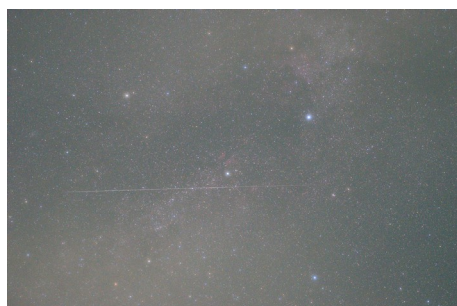
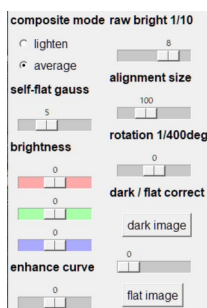




#### 4.7. ガイド撮影した 1 枚に流星が写っていたのでキレイにしたい。

流星が写ったコマの後ろもしくは前に 10 枚以上のガイド撮影した写真がある場合、それらを加算平均(average)モードでコンポジット後に、流星が写ったコマと比較明合成(lighten)モードで再コンポジットすると星空の流星がキレイに仕上がります。

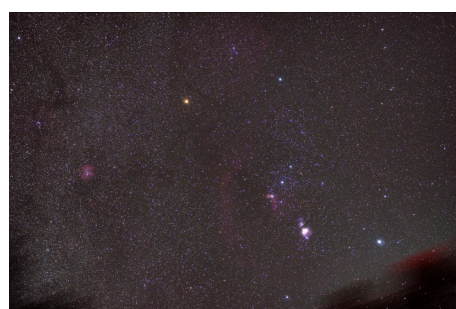
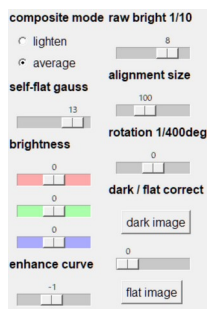
この際のポイントとしては、コンポジットする画像と流星が写ったコマの画像を同じ設定で保存することです。比較明合成の際、ほぼ同じ明るさのピクセルは加算平均処理しているので、2 回目のコンポジットの際に流星が写ったコマのノイズが低減されてキレイになります。



#### 4.8. 暗黒星雲を浮かび上がらせてキレイにしたい。

暗い空で 30 枚以上のガイド撮影をした RAW ファイルを用いる前提ですが、セルフフラット処理が威力を発揮します。raw bright を大きめにしてコントラストをつけて、明るくなった画像を enhance curve で適度な暗さに調整して加算平均(average)モードでコンポジットします。100 枚以上あるとかなりキレイに仕上がります。画像を保存後に、他の画処理ソフトでトーンカーブ調整などすると更にキレイに仕上がります。





#### 4.9. タイムラプス動画をキレイに作成したい。

starflattener の処理中に裏で、コンポジットに使用する画像を使って mp4 動画を作成しています。export ボタンで画像出力する際に Mpeg4 フォーマットを選ぶと、作成した mp4 動画を出力します。

固定撮影した複数枚の写真を入力した場合、比較明合成(lighten)モードのときは星の軌跡が伸びてゆく様子が、加算平均(average)モードのときは星が日周運動する様子が、それぞれ動画として作成されています。10コマで1秒の動画を作りますので、100枚で10秒間の動画、600枚で1分の動画を作成できます。

starflattener 起動時に-vsize オプションを指定すると、作成する動画サイズを default から変更することができます。default は 1080x1080 です。動画の例はドキュメントではお伝えできないので、ぜひご自身でお試ください。

## 5. 主要機能の仕様

### 5.1. コンポジット処理 (composite mode)

複数画像を重ね合わせて新たな画像を作成する処理です。比較明合成モード (lighten) モードか、加算平均 (average) モードを選択できます。

#### (1) 比較明合成モード

複数画像を重ね合わせる際、RGB 各々ピクセル毎の最大値を保存します。ただし、その時点までの最大値と±3%以内の値のときは両者の平均値を保存します。

また、重ね合わせる前処理としてノイズ低減処理を行います。隣接する4ピクセル毎の最小値を新たな値として置き換えています。

#### (2) 加算平均モード

複数画像を重ね合わせる際、RGB 各々ピクセル毎の平均値を保存します。

### 5.2. セルフフラット処理 (self-flat gauss)

画像毎に各々 RGB を gaussian フィルター処理してセルフフラット画像を作成し、元画像から減算します。パラメーターとして gaussian フィルターのサイズを指定できます。フィルターサイズは1が最も小さく、値が大きくなるほど大きくなります。0を指定すると gaussian フィルターの代わりに各々 RGB の平均値をセルフフラット画像とします。15以上を指定した際は、平均値以下の値を平均値に置き換えた後にフィルター処理を行います。10~14の間は置き換える値を0~平均値まで段階的に変更します。

### 5.3. RGB 明度調整 (brightness)

入力画像の RGB 値に係数をかけて調整します。係数  $r$  は-1~1の間の値を指定できます。係数に応じて次の式で明るさを変更しています。

$$y = 1 - (1 - x) ** (1 / (1 - r / 3))$$

## 5.4. トーンカーブ調整 (enhance curve)

入力画像のトーンカーブを調整します。セルフフラット処理と同時にトーンカーブ調整しています。本ツールではセルフフラット処理を減算で実現しているので、補正のためにトーンカーブ調整を行います。本機能は、その際のパラメータを利用したトーンカーブ調整です。ソースコードは次の通りです。array\_org は元画像、array\_flat は元画像から生成したフラット画像です。

```
def fsigmo( x, y, z ):  
    sigmo= (0.5+ x * z) / (0.5+ z + np.exp( -10 * (x - y) ))  
    return sigmo  
  
def fdiff( b, f, r_dark, r_light ):  
    if r_light >= 0:  
        y= r_dark  
        z= 1- r_light  
    else:  
        y= r_dark - 0.5* r_light  
        z= 1  
    r= fsigmo( (b - f) / (1- f), y, z );  
    return r  
  
array_out= fdiff( array_org, array_flat, 0.01, e )
```

## 5.5. 元画像の合成処理 (mix original)

セルフフラット処理後の画像はコントラストがやや強めに仕上がります。シーンによっては、元画像と平均化することで全体のバランスを整えることができるので、本機能を追加しました。なお、合成比率を最大にすると元画像がそのまま使用しますので、元画像のままコンポジット処理や動画作成することもできます。

## 5.6. RAW 現像明度調整 (raw bright 1/10)

本ツールはRAW ファイルを入力して現像後に画像処理する機能があります。現像処理は 16bit 出力で行っていますが、明度調整が必要なのでその値 (bright)を指定します。現像処理には次のコードを使用しています。

```
rawpy.postprocess( output_bps= 16, use_camera_wb= True, bright= bright )
```

## 5.7. アライメント BOX サイズ指定 (alignment size)

コンポジット処理の際にアライメント BOX を指定すると、その枠内の画像でアライメントを取りながら重ね合わせます。アライメント BOX の座標は、GUI モード時は画像をマウスでクリックして、コマンドラインモード時は XY 座標値で指定します。その時のアライメント BOX サイズも指定することができます。

## 5.8. 回転コンポジット機能 (rotation 1/4000deg)

コンポジット処理の際に回転角度を指定すると、指定した角度ずつ画像を回転しながら重ね合わせます。固定撮影で連続撮影したり、経緯台でガイド撮影した場合に、撮影対象の方角に応じて写野が回転するのを補正しながらコンポジットすることができます。回転角度は方角によって全く異なるので、最初は 10 コマ飛ばしで角度を確認して、本番では 1/10 の角度を指定してコンポジットすると早く調整できます。

## 5.9. ダーク/フラット補正処理 (dark/flat correct)

ダーク補正処理は、入力画像とダーク画像の同じ座標で±10%以内の明るさのピクセルを、画像の gaussian フィルター処理後のピクセル値に置き換えます。ダーク画像はあらかじめ複数枚を average モードで加算平均しておくときれいに仕上がります。

フラット処理は、入力画像からフラット画像を引き算します。処理方法はセルフフラット処理とほぼ同じ計算式を使用しています。フラット画像もあらかじめ複数枚を average モードで加算平均しておくときれいに仕上がります。

## 5.10. メタデータ表示 (metadata)

1 枚目に指定した入力画像の、焦点距離・絞り・シャッター速度・ISO 感度・日時を表示します。またこのデータを出力画像のメタデータとして書き込みます。



### 5.11. 処理制御ボタン (preview/composite/export image)

preview ボタンは、調整したパラメータを画像表示画面に反映させます。

composite start/stop ボタンは、一度押すごとにコンポジット処理を開始/停止します。stop ボタンを押した瞬間に行っていた処理は完了してから停止します。

export image ボタンは、画像表示画面の画像をファイルに出力します。出力時に指定できるフォーマットは、Tiff/Jpeg/Mpeg4 の 3 種類です。

Mpeg4 の時は現在表示されている画像ができるまでに生成した画像を使って、1 コマずつ 10fps で動画に処理します。lighten モードの時は星の軌跡が伸びていく動画に、average モードの時は星が動いていく動画になります。

### 5.12. 画像表示画面 (画像/ヒストグラム)

GUI モードで起動した際には 1 枚目の入力画像がそのまま表示されています。preview か composite を押すと処理後の画像に更新されます。また、コンポジット中はその時点での処理済み画像を表示しています。もしコンポジット処理の途中で最適と思えるシーンが表示された場合は、stop ボタンで停止して export することもできます。

ウィンドウを縦に延ばすとヒストグラムが現れます。RGB 明度調整の際の参考にしてください。

## 6. コマンドラインモードの仕様

### 6.1. コマンドライン引数

コマンドライン引数として次のものを指定できます。主要機能に書いた機能と名称を合わせましたので、あらためて説明しなくても概ね分かると思います。指定する値は範囲チェックをしませんので、GUI モードのスライダーでは指定できない値を指定することができます。

-vsize 以外のオプション引数を指定して画像ファイルを引数指定したときに、コマンドラインモードで起動します。オプション引数を一切指定しないか、指定したオプション引数が-vsize だけで画像ファイルを引数指定したときは GUI モードで起動します。

```
usage: starflattener [options] img1 [...imgN]
      to execute w/ gui mode if no option is assigned.
option: -out <image_file>      : output tif/jpg/mp4 filename. (* output.tif)
      -lighten                  * : composite w/ lighten mode.
      -average                  : composite w/ average mode.
      -gauss <D>                : gaussian ratio for self flatten.
      -bright <R> <G> <B>      : brightness control for R/G/B.
      -enhance <R>              : tone curve enhancement ratio.
      -mix_org <R>              : mixing ratio with original image.
      -raw_bright <B>           : brightness in raw processing.
      -align_size <S>           : size of alignment box.
      -align_pos <X> <Y>        : alignment at position (X,Y).
      -rotate <R>               : rotation image R deg / n.
      -dark <dark_file>         : input dark image filename.
      -dark_ratio <R>           : ratio to apply dark correction.
      -flat <flat_file>         : input optical flat image filename.
      -flat_ratio <R>           : ratio to apply optical flat correction.
      -vsize <W> <H>           : set video size when output mp4.
      * : default setting
```

### 6.2. メッセージ表示

コマンドラインモードで実行すると、処理の経過がわかるようにコマンドプロンプト上に次のようなメッセージを表示します。

同じ内容は、GUI モードで export image ボタンを押した際に、指定した画像ファイルと同じ名前で拡張子が.txt のファイルにも出力されます。

```
starflattener (python edition) version 2.0.7
      Copyright (c) 2020-2025 by Kappa v.v.v , All rights reserved.
- CUDA mode is selected. to comment cupy and cupyx if unselect.
- image files...
```

```
1 test1\_DSC0313.ARW
- metadata...
focal-length F-num : 24mm f2.8
exposure-time ISO  : 20s iso3200
date & time       : 24/10/20 19:01
- information...
#image used : 0
date & time  : 2025:12:18 22:45:17
run time    : 4.4s
```

## 6.3. バッチファイルで処理する方法

多くのファイルを処理したい時は、バッチファイルを使用すると便利です。例えば複数ファイルを1枚ずつセルフフラット処理したい場合は、次のようにバッチファイルを書きます。

```
--- exec.bat ---
for %%f in ( input\*.arw ) do ^
starflattener -average -out %%~nf.tif %%f
```

## 6.4. python で処理する方法

バッチファイルの代わりにpythonで処理すると、複雑なバッチ処理も可能です。例えば3枚毎に加算平均したい場合は、次のようなpythonファイルを書いて実行します。

```
--- exec.py ---
import os
import subprocess
if __name__ == "__main__":
    arw= os.listdir( "input" )
    for i in range( 0, len( arw ), 3 ):
        arw3= arw[i:i+3]
        if len( arw3 ) < 3:
            break
        print( "[%03d] %s %s %s"%( i, *arw3 ) )
        patharw3= [ "input\%s"%( f ) for f in arw3 ]
        subprocess.run( [ "python", "starflattener.py", "-average",
            "-out", "test\image%03d.tif"%( i ) ] + patharw3 )
```

以上