

Bonjour,

Le [focus stacking](#) est une technique qui vise à augmenter la profondeur de champ d'une photomicrographie en réalisant la synthèse de plusieurs clichés obtenus en faisant varier la mise au point d'un bout à l'autre du spécimen. En microscopie, elle apporte une solution élégante au problème posé par la faible [profondeur de champ](#) des objectifs à ouverture numérique élevée.

Les utilisateurs de Microsoft Windows disposent de nombreux outils dédiés au focus stacking, notamment [CombineZP](#), [Helicon Focus](#), [Zerene Stacker](#) ou [PhotoAcute Studio](#) ; la plupart d'entre eux ont fait l'objet de présentations détaillées sur le forum. Sous GNU/Linux, ces outils sont utilisables à travers

[Wine](#)

ou des outils de virtualisation comme

[VirtualBox](#)

ou

[VMware](#)

. Cependant, il existe aussi des solutions en

[ligne de commande](#)

, simples d'emploi et surtout faciles à automatiser, qui motivent l'écriture du présent article.

Étant surtout intéressé par la mycologie, j'ai choisi d'appliquer cette technique à des basidiospores de [russule](#). La détermination des russules est généralement considérée comme difficile : elle fait appel, entre autres choses, à l'examen des caractères microscopiques et, en particulier, de l'ornementation sporale. Celle-ci est mise en évidence par le

[réactif de Melzer](#)

, un colorant iodo-ioduré pour lequel un

[protocole de coloration rapide](#)

a été présenté sur le forum. Afin d'aider le déterminateur, les livres de mycologie proposent des schémas détaillés de l'ornementation sporale, réalisés par de talentueux dessinateurs.

Écrit par Eddy

Jeudi, 23 Décembre 2010 20:54 - Mis à jour Dimanche, 09 Janvier 2011 21:00

---

À ce stade, je suppose que des préparations soignées ont été réalisées et que des images ont été prises en vue d'un traitement par focus stacking. Dans la mesure du possible, celles-ci doivent être bien alignées, bien contrastées et comporter des zones nettes légèrement chevauchantes (on recommande environ 30% de recouvrement pour deux images adjacentes). En pratique, entre 5 et 10 images au format JPEG suffisent à donner de bons résultats. Les tests réalisés avant la rédaction de cet article me donnent à penser que, dans le cas présent, le recours au format RAW n'est pas indispensable.

### Conversion des images JPEG au format TIFF

Les logiciels que nous allons utiliser pour empiler les images travaillent avec des fichiers au format TIFF. Or, les appareils photo proposent généralement de sauvegarder les images au format JPEG (c'est en tout cas ce que propose mon Canon EOS 500D, en plus du format RAW). Nous allons donc commencer par convertir nos images JPEG au format TIFF. Il existe pour cela une méthode fastidieuse car répétitive qui consiste à ouvrir chaque image dans un logiciel tel que [Gimp](#), puis à l'enregistrer sous un autre format. Sous GNU/Linux, on peut traiter les images par lots à l'aide d'un outil de la suite logicielle [ImageMagick](#)

appelé

convert

(installation sous Ubuntu :

```
sudo apt-get install imagemagick
```

). Celui-ci peut non seulement convertir une image dans un autre format, mais aussi effectuer des transformations diverses comme un logiciel de retouche photo. Je m'en sers donc pour convertir un lot d'images au format TIFF en leur appliquant des filtres (netteté, contraste) et une correction de type

[flat field](#)

.

```
eddy$ find *.jpg -execdir convert flat-field.jpg {} -unsharp 0x20 -compose divide -composite {}.tif ;
```

Cette longue commande fait beaucoup de choses : d'abord, elle utilise le programme find

Écrit par Eddy

Jeudi, 23 Décembre 2010 20:54 - Mis à jour Dimanche, 09 Janvier 2011 21:00

---

pour trouver tous les fichiers JPEG du répertoire courant. Elle applique ensuite à chacun d'entre eux la commande

convert

, qui effectue les transformations suivantes :

- Augmentation du contraste de l'image JPEG (option `-unsharp 0x20`).
- Correction de type flat field (option `-compose divide -composite`) avec la carte des poussières `flat-field.jpg`.
- Conversion au format TIFF.

Vous aurez sans doute compris que les paires d'accolades `{}` sont remplacées par le nom du fichier JPEG récupéré par la commande

find

, de sorte que la commande est appliquée successivement à tous les fichiers du répertoire.

J'ajoute que l'on peut combiner plein d'autres outils. Je vous renvoie pour cela au manuel d'utilisation de l'outil

convert

[sur cette page](#)

(en anglais).

Une dernière précision importante pour la suite : Les fichiers TIFF produits portent l'extension `.jpg.tif`, car nous n'avons indiqué à aucun moment que nous voulions supprimer l'extension JPEG ! C'est bête mais le savoir évite bien des tâtonnements... 

Écrit par Eddy

Jeudi, 23 Décembre 2010 20:54 - Mis à jour Dimanche, 09 Janvier 2011 21:00

---

### Alignement des images en vue du focus stacking

Nous allons maintenant vérifier l'alignement des images au format TIFF en vue de leur synthèse ultérieure par le programme de focus stacking. Cette étape est **importante** : elle conditionne en grande partie la qualité du résultat final. On utilise pour cela un outil appelé `align_image_stack`

qui fait partie de la suite d'outils

[Hugin](#)

(installation sous Ubuntu :

```
sudo apt-get install hugin-tools
```

). Son utilisation est très simple :

```
eddy$ align_image_stack -m -a zframe_*.jpg.tif
```

Cette commande indique à `align_image_stack` d'utiliser toutes les images au format TIFF du répertoire courant; elle crée des masques qui seront préfixés par

`zframe_`

, de sorte que nous pourrions facilement les réutiliser par la suite. Cette étape est relativement longue si l'on choisit de travailler sur les images en taille réelle, mais on peut aussi demander de travailler sur des miniatures... au prix d'une petite perte de précision. Malgré tout, avec les ordinateurs actuels, le programme vient à bout de 5-10 images en moins d'une minute.

### Le focus stacking proprement dit !

□

Maintenant que les images sont alignées, il ne reste plus qu'à construire l'image finale en utilisant un autre logiciel en ligne de commande appelé [enfuse](#) (installation sous Ubuntu : `sudo apt-get install enfuse`

). On utilise les paramètres suivants :

```
eddy$ enfuse -o output.tif
```

```
--exposure-weight=0
```

```
--saturation-weight=0
```

```
--contrast-weight=1
```

```
--hard-mask
```

```
--contrast-window-size=9 zframe_*.tif
```

Les paramètres passés à enfuse lui indiquent qu'il faut accorder du poids aux pixels bien contrastés, (`--contrast-weight=1`) mais pas aux pixels bien exposés (`--exposure-weight=0`) ou présentant une saturation convenable (`--saturation-weight=0`). Le contraste local est évalué à partir d'une « fenêtre » de 9 pixels (

```
--contrast-window-size=9
```

). Les images utilisées en entrée sont les images alignées produites précédemment (celles qui sont préfixées par `zframe_`

## Focus stacking en ligne de commande sous GNU/Linux

Écrit par Eddy

Jeudi, 23 Décembre 2010 20:54 - Mis à jour Dimanche, 09 Janvier 2011 21:00

---

), et l'image finale est enregistrée dans le fichier  
output.tif

. Elle peut ensuite faire l'objet de post-traitements divers (notamment

[NeatImage](#)

et

[MicroCartouche](#)

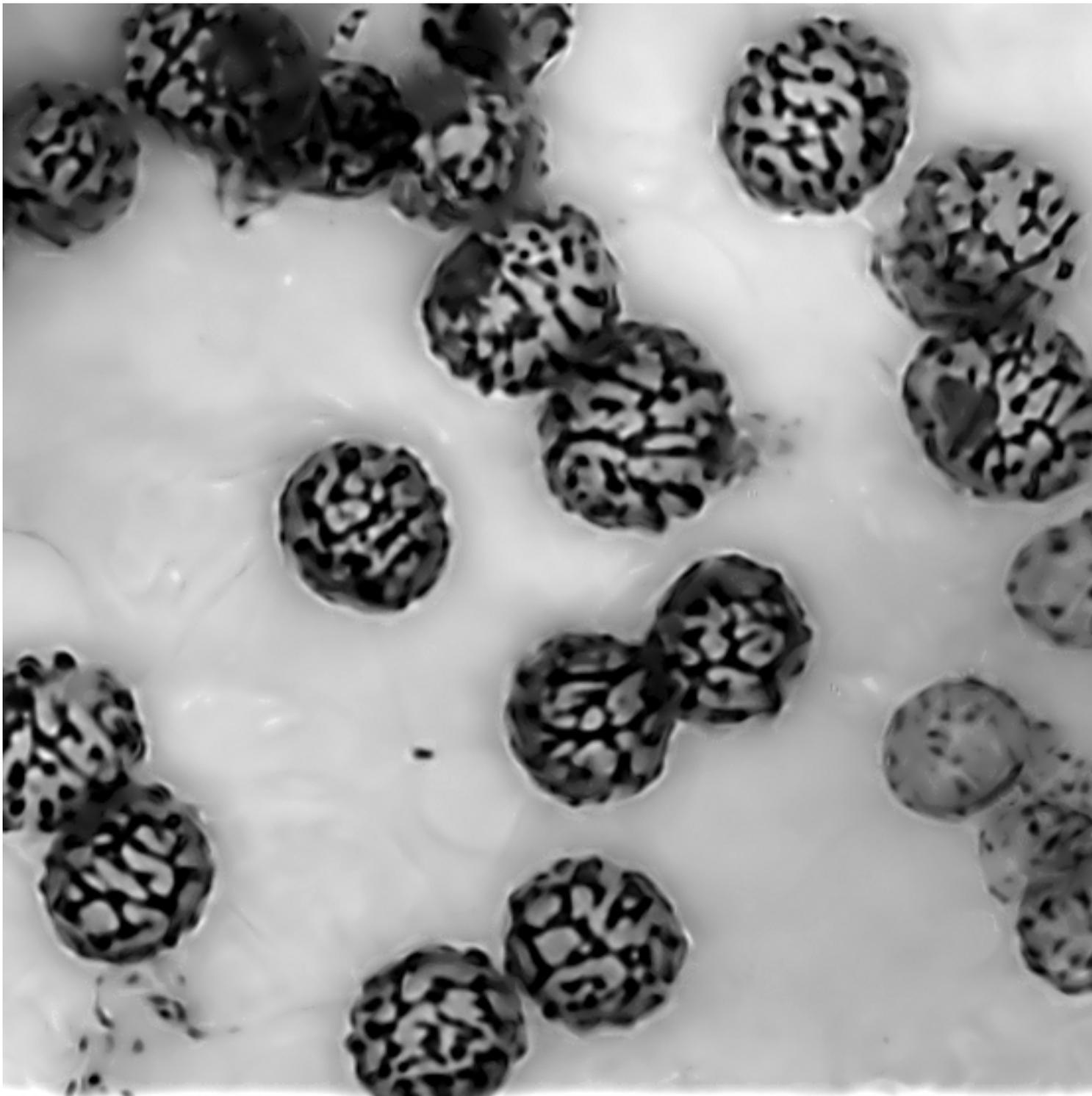
). Voici un exemple d'image finale obtenue à l'aide de ces outils :

# Focus stacking en ligne de commande sous GNU/Linux

Écrit par Eddy

Jeudi, 23 Décembre 2010 20:54 - Mis à jour Dimanche, 09 Janvier 2011 21:00

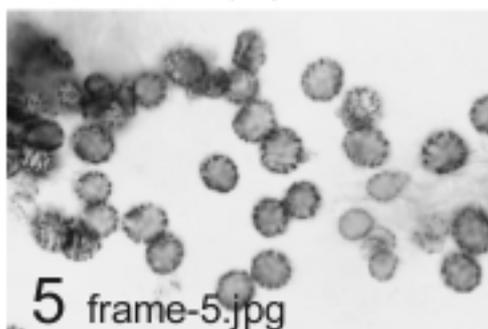
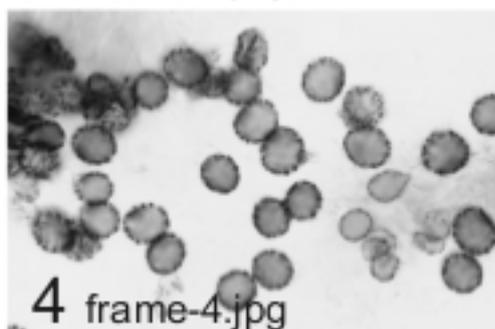
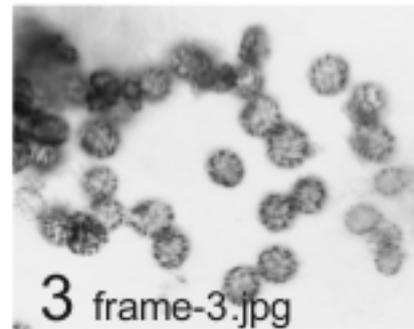
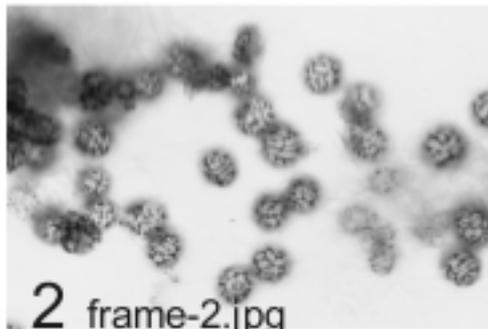
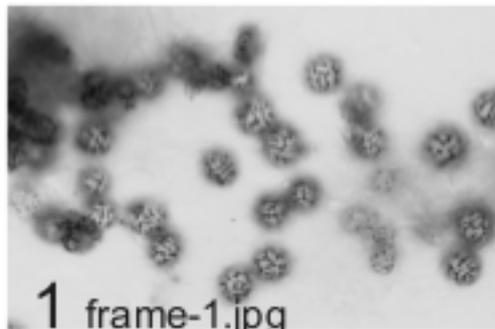
---



10  $\mu$ m obj. x100 R = 1:1

Fond clair - Prép : Chloral

```
eddy$ ls *.jpg
```



```
eddy$ find *.jpg -execdir convert {} {}.tif \;
```

```
eddy$ ls *.tif
```

```
frame-1.jpg.tif  
frame-2.jpg.tif  
frame-3.jpg.tif  
frame-4.jpg.tif  
frame-5.jpg.tif
```

```
eddy$ align_image_stack -m -a zframe_ *.tif
```

```
eddy$ enfuse -o output.tif
```

```
--exposure-weight=0  
--saturation-weight=0  
--contrast-weight=1  
--hard-mask  
--contrast-window-size=9 zframe_*.tif
```

## Image finale : output.tif

- Recadrage de l'image
- Réduction du bruit avec NeatImage

