

# G'MIC 1.5.9.3 : Poisson Blending, Seamcarving, OpenMP, et autres joyusetés !

Posté par [David Tschumperlé \(site web personnel\)](#) le 23/06/14 à 22:30. Édité par 5 contributeurs. Modéré par [NeoX](#). [Licence CC By-SA](#).

Étiquettes : [gimp](#) , [openmp](#) , [g'mic](#) , [traitement\\_d'images](#) + [Étiqueter](#)

À l'occasion de la sortie, vendredi dernier, de la version **1.5.9.3** de [G'MIC \(GREYC's Magic for Image Computing\)](#), qui est un framework libre pour le traitement d'images, je vous propose de continuer ma petite série d'articles de résumés sur ce que permet de faire ce logiciel, qui a vu le jour en 2008, et qui continue d'évoluer jour après jour, à un rythme qu'on pourrait qualifier de "soutenu".

## Sommaire

- [Contexte](#)
- [Nouvelles fonctions de traitement d'images.](#)
  - [1. Poisson Editing / Seamless blending.](#)
  - [2. Seamcarving](#)
  - [3. Synthèse de micro-textures.](#)
  - [4. Montage complexes](#)
  - [5. Reconstruction d'une forme à partir d'un squelette.](#)
  - [6. Simulation de flou de profondeur](#)
  - [7. Fractalize](#)
  - [8. Ascii art](#)
  - [9. Rebouchage de petits trous.](#)
  - [10. Le jeu : 2048](#)
- [Autres améliorations notables](#)
  - [1. Directives OpenMP pour la parallélisation](#)
  - [2. Autres amélioration globales.](#)
- [Communication / Visibilité](#)
  - [1. Libre Graphics Meeting 2014.](#)
  - [2. Pages de tutoriel](#)
  - [3. Tutoriels vidéos](#)
  - [4. Liste de discussion pour développeurs](#)
- [Perspectives et conclusions.](#)

Salut à tous,

## Contexte



**Avertissement:** Je suis le développeur principal du projet G'MIC. Je ne serai donc probablement pas impartial dans l'écriture du texte qui suit (sans chercher à être de mauvaise foi non plus). Vous voilà prévenus !

G'MIC est un logiciel libre (distribué sous licence CeCILL), développé dans l'équipe [IMAGE](#) du laboratoire [GREYC](#), à Caen. Le GREYC est un laboratoire de recherche public ayant des tutelles multiples [CNRS](#), [ENSI-CAEN](#) et [Université de Caen](#) ).

Cet article fait suite aux résumés précédents ([Février 2014](#), [Octobre 2013](#), [Juillet 2013](#), [Mars 2013](#), [Octobre 2012](#), [Avril 2012](#), [Octobre 2011](#), [Mars 2011](#), et y en a d'autres plus anciens, mais bon...). Pour plus d'informations sur ce qu'est G'MIC, je vous incite donc à aller jeter un oeil a ces articles précédents !

Au programme de ce journal, un résumé des fonctionnalités nouvelles introduites depuis la sortie de la version 1.5.8.3 (qui était l'objet de mon [dernier journal](#) sur le sujet). C'est parti !

# Nouvelles fonctions de traitement d'images.

## 1. Poisson Editing / Seamless blending.

La technique du *Poisson Editing* en traitement d'image a été popularisée par *P. Perez, M. Gangnet* et *A. Blake* dans le papier du même nom (disponible [ici](#), au format PDF). En résumé, elle permet de faciliter l'incrustation d'un bout d'une image A dans une image B, en calculant une solution qui détermine automatiquement les changements colorimétriques à appliquer aux pixels de A pour qu'elle s'incruste le mieux possible dans l'image B (d'où le deuxième sobriquet *Seamless blending* sous lequel cette technique est connue). Ceci s'obtient en résolvant [l'équation de Poisson](#)<sup>w</sup> définie sur le domaine de l'image A.

Un exemple pratique d'utilisation de cette technique : ici, nous souhaitons ajouter un requin dans un petit étang inoffensif de la campagne française. Si on copie/colle un requin (découpé d'une autre image) sur l'image de l'étang, on va très probablement obtenir quelque chose comme le résultat de gauche, avec des couleurs qui ne s'accordent pas entre les deux images. Le *Poisson Editing* permet de transformer l'image de gauche en l'image de droite *automatiquement*, sans avoir à manipuler d'outils de retouches de niveaux ou de couleurs.



Aujourd'hui donc, G'MIC permet cela. Dans le [greffon pour GIMP](#) par exemple, vous accédez à cet algorithme par le nouveau filtre **Layers / Blend [seamless]**. Tous les exemples que je montre ici ont été réalisés en 5 minutes maxi, chacun avec ce filtre, sous GIMP. C'est infiniment plus simple que de se taper les réglages des couleurs et des niveaux à la main !

Le *Poisson Editing* a beaucoup d'applications amusantes. Ça marche relativement bien pour copier/coller des bouts de têtes par exemple, ce qui permet d'échanger l'intérieur de visages sur deux photos différentes, comme sur l'exemple ci-dessous :



Ou bien sûr, sur une même photo, ça marche aussi ! Nous avons réalisé un [tutoriel vidéo](#) de 3m23s qui illustre comment ce dernier exemple a été réalisé, en temps réel. Quand j'ai dit 5 minutes, j'étais plutôt large ! :) Une [autre vidéo](#) montre la même opération pour l'échange de visages entre deux images de peintures différentes.



Un dernier exemple pour la route (en théorie assez difficile à réaliser, du fait des différences de couleur de peau). C'est loin d'être parfait, mais si on considère le temps de réalisation (moins de 5 minutes), c'est vraiment pas mal.



À noter que la technique du *Poisson Editing* est prévue pour être implémentée par les développeurs de GIMP en "natif" dans les prochaines versions de GIMP. Mais en attendant que ça fonctionne :, vous avez donc la possibilité d'utiliser le [greffon G'MIC pour GIMP](#) pour réaliser cette opération.

## 2. Seamcarving

Autre algorithme bien connu des traiteurs d'images : le *Seamcarving*, permet de réduire ou d'agrandir une image de manière 'intelligente' en supprimant en priorité les portions d'images qui n'ont pas d'intérêt particulier (fond, ou



couleurs unies). Ceci, afin de pouvoir redimensionner l'image sans modifier l'*aspect ratio* des objets ou des structures d'intérêts présents dans les images. Le papier décrivant cet algorithme est disponible [ici](#).

Une implémentation de l'algorithme du *Seamcarving* a été réalisée dans G'MIC par *Andy*, un sympathique contributeur anglais. Les deux petits exemples d'utilisation ci-dessous illustrent l'utilité de cette approche pour la réduction puis l'agrandissement d'image :



Notez que les objets "importants" dans ces deux images n'ont pas été modifiés durant le redimensionnement, et ceci sans intervention manuelle. Par exemple, les mouettes et le bateau gardent leur taille originale. Ca n'est pas parfait bien sûr (les câbles du pont par exemple), mais ça peut être utile dans pas mal de cas (cela marche bien sur les panoramas par exemple).

Notons que le seamcarving est déjà disponible dans GIMP, via le greffon [Liquid Rescale](#). L'intérêt pour nous de l'avoir dans G'MIC réside surtout dans le fait de pouvoir le combiner facilement avec tous les autres traitements disponibles de G'MIC, et de pouvoir le proposer pour toutes nos interfaces. Par exemple, vous pouvez facilement tester la technique du *Seamcarving* en ligne, via le site [G'MIC Online](#), en choisissant le filtre **Deformations / Seamcarve**.

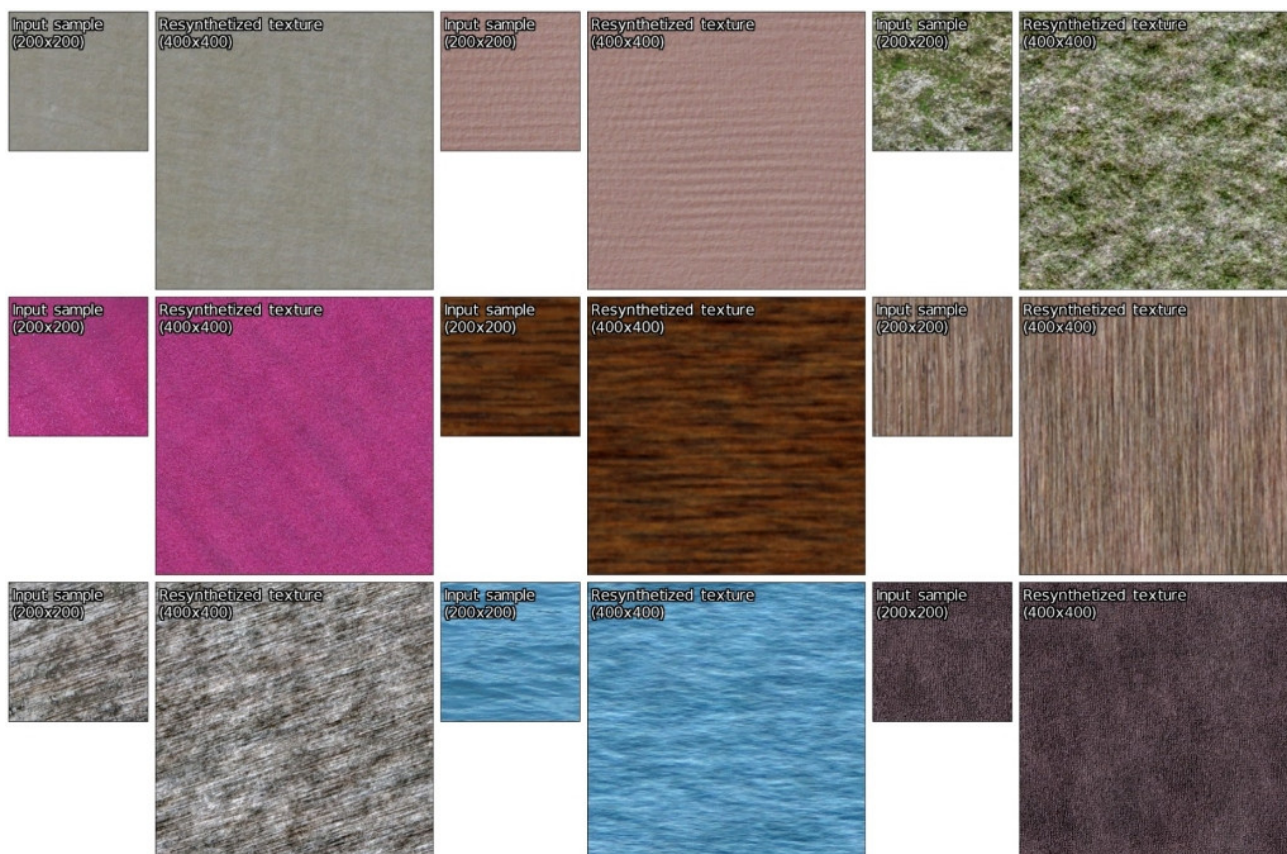
### 3. Synthèse de micro-textures.

La synthèse de micro-textures par *randomisation de phase* est une technique qui a été proposée par *B. Galerne*, *Y. Gousseau* et *J-M. Morel*, et qui permet de synthétiser une image de taille quelconque ayant les mêmes caractéristiques visuelles qu'une image contenant une *micro-texture*.

Une description complète de l'algorithme peut être vue [ici](#).

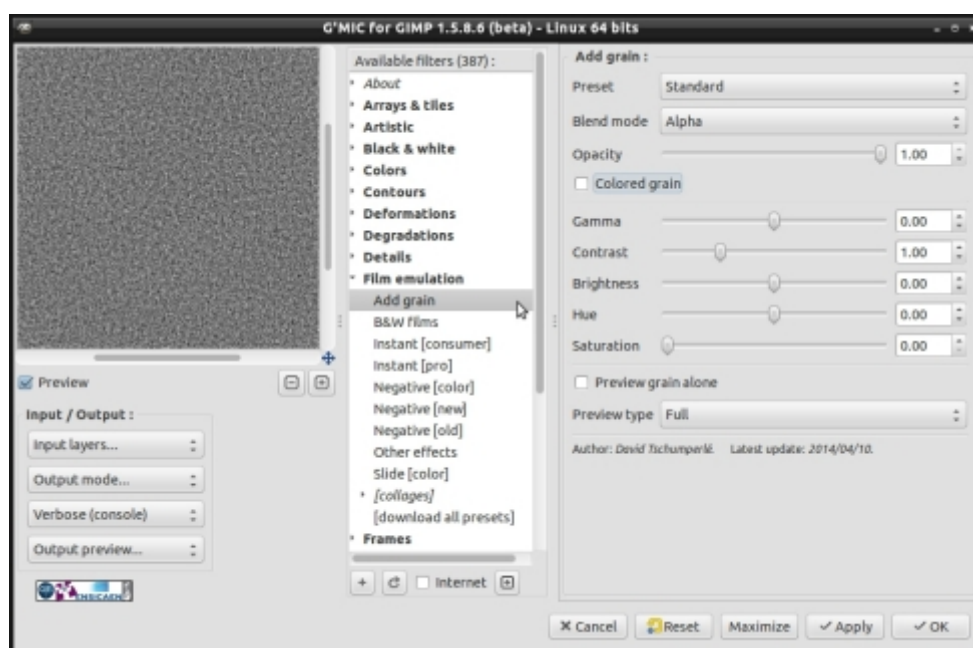
Le qualificatif *micro-* est important ici : la texture originale doit être suffisamment peu structurée pour que la synthèse fonctionne correctement (ça ne marchera pas avec un mur de briques par exemple).

G'MIC implémente maintenant cette technique de synthèse de micro-textures, et vous pouvez vous en servir pour générer des textures plus grandes à partir de textures plus petites, comme illustré par l'image suivante :



Dans le greffon G'MIC pour GIMP, cet algorithme est utilisable via le filtre **Patterns / Resynthesize texture**. Les textures ainsi générées ont en plus la propriété d'être *périodiques*, donc vous pouvez les mettre sous forme de sprites et en faire des tuiles facilement (pour des jeux vidéos par exemple).

Une conséquence directe d'avoir cet algorithme disponible, est de pouvoir proposer un filtre ajoutant du *grain* à des images. Mais attention, pas du grain *complètement* synthétique (type bruit Gaussien ou Poissonien), comme le proposent beaucoup de filtres d'ajout de bruit. Non, ici le grain est resynthétisé à la taille de l'image, à partir d'un échantillon scanné d'une photo argentique contenant du "vrai" grain (localisé sur un niveau de gris constant). En plus du modèle de grain à resynthétiser, on peut bien sûr régler divers paramètres, comme le mode de blending, son opacité, l'échelle du grain, etc...



#### 4. Montage complexes

Nous avons également ajouté une commande très générique *-montage* dans G'MIC pour permettre la réalisation d'un "montage" à partir de plusieurs images d'entrées, telle que celui-ci (ici, à partir de 5 images) :





G'MIC possédait déjà diverses commandes pour réaliser des montages simples, mais cette nouvelle commande ajoute plus de flexibilité. L'originalité provient de la façon dont le "plan" de montage est défini. On définit un montage entre deux images numérotées 0 et 1, par  $H0:1$  ou par  $V0:1$  suivant qu'on veut un alignement horizontal ou vertical de ces deux images. Mais on peut également remplacer les indices 0 ou 1 par des expressions de montage  $Hm:n$  ou  $Vm:n$ , et ceci, à un niveau de récursion quelconque. La commande *-montage* prend ensuite en argument cette expression, et redimensionne automatiquement toutes les images à leur taille optimale pour générer le montage désiré. L'exemple ci-dessus correspond par exemple à un montage du type  $VH0:1:H2:V3:4$  (où 0=Joconde, 1=Jungle, 2=Fenêtre, 3=Voilier et 4=Fille).

Ajoutez à ceci la possibilité d'exprimer aussi des rotations d'images ( $R0$ , pour une rotation de  $90^\circ$ ), ou de les symétriser ( $_M0$ , pour un effet miroir horizontal) et d'ajouter du padding, et vous obtenez une commande vraiment puissante pour réaliser des montages d'images quelconques.

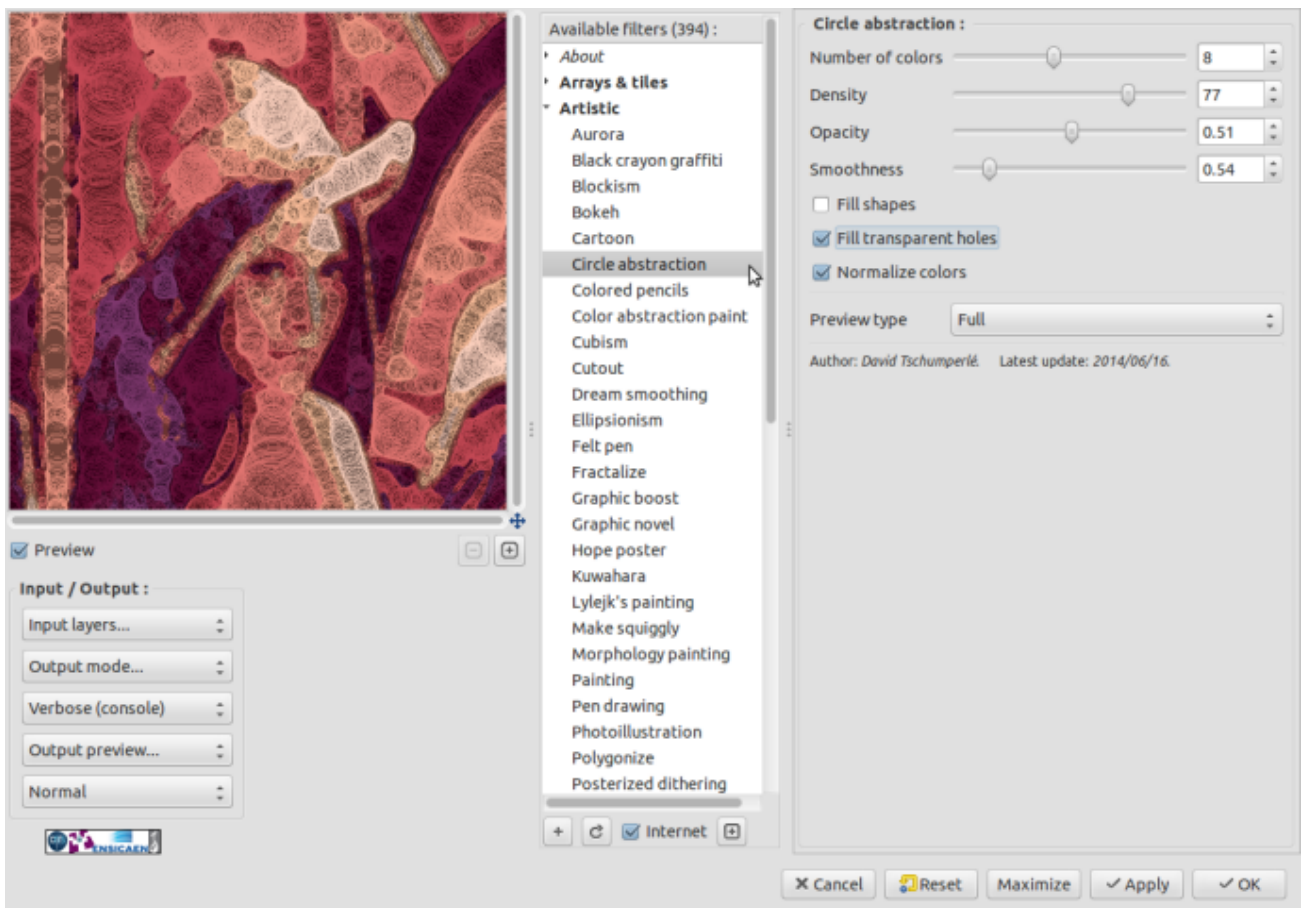
Patrick David, contributeur actif de G'MIC a écrit un [article entier sur son blog](#) (blog dont je conseille vivement la lecture, par ailleurs!), à propos de cette fonctionnalité particulière maintenant disponible dans G'MIC.

## 5. Reconstruction d'une forme à partir d'un squelette.

G'MIC permet également de reconstruire maintenant une forme binaire 2d quelconque à partir de son squelette. Pour le moment, on n'utilise pas beaucoup cette possibilité, mais on peut imaginer faire des choses sympathiques avec ce genre de technique, comme par exemple pouvoir *extruder* intelligemment une forme 2d sous forme de représentation 3d, comme illustré ci-dessous (la forme 2d initiale est en haut à gauche, en rouge).



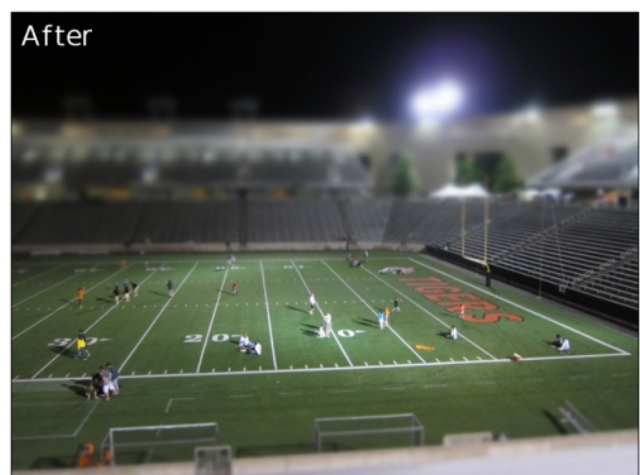
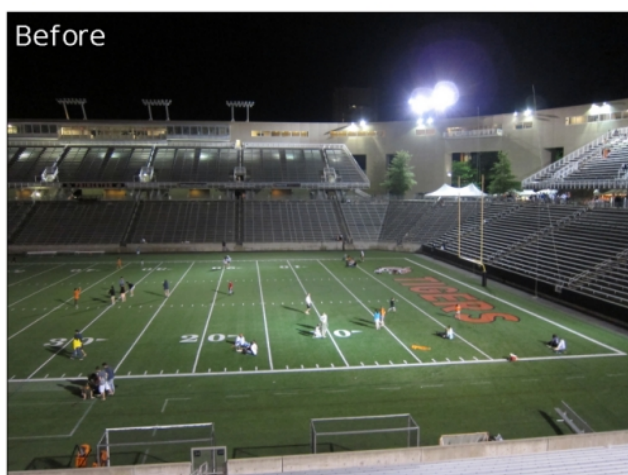
On peut aussi utiliser cette reconstruction à partir de squelettes pour générer des rendus artistiques à base de cercles d'une image, comme avec le filtre **Artistic / Circle abstraction** que nous avons ajouté récemment dans le greffon G'MIC pour GIMP :



À creuser encore, pour peut-être plus d'effets à venir...

## 6. Simulation de flou de profondeur

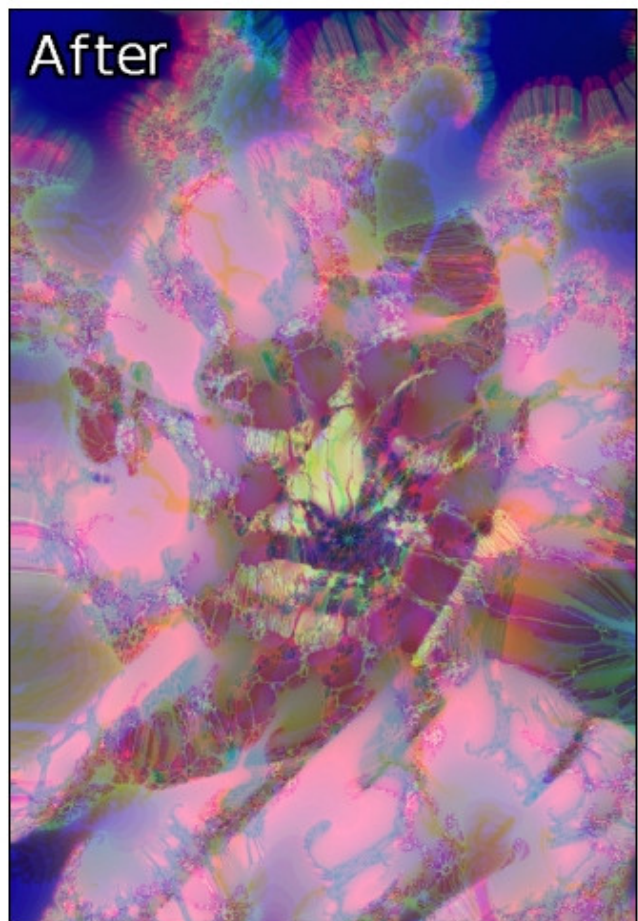
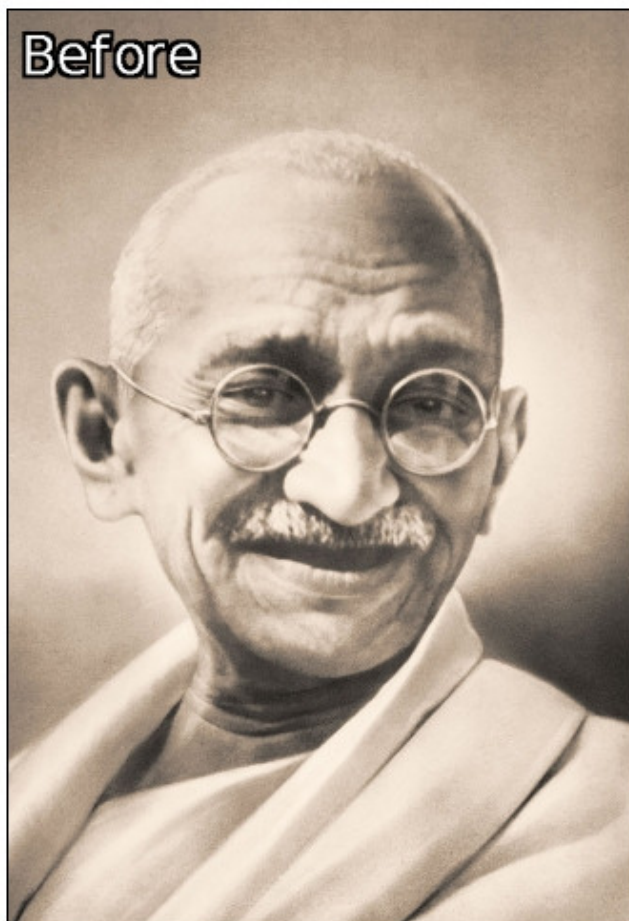
Nous avons également ajouté un filtre **Degradations / Blur [depth-of-field]** qui tente de simuler un flou de profondeur dans une image. Ce type de technique permet de simuler des effets de type [Tilt-shift<sup>W</sup>](#), même si l'effet de flou n'est pas toujours très réaliste. Il y a certainement des choses à améliorer encore sur ce filtre, mais on peut déjà réaliser des choses sympas, comme avec l'exemple ci-dessous, qui donne l'impression qu'une maquette a été photographiée (image de droite).



## 7. Fractalize

Autre petit filtre amusant, **Artistic / Fractalize** qui va tenter de déformer le rendu d'une [fractale de Mandelbrot<sup>W</sup>](#) de telle sorte qu'elle ressemble le plus possible à une image d'entrée. Un exemple ci-dessous :



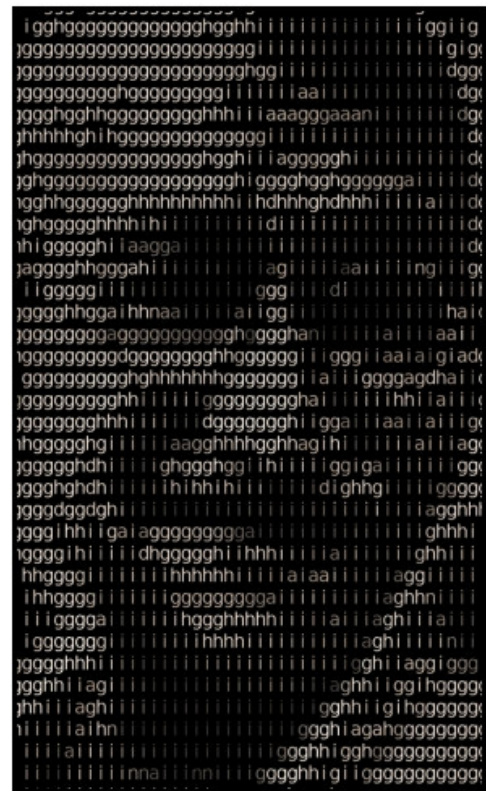


Je me rends compte qu'il aurait été opportun de choisir le portrait de [Benoît Mandelbrot<sup>W</sup>](#) pour l'exemple...

## 8. Ascii art

Là encore, une fonctionnalité assez classique ajoutée dans G'MIC, permettant de générer des images de type [ASCII Art<sup>W</sup>](#). On peut choisir différents types de rendus (Noir et Blanc ou coloré), et différents types de jeux de caractères : binaire, ASCII, symboles de maths, ou même dictionnaire personnalisé. Cette dernière opportunité permet par exemple de générer des images de type ASCII-Art d'un personnage, uniquement avec les lettres qui composent son nom ou son prénom (exemple de Gandhi, à droite ci-dessous).



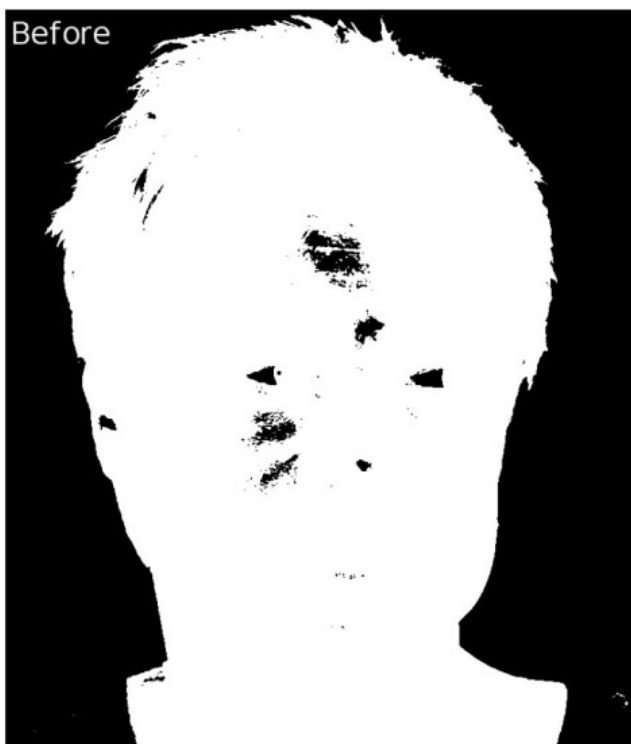
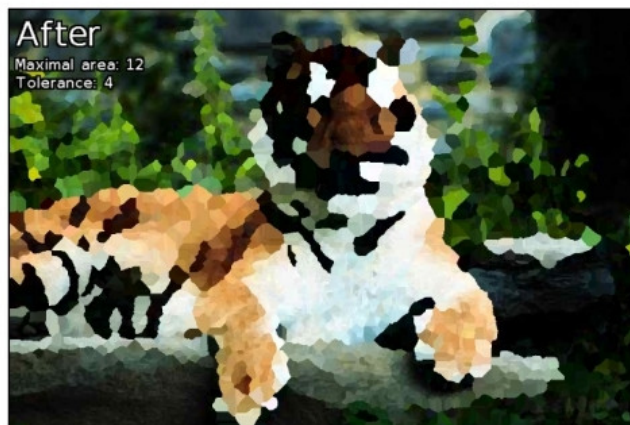
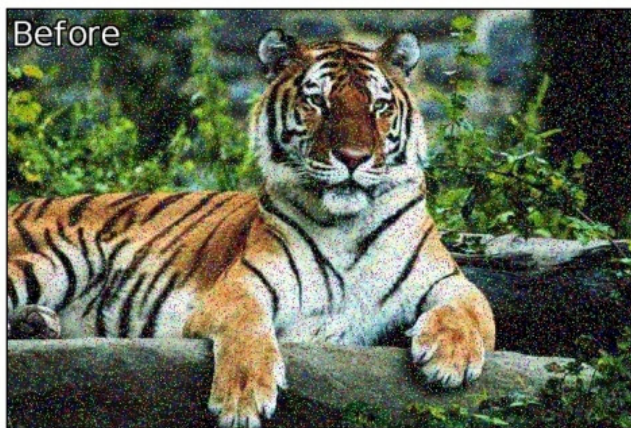


L'utilisation de ce filtre dans GIMP permet naturellement de créer des rendus simultanément sur plusieurs calques d'entrées, et vous pouvez donc facilement générer [des vidéos de type ASCII Art sous GIMP](#).

## 9. Rebouchage de petits trous.

Nous avons également ajouté un filtre **Repair / Inpaint [holes]** qui permet de détecter et de "reboucher" les amas de pixels de même couleur formant des petits "trous" dans des images.

Ce type de configuration peut se rencontrer dans le cas d'images avec du bruit impulsif (ci-dessous première ligne), ou lorsque l'on essaye de créer des masques à partir d'une segmentation automatique d'une image (ci-dessous, dernière ligne). On peut aussi jouer avec ce filtre pour créer des abstractions d'images (ligne du milieu). C'est un filtre qu'on n'utilisera qu'occasionnellement, mais que de temps gagné dans les cas où on en a besoin !



## 10. Le jeu : 2048

Alors là, plus pour le fun, on a ajouté le jeu [2048<sup>W</sup>](#) dans G'MIC. Il est jouable uniquement à partir de [l'interface ligne de commande de G'MIC](#), en lançant la commande `gmic -x_2048`. Rien de spécial à dire, c'est un 2048, assez simple, cela a été écrit surtout pour tester l'interpréteur de langage de script intégré à G'MIC. C'est souvent avec ce genre de petites démos "hors-sujet" qu'on découvre des petits bugs à corriger ou des améliorations possibles à apporter au code.





## Autres améliorations notables

Cela c'était pour les fonctionnalités de traitement d'images "pures", partie la plus visible de l'iceberg. Quelques mots sur les améliorations notables moins visibles, mais ô combien importantes !

### 1. Directives OpenMP pour la parallélisation

Les fonctions de traitement d'images de G'MIC reposent essentiellement sur une bibliothèque C++ de traitement d'image générique, nommée [Climg](#). Pour la version **1.5.9.3**, nous avons apporté de gros changements dans Climg en y ajoutant des directives [OpenMP](#)<sup>w</sup>, afin de paralléliser la plupart des fonctions de calcul et de traitement d'image importantes. Par conséquent, G'MIC en bénéficie aussi. Le résultat immédiat, c'est que les filtres et traitements de G'MIC deviennent plus rapide, car ils peuvent utiliser plusieurs coeurs de calculs si la machine hôte en dispose. C'est particulièrement visible sur des "grosses" images (dès qu'on dépasse une résolution de 1024x1024 en fait). Certaines commandes peuvent aller jusqu'à 6 fois plus rapidement, notamment celles qui utilisent l'évaluateur de formules mathématiques intégré dans G'MIC.

La plupart des opérations arithmétiques et de filtrage des images sont également accélérées. C'est très intéressant pour la retouche photo par exemple. Le support d'OpenMP est aujourd'hui fonctionnel sous Linux et Windows. On a encore quelques problèmes pour que ça tourne sous Mac OS X malheureusement.

En tout cas, l'ajout de directives OpenMP a été un gros travail, mais c'est aussi une grande satisfaction de voir le gain des performances (en même temps, j'ai un PC de bureau à 24 coeurs...).

## 2. Autres amélioration globales.

Plus globalement, pas mal d'efforts ont été fournis pour améliorer les performances de G'MIC. L'empreinte mémoire de l'interpréteur a été réduite, ainsi que son temps de compilation (il faut actuellement moins de cinq minutes pour compiler G'MIC sur une machine récente, avec `make -j`).

Cela reste assez gourmand en mémoire pour le compiler, mais c'est de moins en moins un problème.

## Communication / Visibilité

### 1. Libre Graphics Meeting 2014.

Le projet étant actif, de plus en plus de contributeurs et d'utilisateurs se manifestent, et même si c'est souvent pour des contributions mineures, c'est toujours agréable ! On a pu rencontrer certains utilisateurs au workshop [Libre Graphics Meeting](#) qui a eu lieu à Leipzig en Avril dernier. C'était sympa, on a fait le plein d'idées (qui se sont concrétisées par de nouveaux filtres après coup), et on a pu [présenter notre projet](#).

Pour information, voici les slides de *teaser* qui ont été présentés.



On a rencontré un grand monsieur (par la taille, et par le talent), j'ai nommé *Rolf Steinort*, instigateur de la série de tutoriels vidéos [Meet the GIMP](#), qui a eu la gentillesse de réaliser un [épisode #199](#) entièrement consacré à G'MIC !

### 2. Pages de tutoriel

On a également la chance depuis quelques mois d'avoir quelqu'un (*Garry Osgood*) qui écrit très régulièrement [des pages de tutoriel sur l'utilisation de G'MIC](#), principalement dans le cadre d'une utilisation en ligne de commande. C'est une formidable opportunité pour que les gens découvrent la richesse de G'MIC sans se lancer dans la lecture de la documentation officielle "technique", qui est certes complète mais assez rébarbative. *Garry* décrit des cas d'utilisation réels, et même s'il y a encore beaucoup à dire, ce qu'il a fait est déjà une très bonne documentation. Je ne peux que vous conseiller [d'y jeter un oeil](#).

### 3. Tutoriels vidéos

Cette fois, en visant plus les utilisateurs du [greffon G'MIC pour GIMP](#), nous avons ouvert [une chaîne Youtube](#) avec des petites vidéos de démonstration de l'utilisation du greffon pour des tâches bien précises. Il y a encore assez peu de contenu, mais des choses assez marrantes déjà :



Vous pouvez y apprendre comment [échanger deux visages](#), [créer des logos avec des amas de fruits](#), [comment supprimer la trame dans une photo avec la transformée de Fourier](#), [comment rajeunir quelqu'un sur une photo de portrait](#), [comment réaliser un effet de type "Fractalus" en quelques clics](#) ou encore [comment](#)



[faire un montage d'images facile à partir de plusieurs calques d'entrées](#).

Je débute en montage vidéo, donc ça n'a pas du tout un aspect professionnel. Mais c'est un début !

#### 4. Liste de discussion pour développeurs

Nous avons également ouvert une *mailing list* pour les gens intéressés par le développement du projet G'MIC. Il y a très peu d'inscrits et très peu d'activité pour le moment, donc je me permets de faire un peu de pub pour ce mode de communication plus technique. [Ça se passe ici](#).

### Perspectives et conclusions.

Voilà, je pense avoir fait le tour des nouveautés réalisées depuis ces *4 derniers mois*. G'MIC est un projet **actif** (7 versions en 4 mois), avec encore plein de potentiel d'amélioration. On a encore quelques idées pas encore implémentées, j'espère que ça arrivera dans les prochaines versions.

Actuellement, le greffon G'MIC pour GIMP affiche 394 filtres. Il y en a en réalité plus de 700, mais nous avons arrêté de compter les filtres qui se trouvaient dans le répertoire **Testing/**, qui ne sont pas toujours bien finis. Le nombre de commandes compris par l'interpréteur dépasse 850, malgré une taille du binaire "raisonnable" (environ 5Mo).

Je finirais, comme d'habitude, par vous conseiller de l'installer et de le tester, si le traitement d'images vous intéresse. Sous Ubuntu, un gentil contributeur allemand propose un [PPA toujours mis à jour](#), malgré les versions fréquentes (merci à lui). L'interface en ligne de commande est à mon avis sous-exploitée, alors pourquoi ne pas tester `gmic -demo` et vous laisser séduire ? :)

Ah ! Et je salue votre courage d'avoir été jusqu'au bout de ce journal.

À la prochaine, pour de nouvelles aventures !

### Aller plus loin

■ ■ [Journal à l'origine de la dépêche](#) (339 clics)

#### Maths

Posté par [matheod](#) le 24/06/14 à 01:03. Évalué à 5. Dernière modification le 24/06/14 à 01:03.

| Ca doit être super intéressant du point de vue des mathématiques tout ça ^^

#### Bon boulot !

Posté par [xavier philippon](#) le 24/06/14 à 07:23. Évalué à 5.

| Je dirais même plus, Excellent travail !!!

#### Re: Bon boulot !

Posté par [reynum](#) ([site web personnel](#)) le 24/06/14 à 14:17. Évalué à 2.

| Je dirais même plus, Excellent travail !!!

Ouai c'est en lisant des dépêches comme celle là que je regrette de ne pas être un professionnel de l'image :-  
D