



Débuter avec The Gimp Version 2



Table des matières

Avertissement.....	3
But de ce document.....	5
Script-Fu.....	5
Greffons compilés sous forme de programmes .exe (environnement Windows).....	5
Programmes externes.....	5
« Plug-ins » pour Photoshop.....	5
Greffon pour activer certains « plug-ins » d'Adobe Photoshop.....	6
Panorama Tools , ensemble d'outils pour panoramas.....	7
Paramètres des filtres Adjust, Correct, Perspective, Remap.....	9
Fonction de perspective.....	9
Fonctions de conversion Remap.....	11
Fonctions de correction Correct.....	12
Liens autour de Panorama Tools.....	12
Assembler plusieurs photographies avec Hugin.....	13
PTOpenGui (Panorama Tools Open GUI).....	17
Utilitaire ControlPoints (Faciliter le repérage des points).....	18
Visualiser un panorama dans une page HTML avec PTViewer.....	19
Interactive Java Viewer for HDR-panoramas	23
Visualiser un panorama dans un fichier PDF avec PDFPanorama.....	26
Panoramas, liens divers (non classés).....	27
Pratique.....	28
Consulter les données contenues dans les images (EXIF, IPTC).....	30
Avec le greffon « EXIF Browser » pour Gimp.....	30
Avec XnView (visualiseur et convertisseur d'images universel).....	31
Corriger les yeux rouges ou oranges (prise de vue avec flash).....	32
Script-Fu Red Eye (red-eye.scm).....	32
Orange Eye (non testé).....	32
RedEye (non testé).....	32
Didacticiels.....	32
Script-Fu Camouflage (motif).....	35
Script-Fu Duotone (teinter).....	38
RawPhoto (traitement des images brutes).....	41

Sous Linux	41
Adaptation pour Windows.....	41
Conseil.....	41
Quelques liens.....	42
Greffon « resynthesizer »	43
Contenu du fichier README (en anglais).....	43
Linux.....	45
Windows.....	45
Exemple.....	45
Script-Fu smart-remove.scn.....	46
Script-Fu smart-enlarge.scn.....	48
PhymodMedia.....	51
Linux.....	51
Windows.....	51
Exemples.....	51
Créer des nuages animés	53
Effet de peinture à l'huile sur toile avec le Script-Fu thick-oil-paint.scn.....	55
« Extras » pour Gimp (brosses & motifs).....	57
Greffon Voronoi pour créer des motifs.....	59
Script-Fu layers-view (utilisation).....	60
Utiliser un Script-Fu pour créer un logo.....	63
Script-Fu photo-stack (effet).....	65
Script-Fu Corrosion (effet).....	68
Script-Fu ISO Noise Reduction (retouche photo).....	71
Script-Fu fire-logo (logo texte).....	74
Script-Fu Shadowy (logo texte).....	76
Insertion d'un © ou tout autre texte dans une bordure	79
Boutons Aqua pour pages Web.....	82
Logo avec texte (style iMac).....	88
Digital camera noise filter & Unsharp 2.....	90
Greffon SIOX pour faciliter les détourages.....	91
Traitement par lot, DBP (David's Batch Processor).....	93

Avertissement

Chaque visite sur un lien commercial (Publicité) à partir de notre page Web <http://www.aljacom.com/~gimp/> nous rapporte une petite contribution financière qui, au mois de septembre 2005, représente 7% du prix de revient d'un document téléchargé. Cette publicité permet de poursuivre la mise à jour et la création de nouvelles documentations.

Merci encore pour votre compréhension.

Ce document créé par Aljacom présente quelques fonctions du logiciel libre et gratuit Gimp Version 2.2.x

Vous pouvez, dans un but non commercial, distribuer, modifier des copies de ces pages selon :



This work is licensed under a [Creative Commons License](http://creativecommons.org/licenses/by-nc/2.0/).

<http://creativecommons.org/licenses/by-nc/2.0/>

:O)

Documentation réalisée par **ALJACOM**

<http://www.aljacom.com/~gimp/>

Outil de production :

<http://www.openoffice.org/>



Fichier PDF :

AFPL Ghostscript 8.51

<http://www.cs.wisc.edu/~ghost/>

PDFCreator 0.8.0

http://sector7g.wurzel6-webdesign.de/pdfcreator/index_en.htm

extendedPdf 1.3

<http://www.jdisoftware.co.uk/pages/jdi-about.php>

Historique des versions

Version du 10 septembre 2005	RawPhoto : UFRaw + Liens
Version du 24 août 2005	Traitement par lot, DBP
Version du 21 août 2005	Greffon SIOX pour détourages
Version du 19 mars 2005	Digital camera noise filter
Version du 29, 30 janvier 2005	Avertissement
Version du 6 janvier 2005	Pspi, Gimp 2.2.x
Version du 9 décembre 2004	Mise à jour ISO Noise Reduction
Version du 17 novembre 2004	Page des greffons de M. Schumacher
	Voronoi, red eye. Orange eye
Version du 16 novembre 2004	Technique script abcdugimp p 63
Version du 14 novembre 2004	Liens + copyright-border + Aqua + Logo iMac
Version du 13 novembre 2004	Script-Fu ISO Noise Reduction + fire-logo + Shadowy
Version du 12 novembre 2004	Script-Fu Corrosion
Version du 11 novembre 2004	Script-Fu pour créer un logo + photo-stack + nota page 5
Version du 10 novembre 2004	Script-Fu layers-view.scm
Version du 31 octobre 2004	Voronoi
Version du 30 octobre 2004	« Extras » pour Gimp
Version du 29 octobre 2004	PhymodMedia
Version du 28 octobre 2004	Greffon « resynthesizer »
Version du 15 octobre 2004	Liens + RawPhoto
Version du 10 septembre 2004	duotone
Version du 29 août 2004	Camouflage + Cor. + MaJ liens
Version du 26 août 2004	ajout EXIF Browser
Version du 15 juillet 2004	Origine du document

The GIMP Version 2 (PARTIE 4)

But de ce document

Les présentations dans ce document sont faites avec **Gimp 2.2.x** et **GTK 2.4.x** sous Windows XP.

Ce document présente des fonctionnalités ajoutées à Gimp par des greffons ou des logiciels.

Vous trouverez à la page <http://registry.gimp.org/index.jsp> (en anglais) ou <http://www.gimp-fr.org/greffon.php> (en français) des liens vers des greffons. Les explications sur les différents types de greffons sont données sur le lien précédent.

Simon Budig propose des explications sur les scripts-fu (en anglais) à partir de : <http://www.home.unix-ag.org/simon/gimp/guadec2002/gimp-plugin/html/>

Script-Fu

Les Scripts-Fu sont des fichiers programmes de type texte avec l'extension .scm

Sous Windows vous les trouverez dans le répertoire (voir le nota en bas de la page 5) :

C:\Program Files\GIMP-2.2\share\gimp\2.0\scripts

Dans ce document, sur chaque description de Script-Fu, vous retrouverez le nom de l'auteur ainsi que l'adresse du site où le script est disponible. Vous pouvez consulter ces adresses pour y trouver des informations complémentaires et aussi d'autres Scripts-Fu qui ne sont pas dans ce document.

Dans ce document le code source du script est de couleur rouge avec une police de petite taille. Vous pouvez augmenter la taille d'affichage dans Acrobat Reader pour faciliter la consultation. Tous les scripts présentés sont actifs sur l'image qui est sélectionnée ou par la fenêtre principale.

Greffons compilés sous forme de programmes .exe (environnement Windows)

Ces fichiers exécutables sont dans le répertoire C:\Program Files\GIMP-2.2\lib\gimp\2.0\plug-ins (nota en bas de la page 5). Le greffon EXIF Browser, par exemple, s'y trouve comme indiqué à la page 30.

Michael Schumacher propose beaucoup de greffons compilés pour Gimp 2 version Windows sur la page <http://schumaml.gmxhome.de/downloads/gimp/>

Programmes externes

AFPL Ghostscript est un exemple de programme que Gimp utilise pour ouvrir ou enregistrer des fichiers Postscript ou PDF. Son installation est décrite dans la 1^{ère} partie de notre documentation.

« Plug-ins » pour Photoshop

Ils sont prévus pour fonctionner avec Adobe Photoshop. Avec le greffon proposé à la page 6 certains peuvent fonctionner avec Gimp. Vous trouverez à <http://thepluginsite.com/resources/freeps.htm> une liste de « plug-ins » libres ou gratuits (le fonctionnement n'est pas garanti).

Nota sous Windows : Si vous ajoutez des Scripts, des Greffons, des Motifs, etc. vous pourrez les enregistrer dans le répertoire système indiqué **ou** dans votre répertoire personnel de Gimp accessible depuis C:\Documents and Settings.

Greffon pour activer certains « plug-ins » d'Adobe Photoshop

Tor Lillqvist a compilé ce greffon pour qu'il fonctionne avec Gimp 2 version Windows. Vous pouvez télécharger **gimp-pspi-1.0.3.zip** (N° de version disponible en juillet 2004) à la page :
<http://www.gimp.org/~tml/gimp/win32/downloads.html>

ou directement <http://www.gimp.org/~tml/gimp/win32/gimp-pspi-1.0.3.zip> puis vous décompressez de façon à placer **pspi.exe** dans le répertoire **C:\Program Files\GIMP-2.2\lib\gimp\2.0\plug-ins**

Créer un répertoire pour recevoir les « plug-ins » de Photoshop par exemple C:\ProgramFiles\GIMP-2.2\plugin_ps et **placer vos « plug-ins » dans ce répertoire**

Dans Gimp, configurer le répertoire où sont localisés les « plug-ins » de Photoshop

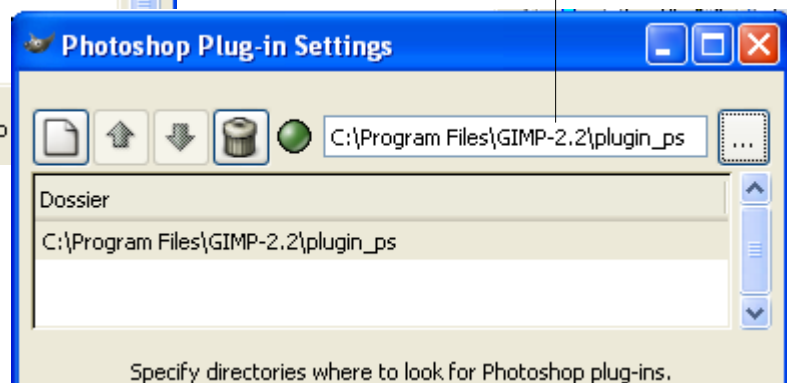
De la fenêtre Principale faire :



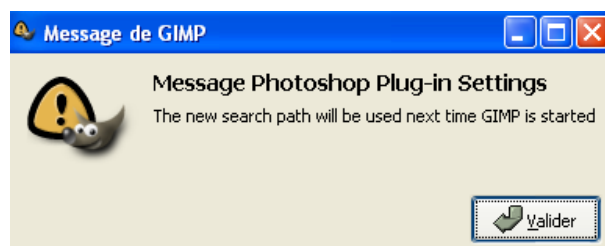
Indiquer le répertoire



Sélection



Valider



Place vos « plug-in » pour Photoshop dans le répertoire, relancer Gimp, l'installation est terminée.

Panorama Tools , ensemble d'outils pour panoramas

Vous trouverez à la page <http://www.tawbaware.com/maxlyons/pano12ml.htm> les indications sur ces outils créés par Helmut Dersch et modifiés par Max Lyons. Ces outils comprennent **un installateur pour Windows** (setup.exe) **que nous nous n'utilisons pas pour cet exemple**.

Le greffon prévu pour Gimp dans le paquet n'est pas compatible avec Gimp version 2.

Nous utiliserons les « plug-ins » prévus pour Photoshop.

Lien pour télécharger le fichier compressé : <http://www.tawbaware.com/maxlyons/panotoolsml.zip>

Contenu de panotoolsml.zip version 2.6.ML12 :

Nom	Type	Taille d...	A u...	Taille	T...	Date
PanoTools	Dossier de fic...	0 Ko		514 Ko	10...	2003-08-18 23:35
PTAssembler.txt	Document texte	1 Ko	Non	1 Ko	30%	2002-11-26 00:58
readme.txt	Document texte	1 Ko	Non	2 Ko	56%	2003-08-19 00:01
setup.exe	Application	512 Ko	Non	541 Ko	6%	2003-08-25 23:53

Répertoire PanoTools de panotoolsml.zip

Nom	Type	Taille d...	A u...	Taille	T...	Date
Copyright	Dossier de fic...	0 Ko		514 Ko	10...	2003-06-14 13:31
Gimp	Dossier de fic...	0 Ko		514 Ko	10...	2003-06-14 13:31
Helpers	Dossier de fic...	0 Ko		514 Ko	10...	2003-06-14 13:31
Photoshop Plugin	Dossier de fic...	0 Ko		514 Ko	10...	2003-08-25 23:49
Sources	Dossier de fic...	0 Ko		514 Ko	10...	2003-08-25 23:48
pano12.dll	Extension de ...	234 Ko	Non	474 Ko	51%	2003-08-25 23:48
ptcrypt.jar	Executable J...	6 Ko	Non	6 Ko	0%	2001-12-02 11:42
pteditor.jar	Executable J...	162 Ko	Non	162 Ko	0%	2001-12-02 11:42
Readme.html	HTML Document	6 Ko	Non	16 Ko	67%	2001-12-02 13:41
readme_fixed.txt	Document texte	1 Ko	Non	1 Ko	35%	2003-08-19 00:00
readme_patch_Lyons.txt	Document texte	2 Ko	Non	4 Ko	54%	2003-08-25 23:44

Répertoire Photoshop Plugin de panotoolsml.zip

Nom	Type	Taille d...	A u...	Taille	T...	Date
adjust.8bf	Adobe Photoshop Plugin	4 Ko	Non	9 Ko	54%	2001-12-02 11:43
correct.8bf	Adobe Photoshop Plugin	4 Ko	Non	9 Ko	59%	2001-12-02 11:42
PanActions.atn	Fichier ATN	1 Ko	Non	1 Ko	66%	1999-01-26 20:23
perspect.8bf	Adobe Photoshop Plugin	4 Ko	Non	9 Ko	54%	2001-12-02 11:43
Readme.txt	Document texte	11 Ko	Non	31 Ko	66%	2000-09-05 19:17
remap.8bf	Adobe Photoshop Plugin	4 Ko	Non	9 Ko	54%	2001-12-02 11:43

Vous décompressez le fichier **panotoolsml.zip** pour que ces fichiers s'installent de la manière suivante :

- pano12.dll dans le répertoire **C:\windows\system**
- les fichiers du répertoire Photoshop Plugin de panotoolsml.zip dans le répertoire créé pour les «plug-ins» de Photoshop à la page 6 (C:\ProgramFiles\GIMP-2.2\plugin_ps).

Voici un lien vers un **très beau site français d'Arnaud Frich sur les panoramas** et toutes ses techniques : <http://www.arnaudfrichphoto.com/>

Les techniques d'assemblage des images : <http://www.arnaudfrichphoto.com/gpass7.htm>

Vous trouverez sur le site d'Adrien <http://adrien.bphoto.free.fr/> des didacticiels (qui peuvent être adaptés à Gimp) autour de PanoramaTools.

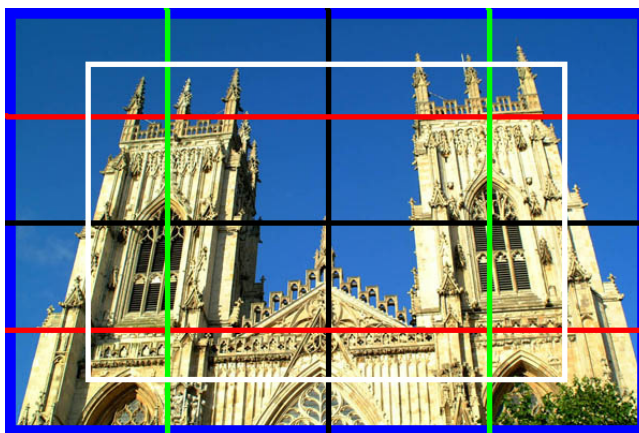
Thomas Niemann propose une très bonne documentation (pdf), en anglais, sur :
<http://www.epaperpress.com/pano/>

Dans les exemples qui suivent nous utilisons une photographie de la cathédrale York Minster faite par Ian Britton et disponible à la page <http://www.freefoto.com/preview.jsp?id=05-05-8>.

Une mire (Rouge, Vert, Bleu, Blanc, Noir) sera ajoutée à cette image pour mieux comprendre les effets des « plug-ins » Photoshop de Panorama Tools.

Vous pouvez télécharger la mire sur ce lien : [mires_papillon_wilber.zip](#)

Résultat :



Pour un fonctionnement correct l'image doit être aplatie par Image, Aplatir l'image.

L'utilisation, les paramètres de « plug-ins » sont dans le fichier Readme.txt

Sur certains filtres vous pouvez sauvegarder vos paramètres dans un fichier avec l'option Save et les recharger avec l'option Load

Nota pour les utilisateurs de Pixia :

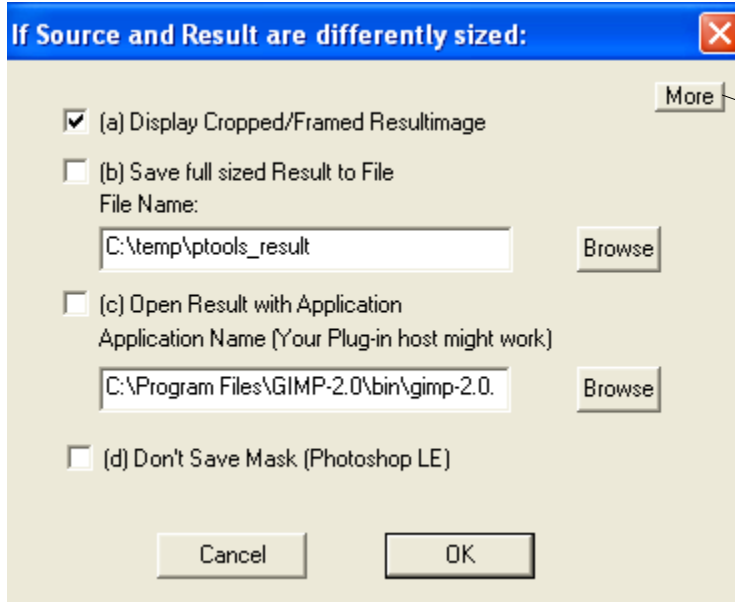
Les filtres Panorama Tools avec l'extension .8bf fonctionnent bien avec Pixia et le « plug-in » Ebisawa disponible à <http://pixia.free.fr/telech.htm>

Paramètres des filtres Adjust, Correct, Perspective, Remap

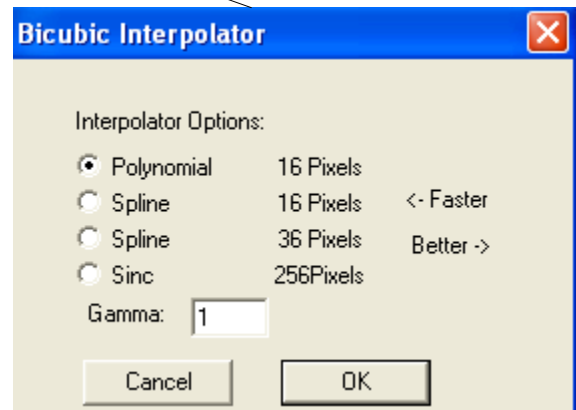
Chaque filtre est accessible sur l'image par Filtres, Panarama Tools.



Sélectionner un des filtres, Appuyer sur le petit bouton « prefs » en haut à droite de la fenêtre :



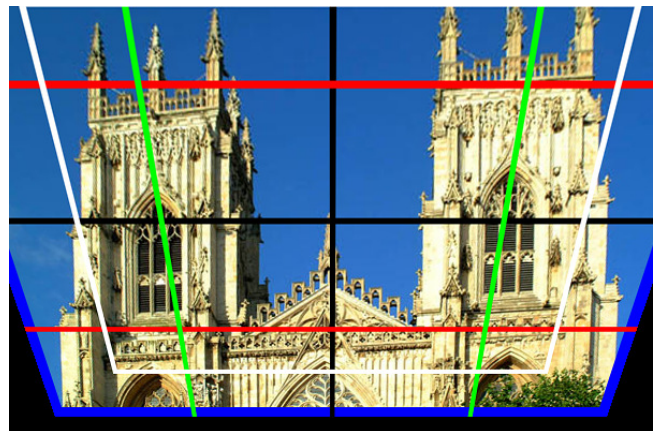
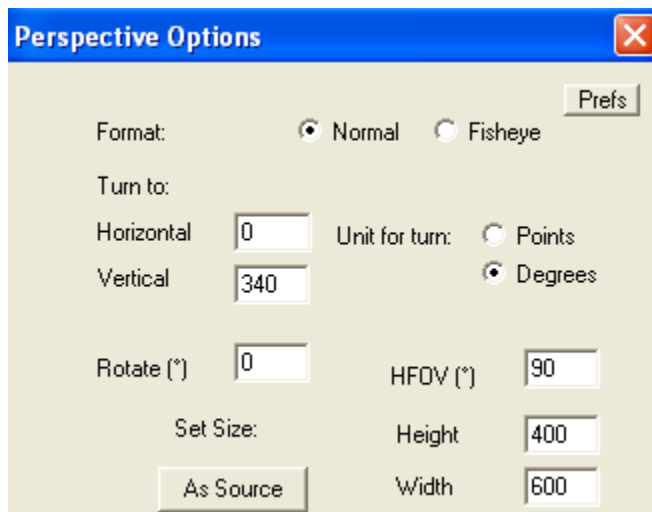
Qualité de l'interpolation



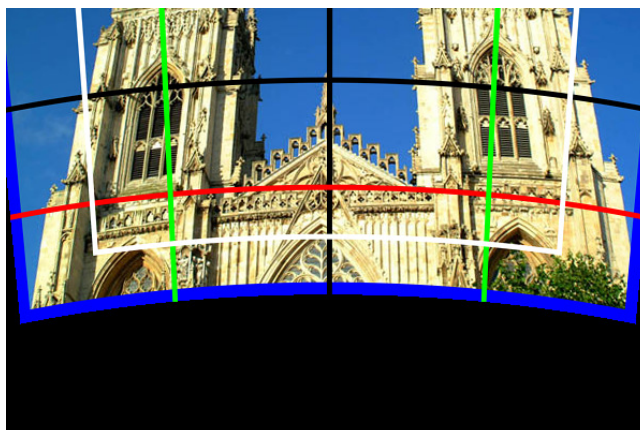
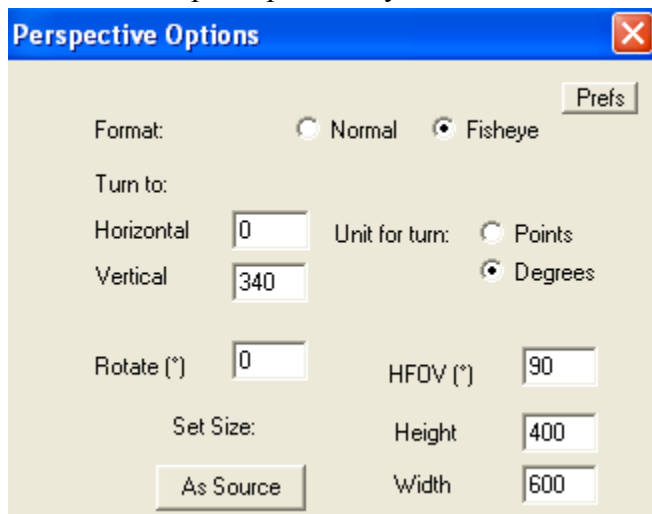
Fonction de perspective

Sur l'image faire : Filtre, PanoramaTools, Perspective.

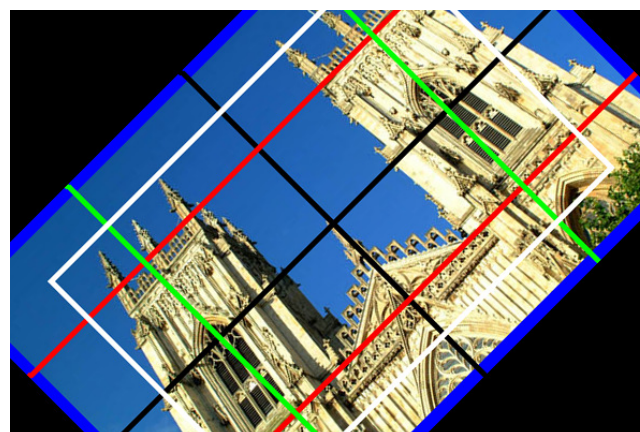
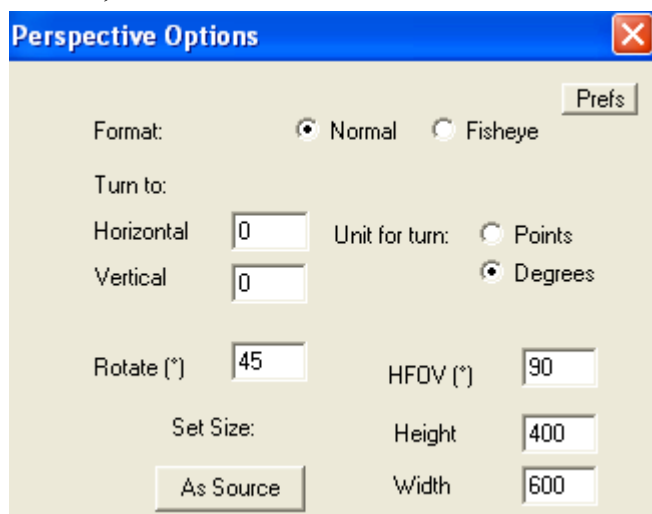
Ici HFOV est déterminé de façon empirique et dépend de l'objectif utilisé lors de la prise de vue.



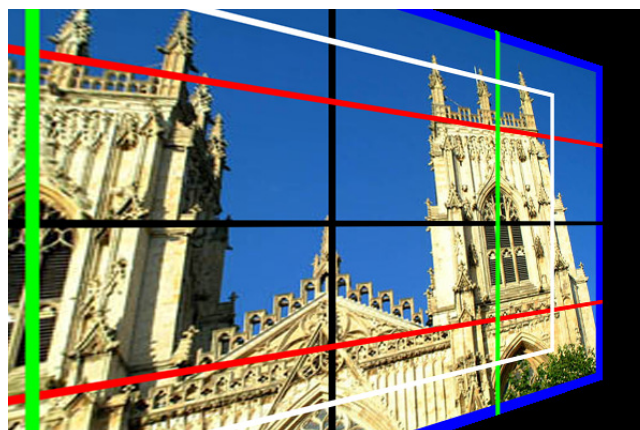
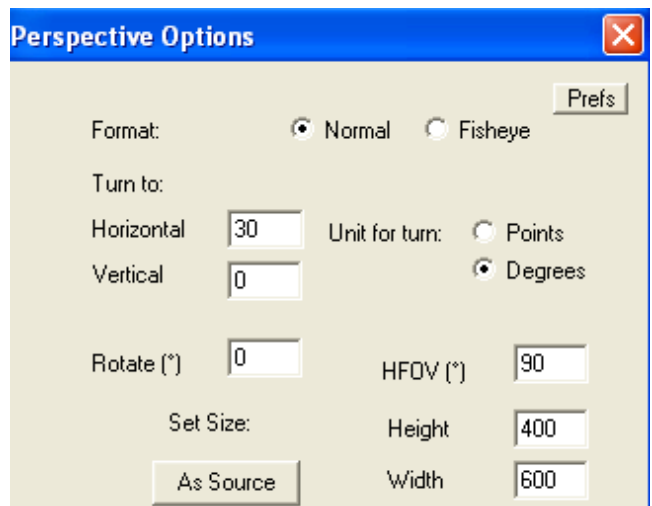
Normal et remplacé par Fisheye



Normal, Rotate

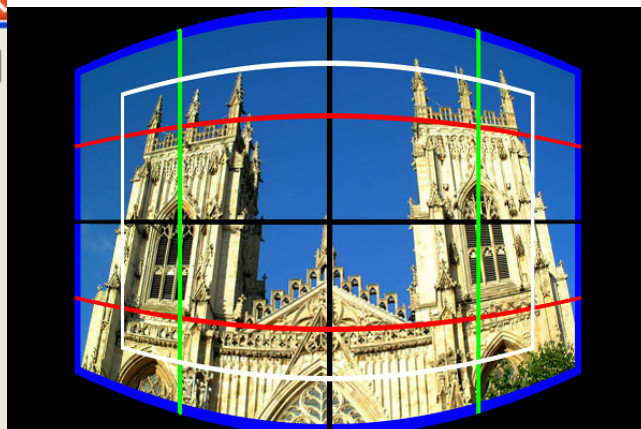
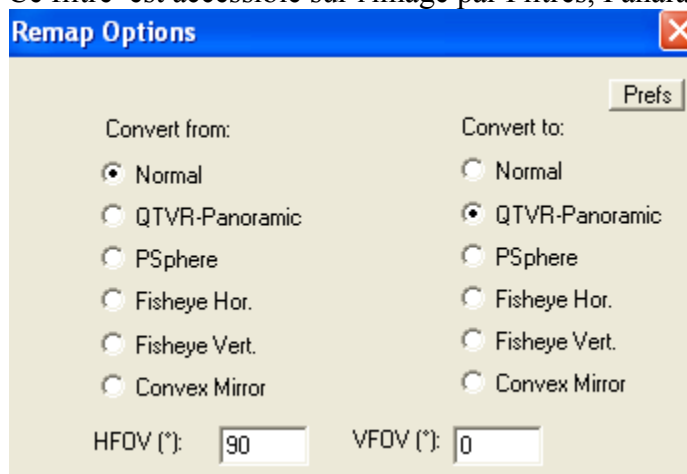


Horizontal

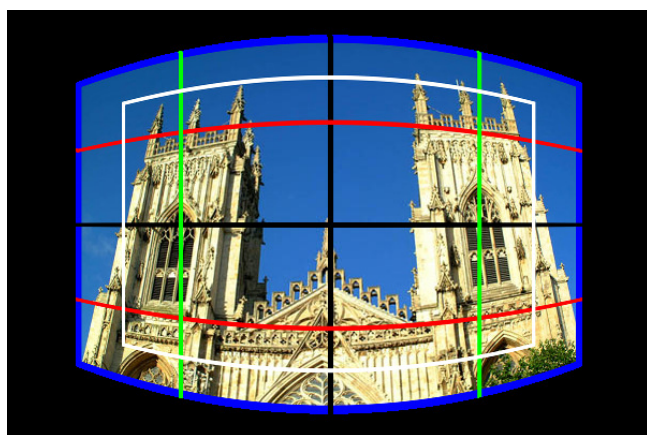


Fonctions de conversion Remap

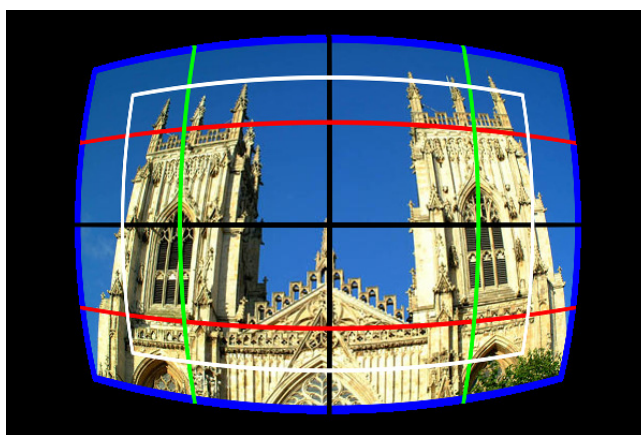
Ce filtre est accessible sur l'image par Filtres, Panarama Tools, Remap



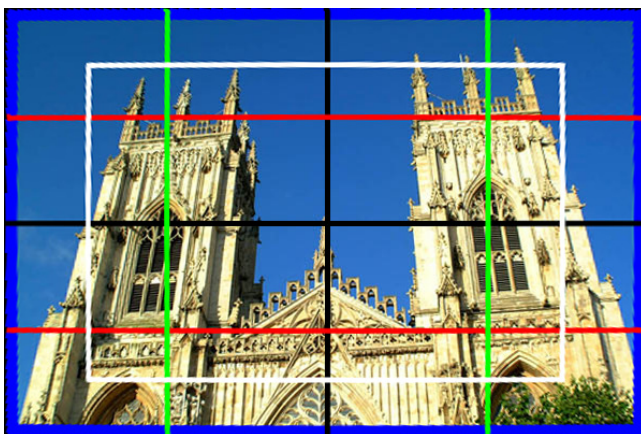
Normal vers PSphere



Normal vers Fisheye Hor.



Pour transformer cette image sensiblement à l'origine faire Fisheye Hor vers Normal, HFOV = 113. Résultat :

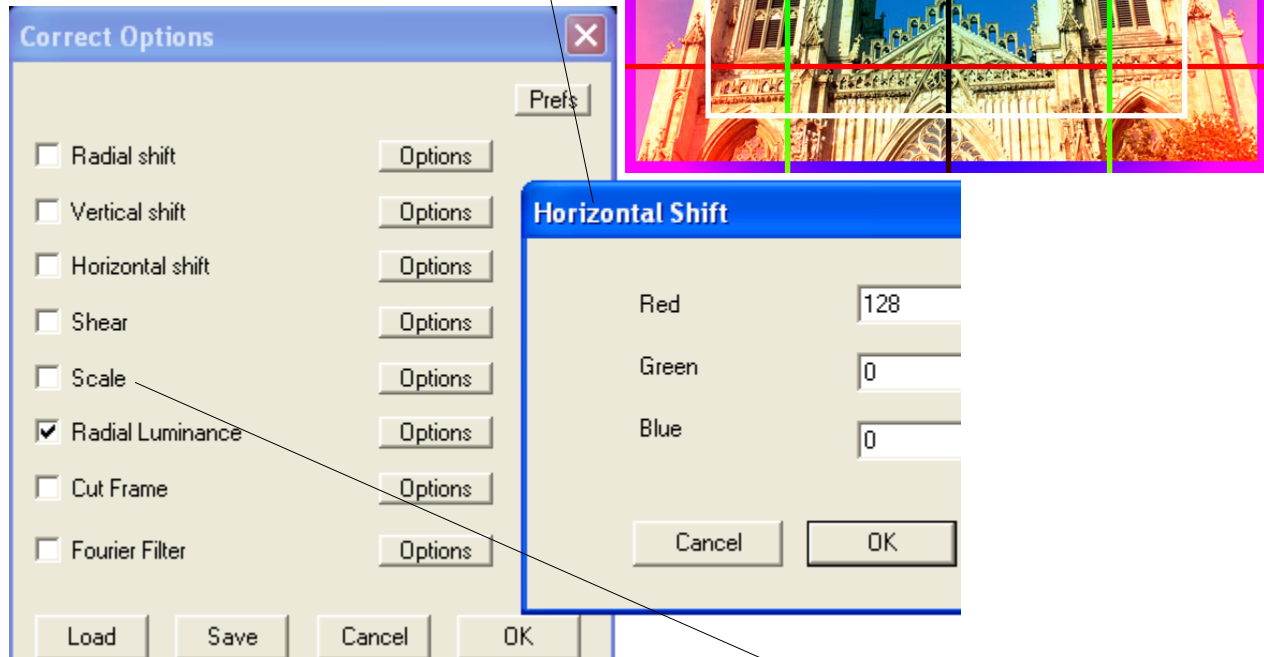


Fonctions de correction Correct

Sur l'image, Filtres, Panorama Tools, Correct. Choisissez le type de correction. S'ouvre alors une autre fenêtre pour paramétrer. Les options sont cumulatives.

Nota : Le nom de cette fenêtre n'est pas bon.

C'est bien « Radial Luminance »



Le filtre de mise à l'échelle Scale possède des qualités remarquables.

Liens autour de Panorama Tools

<http://groups.yahoo.com/group/PanoTools/> : Groupe.

<http://www.path.unimelb.edu.au/~bernardk/tutorials/360/index.html> Documentation, PTVIEWER

<http://hugin.sourceforge.net/> : Graphical User Interface pour Panorama Tools (Linux - Windows).

<http://ptfaq.org/fom-serve/cache/1.html> : FAQ sur Panorama Tools.

<http://perso.club-internet.fr/lionel.maraval/photos/technique/technique.htm> : Exemples d'utilisation.

<http://www.philohome.com/panotutorial/panoramatutorial1.html>: Didacticiel en anglais.

<http://epaperpress.com/pano/> : Didacticiel en anglais.

<http://home.no.net/dmaurer/%7Edersch/Index.htm> : Autres ressources logicielles

<http://www.panoguide.com/products/dersch/panotools/> : Présentation générale en anglais.

<http://www.ptgui.com/> : Graphical User Interface pour Panorama Tools (Payant).

<http://www.tawbaware.com/>: Graphical User Interface pour Panorama Tools (Payant)

<http://www.dplinux.org/twiki/bin/view/DPLinux/PanoramaTools?skin=print> : Linux

<http://sourceforge.net/projects/ptbcbgui/> : GUI pour Panorama Tools (Windows, Linux avec Wine).

<http://panotools.sourceforge.net/>

Assembler plusieurs photographies avec Hugin

Dans cet exemple nous n'utilisons pas Gimp pour réaliser cet assemblage d'images car l'utilisation de GUI pour Panorama Tools est beaucoup plus simple.

Cet interface graphique pour Panorama Tools fonctionne sous Linux et Windows.

Le site d'Hugin : <http://hugin.sourceforge.net/> est très complet et propose des didacticiels en anglais et des traductions en français.

• Installation sous Windows

Nous utilisons Panorama Tools utilisé dans les exemples précédents panotoolsml.zip version 2.6.ML12 dont le téléchargement est indiqué page 7.

Compléter l'installation avec setup.exe, voici les fichiers installés :

	Nom	Taille	Type	Date de modification
PanoTools	PTOptimizer.exe	6 Ko	Application	2001-12-02 11:43
	PTStitcher.exe	50 Ko	Application	2001-12-02 11:43
	readme	1 Ko	Raccourci Internet	2004-07-12 05:51
	readme_patch_Lyons.txt	4 Ko	Document texte	2003-08-25 23:44
	unins000.dat	2 Ko	Fichier DAT	2004-07-12 05:51
	unins000.exe	70 Ko	Application	2003-04-14 03:00

Pour l'étape suivante vous rendre sur le site de Andrew Mihal à la page <http://www-cad.eecs.berkeley.edu/~mihal/enblend/index.htm> et télécharger Emblend version Windows <http://www-cad.eecs.berkeley.edu/~mihal/enblend/enblend-1.3.zip>

Décompresser le fichier Zip pour installer enblend.exe dans le répertoire C:\Program Files\PanoTools.

Pour la dernière étape vous rendre à <http://hugin.sourceforge.net/snapshots/windows/> et télécharger la dernière version d'Hugin (au 12 juillet 2004 : hugin_2004_07_03-13_53_win32.zip).

Décompresser le fichier Zip => création du sous-répertoire hugin-0.4 (avec cette version).

Vous allez dans ce sous-répertoire et lancez hugin.exe. À la première utilisation agrandissez la fenêtre.

Pour assembler 2 images vous pouvez utiliser le didacticiel en français disponible sur le site d'Hugin : <http://hugin.sourceforge.net/tutorials/two-photos/fr.shtml>

Ce didacticiel n'est pas tout à fait à jour et demande une très, très légère adaptation.

Sur la version testée dans la fenêtre « stitcher » il y a un léger décalage entre la sélection sur la liste des formats d'image, par exemple si vous souhaitez avoir le résultat au format Photoshop avec Layers vous devez sélectionner PAN.

Conseil : Vous pouvez sauvegarder aussi votre projet en faisant File, Save. Ceci va créer un fichier de scripts au format texte .pto

Voici un contenu possible d'un fichier .pto avec les 2 images de l'exemple du site d'Hugin :

```
# PTOptimizer script, written by hugin

p f2 w2061 h1302 v87 k0 n"PSD_mask"
m g1 i0 f1

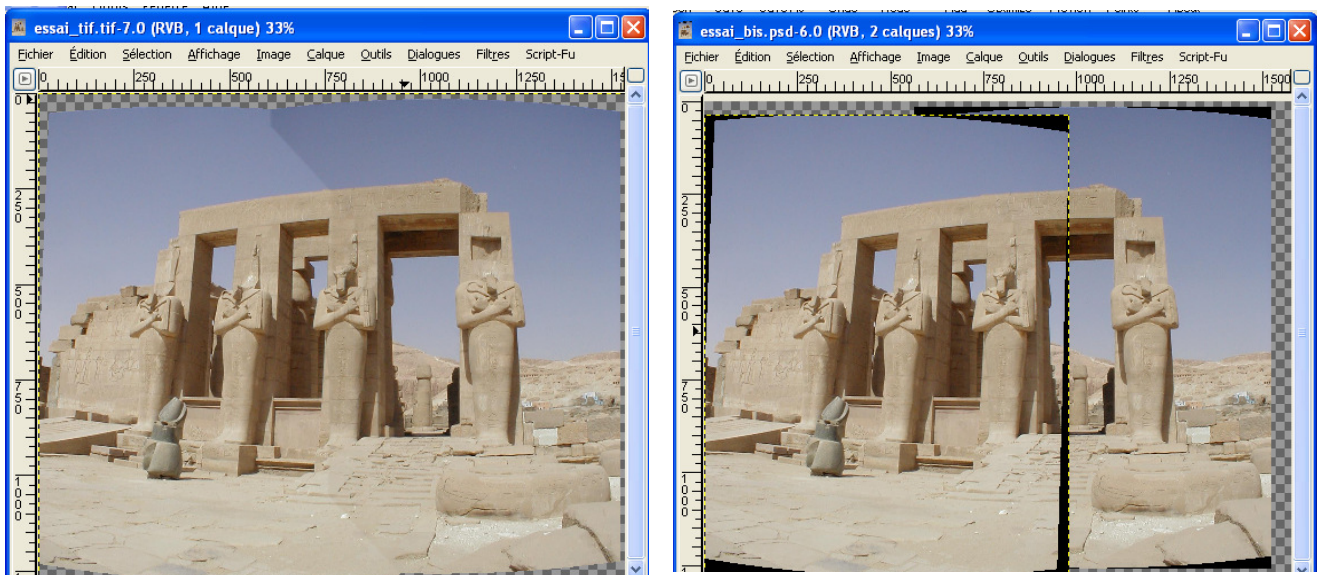
# image lines
i w960 h1280 f0 a0 b0 c0 d0 e0 g0 p0 r0 t0 v40.524 y0 u10 n"045.jpg"
i w960 h1280 f0 a=0 b=0 c=0 d=0 e=0 g=0 p1.62615 r-1.4128 t=0 v=0 y23.6748 u10 n"046.jpg"

# specify variables that should be optimized
v p1 r1
v y1
v

# control points
c n0 N1 x868 y283 X329 Y330 t0
c n0 N1 x891 y868 X332 Y896 t0
c n0 N1 x794 y1090 X235 Y1119 t0
c n0 N1 x942 y1229 X371 Y1236 t0
c n0 N1 x913 y527 X361 Y567 t0

#hugin_options r0
```

Vous pouvez ouvrir les résultats de vos essais dans Gimp. Voici 2 exemples de résultats : TIF sans calque et PSD avec calques :



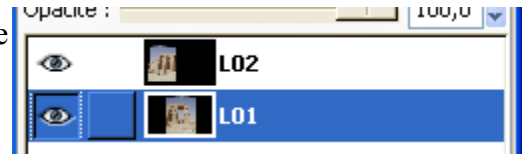
On remarque que Gimp ouvre mal le format PSD avec Layers (avec ou sans masque) avec les grosses zones noires. Malgré tout c'est exploitable. Dans l'exemple suivant nous allons modifier l'image PSD avec Layers issue de Hugin et l'enregistrer au format XCF.

- Adapter une image PSD issue d'Hugin pour Gimp

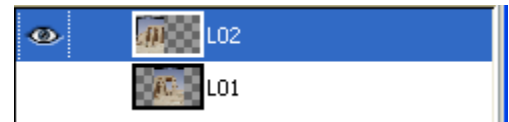
Ouvrir l'image avec calques dans Gimp. L'image créée dans l'exemple précédent comporte 2 calques.

Sur la fenêtre Calque, Canaux, Chemins, sélectionner le calque du bas L01.

Faire un clic droit sur ce calque et sélectionner Ajouter un canal Alpha.

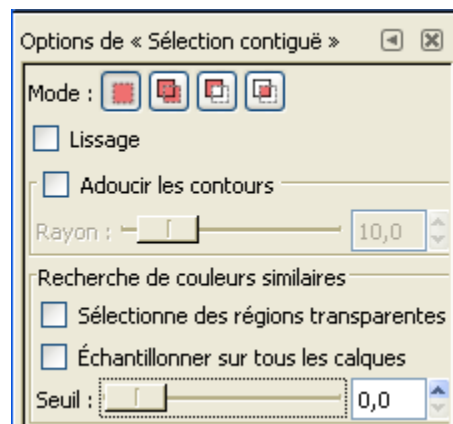


Sélectionner le calque L02, faire un clic droit sur ce calque et sélectionner Ajouter un canal Alpha.

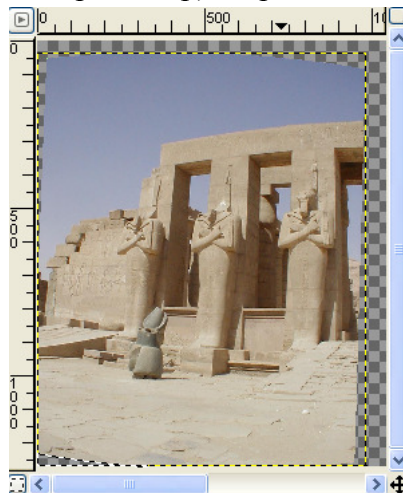


Rendre le calque L01 invisible en faisant un clic sur l'oeil.

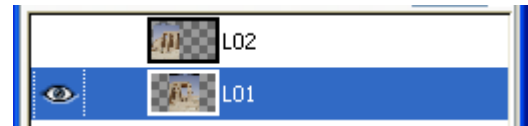
Aller sur la fenêtre principale de Gimp, activer la baguette magique ou Sélection Contiguë (Raccourci Z). Paramétrer les options de l'outil comme suit :



Activer le fenêtre de l'image, appliquer la baguette magique sur la zone noire, effacer en faisant Ctrl+K. La zone devient transparente (Cette manipulation est valable si la photographie ne comporte pas de pixels noirs adjacents à la zone noire créée par Gimp). Répéter cette manipulation sur toutes les zones noires.

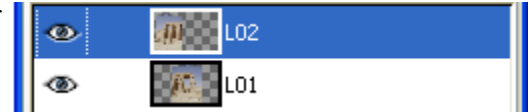


Se rendre sur la fenêtre Calque, Canaux, Chemins, sélectionner le calque du bas L01. Rendre ce calque visible. Mettre le calque L02 invisible.

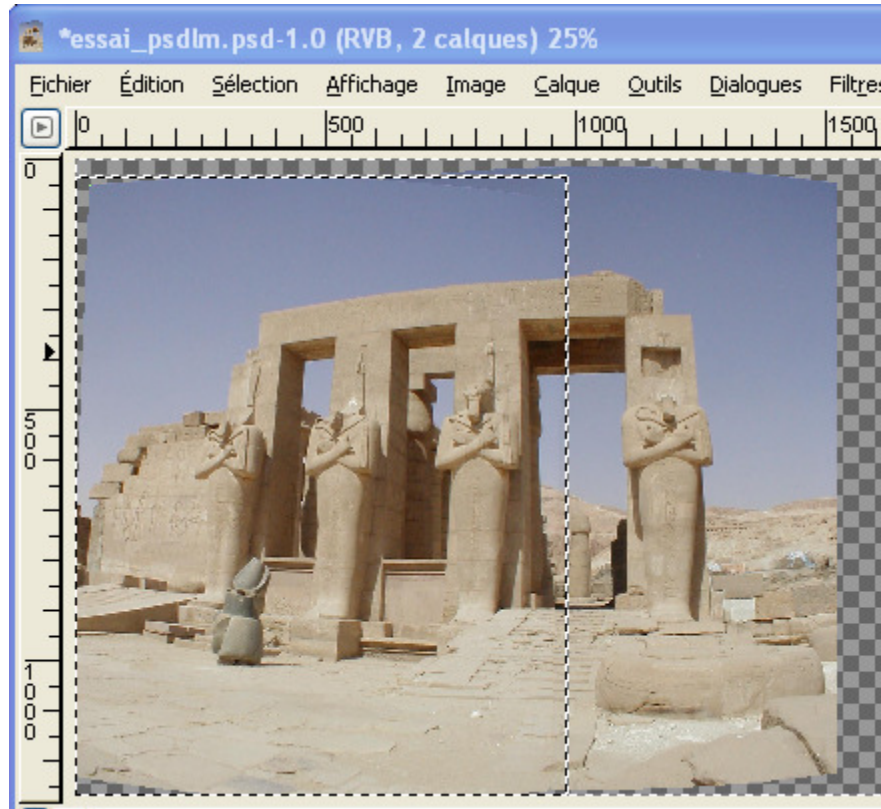


Répéter l'opération avec la baguette magique.

Se rendre sur la fenêtre Calque, Canaux, Chemins sélectionner le calque du bas L02 et rendre ce calque visible.



Résultat :



Enregistrer ce fichier au format XCF de Gimp.

Il est aussi possible d'utiliser ces zones noires pour créer des sélections, etc...

Il y a une différence entre l'image TIF sans calque et les image PSD . Elle provient du logiciel Hugin qui fait un traitement différent entre les types d'images : image aplatie, image avec calques, image avec calques et masques de fusion. D'autre part Gimp ne sait pas restituer correctement les images PSD Photoshop avec calques et masques de fusion (c'est ce qui provoque les parties noires enlevées dans cet exemple).

PTOpenGui (Panorama Tools Open GUI)

Voici un autre très bon GUI (Windows, Linux avec Wine) pour Panorama Tools décrit à <http://www.nic.fi/~juhe/ptbcbgui/>.

La page des téléchargements est : <http://sourceforge.net/projects/ptbcbgui/>

PTOpenGui fonctionne avec Imagemagick disponible à <http://www.imagemagick.org/>

Vous devez télécharger deux fichiers :

Contenu de PTOpenGui.zip

Nom	Type	Taille d...	A u...	Taille	T...	Date
PTOpenGui.exe	Application	171 Ko	Non	414 Ko	59%	2003-09-04 08:01
ReadMe.txt	Document texte	1 Ko	Non	2 Ko	49%	2003-05-08 15:37

Contenu de PTOpenGuiDll.zip

Nom	Type	Taille d...	A u...	Taille	T...	Date
cp3245mt.dll	Extension de ...	316 Ko	Non	888 Ko	65%	1999-01-27 05:00
vcl40.bpl	Fichier BPL	597 Ko	Non	1 84...	68%	1999-01-07 05:02
vcljpg40.bpl	Fichier BPL	52 Ko	Non	105 Ko	51%	1998-10-22 05:01
vclsm40.bpl	Fichier BPL	24 Ko	Non	70 Ko	66%	1998-10-22 05:01
vclx40.bpl	Fichier BPL	77 Ko	Non	247 Ko	70%	1998-06-18 05:00

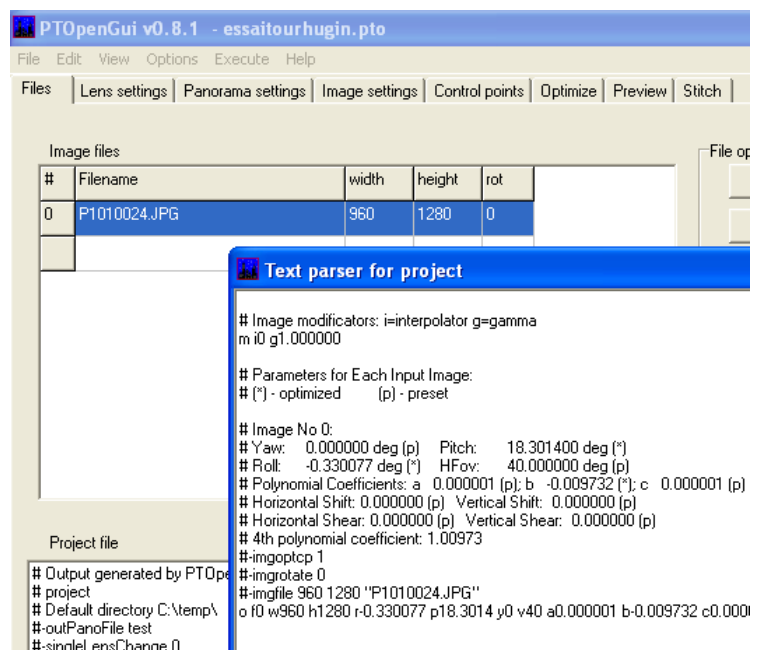
Décompresser les deux fichiers . Zip (dans C:\Program Files\PanoTools par exemple) et lancer PTOpenGui.exe

Vous pouvez, par exemple, charger le fichier .pto décrit à la page 14.

Voici un autre script .pto issu de l'exemple <http://hugin.sourceforge.net/tutorials/architectural/fr.shtml> et importé dans PTOpenGUI :

```
# PTOptimizer script, written by hugin
p f2 w1152 h2112 v48 n"JPEG q90"
m g1 i0 f0
# image lines
i w960 h1280 f0 a0 b-0.009732 c0 d0 e0 g0
p18.3014 r-0.330077 t0 v40 y0 u10
n"P1010024.JPG"
# specify variables that should be optimized
v b0 p0
v r0
v
# control points
c n0 N0 x281 y406 X246 Y1027 t1
c n0 N0 x453 y278 X443 Y1020 t1
c n0 N0 x752 y419 X783 Y886 t1
c n0 N0 x603 y348 X621 Y1055 t1
c n0 N0 x109 y1082 X102 Y1170 t1
#hugin_options r0
```

et son résultat importé dans PTOpenGUI
(Ce script sera modifié par le programme)



Utilitaire ControlPoints (Faciliter le repérage des points)

Voici un outil très pratique sous Windows (Sous Linux un programme équivalent PanoPoints existe <http://panopoints.sourceforge.net/>) pour le repérage des points entre 2 images. Il simplifie l'utilisation des GUI utilisant Panorama Tools. Il génère un fichier texte dont le contenu peut être incorporé aux fichiers de scripts .pto.

Lien vers la page où télécharger ControlPoints:

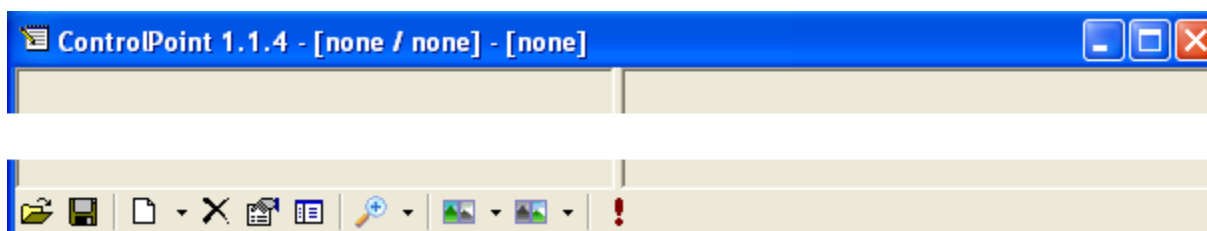
<http://www.smartr.easynet.be/controlpoints/>

Contenu de controlpoints114.zip

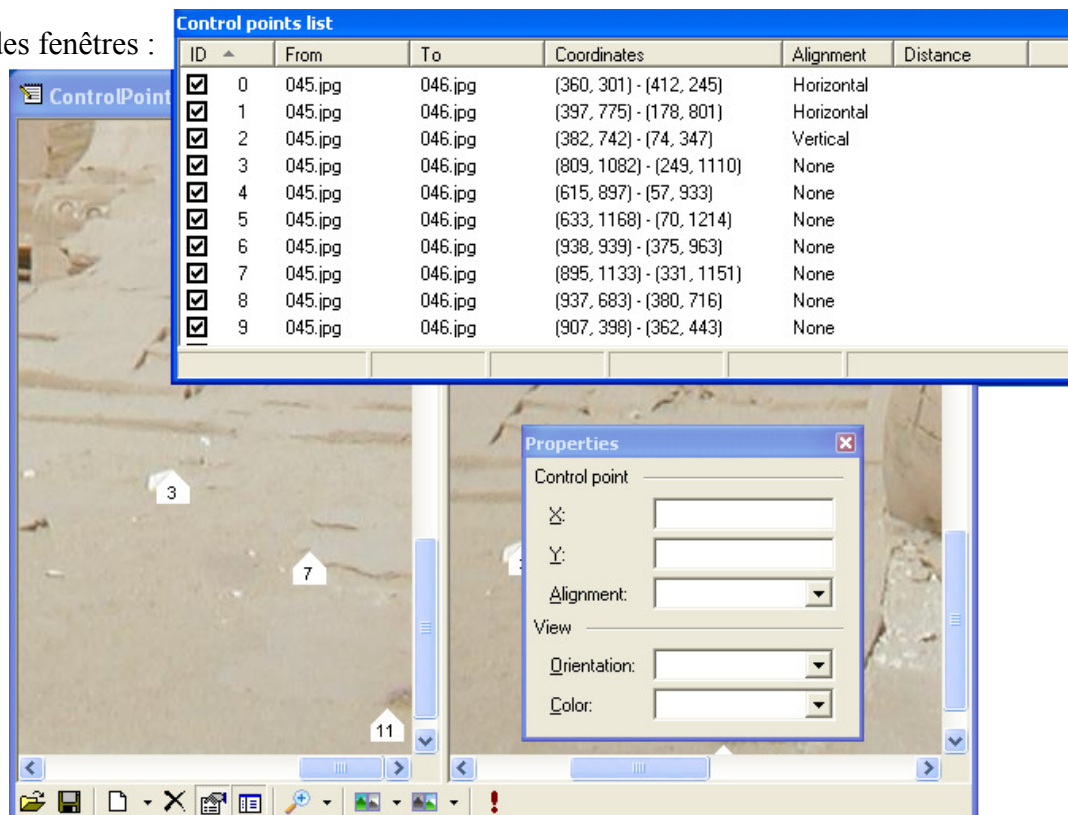
Nom	Type	Taille d...	A u...	Taille	T...	Date
 ControlPoints.exe	Application	409 Ko	Non	875 Ko	54%	2003-04-21 19:57

Décompresser le fichier . Zip (dans C:\Program Files\PanoTools par exemple) et lancer ControlPoints.exe

L'utilisation est très simple et se fait en utilisant les boutons en bas de la fenêtre.



Exemple des fenêtres :



Visualiser un panorama dans une page HTML avec PTVIEWER

Télécharger, par exemple, la dernière version proposée par Ben Kreunen de PTVIEWER (**applet Java**) à la page : <http://www.path.unimelb.edu.au/~bernardk/tutorials/360/ptviewer/files.html> (Lien pour un téléchargement direct : <http://www.path.unimelb.edu.au/~bernardk/tutorials/360/download/ptvj.zip>)

Nota : Voici le lien pour télécharger le didacticiel complet en anglais (17Mo) qui est une image du site Web : <http://www.path.unimelb.edu.au/~bernardk/tutorials/360/download/360.zip>

PTVIEWER créé par Helmut Dersch affiche correctement 2 types de panoramas : « equirectangular » et « rectilinear ». Les panoramas cylindriques doivent être convertis en « equirectangular ». Il utilise 2 types de coordonnées : absolues et relatives (détails sur le site ou le didacticiel).

Vous décompressez le fichier zip téléchargé. Pour l'exemple nous créons un sous-répertoire **essaiptv**. Copier les fichiers **.jar** (compressé) et **.class** du fichier zip décompressé du sous-répertoire **Applet & Class** vers le sous-répertoire **essaiptv** créé pour cet exemple.

Pour les exemples nous utilisons l'image panoramique quadrillée disponible sur le site à l'adresse : <http://www.path.unimelb.edu.au/~bernardk/tutorials/360/ptviewer/vr/pano.jpg> sauvegardée dans le sous-répertoire **C:\essaiptv**.



- Le plus simple à réaliser

Dans cet exemple on affichera l'image du panorama (pano.jpg) dans une fenêtre de 360 pixels * 240 pixels.

Voici le code HTML et la configuration de l'applet Java

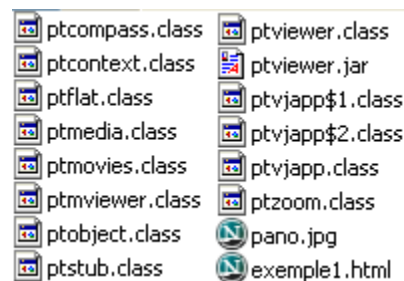
```
<html><head></head><body>

<applet archive="ptviewer.jar" code="ptviewer.class" width="360" height="240">
<param name="file" value="pano.jpg">
</applet>

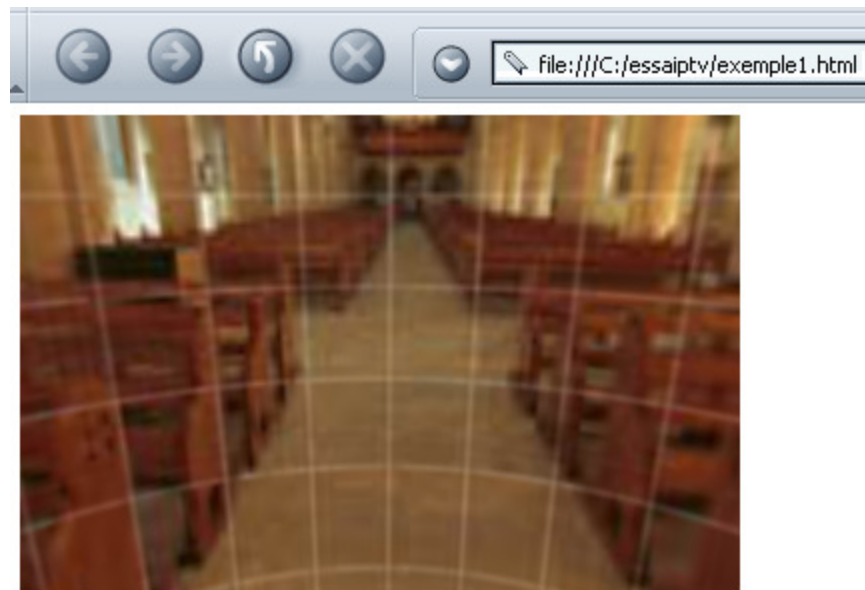
</body></html>
```

Sauvegarder ce code en exemple1.html.

Voici le contenu du sous-répertoire essaipvtv :



Ouvrir le fichier exemple1.html dans votre navigateur. Sur l'image, glisser le curseur en maintenant une touche de la souris enfoncée :



- Paramètres de l'applet java

Vous trouverez à la page d'Helmut Dersch <http://webuser.fh-furtwangen.de/~dersch/PTVJ/doc.html> une liste de tous les paramètres de PTVier ainsi que ses interactions avec le langage JavaScript..

Dans l'exemple suivant nous créons une page HTML avec :

- 2 panoramas de dimensions différentes (pour des raisons pratiques nous utilisons la même image pano.jpg).
- 1 image d'un logo qui s'affiche dans les 2 panoramas de 30*26 pixels. 
- 1 lien vers une page Web dans le second panorama.

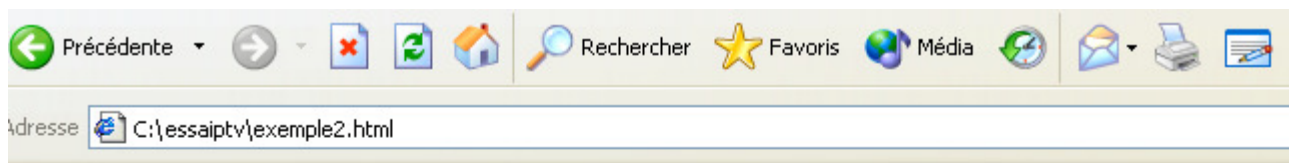
Voici le code HTML et la configuration de l'applet Java

```
<html><head></head><body>

<applet archive="ptviewer.jar" code="ptviewer.class" width="360" height="240">
<param name="file" value="pano.jpg">
<PARAM name=roi0 value=" i'logo.jpg' x250 y120 ">
</applet>
<applet archive="ptviewer.jar" code="ptviewer.class" width="270" height="180" mayscript="true">
<param name="file" value="pano.jpg">
<PARAM name=roi0 value=" i'logo.jpg' x320 y120 ">
<param name=shotspot0 value="x0 y0 a600 b300 u'http://webuser.fh-furtwangen.de/~dersch/PTVJ/doc.html' ">
</applet>

</body></html>
```

Sauvegarder ce code en exemple2.html.



- Ajouter des boutons de défilement dans le panorama

L'exemple est proposé par Helmut Dersch à : <http://webuser.fh-furtwangen.de/%7Edersch/pdfpanorama/Readme.html> .

Pour le réaliser il est nécessaire d'avoir **une image des boutons de commande** <http://www.fh-furtwangen.de/~dersch/pdfpanorama/Control.gif> et un exemple de panorama <http://www.fh-furtwangen.de/~dersch/pdfpanorama/Monp.jpg> à sauvegarder dans le bon répertoire.

Voici le code HTML et la configuration de l'applet Java

```
<html><head></head><body>

<APPLET archive=ptviewer.jar code=ptviewer.class name=ptviewer width=300 height=240>
  <PARAM name=shotspot0 value=" x90 y226 a120 b240 u'ptviewer:panDown()' ">
  <PARAM name=shotspot1 value=" x120 y226 a150 b240 u'ptviewer:panUp()' ">
  <PARAM name=shotspot2 value=" x150 y226 a180 b240 u'ptviewer:panLeft()' ">
  <PARAM name=shotspot3 value=" x180 y226 a210 b240 u'ptviewer:panRight()' ">
  <PARAM name=pan value=-10>
  <PARAM name=tiltmax value=30>
  <PARAM name=frame value="Control.gif">
  <PARAM name=file value="Monp.jpg">
</APPLET>

</body></html>
```

Sauvegarder ce code en exemple3.html.



Vous pouvez regarder le code source de la page <http://www.sudvision.com/previsit/panorama2.htm> comme autre exemple.

Interactive Java Viewer for HDR-panoramas

Le professeur Helmut Dersch propose la version **3.12 de PTViewer** à la page <http://webuser.fh-furtwangen.de/%7Edersch/>

Cette version 3.12 fonctionne avec des images « classiques » et des images au format FJPG. L'applet fait varier la luminosité. Le volume de ces images est augmenté.

Créer un **nouveau** répertoire **C:\essaihdr** . Télécharger <http://webuser.fh-furtwangen.de/%7Edersch/hdr/ptviewer.jar> , le sauvegarder dans le nouveau répertoire. Décompresser ce fichier pour obtenir les .class (sous Windows 7-Zip réalise cette opération sans problème).

Le fonctionnement est détaillé par un exemple à la page <http://webuser.fh-furtwangen.de/%7Edersch/hdr/hdrtest1.html> . Un autre exemple est donné à <http://webuser.fh-furtwangen.de/%7Edersch/hdr/hdrtest2.html>

Voici un code HTML des 2 exemples regroupés et la configuration de l'applet de ces pages :

```
<html><head></head><body>

<li>exemple 1

<li>page :
<a href="http://webuser.fh-furtwangen.de/%7Edersch/hdr/hdrtest1.html">
http://webuser.fh-furtwangen.de/%7Edersch/hdr/hdrtest1.html</a>

<li>image :
<a href="http://webuser.fh-furtwangen.de/%7Edersch/hdr/av.fjpg">
http://webuser.fh-furtwangen.de/%7Edersch/hdr/av.fjpg</a>

<li>applet :

<applet archive="ptviewer.jar" code="ptviewer.class" WIDTH="435" HEIGHT="272">
<param name=fovmin value=3>
<param name=fov value=50>
<param name=fovmax value=54>
<param name=panmin value=-27>
<param name=panmax value=27>
<param name=tiltmax value=18>
<param name=tiltmin value=-18>
<param name=autorange value=30>
<param name=quality value=2>
<param name=applet0 value="{code=
ptzoom.class}{file=av.fjpg}{progress=true}{gamma=1.4}{exposure=6}{pan=0.0}{tilt=0.0}{rot=0.0}{fov=54.0}">
<param name=init value="ptviewer:startApplet(0)">
</applet>
```

exemple 2

Page :

http://webuser.fh-furtwangen.de/%7Edersch/hdr/hdrtest2.html

image :

http://webuser.fh-furtwangen.de/%7Edersch/hdr/pano.fjpg

image boutons :

http://webuser.fh-furtwangen.de/%7Edersch/hdr/frame.gif

applet :

```
<applet name="ptviewer" archive=ptviewer.jar code=ptviewer.class width=420 height=300 mayscript=true>
<param name=file value="pano.fjpg">
<param name=cursor value="crosshair">
<param name=frame value="frame.gif">
<param name=gamma value="1.4">
<param name=exposure value="1">
<param name=autorange value="30">
<param name=quality value=2>
<param name=shotspot0 value=" x336 y286 a350 b300 u'ptviewer:startAutoPan(0.5,0,1)' ">
<param name=shotspot1 value=" x350 y286 a364 b300 u'ptviewer:stopAutoPan()' ">
<param name=shotspot2 value=" x364 y286 a378 b300 u'ptviewer:startAutoPan(0,0,0.97)' ">
<param name=shotspot3 value=" x378 y286 a392 b300 u'ptviewer:startAutoPan(0,0,1.03)' ">
</applet>
```

</body></html>

Autre exemple : <http://www.path.unimelb.edu.au/~bernardk/terrigen/hdr/hdrtest.html>

- Utilisation de l'applet :

La fonction luminosité est valable avec les images « HDR ».

Faire un clic sur le panorama dans la page HTML

Touche du clavier A ou > ou +	==> Zoom avant.
Touche du clavier Z ou < ou -	==> Zoom arrière.
Touche du clavier B	==> Augmentation de la luminosité (brighten).
Touche du clavier D	==> Diminution de la luminosité (darken).
Flèches du clavier	==> Déplacement haut, bas, gauche, droit.
Touche Ctrl + Clic de la souris	==> Zoom arrière continu.
Touche Maj. + Clic de la souris	==> Zoom avant continu.
Clic continu et déplacement de la souris	==> Déplacement continu haut, bas, gauche, droit.

- Créer des fichiers FJPG

Vous pouvez aussi télécharger le convertisseur de format d'image <http://webuser.fh-furtwangen.de/%7Edersch/hdr/ptconvert.zip> , et le décompresser dans C:\essaihdr. Le programme PTConvert.exe fonctionne en ligne de commandes. Les paramètres sont dans le fichier Readme_PTConvert.txt

Par exemple, ce programme utilitaire peut transformer un fichier format HDR (High Dynamic Range) en format FJPG plus léger qui peut être utilisé par l'applet.

Exemple d'une commande : C:\essaihdr\PTConvert.exe fichier.hdr -o resultat.fjpg -q 60

où resultat.fjpg = nom du fichier qui sera créé pour PTVIEWER version 3.12

 fichier.hdr = nom du fichier origine au format HDR.

 60 = qualité du jpeg

Copier et modifier cette ligne de commande, faire : Démarrer, Exécuter, coller cette ligne modifiée dans la fenêtre, OK.

- HDRShop

Le logiciel HDRShop (High Dynamic Range Image Processing and Manipulation) est disponible avec didacticiels à <http://www.debevec.org/HDRShop/> .

C'est avec ce logiciel que vous créez, manipulez, modifiez le format d'images HDR.



Liens sur le HDR :

<http://www.cgtechniques.com/lightbox/tutorials/makehdr/> : Comment faire une image HDR.

<http://kelsolaar.3dvf.net/Ressources/MyArt/SharedRessources/HDRI/HDRI.htm> : Images.

<http://radio.weblogs.com/0110138/stories/2003/02/06/hdrForPanoramas.html> :HDR pour panoramas.

Visualiser un panorama dans un fichier PDF avec PDFPanorama.

Comme pour l'exemple de la page 22 se rendre à la page d'Helmut Dersch <http://webuser.fh-furtwangen.de/%7Edersch/pdfpanorama/Readme.html> et télécharger <http://www.fh-furtwangen.de/~dersch/pdfpanorama/pdfpanorama.zip>.

Décompresser le fichier Zip et copier le fichier **pdfpanorama.exe** dans le répertoire des exemples **C:\essaiptv**

L'exemple de la page 22 (exemple3.html) sera utilisé pour créer le fichier PDF.

PDFPanorama fonctionne en ligne de commandes, par exemple :

```
C:\essaiptv\pdfpanorama.exe -o fichier.pdf exemple3.html -x 10 -y 2
```

où fichier.pdf = nom du fichier qui sera créé.

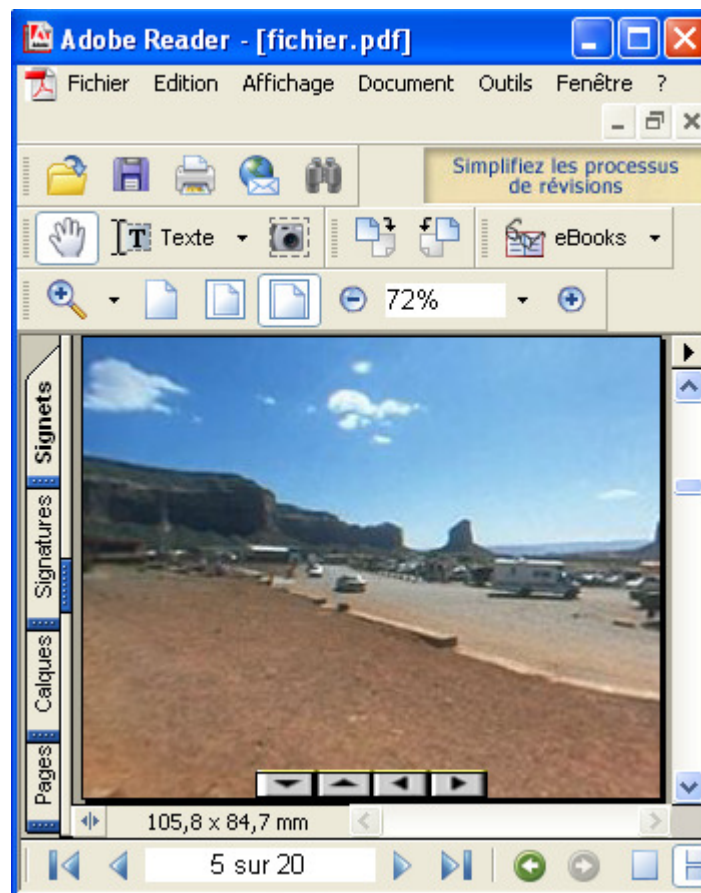
exemple3.html = nom du fichier origine (html contenant les paramètres de PTVIEWER).

2 = nombre de lignes (découpage de Monp.jpg)

10 = nombre de colonnes (découpage de Monp.jpg)

Copier cette ligne de commande, faire : Démarrer, Exécuter, coller cette ligne dans la fenêtre, OK.

Résultat :



Panoramas, liens divers (non classés)

<http://www.philohome.com/> : Site en anglais de Philippe Hurbain sur les techniques de prise de vue et le traitement des images.

<http://www.nyx.net/~cmaarb/images/panorama.html> : Exemples de panoramas créés avec Panorama Tools et Gimp.

<http://www.vjoncheray.com/pano/exemples-java.htm> : Réalisations par un professionnel, présentations faites en utilisant l'applet Java PTVIEWER.

http://www.panavue.com/index_fr.htm : Logiciel + galerie à voir (Québec).

<http://www.tourisme.gouv.qc.ca/francais/phototheque/panoramas/index.html> : Panoramas au format QuickTime du site touristique officiel de gouvernement du Québec.

<http://metrorama.free.fr/> : Panoramas du métro de Paris.

<http://www.htu.at/~sascha/ptguide/index.html> : Technique des panoramas et principes (en anglais).

<http://www.vierpi.de/tif2xcf.html> : Plug-in Gimp pour assembler les images TIF issues des GUI utilisant Panorama Tools (non testé).

<http://www.shallowsky.com/software/pandora/> : Plug-in Gimp pour assembler des images de panoramas (non testé).

<http://user.cs.tu-berlin.de/~nowozin/autopano-sift/> : Programme Autopano (algorithme SIFT).

<http://pnp.chez.tiscali.fr/> : Site français sur les techniques des panoramas.

<http://www.360geographics.com/IQTVRA/IQTVRALinks.html> : Liens.

<http://www.panoram-art.com/> : Réalisations par un professionnel.

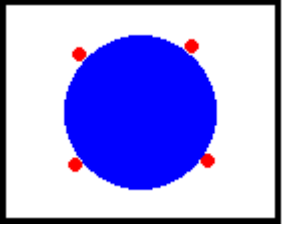
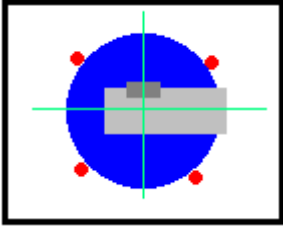
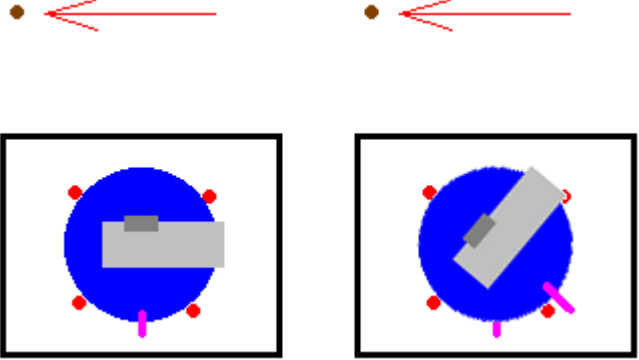
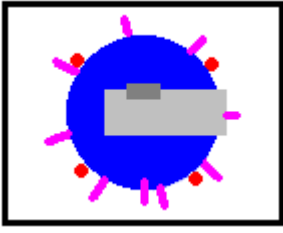
<http://www.linuxfocus.org/English/September2004/article348.shtml> : Création de panoramas sous Linux (article en anglais par Katja Socher).

Pratique

• Matériel

Voici un petit assemblage économique pour faire des montages de panoramas à partir d'un simple appareil de photographie numérique (ou autre) qui donne de très bons résultats.

Le matériel en sus : Une petite planche en bois, 4 clous, une boîte de conserves cylindrique pas trop grosse, un peu de ruban adhésif et un peu de temps pour la réalisation. Le but est de faire pivoter l'appareil photographique, rapidement (pour éviter les gros changements de lumière, le déplacement des nuages), sur un angle d'au moins 360°.

Clouer les 4 clous (en rouge) sur la planche en bois de façon à ce que la boîte de conserve (en bleu) tourne librement.	
Fixer l'appareil photographique (en gris), <u>de niveau</u>, sur la boîte de conserve. Le centre de la boîte de conserve correspond à l'intersection des axes verts. L'appareil photographique et son objectif (en gris foncé) doivent être positionnés comme indiqué. Maintenir l'ensemble avec un morceau de ruban adhésif.	
Déterminer l'angle de visée de votre objectif de façon pratique en faisant un repère sur la boîte de conserve et sur la planche (en violet). Viser un objet (une ligne verticale) et le placer sur une extrémité de l'image, faire tourner la boîte pour que la ligne verticale soit de l'autre côté de l'image. Faire un repère qui correspond à l'angle.	Reporter cet angle sur la planche, sur tout le tour de la boîte en dépassant l'origine.
	
Il suffit d'installer la planche, de niveau, à la bonne hauteur, à l'endroit que vous désirez photographier. Faire pivoter l'appareil à chaque repère de la planche en prenant une photographie (9 photographies dans notre exemple). Avec les repères les prises de vues sont très rapides. Vous pouvez obtenir des effets amusants en insérant le(s) même(s) personnage(s) à chaque ou sur plusieurs phases de la prise de vues.	

- **Assembler le résultat.**

Le résultat peut être facilement assemblé dans Hugin (page 13) avec cette méthode.

Dans l'onglet « Images » charger les photographies une par une en respectant l'ordre des prises de vue.

Sauvegarder souvent car ce logiciel est encore instable (File, Save ou File, Save As).

Dans l'onglet « Camera and lens » indiquer l'angle, etc. Ceci se fera automatiquement si EXCIF (description page 30) est complet.

Dans l'onglet « Control Points » indiquer les points de correspondance des photographies prises 2 à 2 : 0 et 1, 1 et 2, 2 et 3, etc.

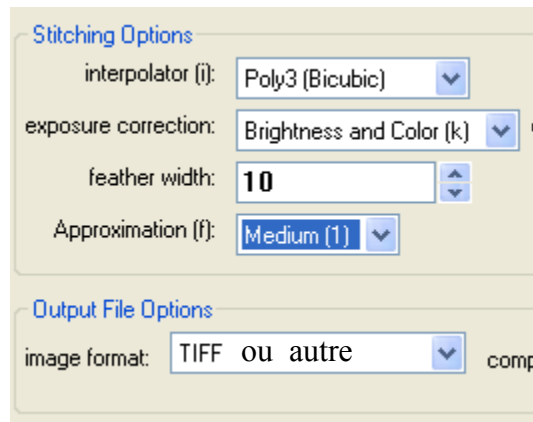
Dans l'onglet « Optimizer » lancer l'optimisation en appuyant sur le bouton avec la première option (Pairwise optim...). Valider.

Dans l'onglet « Stitcher » :

Appuyer sur le bouton « Calculate Field of View ».

Appuyer sur le bouton « Calculate optimal size »

Autres options



Sauvegarder.

Faire « preview ».

Si le résultat est bon appuyer sur le bouton « Stich now ». Fermer Hugin.

Ajuster l'image dans Gimp aux bonnes dimensions (couper les parties noires des extrémités si l'image est trop grande).

Vous pouvez reprendre l'image issue de Gimp dans Hugin comme nouveau projet pour ajuster les verticales (onglet « Control Points », mode « vertical line », etc.).

Consulter les données contenues dans les images (EXIF, IPCT)

Avec le greffon « EXIF Browser » pour Gimp

Le greffon « EXIF Browser » pour Gimp permet de visualiser les données Exif qui caractérisent les paramètres de la prise de vue à partir d'appareils numériques.

Ce greffon fonctionne sous Linux et Windows.

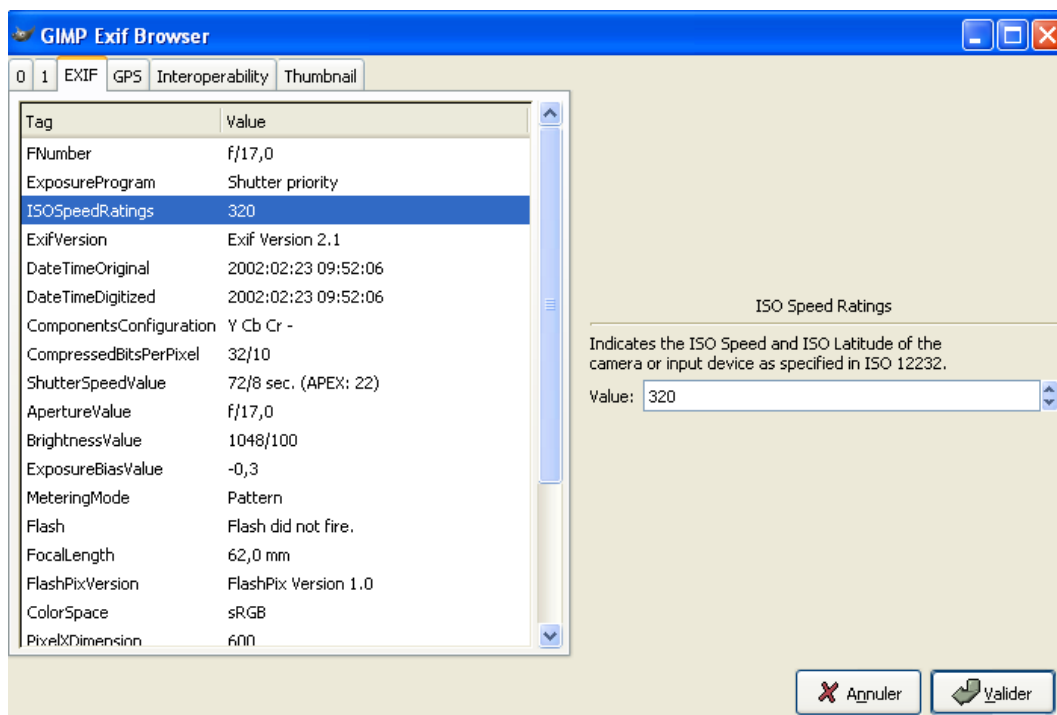
Michael Schumacher a eu la gentillesse de compiler ce greffon pour qu'il fonctionne sous Windows. Le tout est disponible à la page <http://registry.gimp.org/plugin?id=4153>

Sous Windows, vous téléchargez le fichier « exif-browser-bin.zip ». Vous décompressez et placez le fichier « exif-browser.exe » dans le répertoire « plug-ins » de Gimp (C:\Program Files\GIMP-2.2\lib\gimp\2.0\plug-ins) (voir le nota en bas de la page 5)

Voici les caractéristiques :

Nom :	plugin-exif-browser
Description :	Exif data browser
Aide :	Allows the user to view and edit exif data
Auteur :	William Skaggs <weskaggs@primate.ucdavis.edu>
Date :	2004
Copyright :	William Skaggs <weskaggs@primate.ucdavis.edu>

Pour l'utiliser, à partir d'une image contenant des données Exif faire : Filtres, Générique, Exif Browser. Vous pouvez sélectionner les onglets 0, 1, Exif, etc.

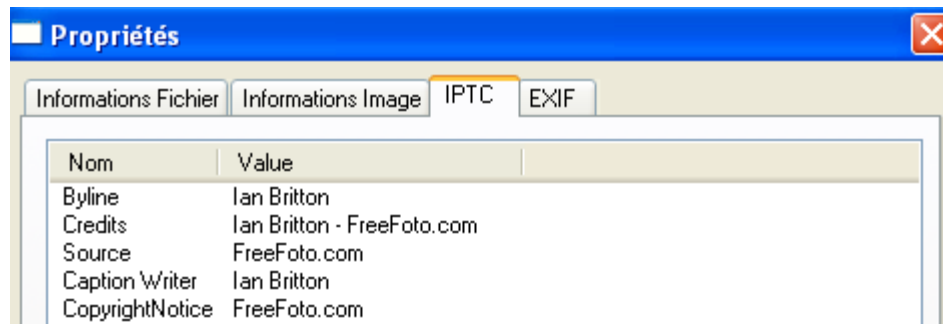


Avec XnView (visualiseur et convertisseur d'images universel)

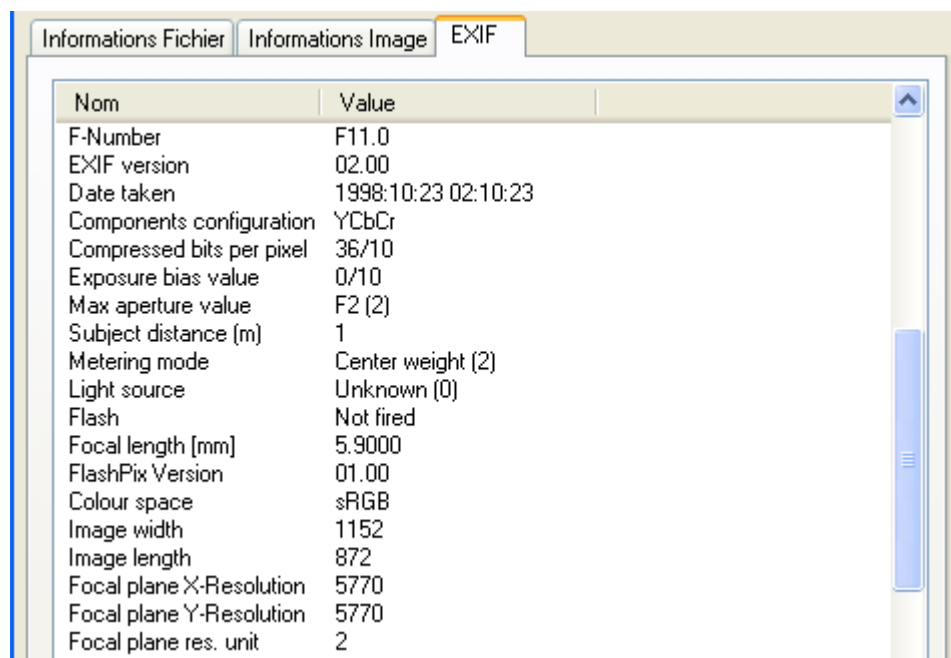
Dans cet exemple, pour lire les données contenues dans les images (quand elles existent) nous utiliserons XnView de Pierre Gougelet (<http://www.xnview.com> ou <http://perso.wanadoo.fr/pierre.g>).

Ouvrir une image dans XnView et faire : Édition, Propriétés. Choisir un onglet.

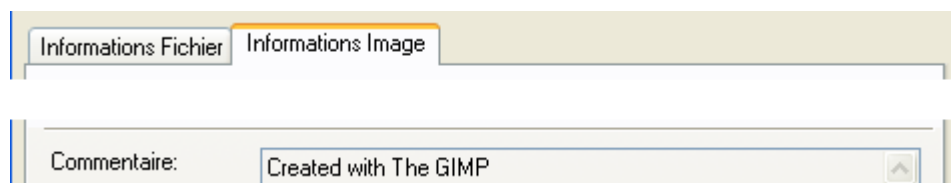
Les données IPTC sont les crédits, par exemple :



Les données EXIF sont les paramètres techniques de la prise de vue, par exemple :



Les commentaires, par exemple :



Corriger les yeux rouges ou oranges (prise de vue avec flash)

Script-Fu Red Eye (red-eye.scm)

Ce script-Fu permet de corriger les yeux rouges. Eric Jeschke propose un didacticiel sur ce sujet à <http://gimpguru.org/Tutorials/RedEye2/> ainsi qu'un article publié dans Linux Journal <http://www.linuxjournal.com/article.php?sid=6567> (en anglais).

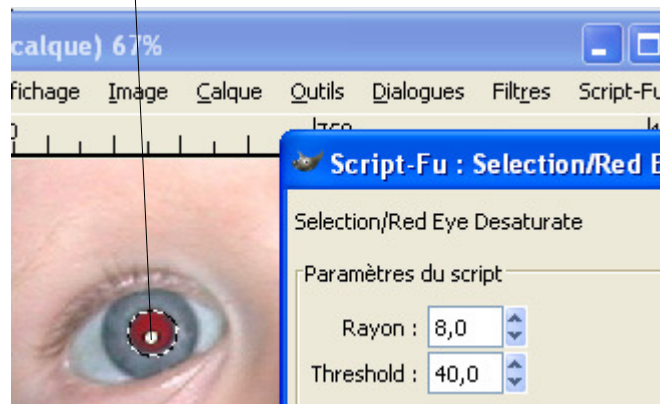
Le Script-Fu est disponible à <http://www.linuxjournal.com/modules.php?op=modload&name=NS-articles/HOWTO&file=656711> ou à la page 33 de ce document. Placer ce script comme indiqué page 5)

Ce Script-Fu a été créé par Martin Guldahl (*auteur du greffon Channel Mixer qui est accessible par Filtres, Couleurs*) .

Ce script est très pratique si le point de lumière existe, sinon, il faut en créer un avec une petite brosse ronde ou carrée avant d'utiliser le Script-Fu. L'effet s'applique sur la sélection dans l'image.

Faire : Script-Fu, Selection, Red Eye Desaturate .

Image © Eric R. Jeschke du didacticiel RedEye2



Orange Eye (non testé)

Linux : <http://registry.gimp.org/plugin?id=4243>

Greffon Windows : <http://schumaml.gmxhome.de/downloads/gimp/>

RedEye (non testé)

Linux : <http://registry.gimp.org/plugin?id=4212>

Greffon Windows : <http://schumaml.gmxhome.de/downloads/gimp/>

Didacticiels

http://www.absolut-photo.com/tutoriels/yeux_rouges_gimp.php en français.

http://mmmmaybe.gimp.org/tutorials/Red_Eye_Removal/ en anglais.

```

; The GIMP -- an image manipulation program
; Copyright (C) 1995 Spencer Kimball and Peter Mattis
;
; This program is free software; you can redistribute it and/or modify
; it under the terms of the GNU General Public License as published by
; the Free Software Foundation; either version 2 of the License, or
; (at your option) any later version.
;
; This program is distributed in the hope that it will be useful,
; but WITHOUT ANY WARRANTY; without even the implied warranty of
; MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
; GNU General Public License for more details.
;
; You should have received a copy of the GNU General Public License
; along with this program; if not, write to the Free Software
; Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
;
;
; red-eye.scm version 0.95 2002/10/06
;
; CHANGE-LOG:
; 0.95 - initial release
;
; Copyright (C) 2002 Martin Guldahl <mguldahl@xmission.com>
;
;
; Removes red eye

(define (script-fu-red-eye image
                                drawable
                                radius
                                threshold
                                red
                                green
                                blue)

  (let* ((select-bounds (gimp-drawable-mask-bounds drawable))
         (x1 (cadr select-bounds))
         (y1 (caddr select-bounds))
         (x2 (cadr (cddr select-bounds)))
         (y2 (caddr (cddr select-bounds)))
         (red-component 0)
         (green-component 1)
         (blue-component 2)
         )

    (gimp-undo-push-group-start image)

    ; Select and view the green channel only as it has the best contrast
    ; between the iris and pupil.
    ; - Deselect the Red and Blue channels

    (gimp-image-set-component-active image red-component FALSE)
    (gimp-image-set-component-active image blue-component FALSE)

    ; Use the fuzzy select tool (select contiguous regions) to select the pupils
    (gimp-fuzzy-select
     drawable
     x1 y1
     threshold 0 TRUE TRUE radius FALSE
     )

    ; Reselect Red and Blue channels
    (gimp-image-set-component-active image red-component TRUE)
    (gimp-image-set-component-active image blue-component TRUE)

    ; Use the channel mixer to make selected area monochrome
    (plug-in-colors-channel-mixer
     ;;run_mode
     1
     image
     drawable
     ;;monochrome
     1
     (/ red 100) (/ green 100) (/ blue 100) 0 0 0 0 0 0
     )

    (gimp-selection-clear image)
    (gimp-undo-push-group-end image)
    (gimp-displays-flush)
  )
)

```

```

(define (script-fu-red-eye-no-cm image
                                     drawable
                                     radius
                                     threshold
                                     )

  (let* ((select-bounds (gimp-drawable-mask-bounds drawable))
         (x1 (cadr select-bounds))
         (y1 (caddr select-bounds))
         (x2 (cadr (cddr select-bounds)))
         (y2 (caddr (cddr select-bounds)))
         (red-component 0)
         (green-component 1)
         (blue-component 2)
         )

    (gimp-undo-push-group-start image)

    ; Select and view the green channel only as it has the best contrast
    ; between the iris and pupil.
    ; - Deselect the Red and Blue channels

    (gimp-image-set-component-active image red-component FALSE)
    (gimp-image-set-component-active image blue-component FALSE)

    ; Use the fuzzy select tool (select contiguous regions) to select the pupils
    (gimp-fuzzy-select
     drawable
     x1 y1
     threshold 0 TRUE TRUE radius FALSE
     )

    ; Reselect Red and Blue channels
    (gimp-image-set-component-active image red-component TRUE)
    (gimp-image-set-component-active image blue-component TRUE)

    ; Now desaturate the Red channel
    (gimp-image-set-component-active image red-component TRUE)
    (gimp-image-set-component-active image green-component FALSE)
    (gimp-image-set-component-active image blue-component FALSE)
    (gimp-desaturate drawable)
    (gimp-image-set-component-active image red-component TRUE)
    (gimp-image-set-component-active image green-component TRUE)
    (gimp-image-set-component-active image blue-component TRUE)

    (gimp-selection-clear image)
    (gimp-undo-push-group-end image)
    (gimp-displays-flush)
  )
)

(script-fu-register "script-fu-red-eye"
  "<Image>/Script-Fu/Selection/Red Eye..."
  "Removes red eye; given selection is seed.\nNeeds the channel-mixer plug-in."
  "Martin Guldahl <mguldahl@xmission.com>"
  "Martin Guldahl"
  "2002/10/05"
  "sg"
  SF-IMAGE "Image" 0
  SF-DRAWABLE "Drawable" 0
  SF-ADJUSTMENT _"Radius" '(8 0 100 .2 1 1 1)
  SF-ADJUSTMENT _"Threshold" '(40 0 255 1 1 1 1)
  SF-ADJUSTMENT _"Red" '(10 0 100 1 1 1 1)
  SF-ADJUSTMENT _"Green" '(60 0 100 1 1 1 1)
  SF-ADJUSTMENT _"Blue" '(30 0 100 1 1 1 1)
  )

(script-fu-register "script-fu-red-eye-no-cm"
  "<Image>/Script-Fu/Selection/Red Eye Desaturate..."
  "Removes red eye; given selection is seed."
  "Martin Guldahl <mguldahl@xmission.com>"
  "Martin Guldahl"
  "2002/10/05"
  "sg"
  SF-IMAGE "Image" 0
  SF-DRAWABLE "Drawable" 0
  SF-ADJUSTMENT _"Radius" '(8 0 100 .2 1 1 1)
  SF-ADJUSTMENT _"Threshold" '(40 0 255 1 1 1 1)
  )

```

Script-Fu Camouflage (motif)

Voici un Script-Fu qui va créer un motif de type camouflage. Pour l'obtenir vous pouvez vous rendre à la page <http://matrix.netsoc.tcd.ie/~horkana/dev/gnome/gimp/script-fu/script-fu.html> de Alan Horkan (sur cette page Alan propose d'autres scripts). Le lien direct pour télécharger ce script est : <http://matrix.netsoc.tcd.ie/~horkana/dev/gnome/gimp/script-fu/scripts/pattern-camouflage.scm>

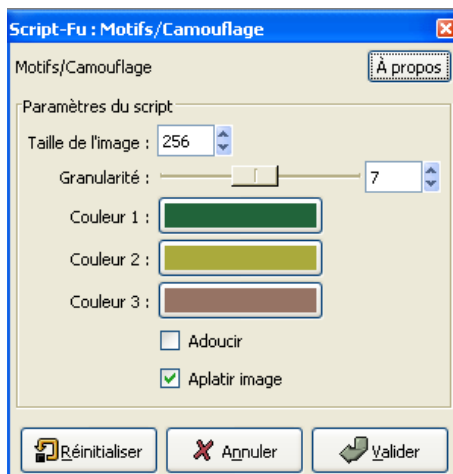
Vous placez « pattern-camouflage.scm » dans le répertoire des scripts (voir début de cette documentation page 5 et le nota en bas de la page 5).

Ce script de Chris Gutteridge (cjg@ecs.soton.ac.uk) At ECS Dept, University of Southampton, England a été modifié par Alan Horkan.

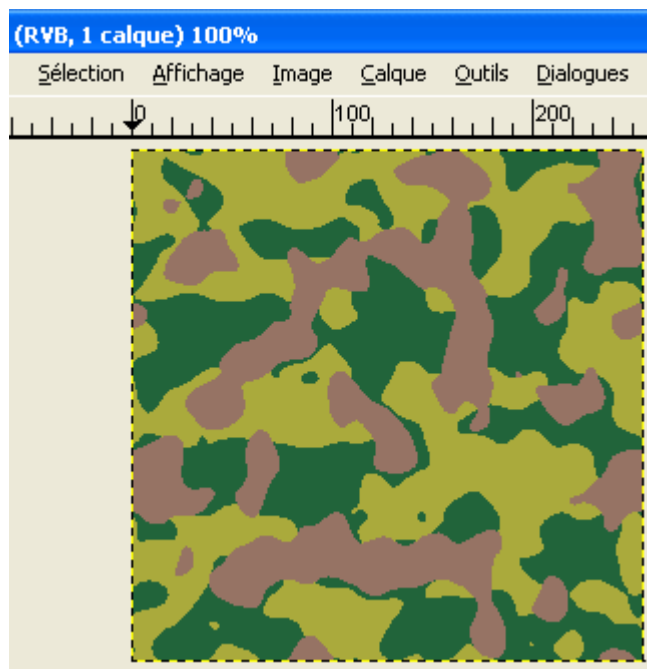
Ce script-fu génère une nouvelle image carrée avec un motif de camouflage.

Pour lancer ce script, à partir de la fenêtre principale, faire : Exts, Script-Fu, Motifs, Camouflage.

La fenêtre des paramètres s'affiche :



Voici le résultat :



Code du script à la page suivante :

```

;; camouflage.scm -*-scheme-*-

;; Changes (gimp2 changes, use current image, don't Flatten!) and fixes by Alan Horkan.
;; Now also works within a selection if there is one.
;; Code restructured and cleanly encapsulated instead of repetition.
;; Alan Horkan claims no copyright on his improvements to this script.

;; Originally by Chris Gutteridge (cjl@ecs.soton.ac.uk)
;; At ECS Dept, University of Southampton, England.

;; This program is free software; you can redistribute it and or modify
;; it under the terms of the GNU General Public License as published by
;; the Free Software Foundation; either version 2 of the License, or
;; (at your option) any later version.
;;
;; This program is distributed in the hope that it will be useful,
;; but WITHOUT ANY WARRANTY; without even the implied warranty of
;; MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
;; GNU General Public License for more details.
;;
;; You should have received a copy of the GNU General Public License
;; along with this program; if not, write to the Free Software
;; Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

(define (script-fu-pattern-camouflage image drawable detail colour1 colour2 colour3 smooth)
  (gimp-image-undo-group-start image) ; enable undo
  ;; save current background colour so it can be restored when finished
  (set! old-bg (car (gimp-palette-get-background)))

  ;; if there is not already a selection then make one
  (if (= (car (gimp-selection-is-empty image)) TRUE)
      (gimp-selection-all image)
      )
  (gimp-edit-copy drawable)

  (set! base-layer (car (gimp-edit-paste drawable FALSE)))
  (gimp-floating-sel-to-layer base-layer)
  (gimp-drawable-set-name base-layer _"Camouflage Base"); not v1.2 compatible

  ;; copy base layer to create other layers, thin layer and thick layer
  (set! thick-layer (car (gimp-layer-copy base-layer TRUE)))
  (gimp-image-add-layer image thick-layer -1)
  (gimp-drawable-set-name thick-layer _"Camouflage Thick Layer"); not v1.2

  (set! thin-layer (car (gimp-layer-copy base-layer TRUE)))
  (gimp-image-add-layer image thin-layer -1)
  (gimp-drawable-set-name thin-layer _"Camouflage Thin Layer"); not v1.2

  (gimp-palette-set-background colour1)
  (gimp-edit-fill base-layer BG-IMAGE-FILL) ;; fill selected, else layer

  (set! blur (- 16 detail))

  ;; ALAN: the real camouflage effect bit, 'where the magic happens'
  (define (camouflage camo-layer colour blur smooth low-threshold)
    ;; ALAN: bugfix gimp1.2.x solid-noise plugin crashed if detail=0
    ; (if (< detail 1) (set! detail 1))
    (plug-in-solid-noise TRUE image camo-layer 1 0 (rand 65536) 1 detail detail) ; create the pattern
    (gimp-threshold camo-layer low-threshold 255) ; seperate to black/white

    (gimp-palette-set-background colour)
    (gimp-by-color-select camo-layer '(0 0 0) 127 REPLACE TRUE FALSE 0 FALSE)
    (gimp-edit-clear camo-layer)
    (gimp-selection-invert image)
    (gimp-edit-fill camo-layer BG-IMAGE-FILL)
    (gimp-selection-none image)
    (if (= smooth TRUE)
        (script-fu-tile-blur image camo-layer blur TRUE TRUE FALSE)
        )
    )

  (camouflage thick-layer colour2 blur smooth 127)
  (camouflage thin-layer colour3 blur smooth 145)

  (gimp-palette-set-background old-bg) ; restore the old background colour
  (gimp-image-undo-group-end image) ; enable undo
  (gimp-displays-flush)
  )

(script-fu-register "script-fu-pattern-camouflage"
  "<Image>/Filters/Render/Pattern/Camouflage..."
  _"Camouflage pattern in three colours.
  Creates three new layers which form a Camouflage like design.
  If there is an active selection the camouflage pattern will be created inside the selection.
  The Smooth option will produce a smoother less jagged pattern but it takes a little longer. "
  "Alan Horkan. Chris Gutteridge. "; author(s)
  "Chris Gutteridge,
  ECS @ University of Southampton, England"; rights
  "2004 09 07 UTC" ;; date, was "April 2000" and "28th April 1998"
  "RGB*"

  SF-IMAGE "Image" 0
  SF-DRAWABLE "Drawable" 0

```

```
:: ALAN: bugfix set min=1, solid-noise plugin crashed if detail=0
:: NOTE: increased max detail to 100, best used only on much larger images
SF-ADJUSTMENT _"Detail" (10 0 100 1 1 0 0)

SF-COLOR _"Colour 1" (33 100 58)
SF-COLOR _"Colour 2" (170 170 60)
SF-COLOR _"Colour 3" (150 115 100)
SF-TOGGLE _"Smooth" FALSE
)

; Alan woz 'ere
```

Script-Fu Duotone (teinter)

Ce Script-Fu créé par Alexios Chouchoulas est une partie d'un didacticiel d' Eric R. Jeschke disponible à la page : <http://gimpguru.org/Tutorials/SepiaToning/>

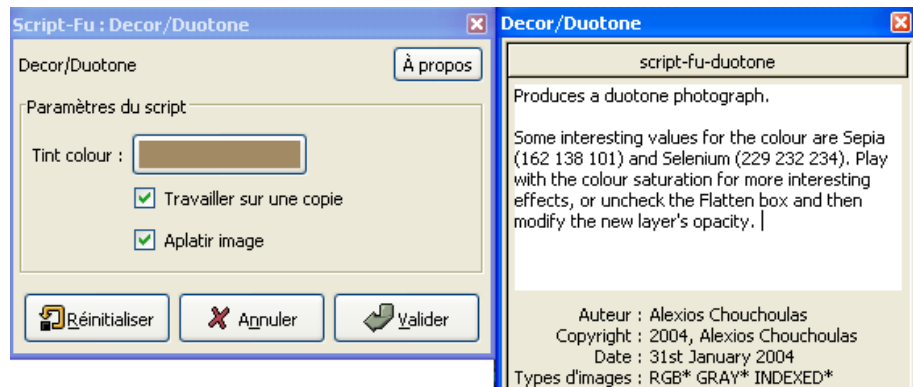
Vous pouvez télécharger directement le Script-Fu duotone-gimp-2.0.scm à partir de ce lien : <http://www.bedroomlan.org/~alexios/files/SOFTWARE/script-fu/duotone-gimp-2.0.scm>

Alexios Chouchoulas propose d'autres Scripts-Fu à la page : <http://www.bedroomlan.org/~alexios/files/SOFTWARE/script-fu/>

Avec ce script vous pouvez teindre une photographie.

Pour activer ce script, à partir d'une image faire : Script-Fu, Decor, Duotone.

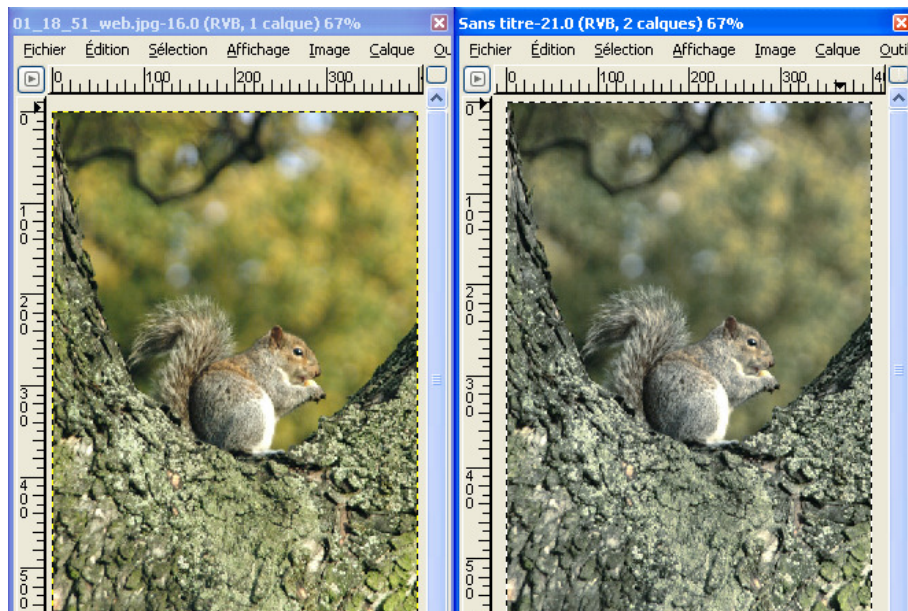
Voici la fenêtre des paramètres et quelques conseils :



Voici le résultat sur une photographie de Ian Britton disponible sur le site <http://www.freefoto.com/> (<http://www.freefoto.org/preview.jsp?id=01-18-51>)

Photographie d'origine

Script-Fu avec couleur Sélénium



```

;
; duotone
;
; $Id: duotone-gimp-2.0.scm,v 1.3 2004/02/12 22:12:29 alexios Exp $
;
; $Log: duotone-gimp-2.0.scm,v $
; Revision 1.3 2004/02/12 22:12:29 alexios
; Another beta version of GIMP 2.0, another release of Duotone.
;
; Revision 1.2 2004/01/31 17:00:21 alexios
; Added support for Indexed images (they're converted to RGB anyway).
;
; Revision 1.1 2004/01/31 16:52:27 alexios
; Initial revision.
;
;
; A simple script to emulate darkroom sepia toning (and other similar
; toning processes). It is based on how actual sepia toning looks
; like, supplemented by Eric R. Jeschke's GIMP tutorial:
;
; http://gimpguru.org/Tutorials/SepiaToning/
;
; This program is free software; you can redistribute it and/or modify
; it under the terms of the GNU General Public License as published by
; the Free Software Foundation; either version 2 of the License, or
; (at your option) any later version.
;
; This program is distributed in the hope that it will be useful,
; but WITHOUT ANY WARRANTY; without even the implied warranty of
; MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
; GNU General Public License for more details.
;
; You should have received a copy of the GNU General Public License
; along with this program; if not, write to the Free Software
; Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
;
; Define the function:
;
(define (alexios-set-pt a index x y)
  (prog1
    (aset a (* index 2) x)
    (aset a (+ (* index 2) 1) y)))

(define (alexios-duotone-spline)
  (let* ((a (cons-array 6 'byte)))
    (set-pt a 0 0 0)
    (set-pt a 1 128 128)
    (set-pt a 2 255 0)
    a))

(define (script-fu-duotone inImage
                           inLayer
                           inTintColour
                           inCopy)
  (inFlatten
   )
  (set! theImage (if (= inCopy TRUE)
                     (car (gimp-image-duplicate inImage))
                     inImage))
  (set! theWidth (car (gimp-image-width theImage)))
  (set! theHeight (car (gimp-image-height theImage)))
  (if (< 0 (car (gimp-image-base-type theImage)))
      (gimp-image-convert-rgb theImage))
  ; Do the actual work.
  ; Copy the image.
  (gimp-selection-all theImage)
  (gimp-edit-copy inLayer)
  ; Make the tint layer
  (set! theLayer (car (gimp-layer-new theImage
                                     theWidth
                                     theHeight
                                     RGBA-IMAGE
                                     "Tint"
                                     100
                                     COLOR-MODE))))
  ; Fill the layer with the tint
  (gimp-palette-set-foreground inTintColour)
  (gimp-edit-fill theLayer FG-IMAGE-FILL)
  ; Add the layer to the image
  (gimp-image-add-layer theImage theLayer 0)
  ; Create a mask for the new layer

```

```

(set! mask (car (gimp-layer-create-mask theLayer ADD-WHITE-MASK)))
(gimp-layer-add-mask theLayer mask)
(gimp-floating-sel-anchor (car (gimp-edit-paste mask TRUE)))
(gimp-curves-spline mask HISTOGRAM-VALUE 6 (alexios-duotone-spline))

; Flatten the image, if we need to.

(if (= inFlatten TRUE) (gimp-image-flatten theImage))

; Have we been working on a copy? If so display the new image.

(if (= inCopy TRUE)
    (begin
      (gimp-image-clean-all theImage)
      (gimp-display-new theImage)
    )
  0
)

; The end.

(gimp-displays-flush)
)

; Register the function with the GIMP:

(script-fu-register
  "script-fu-duotone"
  _ "<Image>/Script-Fu/Decor/_Duotone..."
  "Produces a duotone photograph.

Some interesting values for the colour are Sepia (162 138 101) and Selenium (229 232 234). Play with the colour
saturation for more interesting effects, or uncheck the Flatten box and then modify the new layer's opacity. "
  "Alexios Chouchoulas"
  "2004, Alexios Chouchoulas"
  "31st January 2004"
  "RGB* GRAY* INDEXED*"
  SF-IMAGE      "The Image"      0
  SF-DRAWABLE    "The Layer"      0
  SF-COLOR       _ "Tint colour"  '(162 138 101)
  SF-TOGGLE      _ "Work on Copy" TRUE
  SF-TOGGLE      _ "Flatten Image" TRUE
)

;;; End Of File.

```

RawPhoto (traitement des images brutes)

Nota : Les informations qui suivent ce nota sont d'octobre 2004 et les choses ont évoluées depuis cette date. L'utilitaire Unidentified Flying Raw (UFRaw) est disponible à la page <http://ufraw.sourceforge.net/>. UFRaw permet de lire et manipuler les images au format Raw (non testé). Il est disponible sous Linux (sources à compiler), Debian, Mandrake, Fedora, NetBSD, Mac et Windows en 2 versions (*dummies* et *geeks*). La version « *dummies* » est la plus simple à mettre en oeuvre. La version « *geeks* » fonctionne dans un environnement Cygwin.

Sous Linux

Voici les liens vers le greffon qui permet de lire les images au format Raw (images brutes sans traitements) des appareils numériques supporté par « Dave's Coffin DCRaw utility » (<http://www.cybercom.net/~dcoffin/dccraw/>) dans Gimp. Le code de ce greffon est toujours en développement.

Vous trouverez à la page <http://ptj.rozeta.com.pl/Soft/RawPhoto> le greffon **RawPhoto** proposé par Paweł T. Jochym.

Fedora : RawPhoto by Casper Pedersen (<http://fedoranews.org/casper/rawphoto/>)

Adaptation pour Windows

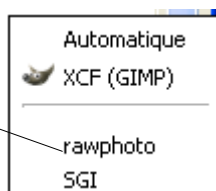
DCRaw pour Windows (suivant le processeur) est adapté par Benjamin Lebsanft (http://home.arcor.de/benjamin_lebsanft/)

Matthew H. Plough a compilé le greffon **RawPhoto** (format Zip) pour Windows et le propose à la page <http://www.princeton.edu/~mplough/rawphoto.html>

Au 14 octobre 2004 la version est <http://www.princeton.edu/~mplough/rawphoto-windows-binary-1.18.zip>

- Vous placez **rawphoto.exe** dans le répertoire **C:\Program Files\GIMP-2.2\lib\gimp\2.0\plug-ins** (voir le nota en bas de la page 5).

- Vous lancez Gimp et par la fonction Fichier, Ouvrir vous avez la possibilité d'importer des images au format Raw



Nota : Le chemin d'accès à l'image (répertoires + sous-répertoires) ne doit pas comporter d'espaces.

Conseil

Installez le greffon RawPhoto uniquement si vous l'utilisez car il peut créer des effets désagréables (ouverture de fenêtre DOS, etc.) lors de l'ouverture des images.

Quelques liens

Articles en français sur le format Raw :

http://www.linternaute.com/photo_numerique/trucs/format/format_4.shtml

http://rmeralli.free.fr/photo-numerique/articles/format_fichier/format_fichier_numerique.html

http://www.megapixel.net/cgi-bin/fs_loader.pl?p=http%3A//www.megapixel.net/html/articles/article-fileformatsf.html

Bruce Fraser, pour la société Adobe, propose deux petites notices explicatives au format PDF, en anglais, sur la façon d'obtenir une image Raw dans un appareil numérique et de la traiter :

http://www.adobe.com/products/photoshop/pdfs/understanding_digitalrawcapture.pdf

http://www.adobe.com/products/photoshop/pdfs/linear_gamma.pdf

Uwe Steinmueller propose une méthodologie, en anglais, pour le traitement d'images Raw :

http://www.outbackphoto.com/artofraw/raw_01/essay.html

Mrwtoppm

Linux programs for converting raw image files from Minolta DiMAGE 5, 7, 7i, 7Hi, and A1 cameras :

<http://home.swbell.net/dobroom/mintoppm/index.html>

<http://home.swbell.net/dobroom/mintoppm/Plugin.html>

Raw saver plugin for the GIMP 1.3/2.0 (Vesa Halttunen) : <http://registry.gimp.org/plugin?id=3457>

Recherche sur Google : <http://www.google.com/search?hl=fr&q=raw+gimp&btnG=Rechercher&meta=>

Greffon « resynthesizer »

Ce greffon proposé par Paul Harrison permet de réaliser des textures, des fonds de pages html à partir d'images, des effets sur des images avec des textures, etc. Dans notre exemple (page 45), il modifie une image pour que les côtés s'ajustent (horizontal et vertical). Ce greffon offre d'autres possibilités décrites à partir de l'adresse : <http://www.logarithmic.net/pfh/resynthesizer>

Contenu du fichier README (en anglais)

Resynthesizer is a GIMP plug-in with many uses:

- Generating a more of a texture

Given a sample of a particular texture, produce a larger version of that texture.

- Making tilable images

Given a sample of a particular texture, produce a tilable version of that texture.

- Removing unwanted features from an image

If there is some unwanted feature in an image (such as a discoloration or a political ally who has fallen out of public favour), Resynthesizer can remove it by extending the surrounding texture over the top of it.

- Applying themes to images

Given a sample texture (for example watercolor brushstrokes), another image can be given that texture while retaining its overall structure. This might be used to make an image look as if it had been painted. This feature acts somewhat like a generic effects plug-in. Given a sample of an effect, that effect can be applied to an image.

- Melting your CPU, filling your swap partition

Resynthesizer is quite resource intensive. What it does is computationally expensive.

Installation (Linux)

Type:

make

make install

To install i18n things, as root type:

make install-mos

Usage

This plug-in installs itself in the Filters/Map menu as "Resynthesize". It generates a texture based on a sample image of the texture. The result can be made to fit the boundary pixels of a region, or to have a specified arrangement of features.

Options:

- Texture source

This can be used to select the image to take the texture from. Any selected regions in this image will *not* be included in the analysis of this texture.

- Make horizontally tilable

- Make vertically tilable

Should the output be tilable, i.e. wrap around at the edges ?

- Fit output to bordering pixels

If this option is checked and there is a selection, Resynthesizer will try to match its output to pixels surrounding the selection.

- Refinement passes

Once resynthesizer has created a synthetic image, it can refine it, fixing up any bits that didn't match up properly in the first pass. Increasing the number of refinement passes will give slightly better quality (but take longer).

- Randomness

Amount of randomness to introduce when resynthesizing the texture. If this is high, the result will tend to contain only small

features. If it is low the result will contain larger features (often copied exactly from the texture source), but may also contain artifacts such as repeating patterns.

- Texture property maps

If this option is set a pair of images (maps), one for the texture source and one for the output, will be used to guide placement of features of the texture in the output. The input and output maps should have the same mode (greyscale or RGB). They should have the same size as the texture source and output image respectively. Resynthesizer will try to select features from the texture source having similarly colored pixels in the input map as the corresponding pixels in the output map. Adjust the map importance slider to specify how slavishly Resynthesizer should follow these maps.

Also included in this package are some scripts that use Resynthesizer:

Script-Fu/Enhance/Smart enlarge

Script-Fu/Enhance/Smart sharpen

Script-Fu/Enhance/Smart remove selection

Authors

Resynthesizer was written by Paul Harrison (pfh@logarithmic.net).

David Rodríguez García (darodga@mail.ono.es) added the « update » button in the preview dialog-box.

Laurent Despeyroux (laurent.despeyroux@free.fr) made some optimizations and most of the changes of the version 0.7.

Ali Akcaagac updated Resynthesizer to work with Gimp 1.3.

Feedback welcome.

Resynthesizer is released under the terms of the GNU General Public licence (see the file "COPYING").

Change log

- 0.10
 - * Added refinement pass/passes
 - * Removed MML stuff (replaced by the number 3)
 - * Bug fixes, code simplification
- 0.9
 - * Improved algorithm (L2 norm, better pixel prioritization scheme)
 - * Updated to work with Gimp 2.0
- 0.8
 - * Updated to work with Gimp 1.3
- 0.7
 - * internationalization (and French version)
 - * optimizations (theoretical)
 - * comments added to code
 - * preview window bigger
 - * « update » button in the preview dialog-box
 - * more messages during calculation
- 0.6
 - * updated stuff to not use compatibility cruft in libgimp
- 0.5
 - * ported to C,
 - * loosing what little readability the code had to begin with in the process
 - * included some script-fu utilities that use resynthesizer
- 0.4
 - * fixed bug selecting which pixels to use with semi-transparent masks
 - * fixed out-by-one bug in partition code
- 0.3
 - * fixed sqrt of negative number problem in Entropy_accumulator
 - * fixed bug that was stuffing up map importance parameter
 - * repeat-last mode now works properly
- 0.2
 - * workaround for strange (presumably) compiler bug involving gimp_message and the ?: operator
- 0.1
 - * initial release

Linux

Télécharger <http://www.logarithmic.net/pfh-files/resynthesizer/resynthesizer-0.10.tar.gz>

Windows

Michael Schumacher a compilé ce greffon qui est disponible à l'adresse suivante

<http://schumaml.gmxhome.de/downloads/gimp/>

Le fichier « resynthesizer-0.10-bin.zip » contient 2 répertoires : plug-ins et scripts.

Dans le premier répertoire vous trouvez le greffon **resynth.exe** que vous placez dans C:\Program Files\GIMP-2.2\lib\gimp\2.0\plug-ins

Dans le second répertoire vous trouvez 2 Scripts-Fu, **smart-enlarge.scm** et **smart-remove.scm** que vous placez dans C:\Program Files\GIMP-2.2\share\gimp\2.0\scripts (voir le nota en bas de la page 5)

Nota : Suivant les caractéristiques de votre micro-ordinateur et de l'image à traiter vous devrez être patient car ce greffon demande beaucoup de ressources pour le calcul.

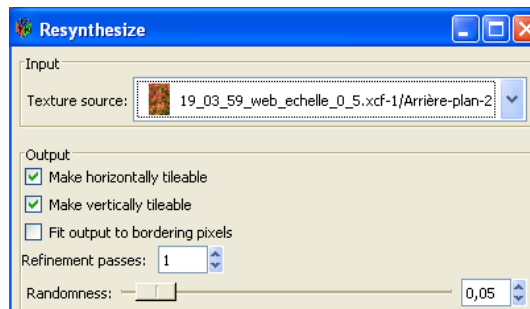
Exemple

Pour l'exemple nous utilisons une photographie de Ian Britton « Picture of Autumn color in Vermont. » disponible sur le site <http://www.freefoto.com/> à la page <http://www.freefoto.org/preview.jsp?id=19-03-59>

Ouvrir cette image « 19_03_59_web.jpg » dans Gimp et la réduire à l'échelle 0,5 (Image, Étirer l'image, Facteur d'échelle X et Y = 0,5 puis Valider). Enregistrer cette image sous un format xcf.

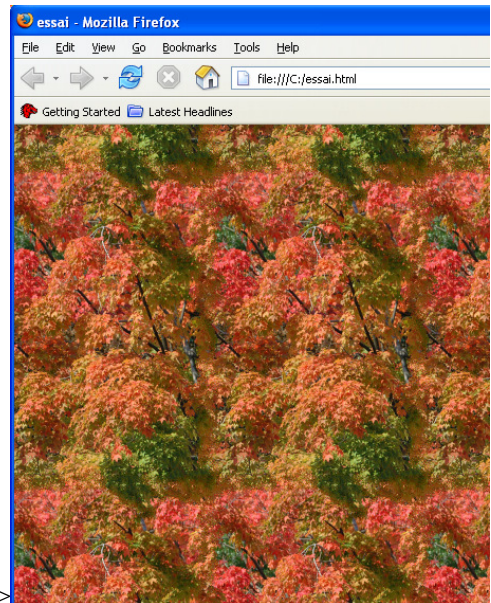
Sur cette image faire : Filtres, Carte, Resynthesize

Paramétrer :



Valider et attendre.

Enregistrer le résultat au format PNG et utiliser l'image comme fond d'une page HTML ==>



Voici quelques liens proposant d'autres exemples :

<http://laurent.despeyroux.free.fr/gimp/texture.html> : Exemple proposé par Laurent Despeyroux.

<http://wiki.puddingbowl.org/phpwiki/index.php/ResynthesizerPlugin> : Supprimer une partie d'image.

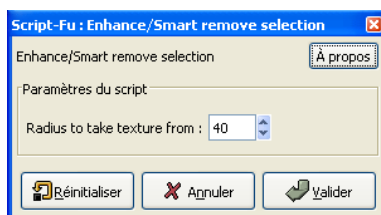
Script-Fu smart-remove.scm

Ce Script-Fu utilise le greffon « resynthesizer » et permet d'enlever une partie d'image. L'exemple utilise une image présentée à la page 38 où l'écureuil va être enlevé.

Ouvrir l'image 01_18_51_web.jpg et l'enregistrer au format xcf.

Utiliser l'outil Sélection à main levée ou lasso (F) autour de l'écureuil. Dans l'exemple la sélection est grossière.

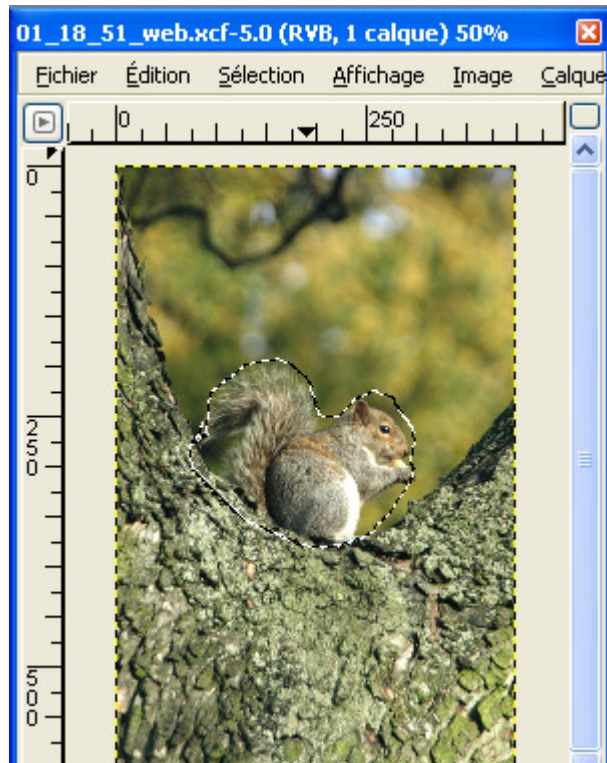
Faire : Scripts-Fu, Enhance, Smart remove selection.



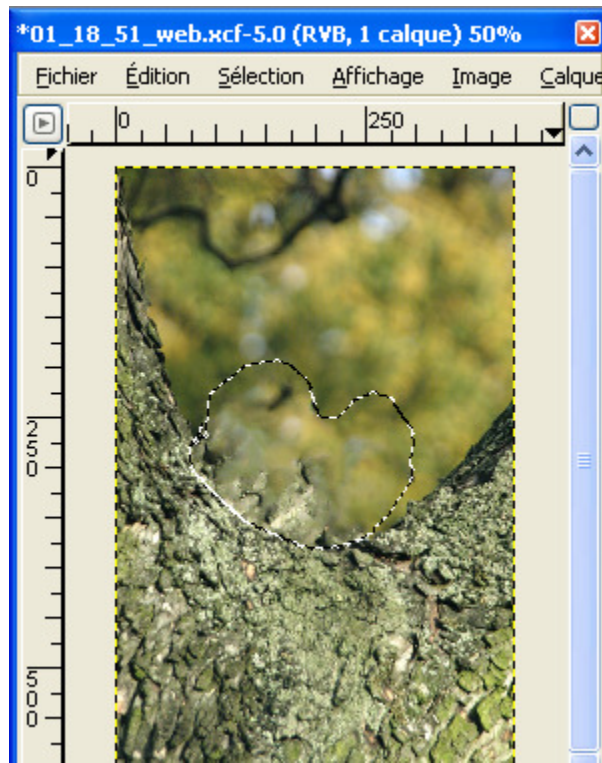
Valider et attendre.

Résultat

Image d'origine



Résultat du Script-Fu



Contenu du Script-Fu smart-remove.scm

```
; Smart Remove version 0.1

; Smart selection eraser.
; Requires resynthesizer plug-in.
; Kinda slow.
; Paul Harrison (pfh@logarithmic.net)

(define (script-fu-smart-remove img layer corpus-border)
  (cond
    ((= 0 (car (gimp-selection-bounds img)))
     (gimp-message "To use this script-fu, first select the region you wish to remove.")
    )
    (t (let*
        (
          (width (car (gimp-drawable-width layer)))
          (height (car (gimp-drawable-height layer)))

          (dupe (car (gimp-image-duplicate img)))

          (channel (car (gimp-selection-save dupe)))
        )

        (gimp-selection-grow dupe corpus-border)
        (gimp-selection-invert dupe)
        (let*
          (
            (old-background (car (gimp-palette-get-background)))
            (channel2 (car (gimp-selection-save dupe)))
          )

          (gimp-selection-load channel)
          (gimp-palette-set-background '(255 255 255))
          (gimp-edit-clear channel2)
          (gimp-palette-set-background old-background)
          (gimp-selection-load channel2)
        )

        (gimp-selection-invert dupe)
        (let*
          (
            (bounds (gimp-selection-bounds dupe))
            (x1 (nth 1 bounds))
            (y1 (nth 2 bounds))
            (x2 (nth 3 bounds))
            (y2 (nth 4 bounds))
          )

          (gimp-image-crop dupe (- x2 x1) (- y2 y1) x1 y1)
        )
        (gimp-selection-invert dupe)

        (plug-in-resynthesizer
          1
          img
          layer
          0
          0
          1
          layer ;hmm, seems to work, was (car (gimp-image-get-active-layer dupe))
          -1
          -1
          0.05 0.0 1)

        (gimp-image-delete dupe)

        (gimp-displays-flush)
      ) ) )

(script-fu-register "script-fu-smart-remove"
  "<Image>/Script-Fu/Enhance/Smart remove selection..."
  "Remove an object from an image by extending surrounding texture to cover it.

Requires resynthesizer plug-in (http://www.logarithmic.net/pfh/resynthesizer).

Yes, it is slow isn't it."
  "Paul Harrison (pfh@logarithmic.net)"
  "Paul Harrison"
  "13/9/2000"
  "RGB* GRAY*"
  SF-IMAGE "Input Image" 0
  SF-DRAWABLE "Input Layer" 0
  SF-ADJUSTMENT "Radius to take texture from" '(40 7 1000 1.0 1.0 0 1)
)
```

Script-Fu smart-enlarge.scn

Ce Script-Fu utilise le greffon « resynthesizer » et propose deux menus accessibles sur l'image par :

- Scripts-Fu, Enhance, Smart enlarge
- Scripts-Fu, Enhance, Smart sharpen

Les exemples utilisent l'écureuil de l'image présentée à la page 38.

Voici le résultat avec une partie de l'écureuil avec **Smart enlarge** (paramètres par défaut) :

Image d'origine

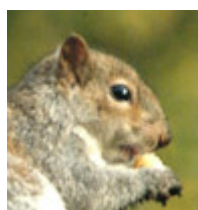


Image à l'échelle 2 **sans** Smart enlarge

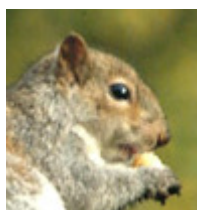


Image à l'échelle 2 **avec** Smart enlarge



Voici le résultat avec une partie de l'écureuil avec **Smart sharpen** (paramètres par défaut) :

Image d'origine



Smart sharpen



Contenu du Script-Fu smart-enlarge.scm

```
; Smart Enlarge version 0.1

; Smart image scaling.
; Requires resynthesizer plug-in.
; Kinda slow.
; Paul Harrison (pfh@yoyo.cc.monash.edu.au)

(define (duplicate img layer)
  (let*
    (
      (width (car (gimp-drawable-width layer)))
      (height (car (gimp-drawable-height layer)))
      (type (car (gimp-drawable-type layer)))
      (base-type (car (gimp-image-base-type img)))

      (new-img (car (gimp-image-new width height base-type)))
      (new-layer (car (gimp-layer-new new-img width height type "Background" 100 0)))
    )

    (gimp-image-add-layer new-img new-layer 0)

    (gimp-edit-copy layer)
    (gimp-floating-sel-anchor (car (gimp-edit-paste new-layer FALSE)))

    new-img
  ))

(define (script-fu-smart-enlarge img layer scale-factor)
  (let*
    (
      (width (car (gimp-drawable-width layer)))
      (height (car (gimp-drawable-height layer)))

      (small (duplicate img layer))
      (large (duplicate img layer))
    )

    (gimp-image-scale small (/ width scale-factor) (/ height scale-factor))
    (gimp-image-scale small width height)

    (gimp-image-scale large (* width scale-factor) (* height scale-factor))

    (gimp-display-new large)

    (plug-in-resynthesizer
      1
      large
      (car (gimp-image-get-active-layer large))
      0
      0
      0
      (car (gimp-image-get-active-layer img))
      (car (gimp-image-get-active-layer small))
      (car (gimp-image-get-active-layer large))
      0.05 0.75 1)

    (gimp-image-delete small)

    (gimp-displays-flush)
  ))

(script-fu-register "script-fu-smart-enlarge"
  "<Image>/Script-Fu/Enhance/Smart enlarge..."
  "Enlarge image, invent high frequency components.
Requires resynthesizer plug-in (http://www.csse.monash.edu.au/~pfh/resynthesizer).
Yes, it is slow isn't it."
  "Paul Harrison (pfh@yoyo.cc.monash.edu.au)"
  "Paul Harrison"
  "27/7/2000"
  "RGB GRAY"
  SF-IMAGE "Input Image" 0
  SF-DRAWABLE "Input Layer" 0
  SF-ADJUSTMENT "Scaling factor" '(2 1 32 0.1 1.0 2 1)
)

(define (script-fu-smart-sharpen img layer scale-factor)
  (let*
    (
      (width (car (gimp-drawable-width layer)))
      (height (car (gimp-drawable-height layer)))

      (small (duplicate img layer))
      (small-blurred (duplicate img layer))
    )

    (gimp-image-scale small (/ width scale-factor) (/ height scale-factor))
    (gimp-image-scale small-blurred (/ width (* 2.0 scale-factor)) (/ height (* 2.0 scale-factor)))
    (gimp-image-scale small-blurred (/ width scale-factor) (/ height scale-factor))
  ))

```

```

(plug-in-resynthesizer
  1
  img
  (car (gimp-image-get-active-layer img))
  0
  0
  0
  (car (gimp-image-get-active-layer small))
  (car (gimp-image-get-active-layer small-blurred))
  (car (gimp-image-get-active-layer img))
  0.05 0.75 1)

(gimp-image-delete small)
(gimp-image-delete small-blurred)

(gimp-displays-flush)
))

(script-fu-register "script-fu-smart-sharpen"
  "<Image>/Script-Fu/Enhance/Smart sharpen..."
  "Sharpen image by inventing high frequency components."

Requires resynthesizer plug-in (http://www.logarithmic.net/pfh/resynthesizer).

Yes, it is slow isn't it."
  "Paul Harrison (pfh@logarithmic.net)"
  "Paul Harrison"
  "27/7/2000"
  "RGB GRAY"
  SF-IMAGE "Input Image" 0
  SF-DRAWABLE "Input Layer" 0
  SF-ADJUSTMENT "Sharpening factor" '(2 1 32 0.1 1.0 2 1)
)

```

PhymodMedia

PhymodMedia est un ensemble d'outils disponibles à <http://www.ordersofn.org/phymod.html> créés par David A. Bartold

Linux

Télécharger <http://www.ordersofn.org/downloads/phymodmedia-0.5.1.tar.gz>

Windows

Michael Schumacher a compilé ce greffon qui est disponible à l'adresse suivante :

<http://schumaml.gmxhome.de/downloads/gimp/phymodmedia-bin.zip>

Ce fichier Zip contient 2 répertoires : plug-ins et scripts.

Dans le premier répertoire vous trouvez le greffon **phymodmedia.exe** que vous placez dans C:\Program Files\GIMP-2.2\lib\gimp\2.0\plug-ins (voir le nota en bas de la page 5)

Dans le second répertoire vous trouvez 2 Scripts-Fu, **cloud-animation.scm** et **thick-oil-paint.scm** .

Vous placez **thick-oil-paint.scm** dans C:\Program Files\GIMP-2.2\share\gimp\2.0\scripts (voir le nota en bas de la page 5)

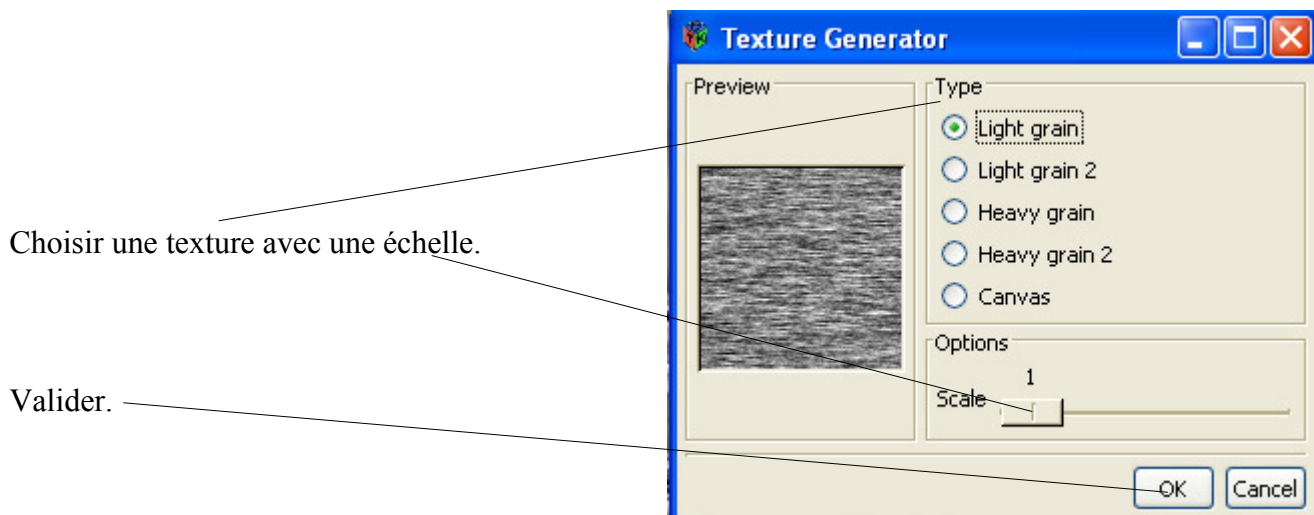
Exemples

Phymomedia offre la possibilité d'utiliser une table à dessins avec différentes options.

L'utilisation se fait en 2 étapes.

La première étape consiste à choisir le support, c'est à dire la texture sur laquelle nous allons dessiner et qui va jouer sur l'aspect des traits. PhymodMedia permet de faire des dessins sur une image.

Pour sélectionner cette texture, sur une image faire : Media, Texture.

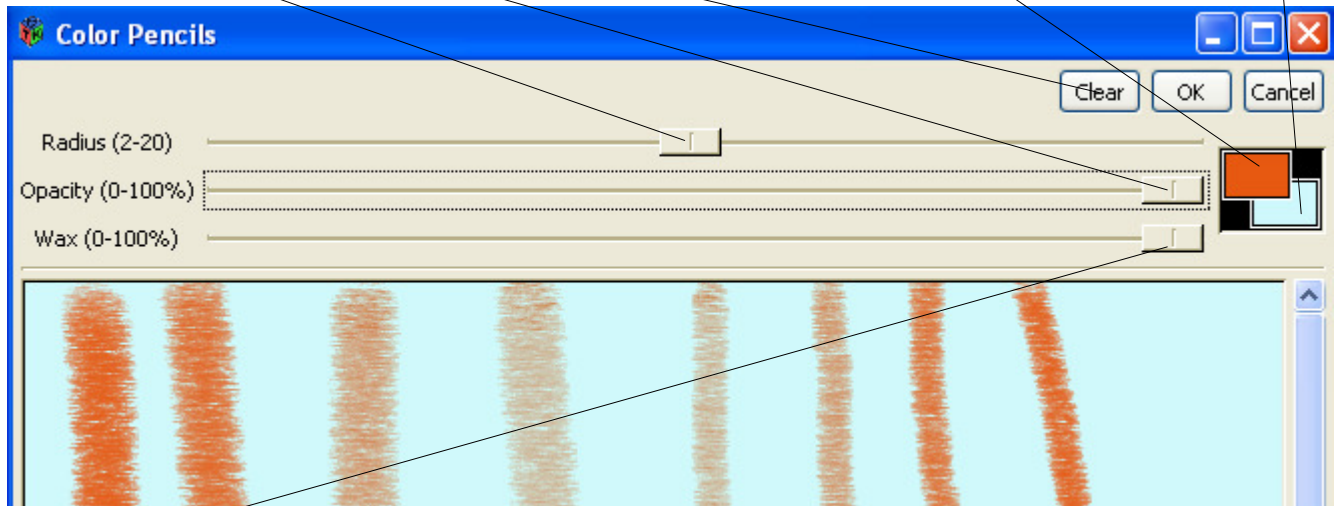


L'étape 2 consiste à dessiner.

Cette possibilité est accessible à partir de l'image aplatie (vide ou non) par le menu : Media, Color Pencils.

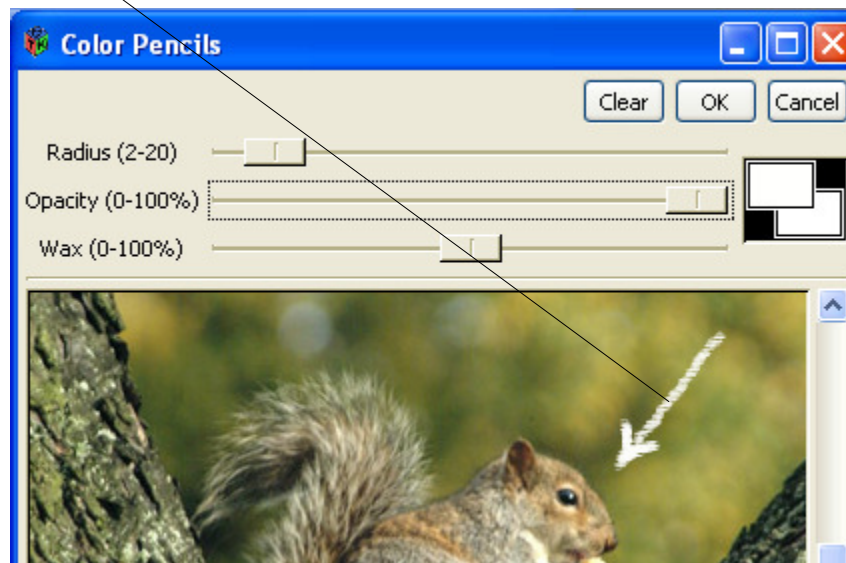
La fenêtre de dessin s'affiche. Lorsque le dessin est terminé valider par OK.

Rayon du crayon Opacité Effacer Sélection de la couleur du crayon Sélection couleur du fond



De gras à dur

PhymodMedia Color Pencils sur l'image utilisée page 38.



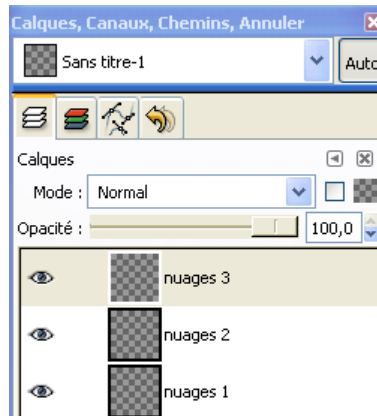
Créer des nuages animés

Le Script-Fu cloud-animation.scm ne fonctionne pas (problème d'utilisation de la fonction gimp-layer-new et de passage de paramètres sur plug-in-sky).

Nous pouvons réaliser manuellement des nuages animés en activant « plug-in-sky » écrit par David A. Bartold qui fait partie de PhymodMedia pour Gimp.

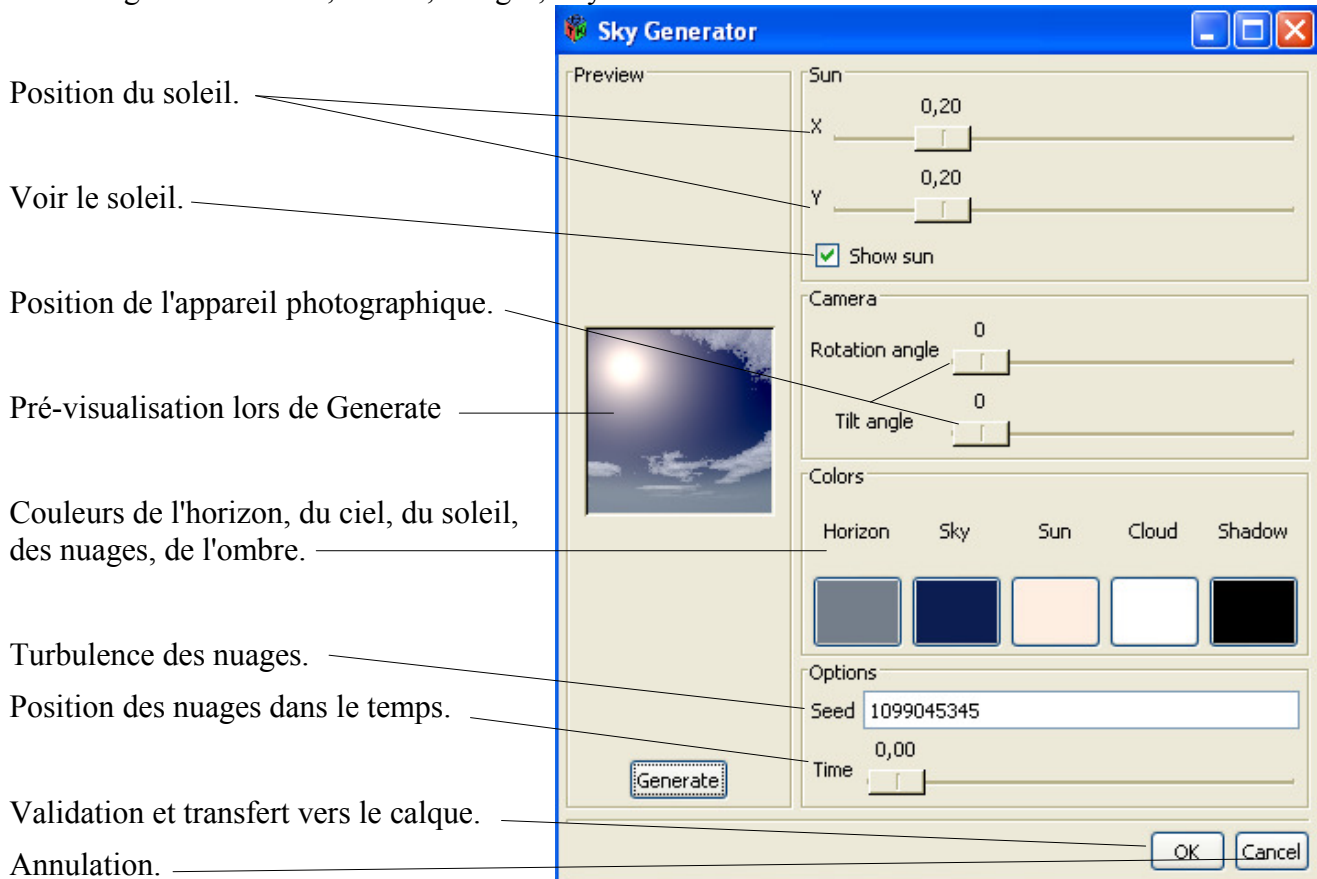
Créer une nouvelle image RVB, dans notre exemple 100*100 pixels.

Créer un nombre de calques correspondant au nombre de séquences de l'animation (3 dans l'exemple) :



Sélectionner le calque nuages1.

Sur l'image faire : Filtres, Rendu, Nuages, Sky. La fenêtre suivante s'affiche :

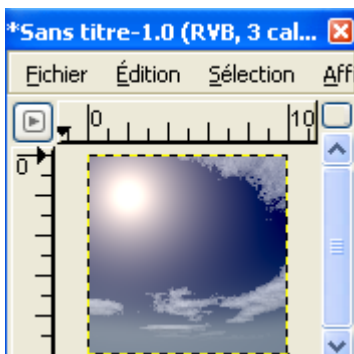


Au départ le curseur Time est positionné à zéro. Lors de la création des 2 autres images de notre exemple il sera positionné à 0.1 et 0.2.

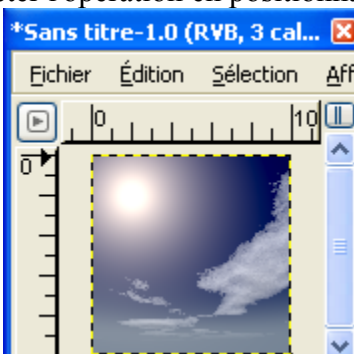
Appuyer sur le bouton Generate pour pré-visualiser la scène.

Appuyer sur OK pour transfert vers le calque nuages1.

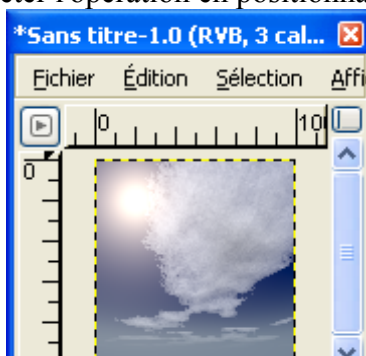
Résultat :



Sélectionner le calque nuages2 et répéter l'opération en positionnant le curseur Time sur 0.1



Sélectionner le calque nuages3 et répéter l'opération en positionnant le curseur Time sur 0.2



Effet de peinture à l'huile sur toile avec le Script-Fu thick-oil-paint.scm

Nous utilisons l'image de l'écureuil présentée à la page 38.

Sur l'image faire, Script-FU, Alchemy, Thick Oil Paint.

Résultat :



Contenu du Script-Fu thick-oil-paint.scm

```
(define (script-fu-thick-oil-paint image
                                     drawable)
  (let* ((original-layer FALSE)
        (glop-layer FALSE)
        (mask-channel FALSE))

    (gimp-undo-push-group-start image)
    (set! original-layer (car (gimp-image-get-active-layer image)))
    (plug-in-oilify 1 image original-layer 7 1)
    (set! glop-layer (car (gimp-layer-copy original-layer TRUE)))
    (gimp-image-add-layer image glop-layer -1)
    (set! mask-channel (car (gimp-layer-create-mask glop-layer WHITE-MASK)))
    (gimp-image-add-layer-mask image glop-layer mask-channel)
    (gimp-edit-copy glop-layer)
    (gimp-floating-sel-anchor (car (gimp-edit-paste mask-channel FALSE)))
    (gimp-invert mask-channel)
    (gimp-brightness-contrast mask-channel 0 85)
    (gimp-image-remove-layer-mask image glop-layer APPLY)
    (plug-in-bump-map 1 image glop-layer glop-layer 160.0 25.0 16 0 0 0 1 0 0)
    (plug-in-apply-canvas 1 image original-layer 0 4)
    (gimp-image-merge-down image glop-layer CLIP-TO-BOTTOM-LAYER)
    (gimp-undo-push-group-end image)
    (gimp-displays-flush)
  ))

(script-fu-register "script-fu-thick-oil-paint"
  "<Image>/Script-Fu/Alchemy/Thick Oil Paint"
  "Applies a heavy oil effect to the specified drawable."
  "David A. Bartold <foxx@mail.utexas.edu>"
  "David A. Bartold"
  "12/17/00"
  "RGB* GRAY*"
  SF-IMAGE "Image" 0
  SF-DRAWABLE "Drawable" 0)
```

« Extras » pour Gimp (brosses & motifs)

Gimp Data Extras (article de gimp-fr.org)

Le paquet gimp-data-extras regroupe un ensemble de brosses et de motifs qui sont distribués séparément du logiciel. Une mise à jour récente a été annoncée et vous pouvez la (re)télécharger depuis <ftp://ftp.gimp.org/pub/gimp/extras/>. Cette mise à jour (<ftp://ftp.gimp.org/pub/gimp/extras/gimp-data-extras-2.0.1.tar.gz>) rajoute 3 motifs qui avaient disparus lors de la mise à jour précédente.

posté par Raymond le 29/10/2004

Sous Windows vous faites une double extraction pour obtenir un sous-répertoire « gimp-data-extras-2.0.1 ».

Dans notre exemple l'extraction s'est faite dans **C:\outils\gimp**

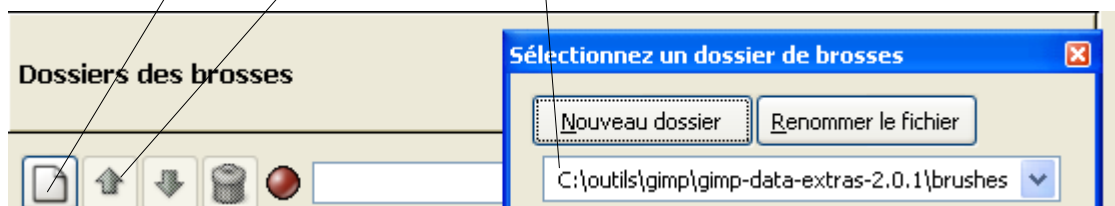
Voici une méthode pour configurer ces nouvelles brosses et nouveaux motifs (Vous pouvez aussi les copier dans un répertoire existant). Dans la fenêtre principale vous faites : Fichier, Préférences.

Une fenêtre s'affiche.

Vous sélectionnez Dossier, Brosses.

Vous appuyez sur ce bouton et indiquez le chemin du nouveau dossier. Valider.

Vous validez et remontez ce chemin pour qu'il arrive en haut de la pile



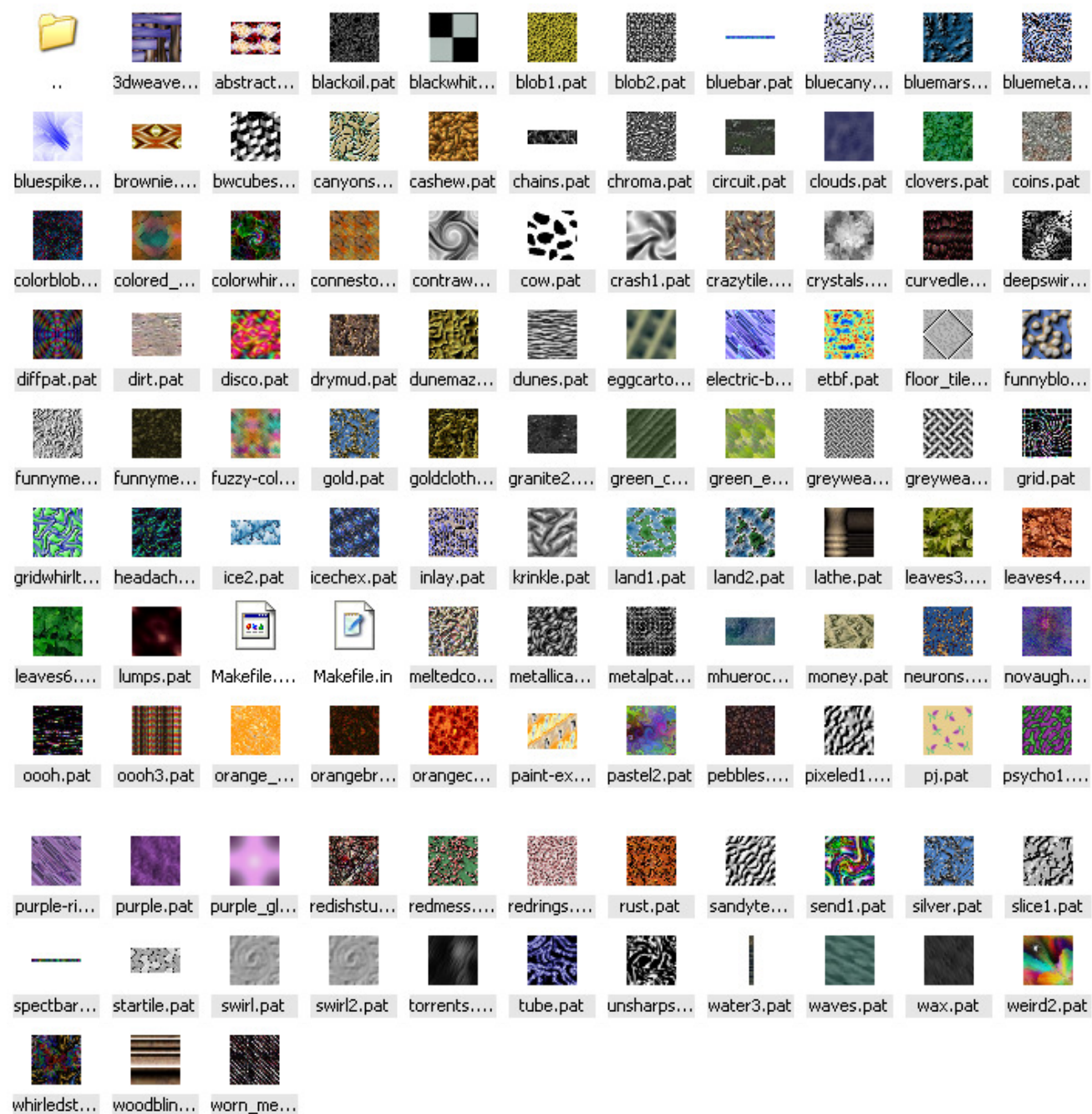
Vous sélectionnez Dossier, Motifs et vous faites la même opération avec le sous-répertoire des motifs C:\outils\gimp\gimp-data-extras-2.0.1\patterns.

Valider et redémarrer Gimp.

Brosses du répertoire C:\outils\gimp\gimp-data-extras-2.0.1\brushes



Motifs du sous-répertoire C:\outils\gimp\gimp-data-extras-2.0.1\patterns



Greffon Voronoi pour créer des motifs

Ce greffon créé par David Nečas permet de créer des motifs.

La page de présentation est <http://trific.ath.cx/software/gimp-plugins/voronoi/> où vous trouverez les explications sur le principe, le fonctionnement ainsi que les téléchargements pour Linux.

Pour Windows, Michael Schumacher a compilé ce greffon qui est disponible sur sa page des greffons : <http://schumaml.gmxhome.de/downloads/gimp/>

Vous décompressez le fichier « voroni.zip » et placez « voroni.exe » dans C:\Program Files\GIMP-2.2\lib\gimp\2.0\plug-ins

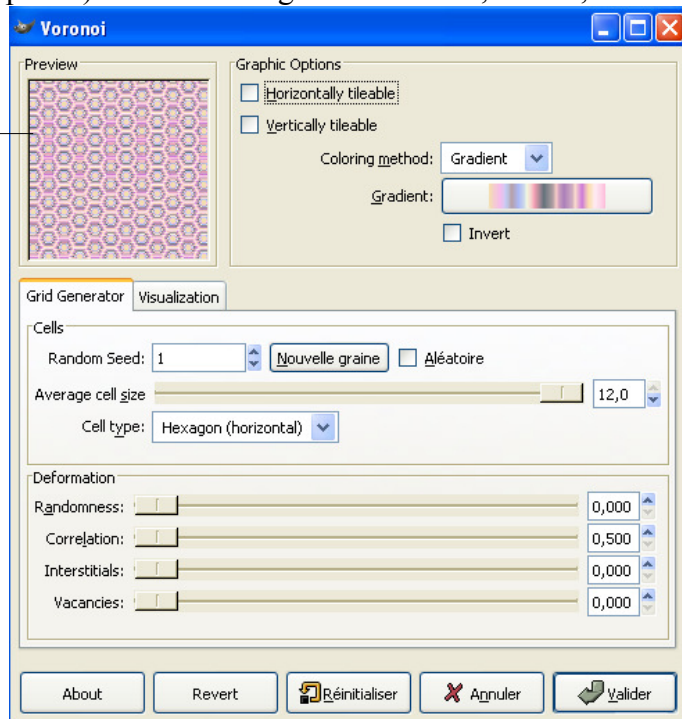
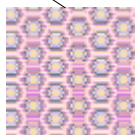
Exemple :

Vous créez une nouvelle image (exemple 64*64 pixels). Sur cette image faire : Filtres, Rendu, Motifs, Voronoi.

La fenêtre de paramétrage s'affiche :

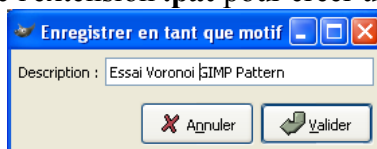
Paramétrer et valider.

Résultat :

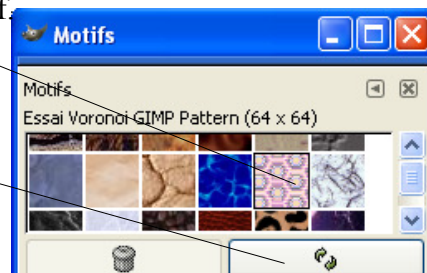


Vous pouvez enregistrer l'image avec l'extension **.pat** pour créer un motif dans un sous répertoire de motifs valide (voir page 57).

Valider.



Dans la fenêtre principale faire un clic sur les motifs pour que la fenêtre de sélection s'affiche. Appuyer sur le bouton de rafraîchissement et sélectionner votre nouveau motif.



Script-Fu layers-view (utilisation)

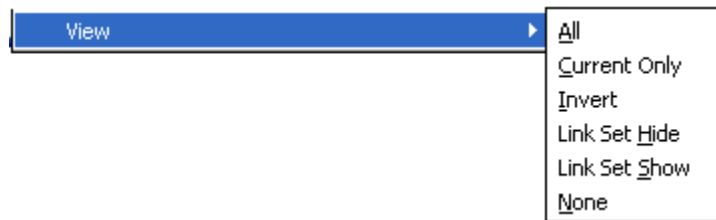
Alan Horkan a écrit le Script-Fu « layers-view.scm » en s'inspirant des possibilités d'un autre logiciel d'images, Jasc Paint Shop Pro 8.

Ce script est disponible a : <http://matrix.netsoc.tcd.ie/~horkana/dev/gnome/gimp/script-fu/scripts/layers-view.scm>

Vous placez **layers-view.scm** dans C:\Program Files\GIMP-2.2\share\gimp\2.0\scripts (voir le nota en bas de la page 5)

Ce Script-Fu est accessible à partir de la fenêtre de l'image par : Calque, View.

Il offre plusieurs possibilités d'agir sur la visibilité des calques à partir de la fenêtre image.



Pour visualiser le fonctionnement vous pouvez ouvrir une image à plusieurs calques ainsi que la fenêtre Calques, canaux & chemins et agir sur les 6 fonctions.

Contenu du Script-Fu

```
:: layers-view.scm -*-scheme-*-  
  
;; Alan Horkan, April 2004. Copyright: Public Domain.  
;; You may do whatever you like with this script but please do share  
;; and try to let me know if you make any interesting changes.  
  
;; Ideas from Jasc Paint Shop Pro 8 (same menu locations used, mostly)  
;; Layers, View: Current Only; All; None; Invert; Show Link Set; Hide Link Set.  
;; Undoable view changes seem odd but i like 'em (and Paint Shop Pro does it too)  
  
; failed to get it working using a straight list  
; looked at copy-visible.scm for inspiration: answer use an array.  
; TODO learn more Scheme, read 'the Wizard book' again ;)  
  
; ideally moving through the layer stack should be abstracted  
; and the specific function passed in a parameter as needed  
; again must read 'the wizard book'  
  
(define (layers-show image boolean)  
  (let* (  
    (layers (gimp-image-get-layers image))  
    (number-layers (car layers))  
    (layer-array (cadr layers))  
  )  
    (gimp-image-undo-group-start image)  
    ; hide all the layers, set visibility FALSE  
    (set! layer-count 0)  
    (while (< layer-count number-layers)  
      (set! layer (aref layer-array layer-count))  
      (gimp-drawable-set-visible layer boolean)  
      (set! layer-count (+ layer-count 1)))  
    (gimp-image-undo-group-end image)  
  )  
  
(define (script-fu-layers-view-all image drawable)  
  (layers-show image TRUE) ; show all the layers  
  (gimp-displays-flush) ; update the display  
)
```

```

(define (script-fu-layers-view-none image drawable)
  (layers-show image FALSE) ; hide all the layers
  (gimp-displays-flush) ; update the display
)

; hide all layers, then (re)show current layer
(define (script-fu-layers-view-current-only image drawable)
  (gimp-image-undo-group-start image)
  (script-fu-layers-view-none image drawable)
  ; drawable could be a channel, must check it.
  (if (not (= 0 (car (gimp-drawable-is-layer drawable))))
    (gimp-drawable-set-visible drawable TRUE)
  )
  (gimp-displays-flush)
  (gimp-image-undo-group-end image)
)

; not the most elegant implementation but it'll do
(define (script-fu-layers-view-invert image drawable)
  (let* (
    (layers (gimp-image-get-layers image))
    (number-layers (car layers))
    (layer-array (cadr layers))
  )
    (gimp-image-undo-group-start image)
    ; hide all the layers, set visibilty FALSE
    (set! layer-count 0)
    (while (< layer-count number-layers)
      (set! layer (aref layer-array layer-count))
      (if (= (car (gimp-drawable-get-visible layer)) TRUE)
        (set! boolean FALSE)
        (set! boolean TRUE)
      )
      (gimp-drawable-set-visible layer boolean)
      (set! layer-count (+ layer-count 1)))
    )
    (gimp-image-undo-group-end image)
    (gimp-displays-flush)
  )
)

(define (layers-view-linked image drawable boolean)
  (let* (
    (layers (gimp-image-get-layers image))
    (number-layers (car layers))
    (layer-array (cadr layers))
  )
    (gimp-image-undo-group-start image)
    ; hide all the layers, set visibilty FALSE
    (set! layer-count 0)
    (while (< layer-count number-layers)
      (set! layer (aref layer-array layer-count))
      ; show/hide linked layer and hide/show the others
      (if (= FALSE (car (gimp-drawable-get-linked layer)))
        ; must use #t or #f booleans, TRUE or FALSE unrecognised by (not)
        ; (set! boolean (not boolean)) ; this isn't working quite right
        ; dirty inelegant hack because (not) wont work as expected
        (set! boolean (if (= boolean TRUE) FALSE TRUE) )
      )
      (gimp-drawable-set-visible layer boolean)
      (set! layer-count (+ layer-count 1)))
    )
    (gimp-image-undo-group-end image)
  )
)

(define (script-fu-layers-view-linked-show image drawable)
  (layers-view-linked image drawable TRUE) ; #t == TRUE
  (gimp-displays-flush)
)

(define (script-fu-layers-view-linked-hide image drawable)
  (layers-view-linked image drawable FALSE) ; #f == FALSE
  (gimp-displays-flush)
)

; this doesn't make as much sense as it does in Paint Shop Pro
; it has many seperate numbered link sets, gimp allows only 1

.....

(set! menu _ "<Image>/Layer/View/")
(set! menu _ "<Image>/Layer/Show ")
;; uncomment above line to move (Show) All/None up to the Layer menu.

;; append/prepend description(s) to about string.
(set! description "
With sincere thanks to Jasc Paint Shop Pro 8 for the inspiration. "
)
(set! authors "Alan Horkan. ")
(set! copyrights "Alan Horkan, 2004. Public Domain. ")
(set! date "2004 04 02 UTC")

(script-fu-register "script-fu-layers-view-current-only"
  _ "<Image>/Layer/View/_Current Only"

```

```

(string-append "Show only the current Layer, hide all other layers. " description)
authors copyrights date "" ; strings set using variables, see above
SF-IMAGE "Image" 0
SF-DRAWABLE "Drawable" 0
)

(script-fu-register "script-fu-layers-view-all"
  (string-append menu _" _All")
  (string-append "Show All Layers. Sets every layer to visible. " description)
  authors copyrights date ""
  SF-IMAGE "Image" 0
  SF-DRAWABLE "Drawable" 0
)

(script-fu-register "script-fu-layers-view-none"
  (string-append menu _" _None")
  (string-append "Show None of the Layers. Hides every layer. " description)
  authors copyrights date ""
  SF-IMAGE "Image" 0
  SF-DRAWABLE "Drawable" 0
)

(script-fu-register "script-fu-layers-view-invert"
  "<Image>/Layer/View/_Invert"
  "Invert the visibility of all Layers.
If a layer is showing, it gets hidden and if a layer is hidden it will be shown. "
  authors copyrights date "" ; strings set using variables, see above
  SF-IMAGE "Image" 0
  SF-DRAWABLE "Drawable" 0
)

;; the names "Link Set Show" and "Link Set Hide" are only a workaround
;; so the two menu items will be beside each other sorted alphabetically sorted

(script-fu-register "script-fu-layers-view-linked-show"
  ; "<Image>/Layer/View/_Show Link Set" ; same as Paint Shop Pro
  "<Image>/Layer/View/Link Set _Show"
  (string-append "Show Link Set. Show only the linked layers and hide all other layers. " description)

  authors copyrights date ""
  SF-IMAGE "Image" 0
  SF-DRAWABLE "Drawable" 0
)

(script-fu-register "script-fu-layers-view-linked-hide"
  ; "<Image>/Layer/View/_Hide Link Set" ; same as Paint Shop Pro
  "<Image>/Layer/View/Link Set _Hide" ; sorted, beside link set show
  (string-append "Hide Link Set. Hide only the linked layers and show all other layers. " description)
  authors copyrights date ""
  SF-IMAGE "Image" 0
  SF-DRAWABLE "Drawable" 0
)

```

Utiliser un Script-Fu pour créer un logo

Walter Higgins propose à la page <http://www.xanadb.com/archive/gimp/> un Script-Fu « ursula-higgins-logo » pour créer un logo.

L'auteur aime la programmation et son script est très intéressant. Voici quelques points forts de sa démarche :

- L'idée d'utiliser un langage de programmation.
- L'utilisation d'un logiciel libre.
- L'utilisation combinée de polices de caractères, de formes géométriques, etc. dans le script.
- La facilité de mise au point du graphisme avec un ajustement précis des coordonnées.
- Script-Fu facilement modifiable pour obtenir un autre rendu.

Vous copiez les lignes du Script dans un éditeur de texte (Bloc-notes par exemple).

Vous enregistrez dans le répertoire des scripts (personnel ou système) avec l'extension .scm (ursula_higgins_logo.scm par exemple). Voir le début de cette documentation à la page 5 et le nota en bas de la page 5 pour l'enregistrement.

Pour activer ce script, depuis la fenêtre de l'image, faite : Script-FU, Alchemy, Ursula Higgins Logo.

Technique

Voici un lien vers la documentation « GIMP Reference Manuals for the GIMP 2.0 API » :

<http://developer.gimp.org/api/2.0/index.html>

Base des procédures : Depuis la « fenêtre Principale » faire ==>

Exts, Explorer la base de procédures.

Programme pour créer des Scripts-Fu sous Windows qui fonctionne dans un environnement Visual Basic (non testé) : http://abcdugimp.free.fr/Script-fu/P_menu_to_gimp.htm

Pour que la modification d'un script-fu soit prise en compte dans Gimp, à partir de la « fenêtre Principale », faire :

Exts, Script-Fu, Rafraîchir les scripts

ou

Xtns, Script-Fu, Refresh Scripts.

Autres sites proposant des logos :

<http://8-p.info/gimp/> (des logos réalisés par KATO Kazuyoshi)

Contenu du Script-Fu « Ursula Higgins Logo »

```
;
; creates an Ursula Higgins Logo
;
(define (script-fu-ursula-higgins-logo image drawable)
  (define (fill-ring drawable color)
    (gimp-ellipse-select image 10 10 128 128 CHANNEL-OP-ADD TRUE 0 0)
    (gimp-ellipse-select image 18 18 112 112 CHANNEL-OP-SUBTRACT TRUE 0 0)
    (gimp-palette-set-foreground color)
    (gimp-edit-fill drawable FOREGROUND-FILL)
  )
  (define (add-uh image drawable)
    (let* ((u-layer (car (gimp-text image drawable 28 24 "u" -1 TRUE 84 PIXELS "" "Bitstream Vera Sans Bold" "" "" "" "" "" "")))
      (h-layer (car (gimp-text image drawable 59 24 "h" -1 TRUE 84 PIXELS "" "Bitstream Vera Sans Bold" "" "" "" "" "" "")))
      )
      (cons u-layer (cons h-layer))
    )
  )
  (gimp-undo-push-group-start image)
  (let* ((width 160)
    (height 160)
    (logo-layer (car (gimp-layer-new image width height RGBA-IMAGE "Ursula Higgins Logo" 100 NORMAL-MODE)))
    )
    (gimp-image-add-layer image logo-layer -1)
    (gimp-image-set-active-layer image logo-layer)
    (gimp-edit-clear logo-layer)
    (fill-ring logo-layer '(70 130 180))
    (gimp-selection-none image)
    (gimp-palette-set-foreground '(0 0 0)) ; black
    (set! text-layers (add-uh image -1))

    (set! u-layer (car text-layers))
    (set! h-layer (car (cdr text-layers)))

    (gimp-layer-set-opacity u-layer 50)
    (gimp-layer-set-opacity h-layer 50)

    (set! merged-layer (car (gimp-image-merge-down image
      (car (gimp-image-merge-down image h-layer 0)) 0)))

    (set! shadow-layer (car (gimp-layer-new image width height RGBA-IMAGE "Logo Shadow" 70 NORMAL-MODE)))
    (gimp-image-add-layer image shadow-layer 1)
    (gimp-image-set-active-layer image shadow-layer)
    (gimp-edit-clear shadow-layer)

    (fill-ring shadow-layer '(0 0 0))

    (gimp-selection-none image)

    ;(gimp-floating-sel-anchor (car (cdr (add-uh image shadow-layer))))

    (plug-in-gauss-iir2 1 image shadow-layer 15 15)
    (gimp-layer-set-offsets shadow-layer 8 8)

    (gimp-undo-push-group-end image)
    (gimp-displays-flush))
  )

(script-fu-register "script-fu-ursula-higgins-logo"
  "~<Image>~/Script-Fu/Alchemy/Ursula Higgins Logo"
  "Ursula Higgins Logo"
  "Walter Higgins"
  "Walter Higgins"
  "Oct 2004"
  "RGB* GRAY* INDEXED*"
  SF-IMAGE "Image" 0
  SF-DRAWABLE "Drawable" 0
  )
```

Script-Fu photo-stack (effet)

Ce script écrit à l'origine par Ian Cameron Smith a été adapté sur le site <http://www.dotsphinx.com/> et présenté à la page <http://www.dotsphinx.com/software/photostack/>.

Ce Script-Fu permet, à partir d'une image, de créer une pile où sont empilées des photographies.

Vous enregistrez <http://www.dotsphinx.com/software/photostack/files/gimp-1.3/photo-stack.scn> dans le répertoire des scripts. Voir le début de cette documentation à la page 5 et le nota en bas de la page 5 pour l'enregistrement.

Ce Script-Fu est accessible à partir de la fenêtre de l'image par : Script-Fu, Modify, Photo Stack.

Origine ==>



Script-Fu photo-stack ==>



Contenu du Script-Fu « photo-stack »

```
; photo-stack
;
; This script turns a simple image into a rendition of a pile of
; photographs, with the original image at the top. Each picture in
; the stack has a white border added, and a shadow behind it (to
; delimit it from the one below). All the underlying pictures are, of
; course, copies of the top one.
;
; The following options are supported (with the given defaults):
; Border Color (white):
;     The color of the border added to the image.
; Border Width (10):
;     The width in pixels of the border added to the image.
; Pictures in Stack (3):
;     The number of pictures in the final stack, including the
;     original image.
; Rotate Sloppiness (100):
;     Amount of rotational sloppiness to add to each picture. 0
;     means don't rotate them.
; Move Sloppiness (10):
;     Amount of sideways sloppiness to add to each picture. 0 means
;     don't move them.
; Top Straight (TRUE):
;     If TRUE, keep the top image straight; otherwise, skew it like
;     the others.
; Add Shadow (TRUE):
;     If TRUE, add a shadow to each layer (recommended).
; Shadow Radius (12.0):
;     Blur radius in pixels used to generate each layer's shadow.
; Shadow Weight (100%):
;     Opacity for the shadow layers: 0 - 100.
; Work on Copy (TRUE):
;     If TRUE, everything is done on a copy of the original image
;     (highly recommended).
; Flatten Layers (FALSE):
;     If TRUE, the image is merged into a single layer when all's
;     done, and autocrops it.
;
; Issues:
; * Unless the user elects to flatten the resulting image, I don't
;   autocrop it, and you're left with a huge image. I need an
;   autocrop which takes into account the sizes and extents of
;   *all* the layers, not just the current one, to fix this.
; * With very large move sloppiness values, you could get a
;   clipped image.
; * Using this on a multi-layered source will discard the layers.
; * Undo doesn't work, I think largely because of the "Add Border"
;   script. I could do the border adding within the script
;   easily enough and fix that.
;
; Thoughts:
; * Currently all the pictures in the stack are the same size. I
;   could add some random cropping and/or resizing, and even 90°
;   rotations...
; * Could have the ability to make multiple images, from separate
;   layers, into a stack... And then maybe add a "permute"
```

```

; option, to generate multiple stacks, one with each image on
; top, to make a whole photo album.

; Main function:
(define (script-fu-photo-stack
  inImage
  inLayer
  inColor
  inSize
  inNum
  inSlop
  inSpill
  inTopStraight
  inShadow
  inShadRadius
  inShadWeight
  inCopy
  inFlatten
)

  (let* (
    theImage theLayer width height newLayer angle
  )

    ; Figure out what image and layer we're working on. If the user
    ; wants to work on a copy, then copy the image; the passed-in
    ; layer handle (inLayer) then becomes invalid.
    (set! theImage
      (if (= inCopy TRUE)
          (car (gimp-image-duplicate inImage))
          inImage
      ))

    (set! theLayer (car (gimp-image-get-active-layer theImage)))

    (gimp-undo-push-group-start theImage)
    (strand (realtime))

    ; Stick a border on the original image; then flatten it, so we've
    ; got a single layer to work on.
    (script-fu-addborder theImage theLayer inSize inSize inColor 0)
    (set! theLayer (car (gimp-image-flatten theImage)))
    (gimp-layer-add-alpha theLayer)

    ; Double the width and height of the image. I can't think of a
    ; better way to figure out how big it needs to be, to handle all
    ; the random resizing and skewing we're going to do.
    (set! width (car (gimp-image-width theImage)))
    (set! height (car (gimp-image-height theImage)))
    (gimp-image-resize theImage
      (* width 2) (* height 2)
      (/ width 2) (/ height 2))

    (gimp-layer-resize theLayer
      (* width 2) (* height 2)
      (/ width 2) (/ height 2))

    ; For every *additional* copy we need, add it.
    (while (> inNum 1)
      ; Copy the original to make the new copy, just under the
      ; original.
      (set! newLayer (car (gimp-layer-copy theLayer TRUE)))
      (gimp-image-add-layer theImage newLayer 1)

      ; Skew it, and add a shadow if required.
      (clint-skew-layer theImage newLayer inSlop inSpill)
      (if (= inShadow TRUE)
          (clint-add-shadow theImage newLayer 1 inShadRadius inShadWeight))

      (set! inNum (- inNum 1))
    )

    ; If we don't need the original (top) image kept straight, skew
    ; it; add a shadow to the original if required.
    (if (= inTopStraight FALSE)
        (clint-skew-layer theImage theLayer inSlop inSpill))
    (if (= inShadow TRUE)
        (clint-add-shadow theImage theLayer 0 inShadRadius inShadWeight))

    ; Flatten the image if requested. Having done this, we can safely
    ; autocrop it.
    (if (= inFlatten TRUE)
        (begin
          (set! theLayer (car (gimp-image-merge-visible-layers theImage 1)))
          (plug-in-autocrop 1 theImage theLayer)
        ))

    ; Tidy up.
    (if (= inCopy TRUE)
        (begin
          (gimp-image-clean-all theImage)
          (gimp-display-new theImage)
        ))

    (gimp-undo-push-group-end theImage)
  )
)

```

```

)

; Skew a layer by the given slop (rotation) and spill (vert/horiz
; shift) factors, randomized.

(define (clint-skew-layer
  image
  layer
  slop
  spill)
  (let* (dx dy angle)
    ; X/Y shift.
    (if (> spill 0)
      (begin
        (set! dx (+ (rand spill) 3))
        (set! dy (+ (rand spill) 3))
        (if (= (rand 2) 1) (set! dx (- 0 dx)))
        (if (= (rand 2) 1) (set! dy (- 0 dy)))
        (gimp-layer-translate layer dx dy)
      ))
      ; Rotational skew.
      (if (> slop 0)
        (begin
          (set! angle (+ (/ (rand slop) 1000) 0.02))
          (if (= (rand 2) 1) (set! angle (- 0 angle)))
          (gimp-rotate layer TRUE angle)
        ))
      ))
  )
)

```

; Add a shadow layer.

```

(define (clint-add-shadow
  image
  layer
  depth
  radius
  opacity)
  (set! layer (car (gimp-layer-copy layer TRUE)))
  (gimp-image-add-layer image layer (+ depth 1))
  (gimp-selection-layer-alpha layer)
  (gimp-palette-set-foreground '(0 0 0))
  (gimp-bucket-fill layer 0 0 100 0 FALSE 0 0)
  (gimp-selection-none image)
  (plug-in-gauss-iir 1 image layer radius TRUE TRUE)
  (gimp-layer-set-opacity layer opacity)
)

```

; Register the function with the GIMP:

```

(script-fu-register
 "script-fu-photo-stack"
 "<Image>/Script-Fu/Modify/Photo Stack"
 "foo"
 "Ian Cameron Smith"
 "1998, Ian Cameron Smith, http://www.hermit.org/"
 "27 May 1998"
 "RGB RGBA GRAY GRAYA"
 SF-IMAGE "The Image" 0
 SF-DRAWABLE "The Layer" 0
 SF-COLOR "Border Color:" '(255 255 255)
 SF-VALUE "Border Width:" "10"
 SF-VALUE "Pictures in Stack:" "3"
 SF-VALUE "Rotate Sloppiness:" "100"
 SF-VALUE "Move Sloppiness:" "10"
 SF-TOGGLE "Top Straight?" TRUE
 SF-TOGGLE "Add Shadow?" TRUE
 SF-VALUE "Shadow Radius:" "12.0"
 SF-VALUE "Shadow Weight (%):" "100"
 SF-TOGGLE "Work on Copy?" TRUE
 SF-TOGGLE "Flatten Layers?" FALSE
)

```

Script-Fu Corrosion (effet)

Voici de jolis effets de corrosion, d'oxydation d'un Script-Fu inspiré par le didacticiel (en anglais) de Ron Scott disponible à la page <http://gug.sunsite.dk/tutorials/ronq1/>

Les auteurs du Script-Fu sont Laetitia Marin et Raymond Ostertag.

Le code source modifié pour Gimp 2 est disponible sur les 2 pages suivantes (texte rouge).

Vous pouvez le copier dans un éditeur de texte et le sauvegarder dans le répertoire des scripts sous corrosion.scn (voir le début de cette documentation à la page 5 et le nota en bas de la page 5 pour l'enregistrement).

Pour accéder au script à partir de la fenêtre de l'image Faire : Script-Fu, Rendu, Corrosion.

L'image de l'exemple est celle utilisée sur le didacticiel de Ron Scott.

Image d'origine ==>



Paramètres

Picker grid spacing	20
Picker Treshold	30
Picker turbulence	30
Bumpmap depth	10

Résultat ==>

```

; Debut du script
; The GIMP -- an image manipulation program
; Copyright (C) 1995 Spencer Kimball and Peter Mattis
; A set of layer effects script for GIMP 1.2
; Copyright (C) 2001 Laetitia Marin titix@gimpforce.org
; Copyright (C) 2001 Ostertag Raymond coordonateur@gimp-fr.org
;-----
; version 0.1 2001-september-01
; - Initial release
; version novembre 2004 -aljacom- adaptation ou remplacement pour Gimp version 2 sur les fonctions
; gimp-layer-new gimp-bucket-fill gimp-fuzzy-select gimp-by-color-select gimp-color-picker
;-----
; This program is free software; you can redistribute it and/or modify
; it under the terms of the GNU General Public License as published by
; the Free Software Foundation; either version 2 of the License, or
; (at your option) any later version.
; This program is distributed in the hope that it will be useful,
; but WITHOUT ANY WARRANTY; without even the implied warranty of
; MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
; GNU General Public License for more details.
; You should have received a copy of the GNU General Public License
; along with this program; if not, write to the Free Software
; Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
;-----
; This is the official English version you'll find a french version at http://www.gimp-fr.org/
; Script-fu corrosion an attempt to realise the Scott-Effect
; Start : a selection in an image or a layer with a transparency area who will be transformed
; in selection
; See the manual at the tutorial section of the gug http://gug.sunsite.dk/
;
;
(define (script-fu-corrosion img calque0 picker_spacing picker_treshold picker_turbulence bumpmap_depth)
(let* ((sizeX (car (gimp-image-width img)))
(sizeY (car (gimp-image-height img)))
(calque1
(car
*****
;
;
; Ligne suivante modification novembre 2004 gimp-layer-new
(gimp-layer-new img sizeX sizeY 1 "plasma" 100 5)))
(masque1
(car
(gimp-layer-create-mask calque1 1)))
(old-fg (car (gimp-palette-get-foreground)))
(old-bg (car (gimp-palette-get-background)))
(blanc '(255 255 255))
)
; undo initialisation
(gimp-undo-push-group-start img)
; picker_spacing must be between 2 and 20 ( change that if you need more )
(while (< picker_spacing 2)
(begin (set! picker_spacing (+ picker_spacing 1))))
(while (> picker_spacing 20)
(begin (set! picker_spacing (- picker_spacing 1))))
; layer 1
(gimp-image-add-layer img calque1 0)
(gimp-image-add-layer-mask img calque1 masque1)
; plasma
(set! seed (* picker_treshold picker_turbulence))
(plug-in-plasma TRUE img calque1 seed 2.5)
(let* ((calque2 (car (gimp-layer-copy calque1 TRUE)))
(masque2 (car (gimp-layer-mask calque2))))
; layer 2
(gimp-image-add-layer img calque2 0)
; fill the layer-mask
(gimp-palette-set-foreground blanc)
(set! activ_selection (car (gimp-selection-is-empty img)))
(cond
((= activ_selection 0) ; selection activ
*****
;
;
; Lignes suivantes modifications novembre 2004 gimp-bucket-fill
(gimp-edit-bucket-fill masque1 0 0 100 0 FALSE 0 0)
(gimp-edit-bucket-fill masque2 0 0 100 0 FALSE 0 0)
(gimp-selection-none img))
((= activ_selection 1) ; no selection activ
(gimp-selection-layer-alpha calque0)
(gimp-edit-bucket-fill masque1 0 0 100 0 FALSE 0 0)
(gimp-edit-bucket-fill masque2 0 0 100 0 FALSE 0 0)
)
) ; end of cond
; picking in the plasma
(gimp-selection-clear img)
(let ((y (/ sizeY picker_spacing)))
(while (< y sizeY)
(begin
(let ((x (/ sizeX picker_spacing)))
(while (< x sizeX)
; back to the initials colours and display the result

```

```

-*****
;
;Ligne suivante modification novembre 2004 gimp-fuzzy-select
;
(gimp-fuzzy-select calque1 x y picker_threshold 0 TRUE FALSE 0 FALSE)
; back to the initials colours and display the result
-*****
;Ligne suivante modification novembre 2004 gimp-color-picker
(set! pick_color (car (gimp-color-picker img calque1 x y FALSE FALSE 10)))
-*****
;Ligne suivante modification novembre 2004 gimp-by-color-select
(gimp-by-color-select calque1 pick_color picker_turbulence 0 FALSE FALSE 15 FALSE)
(set! x (+ x (/ sizeX picker_spacing))))
(set! y (+ y (/ sizeY picker_spacing))))
; clean and fill the layers 1 and 2
(gimp-selection-invert img)
(gimp-edit-clear calque1)
(gimp-edit-clear calque2)
(gimp-selection-clear img)
(gimp-desaturate calque1)
(gimp-desaturate calque2)
; bumpmap on layer 2
(plug-in-bump-map TRUE img calque2 calque2 135 45 bumpmap_depth 0 0 0 TRUE FALSE LINEAR)
; back to the initials colours and display the result
(gimp-palette-set-foreground old-fg)
(gimp-palette-set-background old-bg)
(gimp-undo-push-group-end img)
(gimp-displays-flush))))
(script-fu-register "script-fu-corrosion"
"<Image>/Script-Fu/Render/Corrosion"
"Scott-effect : corrosion"
"titix and raymond"
"2001, titix and raymond"
"12 NOVEMBRE 2004"
""
SF-IMAGE "Image" 0
SF-DRAWABLE "Drawable" 0
SF-VALUE "Picker grid spacing [2-20]" "5"
SF-VALUE "Picker Threshold" "30"
SF-VALUE "Picker turbulence" "30"
SF-VALUE "Bumpmap depth" "10")
; Fin du script

```

Script-Fu ISO Noise Reduction (retouche photo)

Le Script-Fu ISO Noise Reduction V.0 (**réduction du bruit**) est proposé par Martin Egger. Il est disponible à la page <http://www.forum4designers.com/message135333.html> .

La version 1.1 du Script-Fu ISO Noise Reduction est disponible depuis le 8 décembre 2004 sur http://groups-beta.google.com/group/comp.graphics.apps.gimp/browse_frm/thread/18df09889ed695da/22d32c7db1d594ac?_done=%2Fgroup%2Fcomp.graphics.apps.gimp%3Fnum%3D25%26hl%3Den%26lr%3D%26start%3D0%26&_doneTitle=Back+to+topics&_doneTitle=Back&&d#22d32c7db1d594ac

mais il est préférable de copier le code source disponible sur les 2 pages suivantes (texte rouge) dans un éditeur de texte et de le sauvegarder dans le répertoire des scripts sous iso_noise_reduction.scm

Pour accéder au script à partir de la fenêtre de l'image Faire : Script-Fu, Eg, ISO Noise Reduction.

Voici la description donnée par l'auteur : *You can find more about ISO Noise Reduction at <http://www.gimpguru.org/Tutorials/ReducingCCDNoise/> but this script uses a different method (masking the edges and then blurring the individual color channels or the luminance channel only).*

Vous pouvez l'essayer sur l'image du didacticiel d'Eric R. Jeschke :

<http://www.gimpguru.org/Tutorials/ReducingCCDNoise/before-384x512.jpg>

Le résultat est très bon avec l'option "Noise Reduction Method" = "Luminance channel blurring"

Nota :

Vous pouvez voir la comparaison entre plusieurs moyens de réduction du bruit sur notre petit document disponible dans la partie « Divers » de notre page.

Le document présente 3 différents procédés de réduction de bruit sur les images :

- Résultat du didacticiel « Reducing CCD Noise with The GIMP » créé par Eric R. Jeschke.
- Résultat obtenu avec le Script-Fu « ISO Noise Reduction » de Martin Egger.
- Résultat avec le logiciel « GREYCstoration » de David Tschumperlé.

Lien vers ce document: [reduction_du_bruit.zip](#) (environ 300ko)

Contenu du Script-Fu ISO Noise Reduction version 1.1

```
;
;
; ISO Noise Reduction, V1.1
;
; Martin Egger (martin.egger@gmx.net)
; (C) 2004, Bern, Switzerland
;
; You can find more about ISO Noise Reduction at
; http://www.gimpguru.org/Tutorials/ReducingCCDNoise/
; but this script uses a different method (masking the edges and then
; blurring the individual color channels or the luminance channel only).
;
; This plugin was tested with Gimp 2.2
;
; This program is free software; you can redistribute it and/or modify
; it under the terms of the GNU General Public License as published by
; the Free Software Foundation; either version 2 of the License, or
; (at your option) any later version.
;
; This program is distributed in the hope that it will be useful,
; but WITHOUT ANY WARRANTY; without even the implied warranty of
; MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
; GNU General Public License for more details.
;
; You should have received a copy of the GNU General Public License
; along with this program; if not, write to the Free Software
; Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
;
; Define spline values for curves
;
(define (set-pt a index x y)
  (progn
    (aset a (* index 2) x)
    (aset a (+ (* index 2) 1) y)
  )
)
;
; Special curve for layer mask
;
(define (EdgeSpline)
  (let* (
    (a (cons-array 4 'byte))
    )
    (set-pt a 0 100 0)
    (set-pt a 1 190 255)
    a
  )
)
;
; Define the function
;
(define (script-fu-Eg-ISONoiseReduction InImage InLayer InEdge InType
  InRadius InDelta)
  (let* (
    (TempLayer (car (gimp-layer-copy InLayer TRUE)))
    (RadiusRB (* InRadius 1.5))
    (DeltaRB (* InDelta 2))
    )
  )
; Save history
;
  (gimp-image-undo-group-start InImage)
;
  (gimp-image-add-layer InImage TempLayer -1)
;
; Find edges, Radius = 2, Warpmode = Smear (1), Edgemode = Sobel (0)
;
  (plug-in-edge TRUE InImage TempLayer InEdge 1 0)
  (plug-in-gauss TRUE InImage TempLayer 4 4 0)
  (gimp-curves-spline TempLayer HISTOGRAM-VALUE 4 (EdgeSpline))
  (gimp-by-color-select TempLayer '(0 0 0) 80 CHANNEL-OP-ADD FALSE FALSE 0 FALSE)
  (let* (
    (MaskChannel (car (gimp-selection-save InImage)))
    )
    (gimp-image-remove-layer InImage TempLayer)
  )
; Blur separate RGB channels, use different radius/delta for Red/Blue and Green
;
  (if (= InType 0)
    (begin
      (gimp-image-set-component-active InImage RED-CHANNEL TRUE)
      (gimp-image-set-component-active InImage GREEN-CHANNEL FALSE)
      (gimp-image-set-component-active InImage BLUE-CHANNEL FALSE)
      (plug-in-sel-gauss TRUE InImage InLayer RadiusRB DeltaRB)
    )
    (begin
      (gimp-image-set-component-active InImage RED-CHANNEL FALSE)
      (gimp-image-set-component-active InImage GREEN-CHANNEL TRUE)
      (plug-in-sel-gauss TRUE InImage InLayer InRadius InDelta)
    )
  )
)
```

```

;
; (gimp-image-set-component-active InImage GREEN-CHANNEL FALSE)
; (gimp-image-set-component-active InImage BLUE-CHANNEL TRUE)
; (plug-in-sel-gauss TRUE InImage InLayer RadiusRB DeltaRB)
;
; (gimp-image-set-component-active InImage RED-CHANNEL TRUE)
; (gimp-image-set-component-active InImage GREEN-CHANNEL TRUE)
; (gimp-selection-none InImage)
; )
; )
; Blur luminance channel
;
; (if (= InType 1)
;   (begin
;     (let* (
;       (OrigLayer (cadr (gimp-image-get-layers InImage)))
;       (LABImage (car (plug-in-decompose TRUE InImage InLayer "LAB" TRUE)))
;       (LABLayer (cadr (gimp-image-get-layers LABImage)))
;       (LLayer (car (gimp-layer-copy InLayer TRUE)))
;     )
;
;       (gimp-image-add-layer InImage LLayer -1)
;       (gimp-selection-all LABImage)
;       (gimp-edit-copy (aref LABLayer 2))
;       (gimp-floating-sel-anchor (car (gimp-edit-paste LLayer FALSE)))
;       (gimp-selection-load MaskChannel)
;       (plug-in-sel-gauss TRUE InImage LLayer RadiusRB DeltaRB)
;       (gimp-selection-all InImage)
;       (gimp-edit-copy LLayer)
;       (gimp-image-remove-layer InImage LLayer)
;       (gimp-floating-sel-anchor (car (gimp-edit-paste (aref LABLayer 2) FALSE)))
;       (let* (
;         (CompImage (car (plug-in-drawable-compose TRUE LABImage (aref
LABLayer 2) (aref LABLayer 1) (aref LABLayer 0) 0 "LAB"))))
;         (CompLayer (cadr (gimp-image-get-layers CompImage)))
;         )
;         (gimp-selection-all CompImage)
;         (gimp-edit-copy (aref CompLayer 0))
;         (gimp-floating-sel-anchor (car (gimp-edit-paste InLayer FALSE)))
;         (gimp-image-delete CompImage)
;       )
;       (gimp-image-delete LABImage)
;     )
;   )
; )
; (gimp-image-remove-channel InImage MaskChannel)
;
;
; Finish work
;
; (gimp-image-undo-group-end InImage)
; (gimp-displays-flush)
;
; )
; )
;
; (script-fu-register
;   "script-fu-Eg-ISONoiseReduction"
;   "<Image>/Script-Fu/Eg/ISO Noise Reduction"
;   "Reduce sensor noise at high ISO values"
;   "Martin Egger (martin.eg...@gmx.net)"
;   "2004, Martin Egger, Bern, Switzerland"
;   "8.10.2004"
;   "RGB* GRAY*"
;   SF-IMAGE "The Image" 0
;   SF-DRAWABLE "The Layer" 0
;   SF-ADJUSTMENT "Edge search: Radius" '(2.5 1.0 10.0 0.5 0 2 0)
;   SF-OPTION "Noise Reduction Method"
;     (
;       "RBG channel blurring (faster)"
;       "Luminance channel blurring (better)"
;     )
;   SF-ADJUSTMENT "Channel Blur: Radius" '(2.0 1.0 10.0 1.0 0 2 0)
;   SF-ADJUSTMENT "Channel Blur: Delta" '(15.0 1.0 50.0 1.0 0 2 0)
; )
;
;
;

```

Script-Fu fire-logo (logo texte)

Voici un script créé par Vincent Vivien à l'âge de quinze ans qui permet de réaliser un texte avec des flammes.

Le script d'origine est disponible à partir de la page :

<http://perso.wanadoo.fr/bruno.vivien/vincent/gimp/index.htm>

Le code source modifié pour Gimp 2 est disponible sur la page suivante (texte rouge). Vous pouvez copier le code source disponible sur cette page (texte rouge) dans un éditeur de texte et le sauvegarder dans le répertoire des scripts sous fire-logo.scm (voir le début de cette documentation à la page 5 et le nota en bas de la page 5 pour l'enregistrement).

Ce Script-Fu est accessible à partir de la « fenêtre Principale » par : Exts, Script-Fu, Logos, Fire.

Pendant l'exécution de ce script vous devrez paramétrer et agir sur le filtre « Déformation Interactive » . Le fonctionnement de ce filtre est présenté dans la partie 1 de notre documentation dans la rubrique « Animation automatique en déformant une image » .

Exemple :



Contenu du Script-Fu Script-Fu fire-logo.scm

```
; Vincent VIVIEN (15 years old) Student France
; http://perso.wanadoo.fr/bruno.vivien/vincent/gimp/index.htm
; vincent.vivien@wanadoo.fr
; adaptation pour Gimp version 2 aljacom novembre 2004
;
; *****
; *The fire font" v1.1 *
; *script written for The Gimp*
; *****
; Code:
(define (script-fu-fire-logo texte taille police lumin bmp sky)
  *****
  ; adaptation novembre 2004 fonction gimp-image-new
  (let* ((img (car (gimp-image-new 256 256 0)))
        (fig (car (gimp-palette-set-foreground '(0 0 0))))
        (text-layer (car (gimp-text-fontname img -1 0 0 texte 30 TRUE taille PIXELS police)))
        (largeur (car (gimp-drawable-width text-layer)))
        (hauteur (+ (car (gimp-drawable-height text-layer)) 50))
        *****
        ; adaptation novembre 2004 fonction gimp-layer-new
        (logo-layer (car (gimp-layer-new img largeur hauteur 1 "Background" 100 0)))
        (gimp-invert text-layer)
        (gimp-layer-set-offsets text-layer 0 50)
        (gimp-palette-set-background '(0 0 0))
        (gimp-image-add-layer img logo-layer 1)
        (gimp-edit-fill logo-layer FG-IMAGE-FILL)
        (gimp-image-resize img largeur hauteur 0 0)
        (set! feu (car (gimp-image-merge-down img text-layer 0)))
        (plug-in-rotate 1 img feu 1 TRUE)
        (plug-in-wind 1 img feu 9 0 9 0 1)
        (plug-in-rotate 1 img feu 3 TRUE)
        (plug-in-gauss-rl2 1 img feu 1 1)
        (gimp-message "Please make a dissolved letters' effect !")
        (plug-in-iwarp 0 img feu)
        (if (= lumin TRUE)
            (begin (gimp-color-balance feu 0 1 92 -2 -61)
                   (gimp-color-balance feu 1 1 38 -6 -48)
                   (gimp-color-balance feu 2 1 7 -4 -24)))
            (if (= lumin FALSE)
                (begin (gimp-color-balance feu 0 0 92 -2 -61)
                       (gimp-color-balance feu 1 0 38 -6 -48)
                       (gimp-color-balance feu 2 0 7 -4 -24)))
                (if (= bmp TRUE)
                    (begin (plug-in-bump-map 1 img feu feu 53.68 21.70 3 0 0 0 1 0 2)
                           (plug-in-gauss-rl2 1 img feu 1 1)))
                    (if (= sky TRUE)
                        *****
                        ; adaptation novembre 2004 fonction gimp-layer-new
                        (begin (set! cloud-layer (car (gimp-layer-new img largeur hauteur 0 "Cloud" 50 0)))
                               (set! star-layer (car (gimp-layer-new img largeur hauteur 0 "Star" 100 0)))
                               (gimp-image-add-layer img cloud-layer 1)
                               (gimp-image-add-layer img star-layer 2)
                               (gimp-edit-fill cloud-layer FG-IMAGE-FILL)
                               (gimp-edit-fill star-layer FG-IMAGE-FILL)
                               (gimp-by-color-select feu '(0 0 0) 0 2 TRUE 0 0 0)
                               (gimp-edit-clear feu)
                               (gimp-selection-none img)
                               (plug-in-scatter-hsv 1 img star-layer 2 53 46 110)
                               (plug-in-solid-noise 1 img cloud-layer 0 0 1 1 4.0 4.0)
                               (gimp-color-balance cloud-layer 0 0 0 34 57)))
                            (gimp-display-new img)
                            (gimp-message "That's all folks. Thanks you! :-)))")
                        (script-fu-register "script-fu-fire-logo"
                                           "<Toolbox>/Xtns/Script-Fu/Logos/Fire..."
                                           "Creates a burning font's effect :-0 (Excuse me for my English!)"
                                           "Vincent Vivien <vincent.vivien@wanadoo.fr>"
                                           "Vincent Vivien"
                                           "novembre 2004"
                                           "RGBA"
                                           SF-STRING _ "Text" "Fireman"
                                           SF-ADJUSTMENT _ "Font Size (pixels)" '(100 2 1000 1 10 0 1)
                                           *****
                                           ; adaptation novembre 2004 fonction SF-FONT
                                           _ "Font" _"-*-911 Porscha-*-*-*24-*-*-*p-*-*-*"
                                           SF-FONT _ "Font" _ "Sans Bold"
                                           SF-TOGGLE _ "Tick for keep brightness" FALSE
                                           SF-TOGGLE _ "Make a nice embossing effect" FALSE
                                           SF-TOGGLE _ "Add a starry sky with nice subdued clouds" FALSE)
                        *****
                    )
                )
            )
        )
  )
; Fin du script
```

Script-Fu Shadowy (logo texte)

Ce script permet de créer des textes en relief. Son auteur est Terry McKay.

Le code source modifié pour Gimp 2 est disponible sur les 2 pages suivantes. Vous pouvez copier le code source disponible sur ces pages (texte rouge) dans un éditeur de texte et le sauvegarder dans le répertoire des scripts sous shadowy.scn (voir le début de cette documentation à la page 5 et le nota en bas de la page 5 pour l'enregistrement).

Ce Script-Fu est accessible à partir de la « fenêtre Principale » par : Exts, Script-Fu, Logos, Shadowy.

Exemple :



Contenu du Script-Fu shadowy.scm

```
; Debut du Script
;http://www.x5ca.net/~mckay/gimp.html Jun/30/2001
; Adaptation Gimp 2 Aljacom novembre 2004
(define (script-fu-shadowy inText inShadDist inShadFuz inBord inSize inFont inBGGradient inBGColr inBGTog inFGGradient inFGColr inFGTog)
(let* (
(dummy (gimp-palette-set-foreground '(0 0 0)))
(theImage (car (gimp-image-new 100 100 0)))
(lA (car (gimp-text-fontname theImage -1 0 0 inText inBord TRUE inSize 0 inFont)))
(lC (car (gimp-text-fontname theImage -1 0 0 inText inBord TRUE inSize 0 inFont)))
(dummy (gimp-palette-set-foreground '(255 255 255)))
(lB (car (gimp-text-fontname theImage -1 0 0 inText inBord TRUE inSize 0 inFont)))
(lE (car (gimp-text-fontname theImage -1 0 0 inText inBord TRUE inSize 0 inFont)))
(lF (car (gimp-text-fontname theImage -1 0 0 inText inBord TRUE inSize 0 inFont)))
(width (car (gimp-drawable-width lA)))
(height (car (gimp-drawable-height lA)))
(lD (car (gimp-layer-new theImage width height 0 "Temp" 100 0)))
(lG (car (gimp-layer-new theImage width height 0 "Temp" 100 0)))
(lH (car (gimp-text-fontname theImage -1 0 0 inText inBord TRUE inSize 0 inFont)))
(lOutline)
(lOutMask)
(lShad)
(lShMask)
)
(gimp-image-resize theImage width height 0 0)
(gimp-image-add-layer theImage lD 0)
(gimp-image-add-layer theImage lG 0)
(gimp-layer-set-visible lA 1)
(gimp-layer-set-visible lB 1)
(gimp-layer-set-visible lC 0)
(gimp-layer-set-visible lD 0)
(gimp-layer-set-visible lE 0)
(gimp-layer-set-visible lF 0)
(gimp-layer-set-visible lG 0)
(gimp-layer-set-visible lH 0)
(gimp-layer-set-offsets lA -1 -1)
(gimp-layer-set-offsets lB 1 1)
(gimp-image-lower-layer-to-bottom theImage lB)
(set! lOutline (car (gimp-image-merge-visible-layers theImage 1)))
;*****
;Modification novembre 2004
;(gimp-layer-set-visible lOutline 0)
(gimp-drawable-set-visible lOutline 0)
(set! lOutMask (car (gimp-layer-create-mask lOutline 0)))
(gimp-image-add-layer-mask theImage lOutline lOutMask)
(gimp-layer-set-edit-mask lOutline TRUE)
(gimp-edit-copy lC)
(gimp-floating-sel-anchor (car (gimp-edit-paste lOutMask TRUE)))
(gimp-image-remove-layer theImage lC)
(gimp-layer-set-visible lD 1)
(gimp-layer-set-visible lE 1)
(gimp-image-lower-layer-to-bottom theImage lD)
(gimp-layer-set-offsets lE inShadDist inShadFuz)
(gimp-palette-set-background '(0 0 0))
(gimp-edit-clear lD)
(set! lShad (car (gimp-image-merge-visible-layers theImage 1)))
(gimp-layer-set-visible lShad 0)
(plug-in-gauss-rl 1 theImage lShad inShadFuz TRUE TRUE)
(set! lShMask (car (gimp-layer-create-mask lShad 1)))
(gimp-image-add-layer-mask theImage lShad lShMask)
(gimp-layer-set-edit-mask lShad TRUE)
(gimp-edit-copy lF)
(gimp-floating-sel-anchor (car (gimp-edit-paste lShMask TRUE)))
(gimp-layer-set-mode lShad MULTIPLY)
(set! lFGMask (car (gimp-layer-create-mask lH 1)))
(gimp-image-add-layer-mask theImage lH lFGMask)
(gimp-layer-set-edit-mask lH TRUE)
(gimp-edit-copy lF)
(gimp-floating-sel-anchor (car (gimp-edit-paste lFGMask TRUE)))
(gimp-image-remove-layer theImage lF)
(gimp-palette-set-background inBGColr)
(gimp-edit-clear lG)
(gimp-palette-set-background inFGColr)
(gimp-edit-clear lH)
(gimp-edit-fill lH 1)
(if (= inBGTog TRUE)
(begin
;*****
; Mofdification novembre 2004
;(gimp-gradients-set-active inBGGradient)
(gimp-gradients-set-gradient inBGGradient)
;*****
; Mofdification novembre 2004
;(gimp-blend lG 3 0 0 100 0 0 FALSE 0 0 0 0 height)
(gimp-blend lG 3 0 0 100 0 0 FALSE FALSE 0 0 0 0 height)
)
)
(if (= inFGTog TRUE)
(begin
```

```

;
; *****
; Modification novembre 2004
;(gimp-gradients-set-active inFGGradient)
(gimp-gradients-set-gradient inFGGradient)
; *****
;Modification novembre 2004
;(gimp-blend IH 3 0 0 100 0 0 FALSE 0 0 0 0 height)
(gimp-blend IH 3 0 0 100 0 0 FALSE FALSE 0 0 0 0 height)
)
)
(gimp-image-lower-layer-to-bottom theImage IH)
(gimp-image-lower-layer-to-bottom theImage IG)
(gimp-layer-set-visible lOutline 1)
(gimp-layer-set-visible lShad 1)
(gimp-layer-set-visible IH 1)
(gimp-layer-set-visible IG 1)
(gimp-display-new theImage)
)
)
(script-fu-register "script-fu-shadowy"
"<Toolbox>/Xtns/Script-Fu/Logos/Shadowy"
"....."<Toolbox>/Xtns/Script-Fu/Logos-Suite/Shadowy"
"Simple hollowed out text Using Masks"
"Terry McKay"
"Terry McKay"
"novembre 2004"
""
SF-STRING "Text String" "The Gimp"
SF-VALUE "Shadow Spacing" "3"
SF-VALUE "Shadow Fuzziness" "5.0"
SF-VALUE "Text Border (pixels)" "40"
SF-VALUE "Font Size (pixels)" "60"
; *****
;Modification novembre 2004
;SF-FONT "Font Name" "-2rebels-adventure-semi bold-r-normal--0-0-0-0-p-0-0-ascii-0"
SF-FONT "Font Name" "Sans Bold"
; *****
;Modification novembre 2004
;SF-GRADIENT "Background Gradient" "Brushed_Aluminium"
SF-GRADIENT "Background Gradient" "Golden"
; *****
;Modification novembre 2004
;SF-COLOR "Background Colour" '(0 0 200)
SF-COLOR "Background Colour" '(0 0 0)
SF-TOGGLE "Background Gradient" TRUE
; *****
;Modification novembre 2004
;SF-GRADIENT "Foreground Gradient" "Brushed_Aluminium"
SF-GRADIENT "Foreground Gradient" "Golden"
; *****
;Modification novembre 2004
;SF-COLOR "Foreground Colour" '(0 0 200)
SF-COLOR "Foreground Colour" '(0 0 0)
SF-TOGGLE "Foreground Gradient" TRUE
)
; Fin du Script

```

Insertion d'un © ou tout autre texte dans une bordure

Voici un Script-Fu proposé par Pavel Antokolsky qui permet de créer une bordure autour d'une image et d'ajouter un © , un texte dans cette bordure. Ce script est une adaptation de « photo-border » fait par Zigmar.

Le script est disponible à <http://www.boblycat.org/~stig/copyright-border.scn>

Sauvegarder ce script dans le répertoire des scripts comme indiqué à la page 5.

Pour accéder au script à partir de la fenêtre de l'image Faire : Script-Fu, Decor, Double Border with Copyright

Exemple



Contenu du Script-Fu copyright-border.scm

```
;;
;;
;; Freely distributed
;;
;;
;; Version 1.0 by Pavel Antokolsky aka Zigmar (antokol@isdn.net.il)
;;
;;
;; Version 1.1 Fufie (fufie@boblycat.org)
;;
;;
;;   - Improved error-checking and undo works
;;   - Added support for copyright-notice
;;   - Moved script to S-Fu/Decor
;;
;;
;; Version 1.2 Fufie (fufie@boblycat.org)
;;
;;
;;   - Added blur-effect on copyright-notice
;;   as suggested by Eugene

(define (script-fu-border-with-copyright aimg adraw ib-width ib-colour ob-width ob-colour
                                         text font-name font-size font-colour blur-text
                                         work-on-copy flatten-image)

  (let* ((type (car (gimp-drawable-type-with-alpha adraw)))
         (img (if (= work-on-copy TRUE)
                   (car (gimp-image-duplicate aimg))
                   aimg))
         (old-width (car (gimp-image-width img)))
         (old-height (car (gimp-image-height img)))
         (old-fg (car (gimp-palette-get-foreground)))
         (border-width (+ ib-width ob-width))
         (new-width (+ old-width (* border-width 2)))
         (new-height (+ old-height (* border-width 2)))
         )

    ;; Start undo, make it simple.
    (gimp-image-undo-group-start img)

    ;;Add space for the border ("canvas resize")
    (gimp-image-resize img new-width new-height border-width border-width)

    ;; Create new layer for the border
    (set! border-layer (car (gimp-layer-new img
                                             new-width
                                             new-height
                                             type
                                             "Photo-border"
                                             100
                                             NORMAL
                                             )))

    ;; Clear newly created layer
    (gimp-edit-clear border-layer)
    ;; Add it to image
    (gimp-image-add-layer img border-layer -1)

    ;; Outer border fill, if it is bigger than zero
    (cond ((> ob-width 0)
           (gimp-rect-select img
                             0 0
                             new-width new-height
                             REPLACE
                             0 0)
           (gimp-rect-select img
                             ob-width ob-width
                             (- new-width (* ob-width 2)) (- new-height (* ob-width 2))
                             SUB
                             0 0)
           (gimp-palette-set-foreground ob-colour)
           (gimp-edit-fill border-layer 0)
           ))

    ;; Inner border fill if width is > 0
    (cond ((> ib-width 0)
           (gimp-rect-select img
                             ob-width ob-width
                             (- new-width (* ob-width 2)) (- new-height (* ob-width 2))
                             REPLACE
                             0 0)
           (gimp-rect-select img
                             border-width border-width
                             old-width old-height
                             SUB
                             0 0)
           (gimp-palette-set-foreground ib-colour)
           (gimp-edit-fill border-layer 0)
           ))

    ;; add the text if the text has size
    ;; doing it very iteratively and unschemy
```

```

(cond ((> (string-length text) 0)
      (let (((text-size '())
              (text-width '())
              (text-height '())
              (written-text '())
              )
            (gimp-palette-set-foreground font-colour)
            ;; get the extent of text when written out
            (set! text-size (gimp-text-get-extents-fontname text font-size PIXELS font-name))
            (set! text-width (car text-size))
            (set! text-height (cadr text-size))

            (set! written-text (gimp-text-fontname img
                                                    -1 ;; new layer
                                                    (- new-width text-width ob-width) ;; x-coord
                                                    (- new-height text-height) ;; y-coord
                                                    text
                                                    -1
                                                    TRUE ;; anti-alias
                                                    font-size
                                                    PIXELS ;; size is in pixels
                                                    font-name))

            ;; if we're to blur anything, this is the time
            (if (= blur-text TRUE)
                (plug-in-gauss-rle TRUE img (car written-text) 1.5 TRUE TRUE))

            )))

;; flatten
(if (= flatten-image TRUE) (gimp-image-flatten img))

;; Finalisation
(gimp-selection-none img)
(gimp-palette-set-foreground old-fg)

;; End UNDO
(gimp-image-undo-group-end img)

;; Display our work
(if (= work-on-copy TRUE) (gimp-display-new img))
(gimp-displays-flush)
))

;; Default values are for really big pictures, scale down if you work with
;; small pictures. Unfortunately these defaults can't figure out the size
;; of your image due to limitations in gimp.

(script-fu-register "script-fu-border-with-copyright"
  "<Image>/Script-Fu/Decor/Double Border with Copyright..."
  "Add a border around an image with copyright note"
  "Frank Ufie (fufie@boblycat.org) [Based on photo-border by: Zigmar (antokol@isdn.net.il)]"
  "Fufie (was: Zigmar)"
  "28/10/2004"
  ""
  SF-IMAGE "Input Image" 0
  SF-DRAWABLE "Input Drawable" 0
  SF-ADJUSTMENT "Inner border width" '(8 0 2000 1 10 0 1)
  SF-COLOR "Inner border colour" '(255 255 255)
  SF-ADJUSTMENT "Outer border width" '(50 0 2000 1 10 0 1)
  SF-COLOR "Outer border colour" '(0 0 0)
  SF-STRING "Text" "302/251 fufie, 2004"
  SF-FONT "Font" "Utopia"
  SF-ADJUSTMENT "Font Size (pixels)" '(40 2 1000 1 10 0 1)
  SF-COLOR "Text colour" '(255 255 255)
  SF-TOGGLE "Blur text" TRUE
  SF-TOGGLE "Work on copy" FALSE
  SF-TOGGLE "Flatten image" TRUE
  )

```

Boutons Aqua pour pages Web

Voici un script pour créer des boutons Aqua. Ce script a été conçu par Iccii et adapté pour fonctionner avec Gimp 2 par Graham Smith.

Ce script est disponible sur la page http://www.crazysquirrel.com/software/gimp/aqua_gimp_2_0.scm

Sauvegarder ce script dans le répertoire des scripts comme indiqué à la page 5.

Le très volumineux code source est disponible sur les 5 pages suivantes.

Ce script dispose de trois fonctions. Pour accéder à toutes ces fonctions, sur la « fenêtre Principale » faire : Exts, Script-Fu, Thèmes de pages Web, AquaPill et choisir une des 3 options Button, Hrule, Round.

Button



Hrule



Round



Nota : Voir le script suivant (page 88) pour créer des textes, des logos (style iMac).

Contenu de aqua_gimp_2_0.scm

```
; The GIMP -- an image manipulation program
; Copyright (C) 1995 Spencer Kimball and Peter Mattis
;
; Aqua Pill --- create an aqua pill web button
; Copyright (C) 2001 Iccii <iccii@hotmail.com>
;
; Additions and Gimp 2.0 Upgrade
; Copyright (C) 2003 Graham Smith <graham-gimp@crazysquirrel.com>
;
;-----
; - Changelog -
; version 0.1 2001/03/06 Iccii <iccii@hotmail.com>
;   - Initial release
; version 0.2 2001/03/08 Iccii <iccii@hotmail.com>
;   - Move the logo generator function from base code to extra code
; version 0.3 2001/03/13 Iccii <iccii@hotmail.com>
;   - Create an Aqua Pill hrule
; version 0.4 2001/03/21 Iccii <iccii@hotmail.com>
;   - Create a Round Aqua Pill
; version 0.5 2001/03/25 Iccii <iccii@hotmail.com>
;   - Add the "remove-dust" function to remove the dust images
; version 0.5a 2001/04/02 Iccii <iccii@hotmail.com>
;   - Add antialias option
;   - Font size calculation make better
; version 0.5b 2001/05/25 Iccii <iccii@hotmail.com>
;   - A bit optimize
; version 0.5c 2001/06/21 Iccii <iccii@hotmail.com>
;   - Minor fix
; version 0.5d 2001/08/10 Iccii <iccii@hotmail.com>
;   - Delete the "remove-dust" function
; version 0.6 2001/09/17 lasm <lasma@rocketmail.com>
;   - Fixed the "Flatten" bug
; version 0.7 2003/08/09 graham <graham@crazysquirrel.com>
;   - Added commenting
;   - Allowed setting of widget back ground color.
;   - Changed bg-color to widget-color and added background-color
;   - Removed unused parameters from script-fu-aqua-pill function
; version 0.8 2004/04/08 graham <graham@crazysquirrel.com>
;   - Upgraded script to work with Gimp 2.0 buttons look slightly
;     different but are pretty close
;   - Added ability to set how many pixels of padding are draw
;     around each of the 'buttons' rather than an abitary amount
;   - Added a bunch of comments
;
;-----
;
; This program is free software; you can redistribute it and/or modify
; it under the terms of the GNU General Public License as published by
; the Free Software Foundation; either version 2 of the License, or
; (at your option) any later version.
;
; This program is distributed in the hope that it will be useful,
; but WITHOUT ANY WARRANTY; without even the implied warranty of
; MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
; GNU General Public License for more details.
;
; You should have received a copy of the GNU General Public License
; along with this program; if not, write to the Free Software
; Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
;
;#####
; START OF PILL DEFINITION
;#####

(define
  (script-fu-aqua-pill
    img
    bg-color
    background-color
    width
    height
    ratio
    blur
    shadow
    flatten
    antialias
    button-padding
  )

  (define (remove-selection-corner img amount)
    (gimp-selection-feather img amount)
    (gimp-selection-sharpen img))

  3

  (define (round-select img x y width height ratio)
    (let* ((diameter (* ratio height)))
```

```

(gimp-ellipse-select img x y diameter height ADD FALSE 0 0)
(gimp-ellipse-select img (+ x (- width diameter)) y
                        diameter height ADD FALSE 0 0)
(gimp-rect-select img (+ x (/ diameter 2)) y
                    (- width diameter) height ADD FALSE 0)))

(let* (
    (old-fg-color (car (gimp-palette-get-foreground)))
    (old-bg-color (car (gimp-palette-get-background)))
    (shadow-height (if (eqv? shadow TRUE) 1 0))
    (bluramount (if (> width height) (* blur height) (* blur width)))
    (shiftx button-padding)
    (shifty button-padding)
    (width-all (+ width (* 1 shiftx) (* button-padding 2)))
    (height-all (+ height (* 1 shifty) (* height shadow-height 0.5) (* button-padding 2)))
    (whratio (sqrt (/ height width)))

    (highlight-upper1-layer (car (gimp-layer-new img width-all height-all 1 "Highlight Upper1" 75 SCREEN)))
    (highlight-upper2-layer (car (gimp-layer-new img width-all height-all 1 "Highlight Upper2" 40 SCREEN)))
    (highlight-lower (car (gimp-layer-new img width-all height-all 1 "Highlight Lower" 100 OVERLAY)))
    (base1-layer (car (gimp-layer-new img width-all height-all 1 "Base Layer1" 75 SCREEN)))
    (base2-layer (car (gimp-layer-new img width-all height-all 1 "Base Layer2" 100 NORMAL)))
  ) ; end variable definition

  (gimp-image-undo-disable img)
  (gimp-selection-none img)
  (gimp-palette-set-foreground bg-color)
  (gimp-palette-set-background background-color)
  (gimp-drawable-fill base2-layer TRANS-IMAGE-FILL)
  (gimp-drawable-fill base1-layer TRANS-IMAGE-FILL)
  (gimp-drawable-fill highlight-lower TRANS-IMAGE-FILL)
  (gimp-drawable-fill highlight-upper2-layer TRANS-IMAGE-FILL)
  (gimp-drawable-fill highlight-upper1-layer TRANS-IMAGE-FILL)

  ;----- START - DROP SHADOW EFFECT -----
  (if (eqv? shadow TRUE)
    (begin
      (set! shadow1-layer (car (gimp-layer-new img width-all height-all 1 "Shadow Layer1" 100 SCREEN)))
      (set! shadow2-layer (car (gimp-layer-new img width-all height-all 1 "Shadow Layer2" 100 NORMAL)))
      (gimp-image-add-layer img shadow2-layer -1)
      (gimp-image-add-layer img shadow1-layer -1)
      (gimp-drawable-fill shadow2-layer TRANS-IMAGE-FILL)
      (gimp-drawable-fill shadow1-layer TRANS-IMAGE-FILL)

      (round-select img shiftx shifty width height ratio)
      (gimp-selection-shrink img (* height 0.1))
      (gimp-edit-fill shadow1-layer FG-IMAGE-FILL)
      (gimp-hue-saturation shadow1-layer 0 0 75 0)
      (gimp-selection-grow img (+ 1 (* bluramount 0.3)))
      (plug-in-gauss-iir2 1 img shadow1-layer (+ 1 (* bluramount 0.3)) (+ 1 (* bluramount 0.3)))
      (gimp-drawable-offset shadow1-layer 0 1 0 (* height 0.4))
      (gimp-selection-none img)

      (round-select img shiftx shifty width height ratio)
      (gimp-edit-fill shadow2-layer FG-IMAGE-FILL)
      (gimp-hue-saturation shadow2-layer 0 0 0 -25)
      (gimp-selection-grow img (+ 1 (* bluramount 0.3)))
      (plug-in-gauss-iir2 1 img shadow2-layer (+ 1 (* bluramount 0.3)) (+ 1 (* bluramount 0.3)))
      (gimp-drawable-offset shadow2-layer 0 1 0 (* height 0.4))
      (gimp-selection-none img)

    ))

  ;----- END - DROP SHADOW EFFECT -----

  (round-select img shifty shifty width height ratio)
  (gimp-edit-fill base2-layer FG-IMAGE-FILL)
  (gimp-hue-saturation base2-layer 0 0 -75 -50)
  (gimp-selection-shrink img (* height 0.01))
  (gimp-edit-fill base1-layer FG-IMAGE-FILL)
  (gimp-hue-saturation base1-layer 0 0 -50 -50)
  (gimp-selection-grow img (* height 0.01)) ; ,Ú,©,µ,İ?,'~%«»?
  (plug-in-gauss-iir2 1 img base1-layer (+ 1 (* bluramount 0.05)) (+ 1 (* bluramount 0.05)))
  (gimp-selection-none img)
  (gimp-image-add-layer img base2-layer -1)
  (gimp-image-add-layer img base1-layer -1)
  (if (eqv? antialias TRUE)
    (let ((layer-width (car (gimp-drawable-width base2-layer)))
          (layer-height (car (gimp-drawable-height base2-layer))))
      (gimp-layer-scale base2-layer (* 2 layer-width) (* 2 layer-height) 0)
      (gimp-layer-scale base2-layer layer-width layer-height 0)))

```

```

(round-select img shifty shifty width height ratio)
  (gimp-rect-select img shifty shifty width (* height 0.5) SUB 0 0)
  (gimp-selection-shrink img (* height 0.05))
  (remove-selection-corner img (* 0.4 height))
  (gimp-edit-fill highlight-lower BG-IMAGE-FILL)
  (gimp-selection-grow img (+ 1 (* bluramount 0.4)))
  (plug-in-gauss-iir2 1 img highlight-lower (+ 1 (* bluramount 0.4)) (+ 1 (* bluramount 0.4)))
  (set! highlight-lower-copy (car (gimp-layer-copy highlight-lower FALSE)))
  (gimp-selection-none img)
  (gimp-image-add-layer img highlight-lower-copy -1)
  (gimp-image-add-layer img highlight-lower -1)

(round-select img shiftx shifty width height ratio)
  (gimp-rect-select img shiftx (+ shifty (* height 0.35)) width height SUB 0 0)
  (gimp-selection-shrink img (* height 0.02 whratio))
  (remove-selection-corner img (+ (* height 0.4)))
  (gimp-edit-fill highlight-upper2-layer BG-IMAGE-FILL)
  (gimp-selection-grow img (+ 1 (* bluramount 0.08)))
  (plug-in-gauss-iir2 1 img highlight-upper2-layer (+ 1 (* bluramount 0.08)) (+ 1 (* bluramount 0.08)))
  (gimp-selection-none img)
  (gimp-image-add-layer img highlight-upper2-layer -1)

(round-select img shiftx shifty width height ratio)
  (gimp-rect-select img shiftx (+ shifty (* height 0.28)) width height SUB 0 0)
  (gimp-selection-shrink img (* height 0.09))
  (remove-selection-corner img (* 0.15 height))
  (gimp-edit-fill highlight-upper1-layer BG-IMAGE-FILL)
  (gimp-selection-grow img (+ 1 (* bluramount 0.05))) ; ,Ú,©,µ,İ?,'~%oo?
  (plug-in-gauss-iir2 1 img highlight-upper1-layer (+ 1 (* bluramount 0.05)) (+ 1 (* bluramount 0.05)))
  (gimp-selection-none img)
  (gimp-image-add-layer img highlight-upper1-layer -1)

  (gimp-image-undo-enable img)
  (gimp-undo-push-group-start img)
  (if (eqv? flatten TRUE)
    (gimp-image-flatten img))
  (gimp-undo-push-group-end img)
  (gimp-palette-set-foreground old-fg-color)
  (gimp-palette-set-background old-bg-color)
  (gimp-display-new img)
  (gimp-displays-flush)

))

;#####
; END OF PILL DEFINITION
;#####

;#####
; START OF ROUND DEFINITION AND REGISTRATION
;#####

(define
  (script-fu-aqua-pill-round
    baseradius
    button-padding
    widget-color
    background-color
    ratio
    blur
    shadow
    flatten
    antialias
  )
  ; (let* ((flatten TRUE)
  (let* (
    (shadow-height (if (eqv? shadow TRUE) 1 0))
    (height baseradius)
    (radius (/ (* ratio height) 4))
    (width (+ 1 (* ratio height)))
    (shiftx button-padding)
    (shifty button-padding)
    (height-all (+ height shiftx (* height shadow-height 0.5) (* button-padding 2)))
    (width-all (+ width shifty (* button-padding 2)))
    (img (car (gimp-image-new width-all height-all RGB))))
    (script-fu-aqua-pill img widget-color background-color width height ratio blur shadow flatten antialias button-padding)
  ))

(script-fu-register "script-fu-aqua-pill-round"
  "<Toolbox>/Xtns/Script-Fu/Web Page Themes/Aqua Pill/Round..."
  "Create a round aqua pill image"
  "Iccii <icci@hotmail.com>"
  "Iccii"
  "Jun 2001"
  ""
  SF-ADJUSTMENT _"Base Radius (pixel)" '(100 2 500 1 1 0 1)

```

```

SF-ADJUSTMENT _"Padding"
SF-COLOR _"Widget Color" '(121 161 248) '(1 0 100 1 1 0 1)
SF-COLOR _"Background Color" '(229 238 255)
SF-ADJUSTMENT _"Round Ratio" '(1 0.05 9 0.05 0.5 2 0)
SF-ADJUSTMENT _"Blur Amount" '(1 0.05 5 0.05 0.5 2 0)
SF-TOGGLE _"Drop Shadow" TRUE
SF-TOGGLE _"Flatten" TRUE
SF-TOGGLE _"Antialias" TRUE

;#####
; END OF ROUND DEFINITION AND REGISTRATION
;#####

;#####
; START OF HRULE DEFINITION AND REGISTRATION
;#####

(define
  (script-fu-aqua-pill-hrule
    width
    height
    button-padding
    widget-color
    background-color
    ratio
    blur
    shadow
    flatten
    antialias
  )
  (let* ((flatten TRUE)
    (let* (
      (shadow-height (if (eqv? shadow TRUE) 1 0))
      (radius (/ (* ratio height) 4))
      (shiftx button-padding)
      (shifty button-padding)
      (height-all (+ height shiftx (* height shadow-height 0.5) (* button-padding 2)))
      (width-all (+ width shifty (* button-padding 2)))
      (img (car (gimp-image-new width-all height-all RGB))))
      (if (< (* height ratio) width)
        (script-fu-aqua-pill img widget-color background-color width height ratio blur shadow flatten antialias button-padding)
        (gimp-message "Warning: Bar Length is too short to create your image!"))
    ))

(script-fu-register "script-fu-aqua-pill-hrule"
  "<Toolbox>/Xtns/Script-Fu/Web Page Themes/Aqua Pill/Hrule..."
  "Create an Hrule with the aqua image"
  "Iccii <icci@hotmail.com>"
  "Iccii"
  "Jun 2001"
  ""
  SF-ADJUSTMENT _"Bar Length" '(500 2 1500 1 1 0 1)
  SF-ADJUSTMENT _"Bar Height" '(25 2 500 1 1 0 1)
  SF-ADJUSTMENT _"Bar Padding" '(1 0 100 1 1 0 1)
  SF-COLOR _"Widget Color" '(121 161 248)
  SF-COLOR _"Background Color" '(229 238 255)
  SF-ADJUSTMENT _"Round Ratio" '(1 0.05 5 0.05 0.5 2 0)
  SF-ADJUSTMENT _"Blur Amount" '(1 0.05 5 0.05 0.5 2 0)
  SF-TOGGLE _"Drop Shadow" TRUE
  SF-TOGGLE _"Flatten" TRUE
  SF-TOGGLE _"Antialias" TRUE)

;#####
; END OF HRULE DEFINITION AND REGISTRATION
;#####

;#####
; START OF BUTTON DEFINITION AND REGISTRATION
;#####

(define
  (script-fu-aqua-pill-button
    text
    size
    font
    text-color
    widget-color
    background-color
    xpadding
    ypadding
    button-padding
    ratio
    blur
    shadow
    flatten
    antialias
  )
  (let*
    (
      (flatten FALSE) ;switches flattening off because it blows up if it is switched on :oD
      (shadow-height (if (eqv? shadow TRUE) 1 0))
      (old-fg-color (car (gimp-palette-get-foreground))) ;save the old foreground color
      (old-bg-color (car (gimp-palette-get-background))) ;save the old background color
      (img (car (gimp-image-new 256 256 RGB))) ;create and aquire a new image
    )
  )

```

```

        (tmp (gimp-palette-set-foreground text-color)) ;only change fg-color for text color
        (text-layer (car (gimp-text-fontname img -1 0 0 text 0 TRUE size PIXELS font)))
        (text-width (car (gimp-drawable-width text-layer))) ;width of the text
        (text-height (car (gimp-drawable-height text-layer))) ;height of the text
        (radius (/ (* ratio text-height 4)) ; radius of the end caps

;this calculates how much space the text alone needs
        (height (+ (* 2 ypadding) text-height))
        (width (+ (* 2 (+ radius xpadding)) text-width))

;shifts the text to the center of the image
;the button-padding tell use how much space should be around the button
        (shiftx button-padding)
        (shifty button-padding)

;this is where the overall size of the drawing is calculated
;note that this calculation all occus elsewhere
;you need to use twice the padding in order to make sure there
;are the same number of pixels onthe bottom and right as
;are on the top and left
        (height-all (+ height shiftx (* height shadow-height 0.5) (* button-padding 2)))
        (width-all (+ width shifty (* button-padding 2)))
    )
    (gimp-palette-set-foreground text-color)
    (gimp-palette-set-background background-color)

;the image is resized to width-all x height-all so that the button will fit on it
;and the text is shifted to the center
    (gimp-image-resize img width-all height-all (+ shiftx xpadding radius) (+ shifty ypadding))
    (gimp-layer-set-offsets text-layer (+ shiftx xpadding radius) (+ shifty ypadding))

;(script-fu-aqua-pill img widget-color background-color width height ratio blur shadow flatten antialias)
(script-fu-aqua-pill img widget-color background-color width height ratio blur shadow flatten antialias button-padding)

(gimp-undo-push-group-start img)
(gimp-image-raise-layer-to-top img text-layer)
(gimp-image-lower-layer img text-layer)
(gimp-image-lower-layer img text-layer)
(set! text-layer-copy (car (gimp-layer-copy text-layer FALSE)))
(set! text-layer-shadow (car (gimp-layer-copy text-layer FALSE)))
(gimp-image-add-layer img text-layer-copy 3)
(gimp-image-add-layer img text-layer-shadow 4)
(gimp-layer-set-mode text-layer-shadow MULTIPLY-MODE)
(gimp-drawable-offset text-layer-copy 0 1 (- (* xpadding 0.1)) (- (* ypadding 0.1)))
(gimp-drawable-offset text-layer-shadow 0 1 (- (* xpadding 0.1)) (* height 0.1))
(gimp-invert text-layer-copy)
(gimp-selection-layer-alpha text-layer-shadow)
(gimp-palette-set-foreground widget-color)
(gimp-edit-fill text-layer-shadow FG-IMAGE-FILL)
(gimp-selection-grow img (+ 1 (* height blur 0.1)))
(plug-in-gauss-irr2 1 img text-layer-shadow (+ 1 (* height blur 0.1)) (+ 1 (* height blur 0.1)))
(gimp-selection-none img)
(gimp-undo-push-group-end img)
(gimp-undo-push-group-start img)
(set! flatten TRUE)
(if (eqv? flatten TRUE)
    (gimp-layer-set-name (car (gimp-image-flatten img)) text))

(gimp-palette-set-foreground old-fg-color)
(gimp-palette-set-background old-bg-color)
(gimp-undo-push-group-end img)
(gimp-displays-flush)

)

)

(script-fu-register
    "script-fu-aqua-pill-button"
    "<Toolbox>/Xtns/Script-Fu/Web Page Themes/Aqua Pill/Button..."
    "Create an Aqua Pill Shaped Button"
    "icci@icci@hotmail.com"
    "icci"
    "Aug 2003"
    ""
    SF-STRING      _ "Text"
    SF-ADJUSTMENT  _ "Font Size (pixels)"
    SF-FONT        _ "Font"
    SF-COLOR       _ "Text Color"
    SF-COLOR       _ "Widget Color"
    SF-COLOR       _ "Background Color"
    SF-ADJUSTMENT  _ "Text Padding X"
    SF-ADJUSTMENT  _ "Text Padding Y"
    SF-ADJUSTMENT  _ "Button Padding"
    SF-ADJUSTMENT  _ "Round Ratio"
    SF-ADJUSTMENT  _ "Blur Amount"
    SF-TOGGLE      _ "Drop Shadow"
    SF-TOGGLE      _ "Flatten"
    SF-TOGGLE      _ "Antialias"

    "The GIMP"
    '(16 2 500 1 1 0 1)
    "Arial Black"
    '(0 0 0)
    '(121 161 248)
    '(229 238 255)
    '(1 0 100 1 10 0 1)
    '(1 0 100 1 10 0 1)
    '(1 0 100 1 10 0 1)
    '(1 0.05 5 0.05 0.5 2 0)
    '(1 0.05 5 0.05 0.5 2 0)
    TRUE
    TRUE
    TRUE

)

;#####
; END OF BUTTON DEFINITION AND REGISTRATION
;#####

```

Logo avec texte (style iMac)

Un peu dans la suite du script précédent (page 82) voici une création de Ivan Zenkov basée sur un didacticiel de Craig Marshall.

Ce Script-Fu est disponible à la page <http://www.mintis.lt/~gimp/skriptai/skriptai/logotipas-shrek.scm>

Sauvegarder ce script dans le répertoire des scripts comme indiqué à la page 5.

Pour accéder à ce script, sur la « fenêtre Principale » faire : Exts, Script-Fu, Papildomi, Logotipai, Šrekas

Exemple :



Contenu du script logotipas-shrek.scm

```
; Shrek text
; Copyright (c) 2004 Ivan Zenkov <ivan.zenkov@gmail.com>
;
;
; Based on iText tutorial by Craig Marshall
;
; This program is free software; you can redistribute it and/or modify
; it under the terms of the GNU General Public License as published by
; the Free Software Foundation; either version 2 of the License, or
; (at your option) any later version.
;
; This program is distributed in the hope that it will be useful,
; but WITHOUT ANY WARRANTY; without even the implied warranty of
; MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
; GNU General Public License for more details.
;
; You should have received a copy of the GNU General Public License
; along with this program; if not, write to the Free Software
; Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

(define (script-fu-shrek-text string font-name font-size text-color)

  (let* ((text-ext (gimp-text-get-extents font-name string font-size 0 font-name))
         (wid (+ (car text-ext) 20))
         (hig (+ (cadr text-ext) 20))
         (img (car (gimp-image-new wid hig 0))))
    (bg-layer (car (gimp-layer-new img wid hig 0 "Background" 100 0)))
    (shadow-layer (car (gimp-layer-new img wid hig 1 "Shadow" 100 0)))
    (text-layer (car (gimp-layer-new img wid hig 1 "Text" 100 0)))
    (glow-layer (car (gimp-layer-new img wid hig 1 "Glow layer (first)" 100 0)))
    (plastic-layer (car (gimp-layer-new img wid hig 1 "Plastic effect" 100 0)))
    (text-channel (car (gimp-channel-new img wid hig "Text channel" 50 '(0 0 0))))
    (shrink-num (/ (caddr text-ext) (* (/ (/ (caddr text-ext) 3.3) (caddr text-ext)) 100)))
    (blur-num (/ (caddr text-ext) 10))
    (old-fg (car (gimp-palette-get-foreground)))
    (old-bg (car (gimp-palette-get-background))))

  (gimp-image-undo-disable img)

  (gimp-image-add-layer img bg-layer 1)
  (gimp-image-add-layer img shadow-layer -1)
  (gimp-image-add-layer img text-layer -1)
  (gimp-image-add-layer img glow-layer -1)
  (gimp-image-add-layer img plastic-layer -1)
  (gimp-image-add-channel img text-channel 0)

  (gimp-palette-set-background '(255 255 255))
  (gimp-edit-clear bg-layer)
  (gimp-edit-clear shadow-layer)
  (gimp-edit-clear text-layer)
  (gimp-edit-clear glow-layer)
  (gimp-edit-clear plastic-layer)
```

```

(gimp-palette-set-background '(0 0 0))
(gimp-edit-clear text-channel)

(gimp-palette-set-foreground text-color)
(gimp-floating-sel-anchor (car (gimp-text-fontname img text-layer 10 10 string 0 TRUE font-size PIXELS font-name)))

; glow effect
(gimp-selection-layer-alpha text-layer)
(gimp-selection-shrink img shrink-num)
(gimp-edit-fill glow-layer 2)
(gimp-layer-set-mode glow-layer 5)
(gimp-selection-none img)
(plug-in-gauss-iir2 1 img glow-layer blur-num blur-num)
;; ugly code
(set! glow-layer-second (car (gimp-layer-copy glow-layer 0)))
(gimp-layer-set-name glow-layer-second "Glow layer (second)")
(gimp-image-add-layer img glow-layer-second -1)
(gimp-layer-set-mode glow-layer-second 5)
(set! text-layer (car (gimp-image-merge-down img glow-layer-second 2)))
(set! text-layer (car (gimp-image-merge-down img glow-layer 2)))

; blur channel
(gimp-edit-copy text-layer)
(gimp-floating-sel-anchor (car (gimp-edit-paste text-channel 0)))
(plug-in-gauss-iir2 1 img text-channel (/ blur-num 2) (/ blur-num 2))
(gimp-drawable-set-visible text-channel 0)

; plastic effect
(gimp-edit-fill plastic-layer 1)
(gimp-layer-set-mode plastic-layer 4)
(plug-in-lighting 1 img plastic-layer text-channel 0 TRUE FALSE 0 0 '(255 255 255) 1 0 1 -1 -1 1 0.30 1 0.40 0.60 27 TRUE FALSE FALSE)
(set! text-layer (car (gimp-image-merge-down img plastic-layer 2)))

; shadow
(gimp-selection-layer-alpha text-layer)
(gimp-edit-fill shadow-layer 1)
(gimp-selection-none img)
(plug-in-gauss-iir2 1 img shadow-layer 1 1)

; I can't use script-fu-drop-shadow :-
; ERROR: wta(1st) to min (see errobj)
; Please fix it.
; (script-fu-drop-shadow 0 img bg-layer 0 0 1 '(0 0 0) 100 1)

; I can't delete channel
; ERROR: Procedural database execution failed:
; (gimp_drawable_delete 66)
; Please fix it.
; (gimp-drawable-delete text-channel)

(gimp-palette-set-foreground old-fg)
(gimp-palette-set-background old-bg)

(gimp-image-undo-enable img)
(gimp-display-new img)))

(script-fu-register "script-fu-shrek-text"
  _ "<Toolbox>/Xtns/Script-Fu/Papildomi/Logotipai/$rekas..."
  "Suteikia tekstui iMac efekta"
  "Ivan Zenkov"
  "Ivan Zenkov"
  "September 17, 2004"
  "Vertė: Giedrius Naudžiūnas <gimp@akl.lt> www.gimp.projektas.lt"
  SF-STRING _ "Text" "$rekas"
  SF-FONT _ "Font" "Serif Bold"
  SF-ADJUSTMENT _ "Font Size (pixels)" '(100 2 1000 1 10 0 1)
  SF-COLOR _ "Text Color" '(84 155 8))

```

Digital camera noise filter & Unsharp 2

Peter Heckert a écrit une nouvelle version de « Digital camera noise filter » disponible en C à <http://home.arcor.de/peter.heckert/dcamnoise2.c> . Les instructions d'installation sous Linux sont dans le code source. D'autre part, Unsharp 2 est disponible à <http://home.arcor.de/peter.heckert/unsharp2.c> (C'est unsharp-plugin pour gimp-2.2 de Winston Chang adapté par Peter Heckert).

Michael Schumacher a compilé ces codes pour un fonctionnement sous Windows disponible à :

<http://schumaml.gmxhome.de/downloads/gimp/dcamnoise2-0.10hack.zip> et

<http://schumaml.gmxhome.de/downloads/gimp/unsharp2-0.10hack.zip>

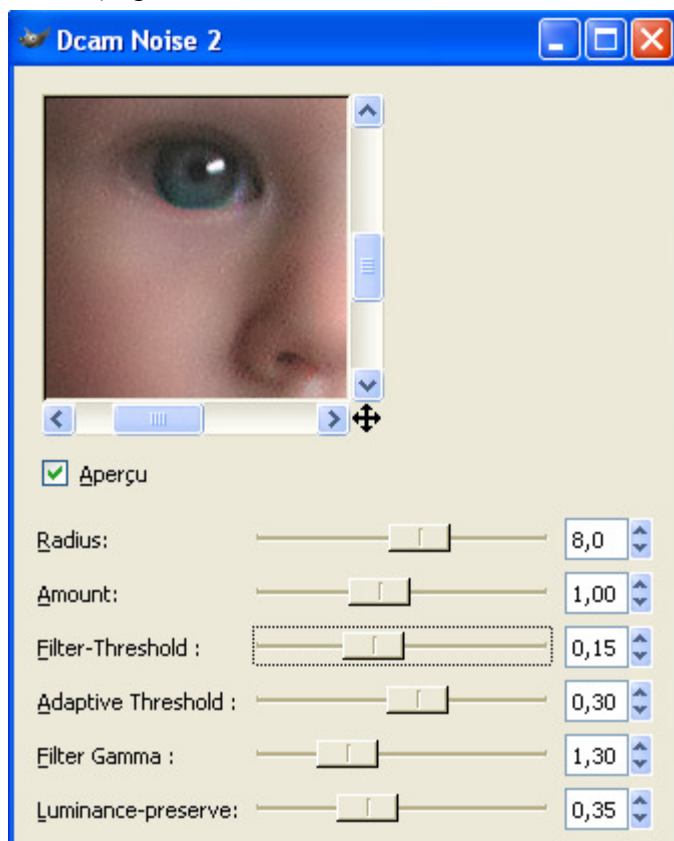
Installation sous Windows : Décompresser les 2 fichiers ZIP et placer dcamnoise2.exe et unsharp2.exe dans C:\Program Files\GIMP-2.2\lib\gimp\2.0 puis démarrer Gimp.

Vous pouvez utiliser l'image <http://www.gimpguru.org/Tutorials/ReducingCCDNoise/before-384x512.jpg> du tutoriel de Eric R. Jeschke (adresse du didacticiel : <http://www.gimpguru.org/Tutorials/ReducingCCDNoise/>) pour tester le fonctionnement.

Sur l'image : Filtres, Amélioration, Dcam noise 2. Paramétrer.

Exemple sur l'image before-384x512.jpg de Eric R. Jeschke avec ces paramètres ==>

Vous pouvez comparer ce résultat avec les résultats d'autres filtres sur notre documentation : [reduction_du_bruit.zip](#)



Greffon SIOX pour faciliter les détournages

SIOX (Simple Interactive Object Extraction) est un projet universitaire du « Department of Computer Science » de la « Freie Universität Berlin » présenté à l'adresse <http://www.siox.org/>.

SIOX sera un outil dans la future version 2.4.x de Gimp. La version actuelle du greffon pour Gimp 2.2.x réalise 50% des possibilités du futur outil.

Le greffon est disponible pour les versions Gimp 2.2.x pour Linux et Windows (<http://www.siox.org/downloads.html>).

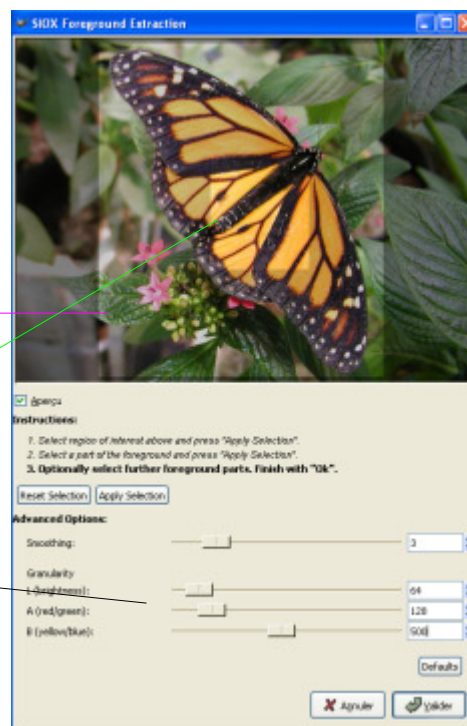
Pour Windows télécharger *siox4gimp.exe* (version disponible le 21 août 2005) directement dans le répertoire **C:\Program Files\GIMP-2.0\lib\gimp\2.0\plug-ins**, lien direct : <http://www.siox.org/downloads/siox4gimp.exe>

Le détournage est une opération délicate et un automatisme comme SIOX a beaucoup de difficultés à réaliser un travail correct. Pour que ce greffon fonctionne il faut respecter ces conditions :

- Affichage – Résolution de l'écran d'au moins 1280*1024 pixels.
- Image RVB (convertir si nécessaire par Image, Mode, RVB).
- Beaucoup de couleurs.
- Beaucoup de contrastes.
- 2 bonnes sélections des plans.

Principe du détournage dans SIOX :

- Sur l'image du papillon (<http://pdphoto.org/PictureDetail.php?mat=pdef&pg=6533>) faire Filtres, Misc, Foreground extraction.
- La fenêtre du filtre s'affiche.
- Faire une première sélection dans la fenêtre du filtre englobant la totalité du papillon.
- Appuyer sur le bouton « Apply selection ».
- Faire une seconde sélection à l'intérieur du papillon.
- Appuyer sur le bouton « Apply selection ».
- Paramétrer.
- Valider.

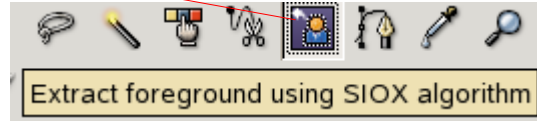


Résultat



En conclusion, pour avoir une idée plus précise des possibilités de SIOX, nous attendrons la future version de Gimp 2.4.x et les améliorations de SIOX qui est toujours en phase de développement.

Sous Linux vous pouvez tester les versions de développement de Gimp (2.3.x) où SIOX est présent comme outil (<http://www.siox.org/preview.html>).



Extrait d'un article (<http://www.gimp-fr.org/news.php>) sur SIOX de Raymond Ostertag sur la version Gimp 2.3.3

...

La grande nouveauté de cette nouvelle version est un l'algorithme  SIOX de sélection d'un objet dans une image. On commence par entourer l'objet à isoler d'une sélection à main levée puis avec une brosse on indique grossièrement où se trouve l'objet dans l'image (on peut aussi indiquer où se trouve le fond dans l'image). À tout moment on peut visualiser ce qui est sélectionné de ce qui ne l'est pas grâce à un masque coloré appliqué sur l'image. Une fois la sélection de l'objet effectuée on valide en appuyant la touche Entrée.

L'idée est évidemment intéressante mais les tests que j'ai effectué ne m'ont pas franchement convaincus de l'efficacité de l'algorithme. Une simple question de temps probablement.

posté par Raymond le 22/08/2005

Traitement par lot, DBP (David's Batch Processor)

DBP (David's Batch Processor) permet d'automatiser des tâches sur un lot d'images :

<http://members.ozemail.com.au/~hodsondbp/dbp.html>

Vous pouvez télécharger le binaire pour Gimp version Windows compilé par Michael Schumacher :

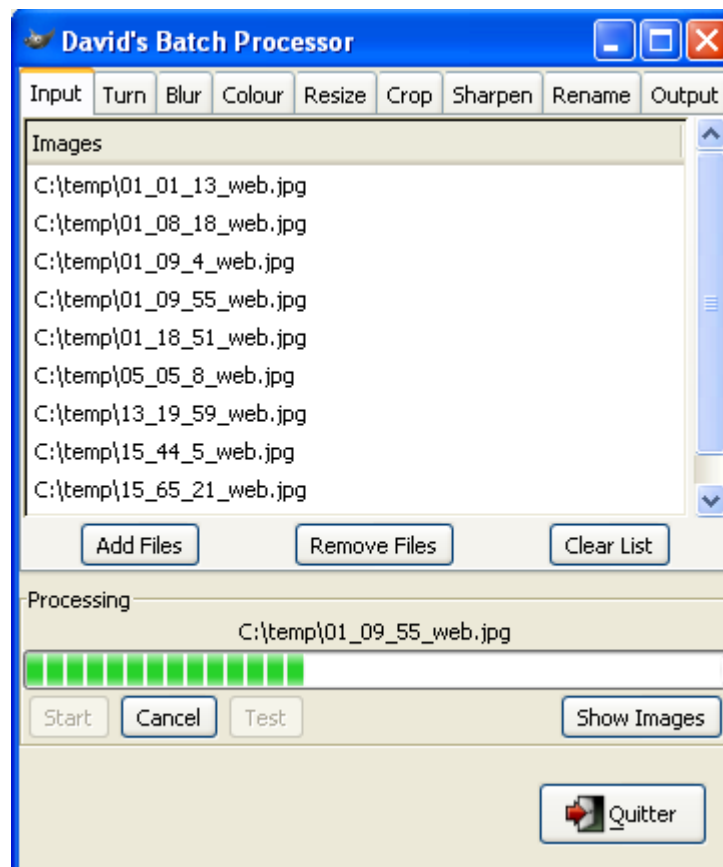
<http://www.ozemail.com.au/~hodsondbp-1.1.zip>

Décompresser et installer *dbp.exe* dans le répertoire C:\Program Files\GIMP-2.0\lib\gimp\2.0\plug-ins\

À partir de la fenêtre principale de Gimp faire Exts, Batch Process.

Onglets :

Sélection des images, Retourner, Flou, Contrastes & Niveaux de gris, Échelle, Découpage, Netteté, Renommer, Type de fichier de sortie.



Choisir les modifications à apportées sur le lot des images dans les différents onglets.

Appuyer sur le bouton Start.

Nota : Selon l'actualité, cette documentation sera actualisée et complétée avec d'autres exemples.

S. Alexandre sylviale@bigfoot.com