Astro-Phot

Planétaire

TRAITEMENT DE LA LUNE

GAD – MEYLAN - 02 / 2023

Sébastien Valat

https://png.is/f/telescope-png-images-free-download/comdlpng6936190-201911182258.html

Disclamer

- Mon expérience se limite à **quelques jeux de lune** avec mon Reflex sur un trepied et avec la lunette du club !
- On peut pour sur faire mieux !
- Je présente avec une lunette, monture et camera
- On peut appliquer avec un simple **Reflex** + **Objectif** + **trepied** !

Fichiers utilisés

- http://progranet.ovh/shares/gad/2022-05-11-lune/
  - Login : **gad**
  - Pass : **2022**

- Prise : **7 mai 2022**

- Fichiers:

Fichier	Images
22_33_42	500
22_41_59	500
22_45_55	1000

Matériel

- Telescope: SkyWatcher Black Diamond 120 ED
- Mount: SkyWatcher NEQ6
- Camera: Atik Infinity
  (1392 x 1040)
- Capture: SharpCap

Le but en planétaire

- On veux **compenser** les **mouvements atmosphériques**
- On va prendre les images **le plus rapidement possible**
- Configuration de la prise :

```
#Format de sortie
Output Format=SER file (*.ser)(Auto)
#Exposition
Exposure=1,6412ms
```

Traitement logiciels
(sous windows)

- PIPP (**Planetary Imaging PreProcessor**)
  https://sites.google.com/site/astropipp/
- Autostakkert (**stacking avec correction des mouvements atmosphériques**)
  https://www.autostakkert.com/
- Astro Surface (**post traitement / déconvolution**)
  http://astrosurface.com/

Full pipeline

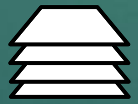
Fusion *.SER et **CROP**

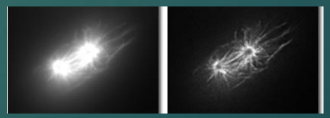
Stacking (turbulences atmosphériques)

Post-traitements (déconvolution…)

PIPP → Autostakkert → Astro surface

&





https://www.onlinewebfonts.com/icon/3917
https://seekicon.com/free-icon/crop-image_1
https://i.stack.imgur.com/xEv1Y.jpg

Pre-Processing : PIPP

FUSION & ALIGNEMENT & CROP

Fusion des .ser

On va aussi faire un « **CROP** » pour **accélérer** les traitements suivants

22_33_42.ser

22_33_42.ser

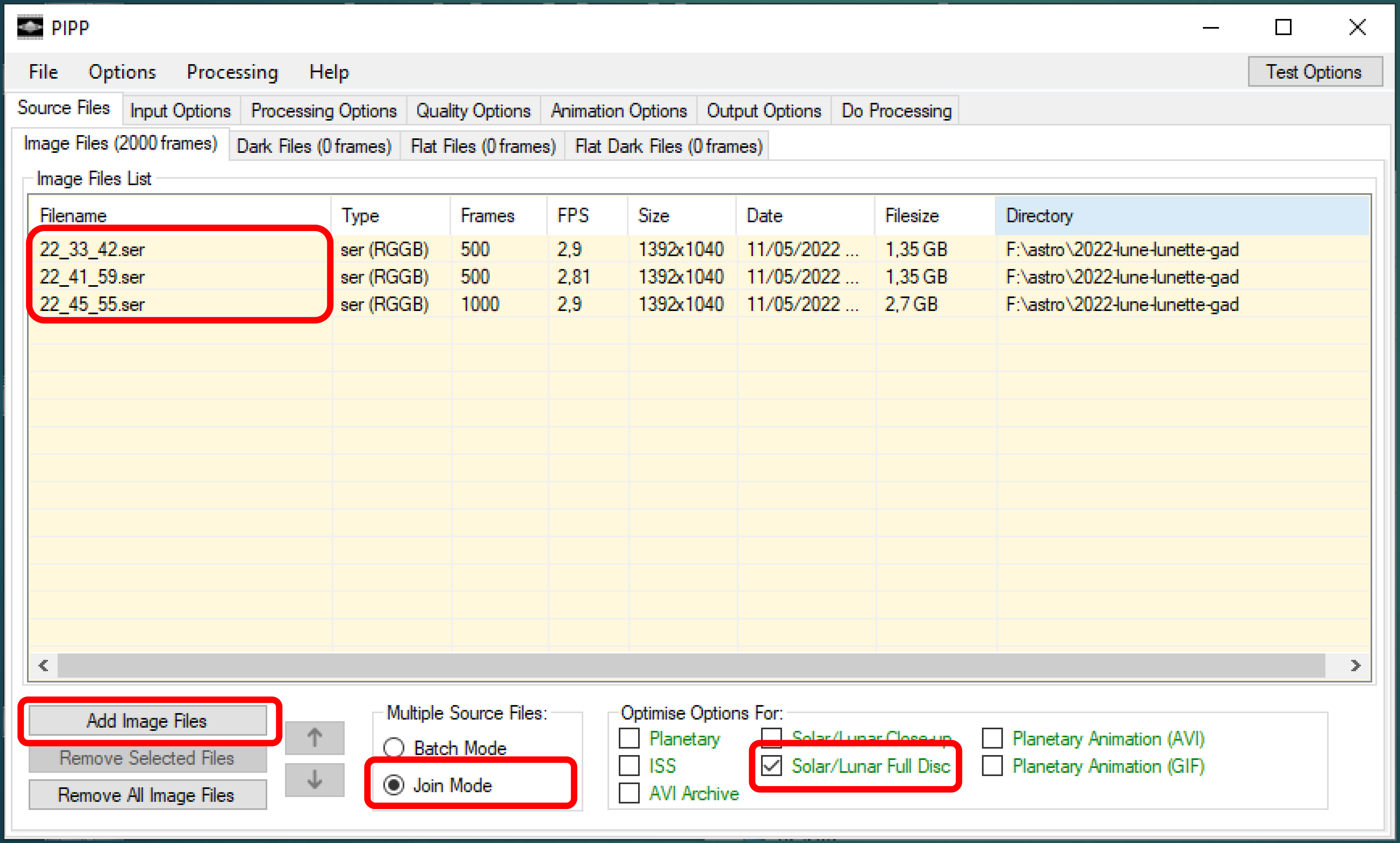
22_33_42.ser

PIPP

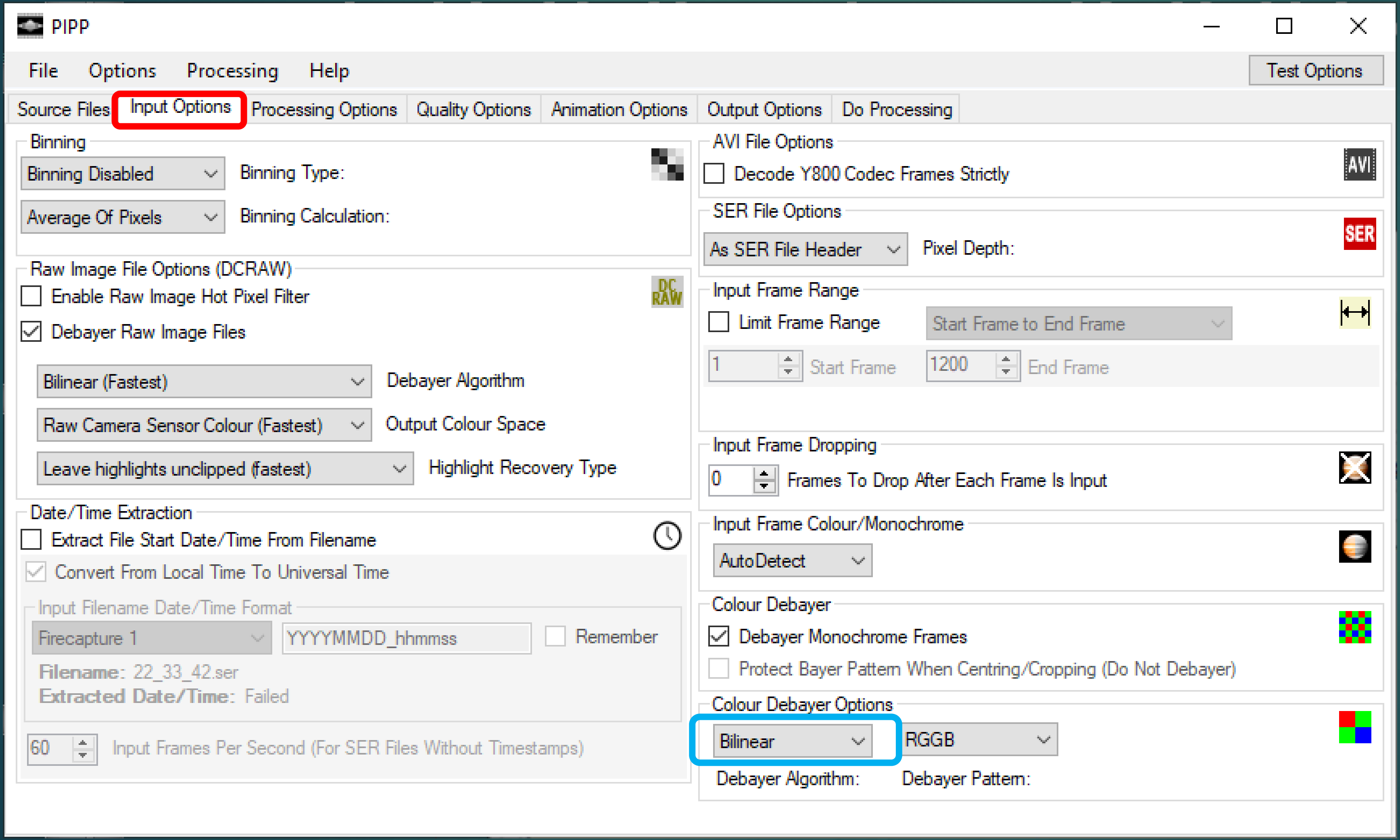
2023-05-07-lune.ser

+ **Debayer** + **N&B**

Etape 1 : Source Files



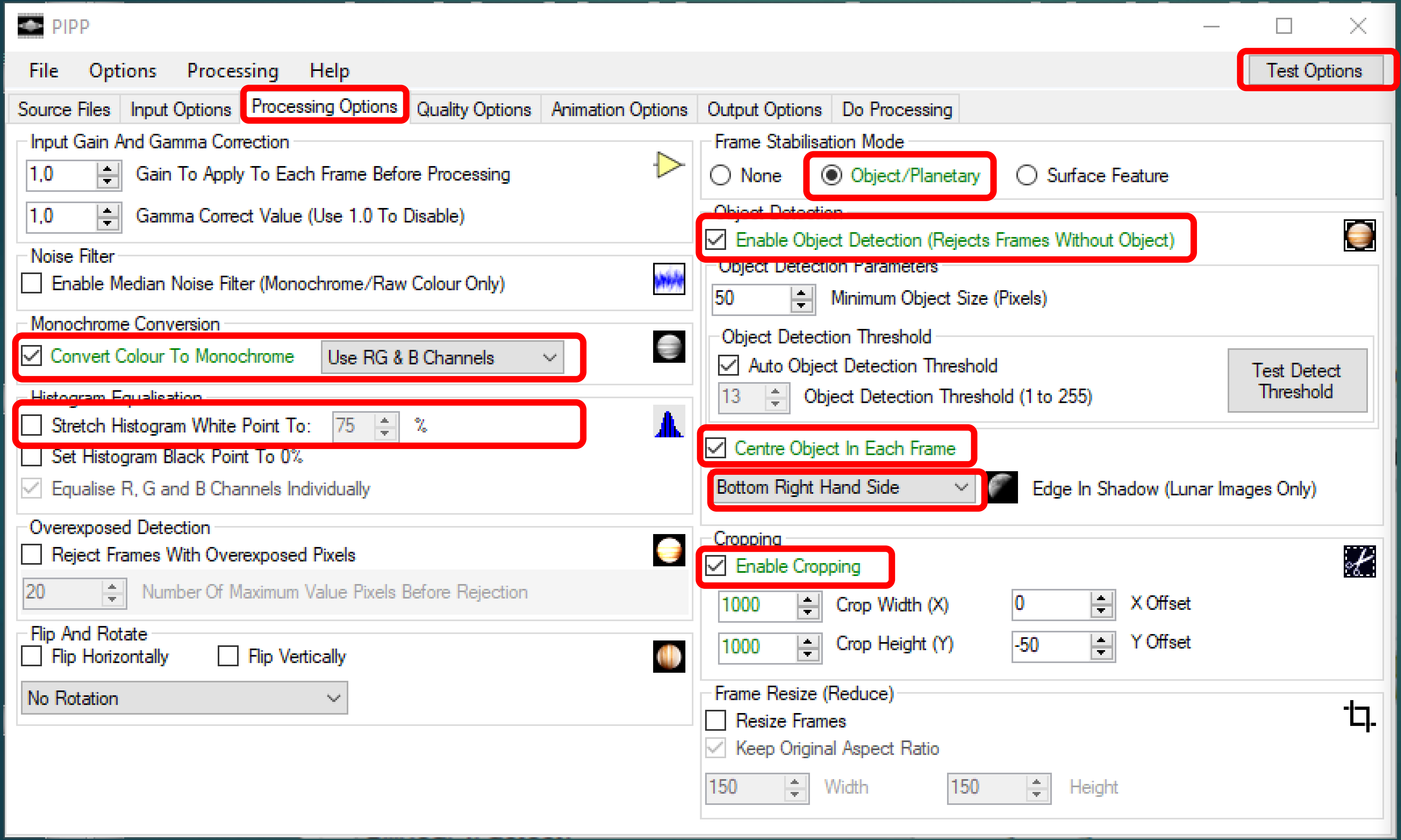
Etape 2 : Input Option

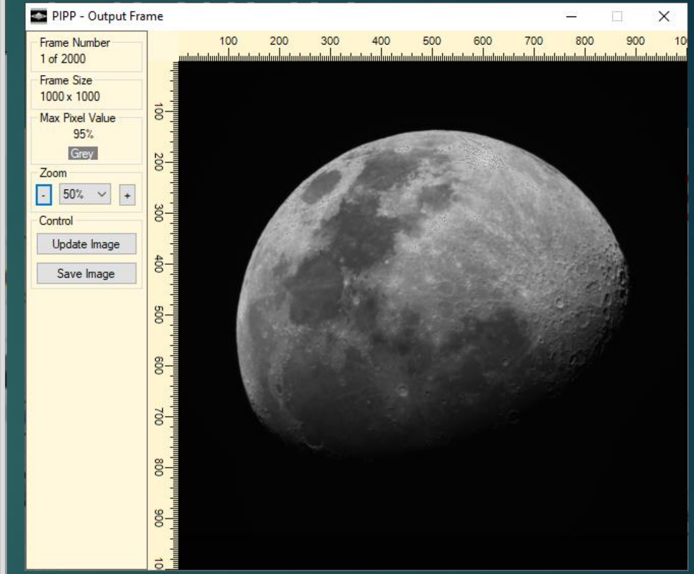


Pas de changement !

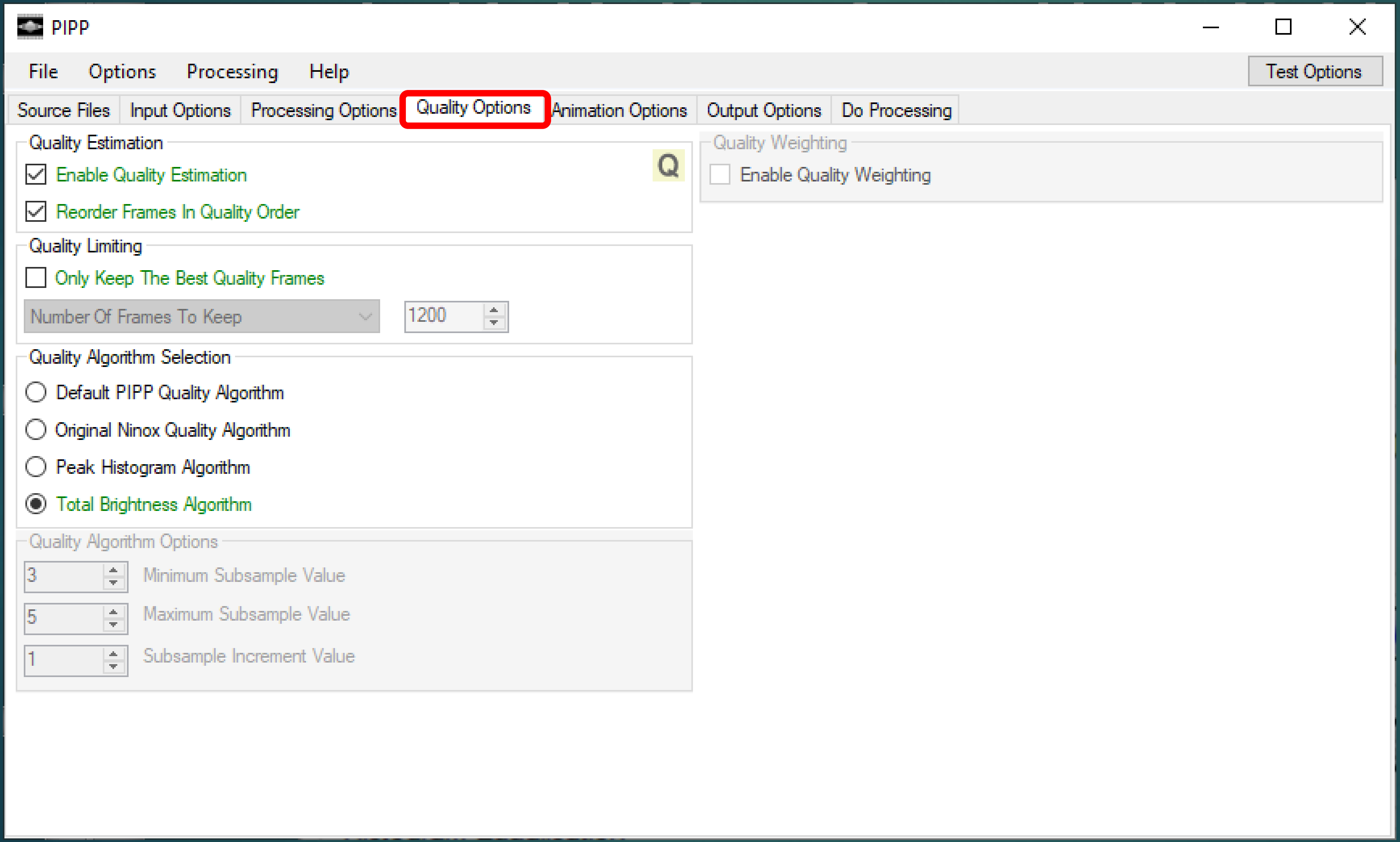
Eventuellement **Debayer HQ Linear** ou **Smooth Hue**

Etape 3 : Processing Options

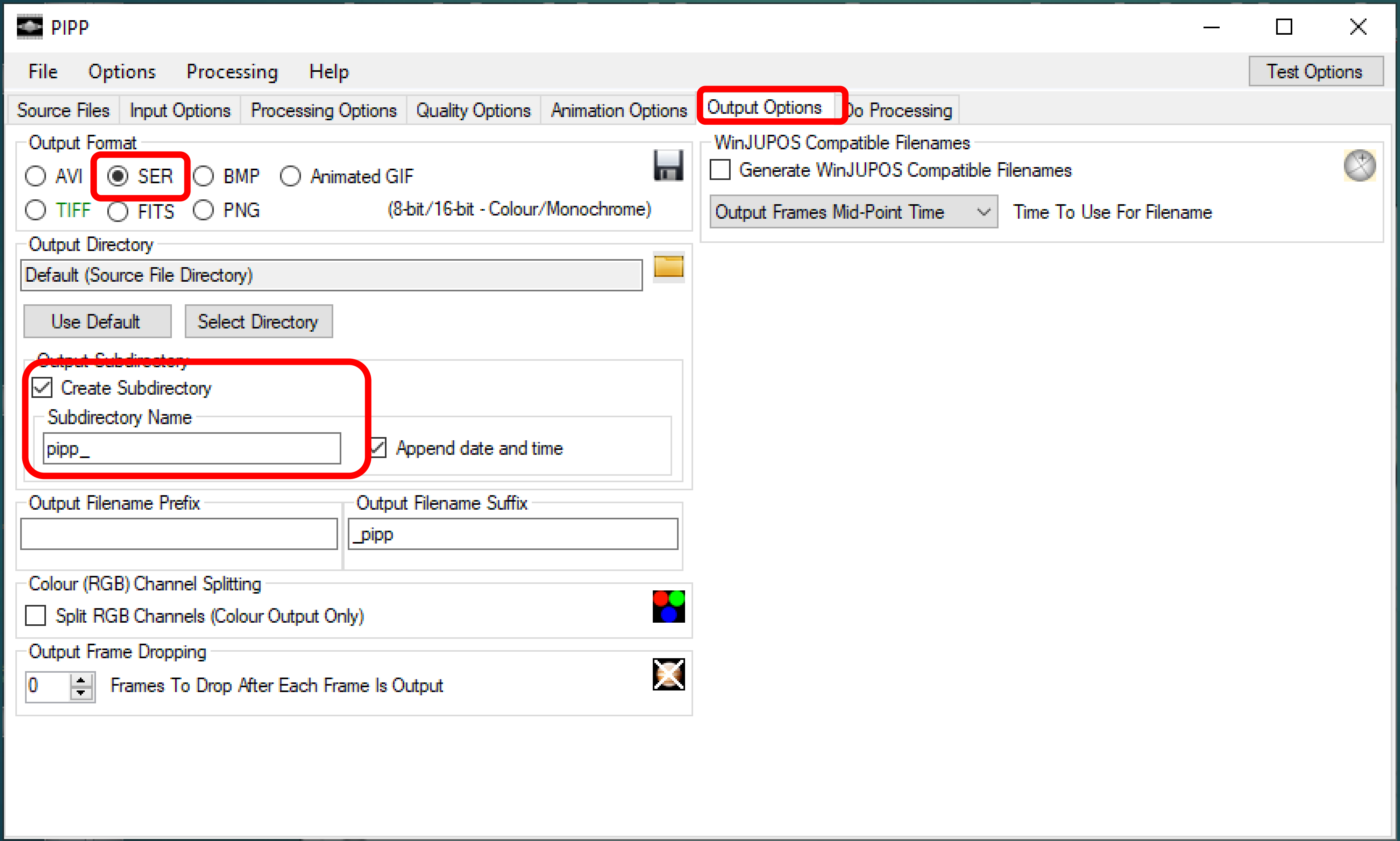




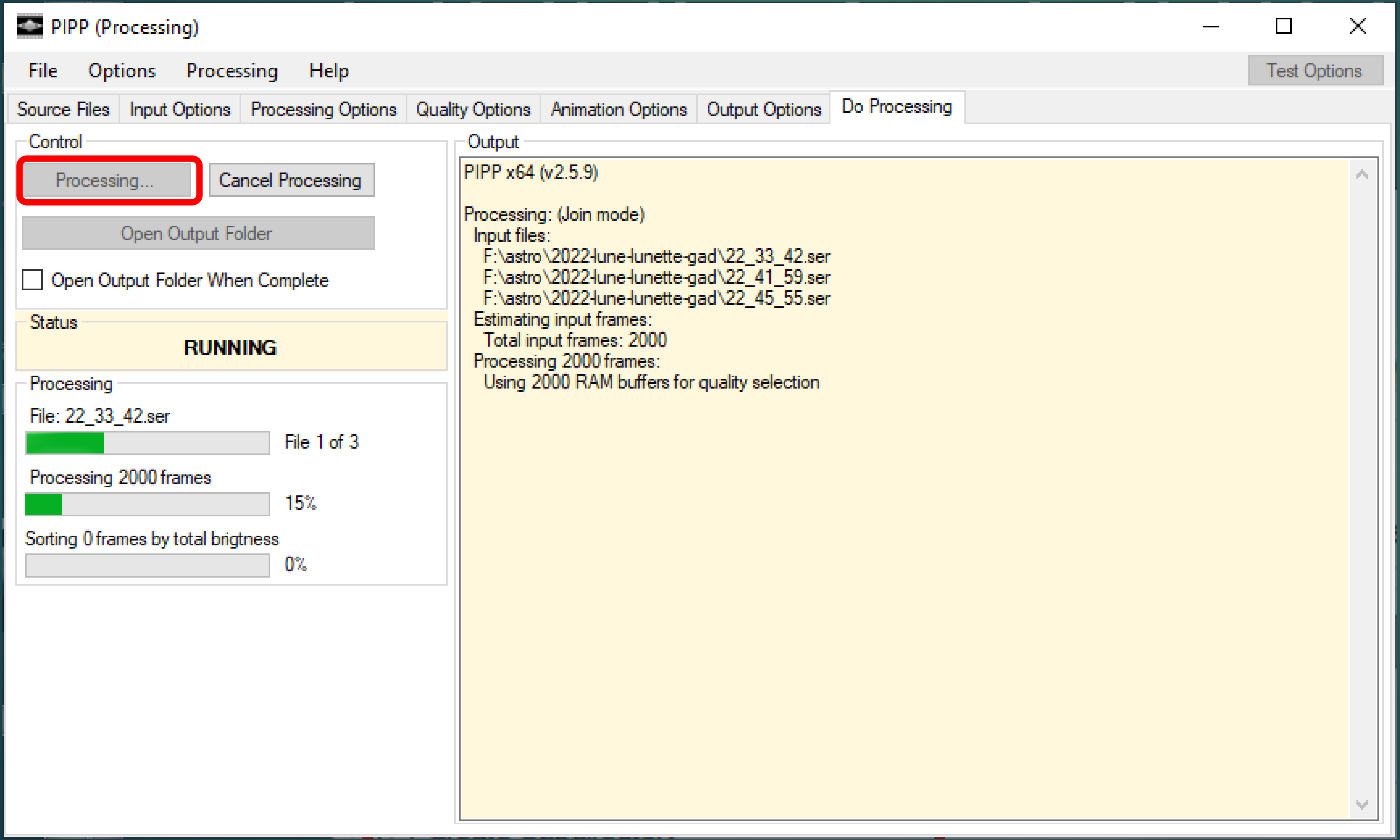
Etape 4 : Processing Options



Etape 5 : Output Options



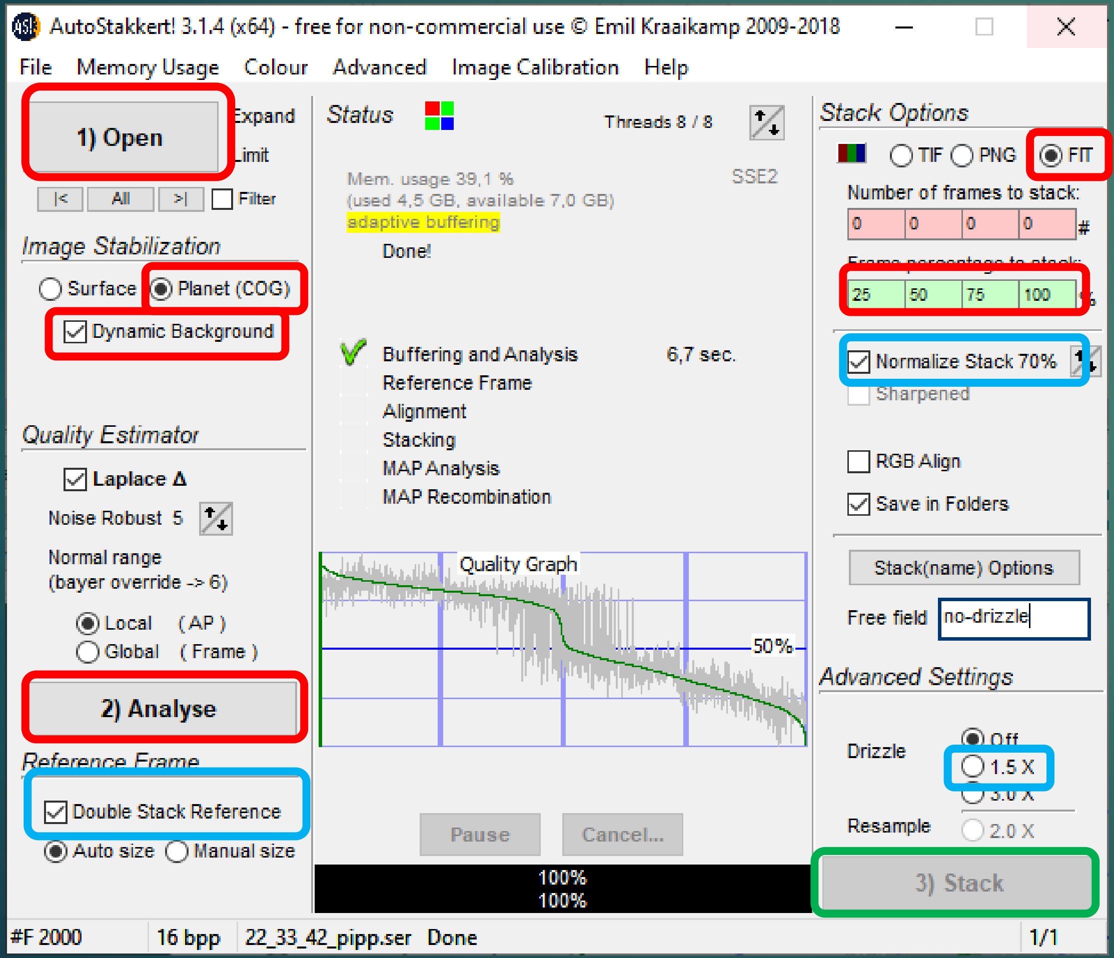
Etape 6 : Processing

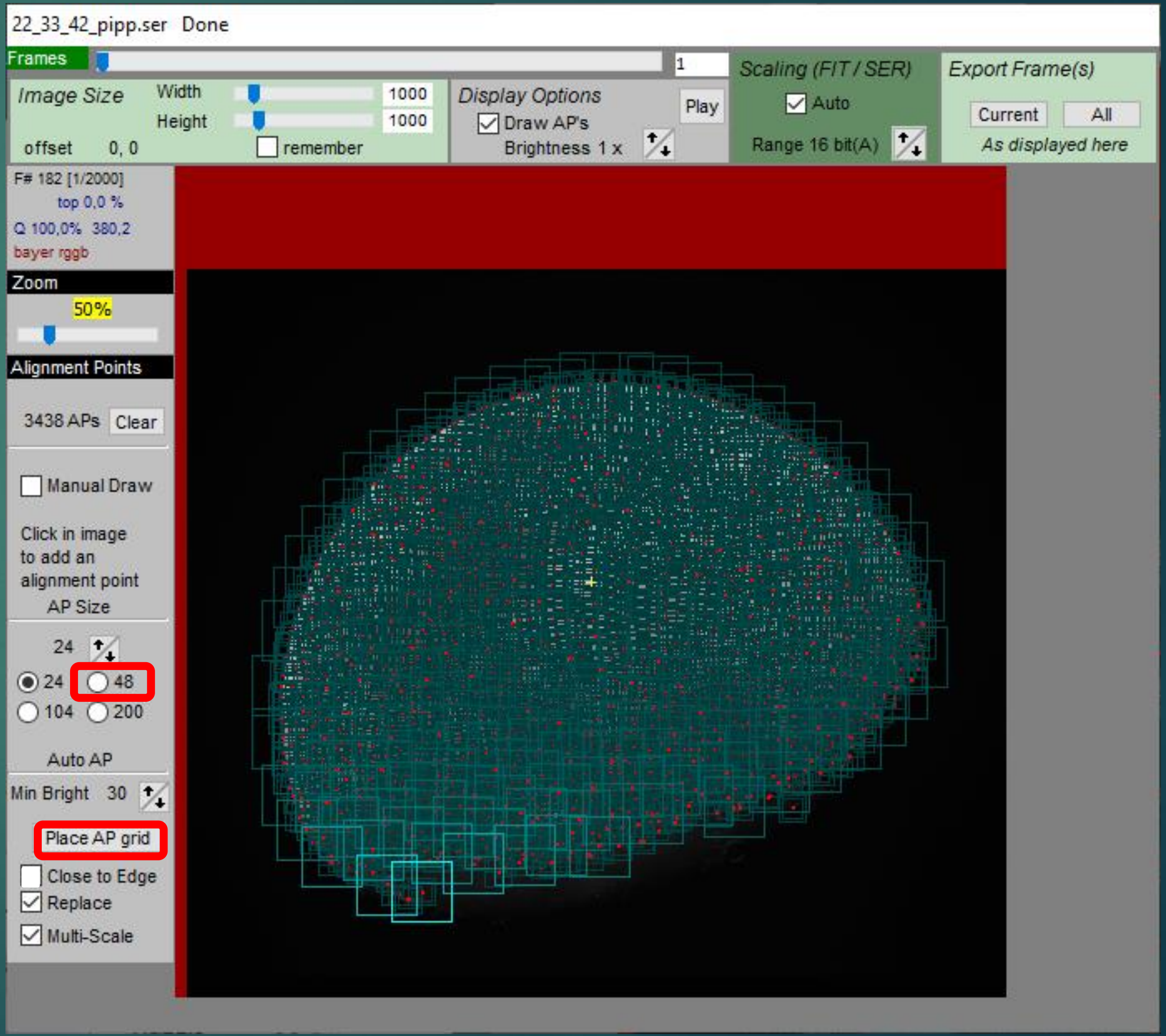


Processing : Autostakkert

TURBULENCES ATMOSPHÉRIQUES

Stacking et correction atosmphère





Impression de rien avoir gagné….., on **verra plus loin**

Image unique

Post Autostakkert

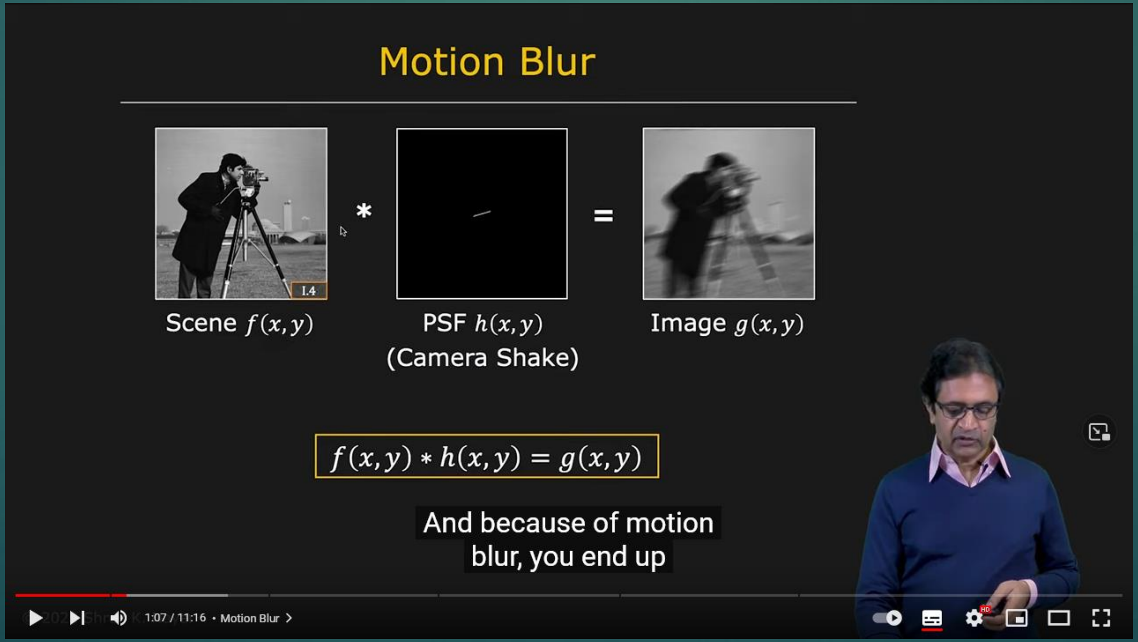
1.5x
(drizzle)

Image unique

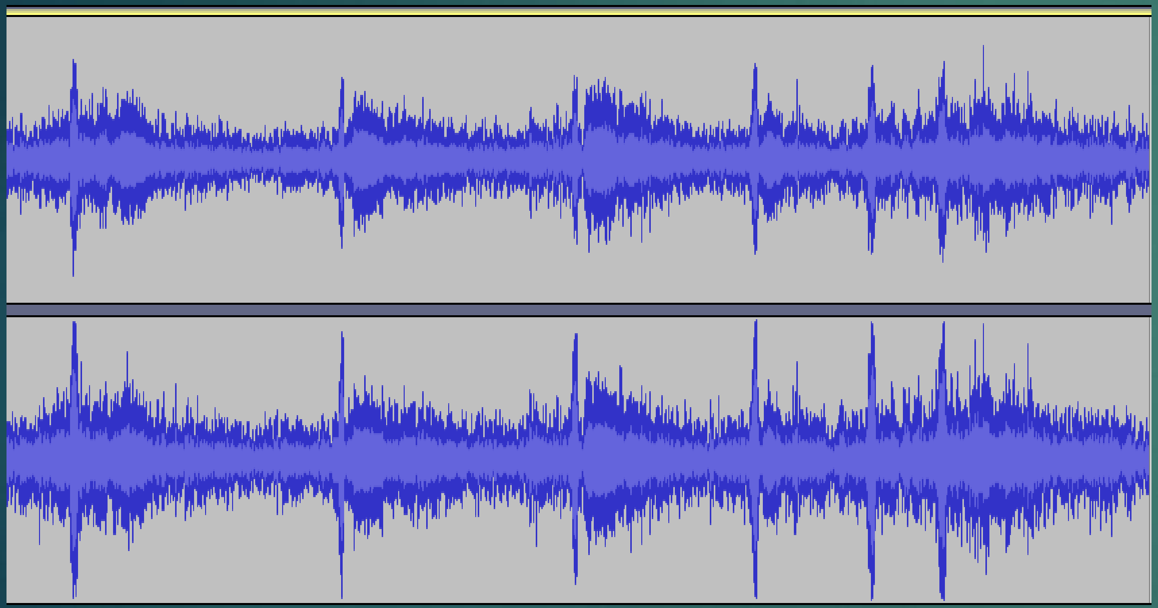
Post Autostakkert

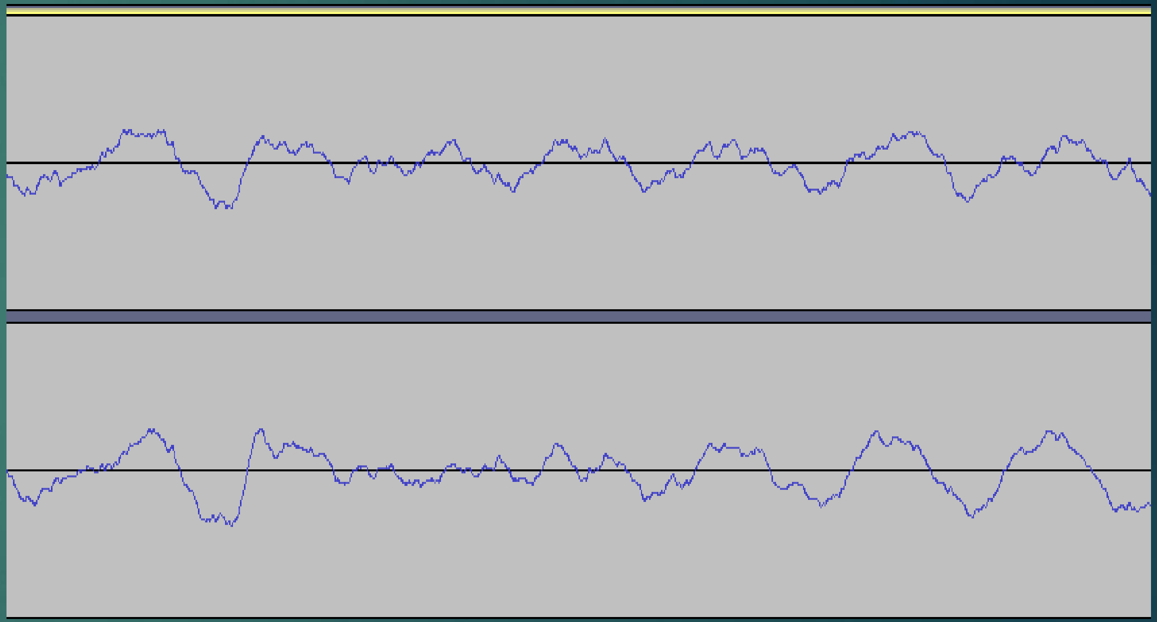
Un mot sur la dé-convolution
Un peu de théorie

- Un très bon tuto bien pédagogique (mais en anglais): https://youtu.be/f-lINpceX6k
  De **Sheer Nayar** (Columbia University)
- Les slides qui suivent sont extrait directement de cette vidéo



La tranformé de Fourier





La tranformé de Fourier

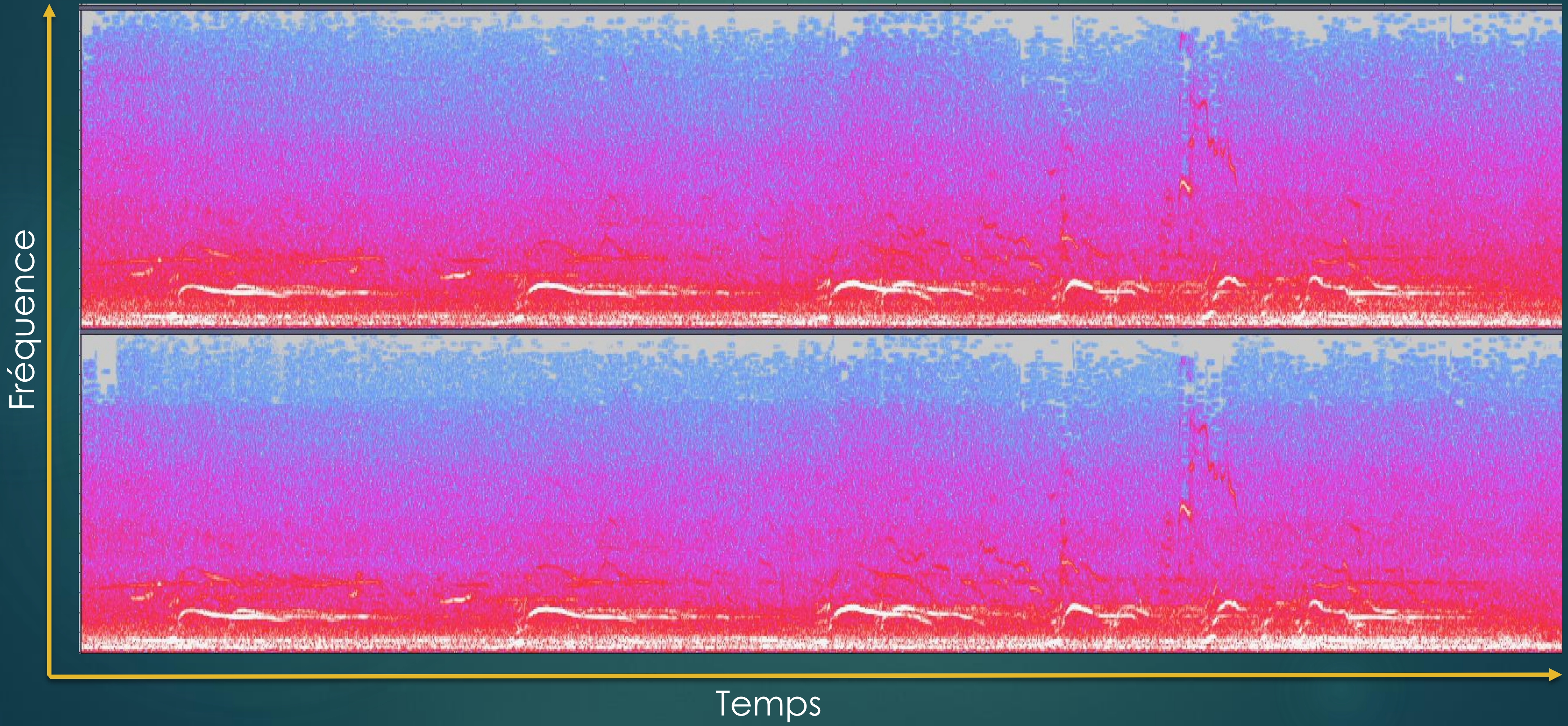
- **La transformée de Fourier** :
  - On passe dans l'**espace des fréquences**.
- **Transformée de Fourier inverse**
  - Retrouver le signal depuis le spectrogramme.
- Très utilize en **traitement du signal** !

$$F(u,v) \overset{\text{def}}{=} \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\, e^{-j2\pi(ux+vy)}\, dx\, dy$$

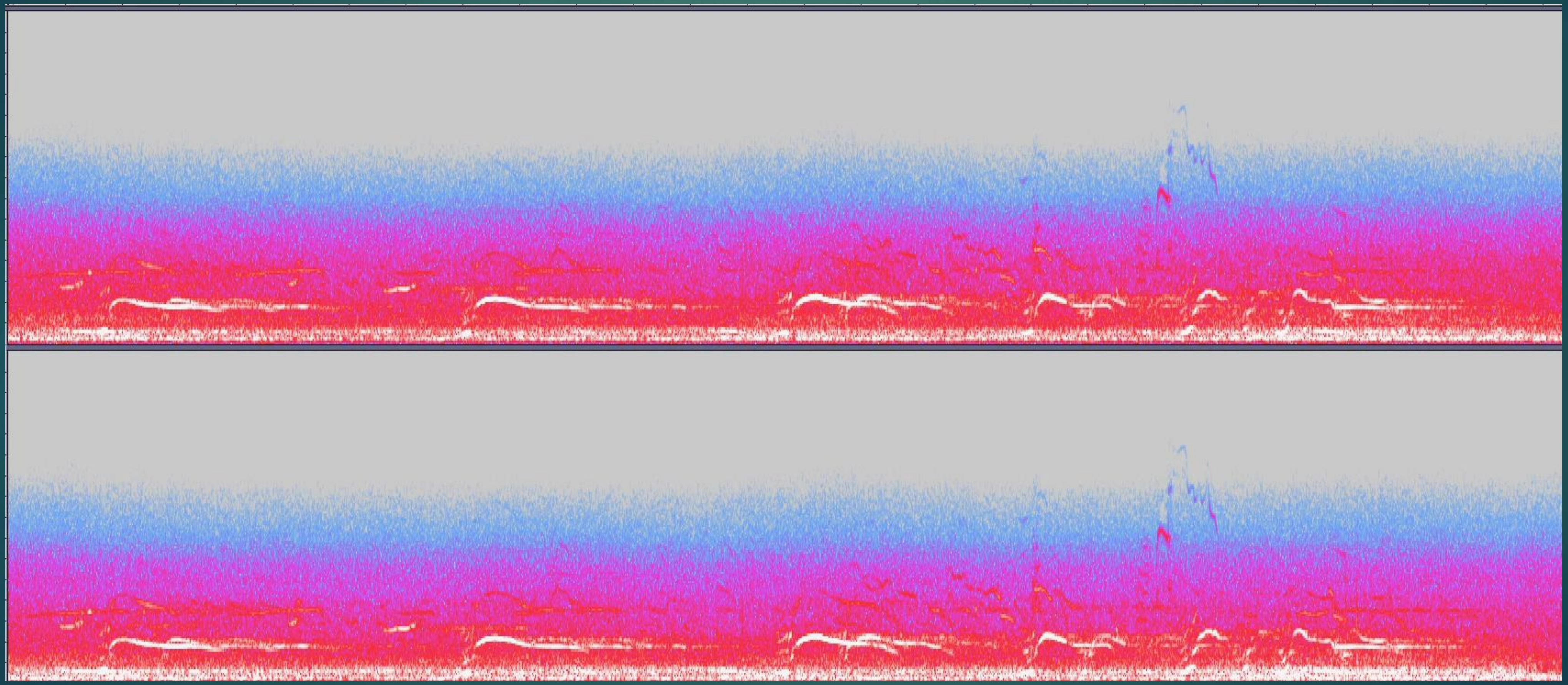
$$f(x,y) \overset{\text{def}}{=} \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} F(u,v)\, e^{j2\pi(ux+vy)}\, du\, dv$$

La tranformé de Fourier

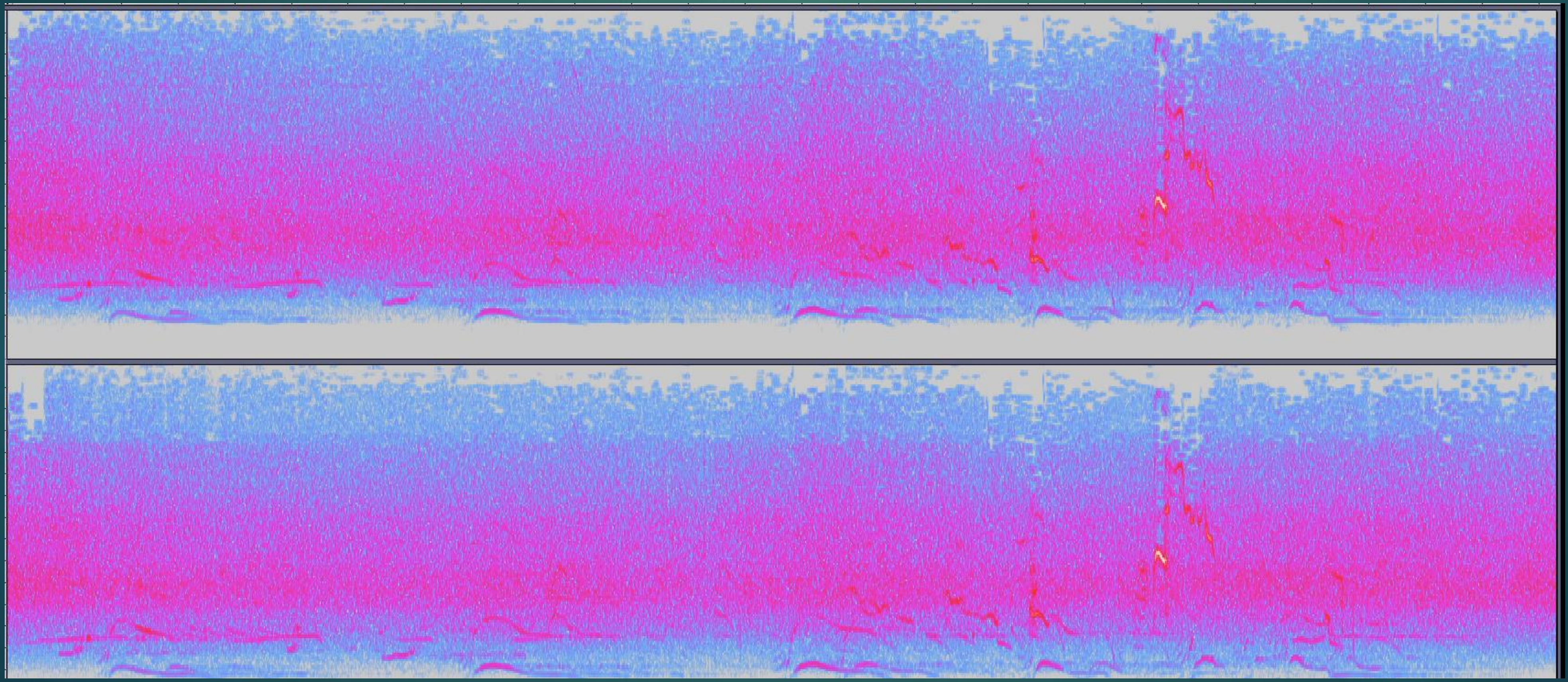
Son de : https://www.universal-soundbank.com/baleines.htm



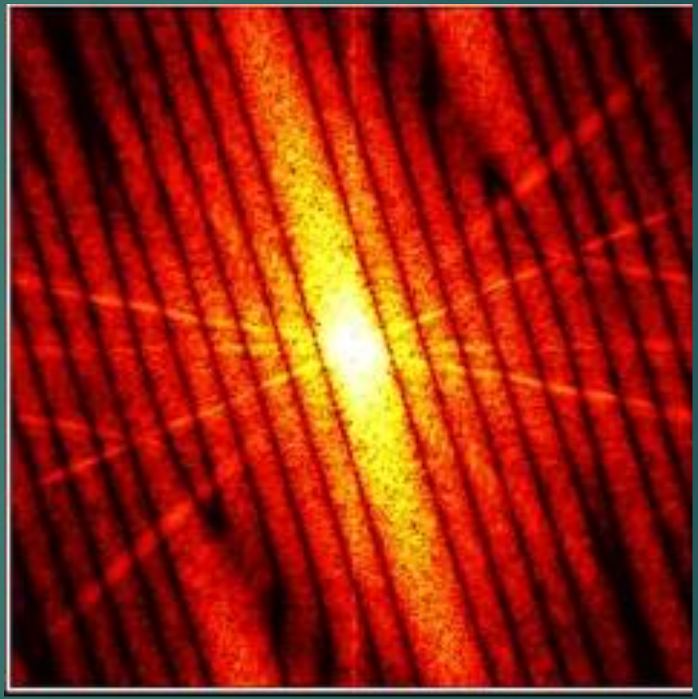
Filtrage : exemple filtre passe bas Coupe à 2.5 KHz



Filtrage : exemple filtre passe haut Coupe à 2.5 KHz



Sur une image

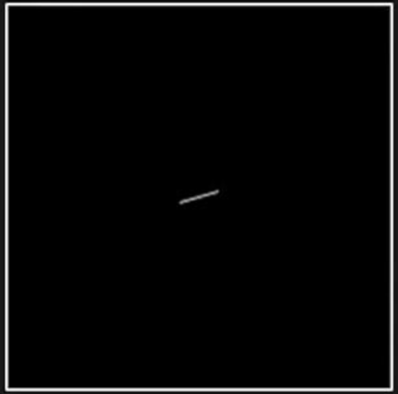


https://youtu.be/f-lINpceX6k

Motion Blur

Scene $f(x,y)$

$*$

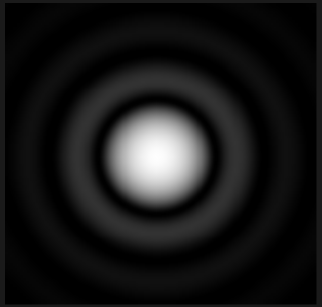


PSF $h(x,y)$
(Camera Shake)

$=$

Image $g(x,y)$

Point Spread Function
(Fonction d'étalement)



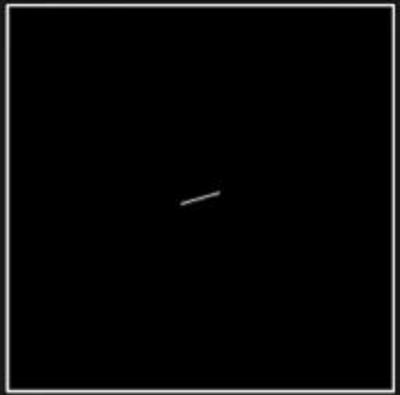
$$\boxed{f(x,y) * h(x,y) = g(x,y)}$$

https://youtu.be/f-IINpceX6k

Motion Blur

Scene $f(x,y)$

$*$



PSF $h(x,y)$
(Camera Shake)

$=$

Image $g(x,y)$

$$f(x,y) * h(x,y) = g(x,y)$$

Given captured image $g(x,y)$ and PSF $h(x,y)$,
can we estimate actual scene $f(x,y)$?

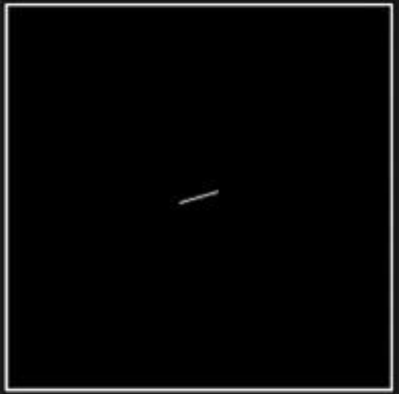
https://youtu.be/f-IlNpceX6k

https://youtu.be/f-lINpceX6k

Motion Deblur: Deconvolution

Scene $f(x,y)$

$*$



PSF $h(x,y)$
(Camera Shake)

$=$

Image $g(x,y)$

Let f' be the recovered scene.

$$f'(x,y) * h(x,y) = g(x,y)$$

$$F'(u,v)H(u,v) = G(u,v)$$

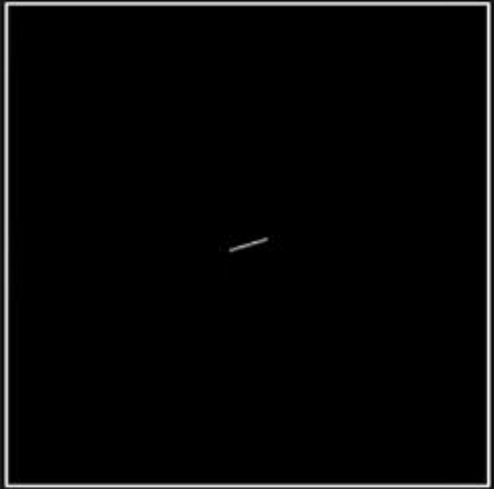
$$F'(u,v) = \frac{G(u,v)}{H(u,v)} \longrightarrow \boxed{\text{IFT}} \longrightarrow f'(x,y$$

Motion Deblur: Deconvolution

$$F'(u,v) = \frac{G(u,v)}{H(u,v)} \longrightarrow \boxed{\text{IFT}} \longrightarrow f'(x,y)$$

Image $g(x,y)$

deconvolve



PSF $h(x,y)$

=

?

Recovered $f'(x,y)$

https://youtu.be/f-lINpceX6k

Motion Deblur: Deconvolution

$$F'(u,v) = \frac{G(u,v)}{H(u,v)} \rightarrow \boxed{\text{IFT}} \rightarrow f'(x,y)$$

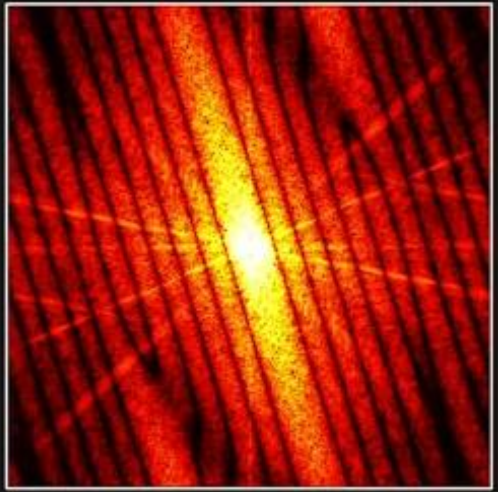
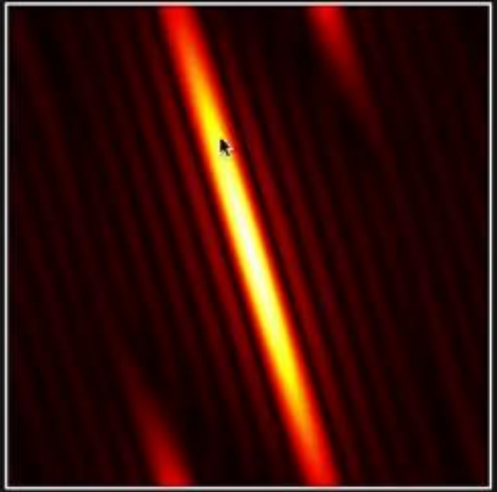


Image $G(u,v)$

/



PSF $H(u,v)$

=

?

Recovered $F'(u,v)$

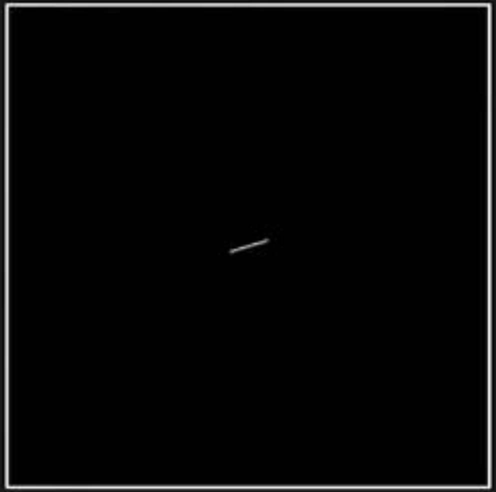
Step 1: Recover $F'(u,v)$ in Fourier Domain

https://youtu.be/f-lINpceX6k

Motion Deblur: Deconvolution

$$F'(u,v) = \frac{G(u,v)}{H(u,v)} \longrightarrow \boxed{\mathrm{IFT}} \longrightarrow f'(x,y)$$

deconvolve



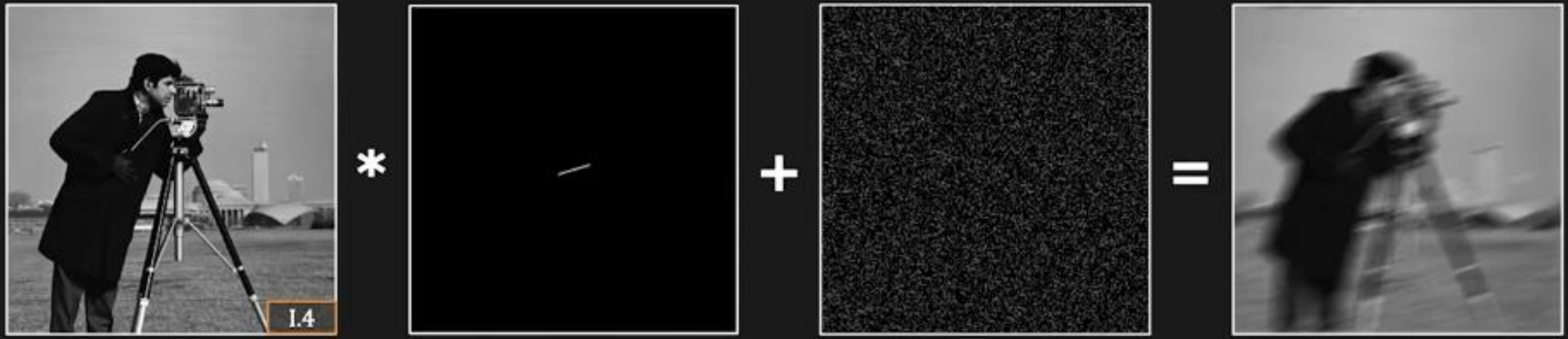
=

Image $g(x,y)$ | PSF $h(x,y)$ | Recovered $f'(x,y)$

Step 2: Compute IFT of $F'(u,v)$ to recover scene

https://youtu.be/f-lINpceX6k

Adding Noise to the Problem



Scene $f(x,y)$ | PSF $h(x,y)$ (Camera Shake) | Noise $\eta(x,y)$ | Image $g(x,y)$

$$\boxed{f(x,y) * h(x,y) + \eta(x,y) = g(x,y)}$$

Can we afford to ignore noise?

https://youtu.be/f-lINpceX6k

Motion Deblur: Deconvolution

If we ignore the noise ($\eta(x,y)$):

$$\frac{G(u,v)}{H(u,v)} = F'(u,v) \longrightarrow \boxed{\text{IFT}} \longrightarrow f'(x,y)$$

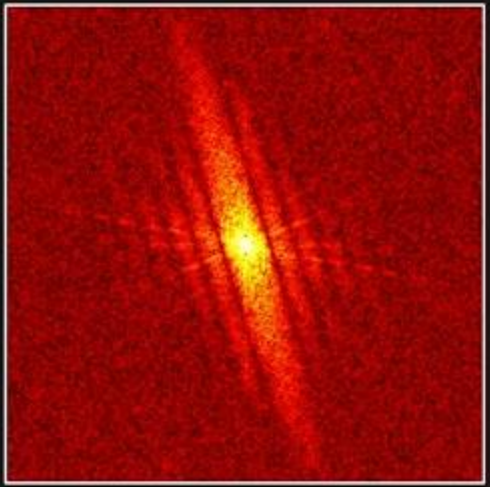
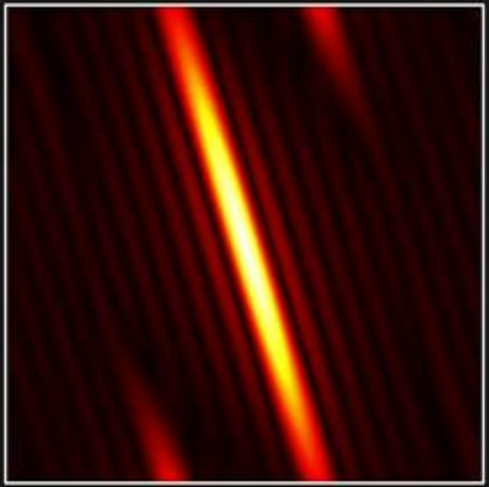


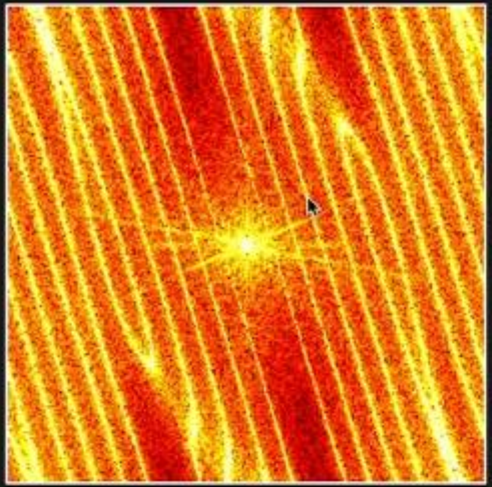
Image $G(u,v)$

/



PSF $H(u,v)$

=



Recovered $F'(u,v)$

Higher frequencies in $F'(u,v)$ are amplified

https://youtu.be/f-lINpceX6k

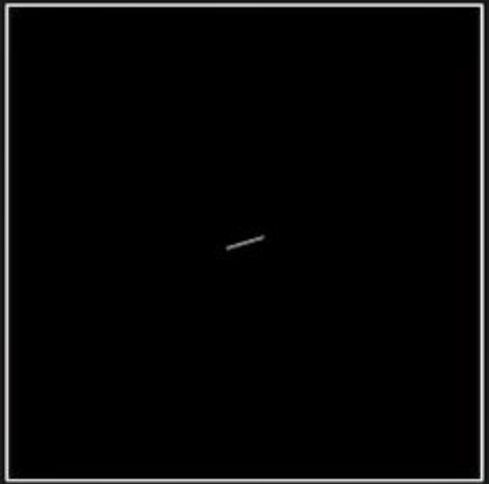
Motion Deblur: Deconvolution

If we ignore the noise ($\eta(x,y)$):

$$\frac{G(u,v)}{H(u,v)} = F'(u,v) \longrightarrow \boxed{\text{IFT}} \longrightarrow f'(x,y)$$

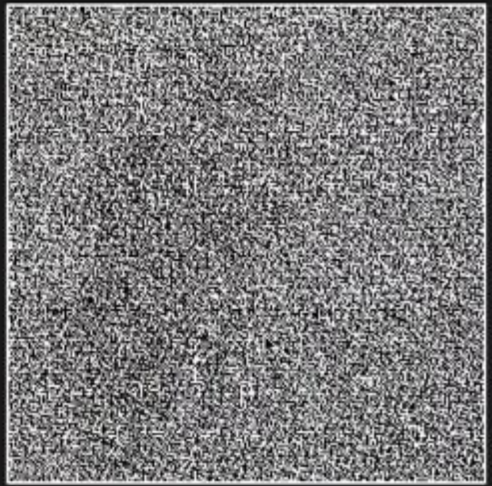
Image $g(x,y)$
(with noise)

deconvolve



PSF $h(x,y)$

=



Recovered $f'(x,y)$

Noise is significantly amplified

https://youtu.be/f-lINpceX6k

Deconvolution: Issues

$$\frac{G(u,v)}{H(u,v)} = F'(u,v) \longrightarrow \boxed{\text{IFT}} \longrightarrow f'(x,y)$$

1. Where $H(u,v) = 0$, $F'(u,v) = \infty \rightarrow$ Not recoverable

2. Motion blur filter $H(u,v)$ is a low pass filter.

   For high frequencies (u,v):
   - Noise $N(u,v)$ in $G(u,v)$ is high
   - Filter $H(u,v) \approx 0$

   } Noise in $G(u,v)$ is amplified

https://youtu.be/f-lINpceX6k

Noise Suppression: Weiner Deconvolution

$$\boxed{F'(u,v) = \frac{G(u,v)}{H(u,v)} \left[\frac{1}{1 + \frac{NSR(u,v)}{|H(u,v)|^2}} \right]}$$

- Determining NSR requires us to have prior knowledge of the noise "pattern" and the scene (or of a similar scene).

$$NSR(u,v) = \frac{|N(u,v)|^2}{|F(u,v)|^2}$$

- Often NSR is set to a single suitable constant λ.

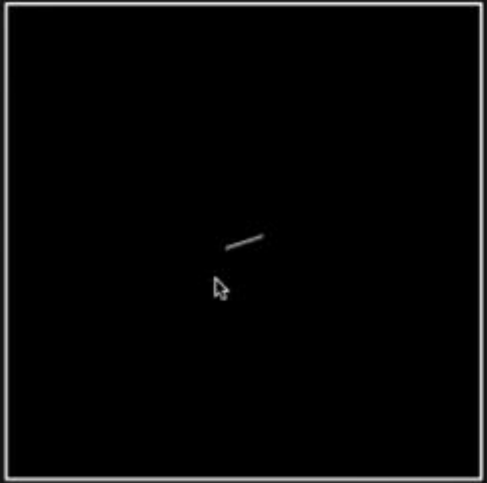
$$\boxed{NSR(u,v) = \lambda}$$

https://youtu.be/f-IlNpceX6k

Noise Suppression: Weiner Deconvolution

Noisy, Blurred Image $g(x,y)$

/



PSF $h(x,y)$

=

Recovered $f'(x,y)$

$NSR(u,v) = \lambda = 0.002$ was used to recover image

https://youtu.be/f-IlNpceX6k

Allez plus loin

- « Analyse d'images IMN 259 » (slides en français)
  https://info.usherbrooke.ca/pmjodoin/cours/IMN259/notes/Filtrage_IMN259_2018_2pages.pdf
- Par **Pierre-Marc Jodoin –** Université Usherbrook (CA)
- Traite la question du flou gaussien
  - Explication de : **Lucy-Richardson**
  - Explication **auto-estimation** de la **PSF (déconvolution aveugle)**

Vraiement plus loin

- Papier sur la dé-convolition RL aveugle (1995) [EN]
  Blind deconvolution by means of the Richardson–Lucy algorithm
  http://prancer.physics.louisville.edu/classes/650/deconvolution/fish_blind_lucy_richardson_deconvolution_josa1995.pdf

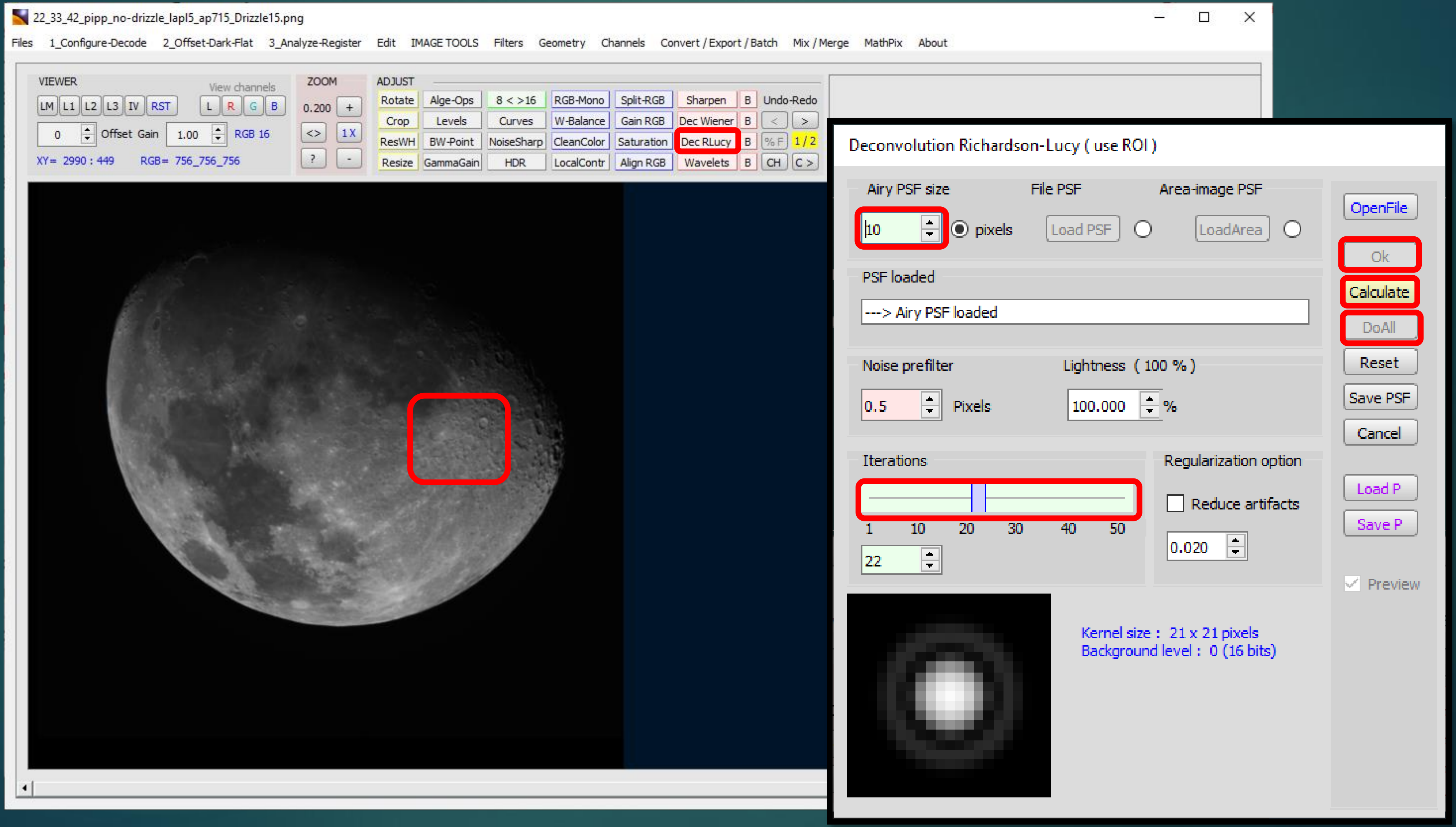
- Papier sur la deconvolution aveule (1995) [FR]
  Comparaison de différentes techniques de déconvolution aveugle
  https://www.researchgate.net/publication/27607499_Comparaison_de_differentes_techniques_de_deconvolution_aveugle

Autour du sujet

- **Sheer Nayar** a plusieurs vidéos très intéressantes
  - CCD versus CMOS (comment ça marche) https://www.youtube.com/watch?v=nsPvcX-_4KU
  - Perception des couleurs https://www.youtube.com/watch?v=V4y3K6zoUQs

Post-Processing : AstroSurface

DE-CONVOLUTION


22_33_42_pipp_no-drizzle_lapl5_ap715_Drizzle15.png
Files 1_Configure-Decode 2_Offset-Dark-Flat 3_Analyze-Register Edit IMAGE TOOLS Filters Geometry Channels Convert / Export / Batch Mix / Merge MathPix About
VIEWER
View channels
LM L1 L2 L3 IV RST
L R G B
0 Offset Gain 1.00 RGB 16
XY= 2990 : 449 RGB= 756_756_756
ZOOM
0.200 + <> 1X ? -
ADJUST
Rotate Alge-Ops 8 < >16 RGB-Mono Split-RGB Sharpen B Undo-Redo
Crop Levels Curves W-Balance Gain RGB Dec Wiener B < >
ResWH BW-Point NoiseSharp CleanColor Saturation Dec RLucy B %F 1/2
Resize GammaGain HDR LocalContr Align RGB Wavelets B CH C >
Deconvolution Richardson-Lucy (use ROI)
Airy PSF size
File PSF
Area-image PSF
10 pixels
Load PSF
LoadArea
OpenFile
Ok
Calculate
DoAll
Reset
Save PSF
Cancel
Load P
Save P
Preview
PSF loaded
---> Airy PSF loaded
Noise prefilter
0.5 Pixels
Lightness (100 %)
100.000 %
Iterations
1 10 20 30 40 50
22
Regularization option
Reduce artifacts
0.020
Kernel size : 21 x 21 pixels
Background level : 0 (16 bits)

Deconvolution RLucy

Avant

Après

Image unique

Post Autostakkert

AS2 dé-convolution RLucy

Les couleurs (contrast & HDR)

Avant

Après

Si on avait une unique image....

Image unique + RL (pas AS2)

AS2 Stacked + RL

Full AstroSurface

- On peut aussi **stacker** avec **AstroSurface**
- Je trouve **moins intuitif**…..
- **AstroSurface** ne **permet de pas de Drizzle 1.5x** pour le mode **multi-point**
  - On perds en capacité de post-traitement (deconvolution)

A vous de jouer

Backup

Noise Suppression: Weiner Deconvolution

$$F'(u,v) = \frac{G(u,v)}{H(u,v)} \left[\frac{1}{1 + \frac{NSR(u,v)}{|H(u,v)|^2}} \right]$$

Where:

Weiner Filter $\stackrel{\text{def}}{=}$ $$W(u,v) = \frac{1}{H(u,v)} \left[\frac{1}{1 + \frac{NSR(u,v)}{|H(u,v)|^2}} \right]$$

Noise-to-Signal Ratio, $NSR(u,v)$:

$$NSR(u,v) = \frac{\text{Power of Noise at } (u,v)}{\text{Power of Signal (Scene) at } (u,v)} = \frac{|N(u,v)|^2}{|F(u,v)|^2}$$

https://youtu.be/f-lINpceX6k