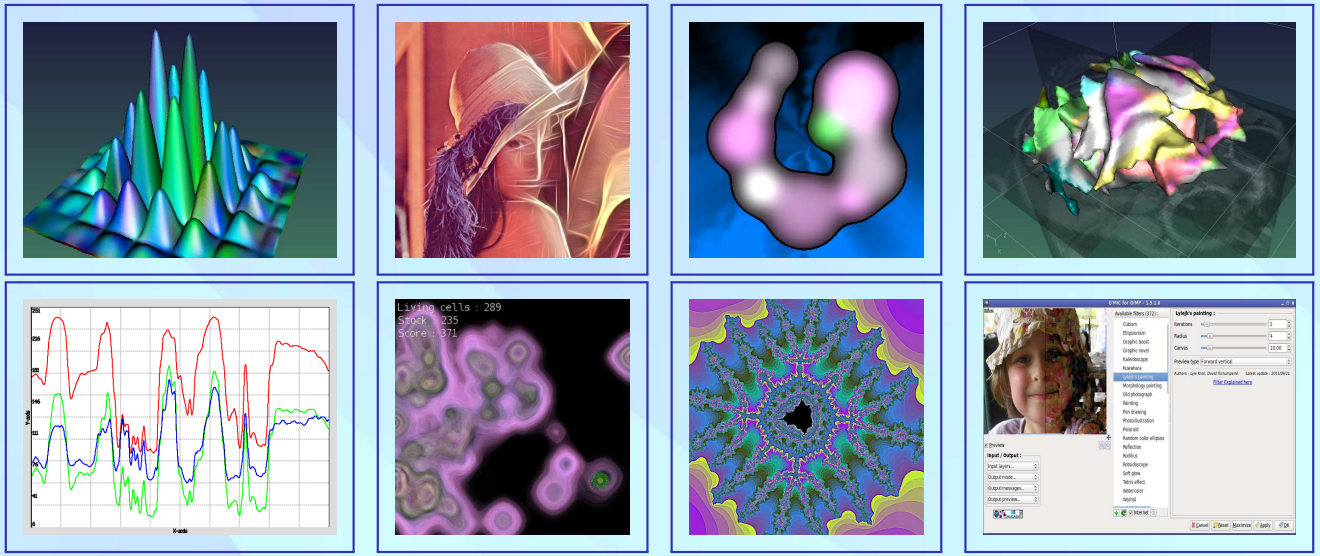
G'MIC

GREYC's Magic Image Converter



David Tschumperlé

Image Team, GREYC / CNRS (UMR 6072)

Capitole du Libre 2012, Toulouse/France, Novembre 2012

Plan de la présentation

1. Contexte & motivations
2. Buts & structure du projet G'MIC
3. Démonstrations

Plan de l'exposé

1. Contexte & motivations
2. Buts & structure du projet G'MIC
3. Démonstrations

Contexte

- Recherche en traitement d'image, au laboratoire GREYC de l'ENSICAEN / CNRS / Université de Caen.

⇒ Développement d'algorithmes (que l'on espère) innovants pour résoudre des problèmes liés au traitement d'image (amélioration, segmentation, détection,...).

Contexte

- Collaborations fréquentes avec des entreprises / laboratoires possédant des images à traiter.

...

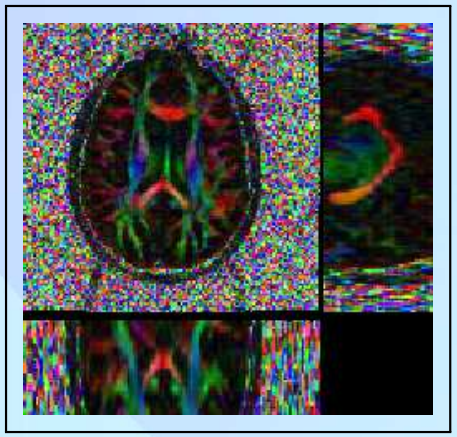
⇒ Données images provenant de nombreuses sources différentes.

Contexte

- Images à traiter très variées : 2D, 2D+t, 3D, 3D+t, pixels à valeurs flottantes et vectorielles ou matricielles.

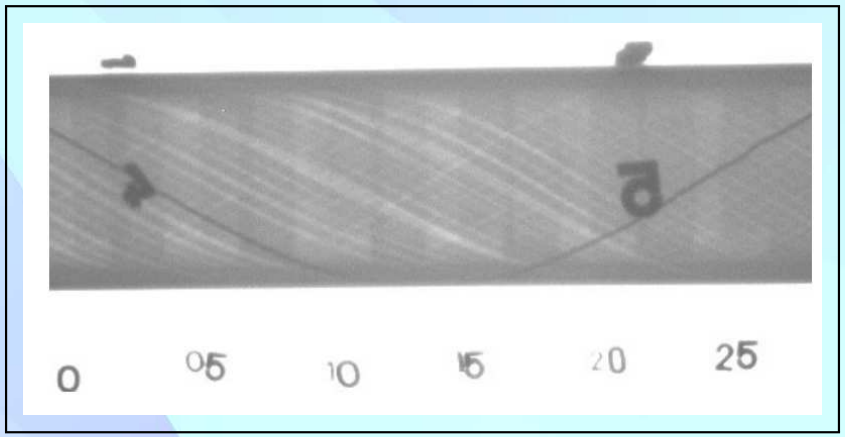
$\Rightarrow$ On peut être assez loin des photos couleurs !

(a) $I_1 : W \times H \rightarrow [0, 255]^3$



(b) $I_2 : W \times H \times D \rightarrow [0, 65535]^{32}$

(c) $I_3 : W \times H \times T \rightarrow [0, 4095]$



(d) $I_4 : W \times H \times T \rightarrow [0, 4095]$

Contexte

- Besoins d'outils de visualisation / exploration, de conversion entre formats, d'application d'opérateurs classiques de TI (filtrage, transformations géométriques, analyse fréquentielle, ...) pour des images très génériques, quelquefois sur des milliers d'images en même temps.

- Question typique entendue au labo : "Comment je peux faire pour convoluer facilement une image 3d à 32 canaux par un noyau gaussien 3d anisotrope ?"

Contexte

⇒ Peu d'outils libres existants pour ces tâches.

- Soit faciles à utiliser, mais pas assez génériques (ImageMagick, GraphicsMagick, ...).
- Soit très flexibles, mais réservés à des programmeurs avertis (bibliothèques externes pour langages de programmation divers).

- Depuis 1999, développement d'une bibliothèque C++ générique pour le traitement d'image.

http://cimg.sourceforge.net

Motivations

CImg est une bibliothèque C++ qui est :

Facile à installer et à utiliser.

Suffisamment générique pour traiter une grande variété de types d'images (basée template).

Fournit les algorithmes classiques (et indispensables) du traitement d'image.

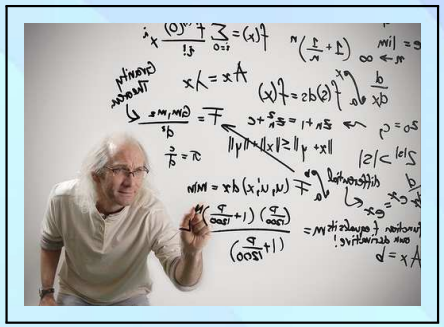
Bibliothèque extensible par nature.

Portable sur plusieurs OS et architectures.

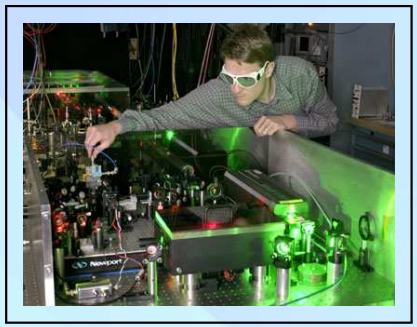
Distribuée sous licence open-source.

Motivations

- Problème résolu ? Non ⇒ généricité des personnes ! ☺
- Le monde de la recherche en traitement d'image est composé de gens très différents :



Mathématiciens



Physiciens

Geeks

Biologistes ...

⇒ Proposer une bibliothèque C++ pour le TI est une approche encore trop restrictive pour satisfaire tout le monde !

Plan de l'exposé

1. Contexte & motivations
2. Buts & structure du projet G'MIC
3. Démonstrations

Buts du projet G'MIC

http://gmic.sourceforge.net

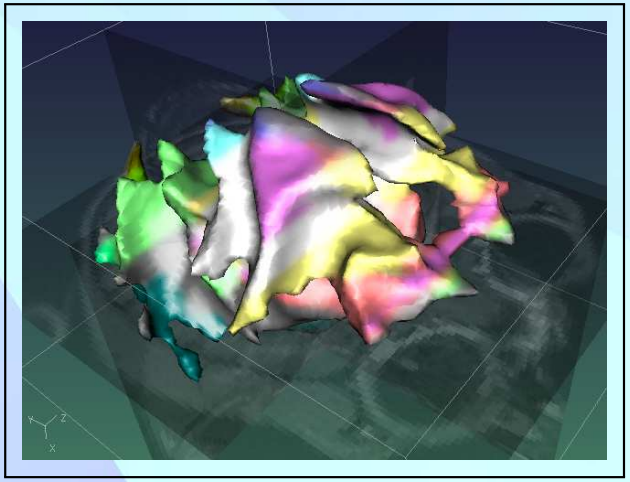
- **Objectifs :** G'MIC se propose de définir différentes interfaces d'utilisation pour accéder aux fonctionnalités de CImg.
- Interfaces plus ou moins complexes (et puissantes) pour différents publics visés.

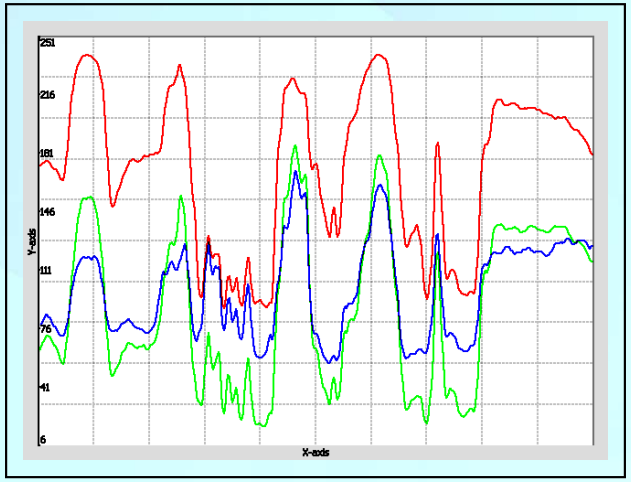
- **Moyen technique :** Définition d'un langage de script **complet** et **concis** pour le **traitement d'images génériques** (langage *G'MIC*), utilisé comme couche de base dans les interfaces d'utilisation.

Buts du projet G'MIC

1. Définition d'un langage de script **complet** et **concis** pour le **traitement d'images génériques** (langage *G'MIC*).
   - **Complet :** Environ 750 commandes disponibles à ce jour pour la visualisation, le filtrage, les transformations géométriques / colorimétriques, l'extraction de caractéristiques, le rendu 3d, le calcul matriciel, le tracé de courbes, ...

$\longrightarrow$ Documentation (.pdf) de plus de 300 pages.

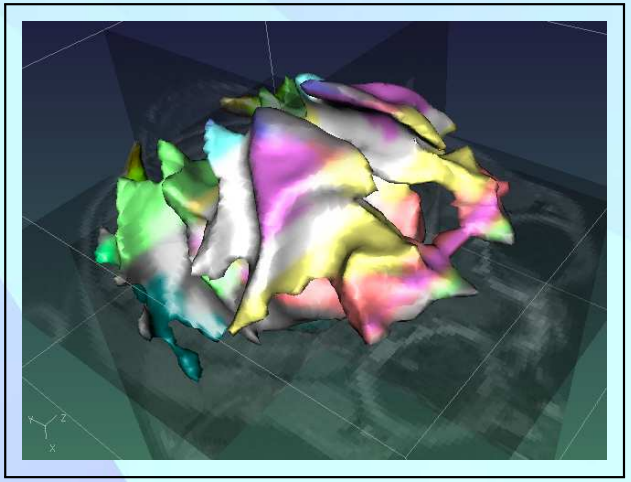


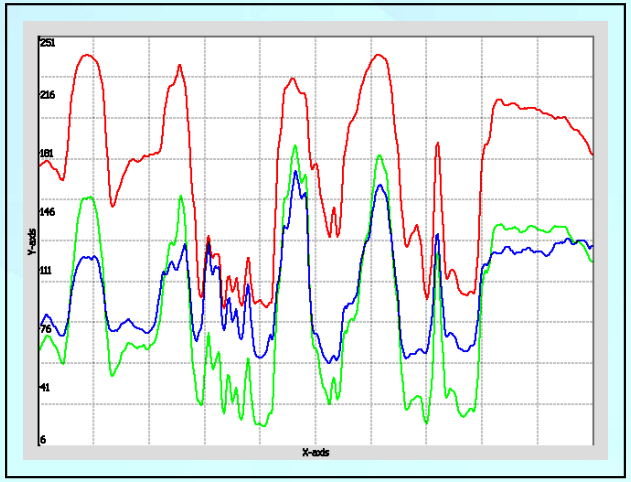


Buts du projet G'MIC

1. Définition d'un langage de script **complet** et **concis** pour le **traitement d'images génériques** (langage *G'MIC*).
   - **Concis :** Langage élaboré spécialement pour écrire des séquences d'opérations de traitement d'images (pipelines). Langage interprété et extensible, par l'ajout de fonctions personnalisées, très courtes à écrire.

$\longrightarrow$ Initialement optimisé pour une utilisation en ligne de commande.





Buts du projet G'MIC

2. Distribution d'une **implémentation libre** d'un interpréteur de ce langage (sous forme d'une bibliothèque C++).
   - **Intégration :** Possibilité d'intégrer les fonctionnalités de *G'MIC* dans des logiciels libres tiers (logiciels de retouches photographiques, logiciels de dessin, ...).
   - **Licence libre :** Bibliothèque distribuée sous licence CeCILL (compatible GPL).

$\longrightarrow$ Peu d'intégrations "extérieures" réalisées à ce jour :

- ★ *EKD*, logiciel de post-production vidéo.
- ★ **En projet :** *Krita* (plug-in), logiciel de dessin.
- ★ **En projet :** *Delaboratory*, logiciel de développement de photos RAW.

Buts du projet G'MIC

3. Fournir des interfaces **utilisables par tous**, embarquant l'interpréteur *G'MIC* (multi-plateforme).
   - **gmic :** Outil permettant de manipuler des images génériques en ligne de commande (CLI). Concurrent aux outils CLI de ImageMagick / GraphicsMagick.

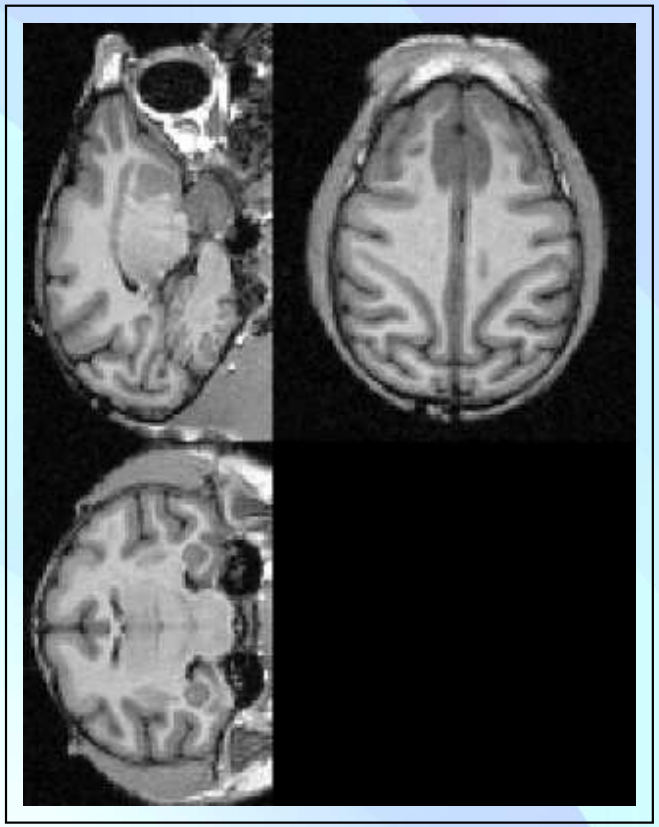
```
dtschump@[illegible]:~$ gmic ~/work/img/lena.bmp -blur 3 -mirror x
[gmic]-0./ Start G'MIC parser.
[gmic]-0./ Input custom commands file '/home/dtschump/work/src/resources.gmic' (added 14 commands, total 1121).
[gmic]-0./ Input custom commands file '/home/dtschump/work/src/gmic/src/gmic_def.gmic' (added 1107 commands, total 222
8).
[gmic]-0./ Set dynamic 3d rendering mode to flat-shaded.
[gmic]-0./ Input file '/home/dtschump/work/img/lena.bmp' at position [0] (1 image 512x512x1x3).
[gmic]-1./ Blur image [0], with standard deviation 3 and neumann boundary.
[gmic]-1./ Mirror image [0] along the 'x'-axis.
[gmic]-1./ Display image [0] = 'lena.bmp*'.
lena.bmp* (512x512x1x3) : this = 0xbf9852e4, size = 1/16 [3072 Kb], data = (CImg<float>*)0xa027a44..0xa027a5b.
  [0] : this = 0xa027a44, size = (512,512,1,3) [3072 Kb], data = (float*)0xb73ba008..0xb76ba007 (non-shared) = [ 203.2
86 207.053 210.367 212.452 212.914 211.713 209 205.179 ... 65.5009 63.9167 62.7955 61.9959 61.279 60.5754 59.9074 59.3
564 ], min = 9.43401, max = 250.19, mean = 128.229, std = 55.711, coords_min = (511,440,0,1), coords_max = (68,57,0,0)
.
[gmic]-1./ End G'MIC parser.
dtschump@[illegible]:~$
```

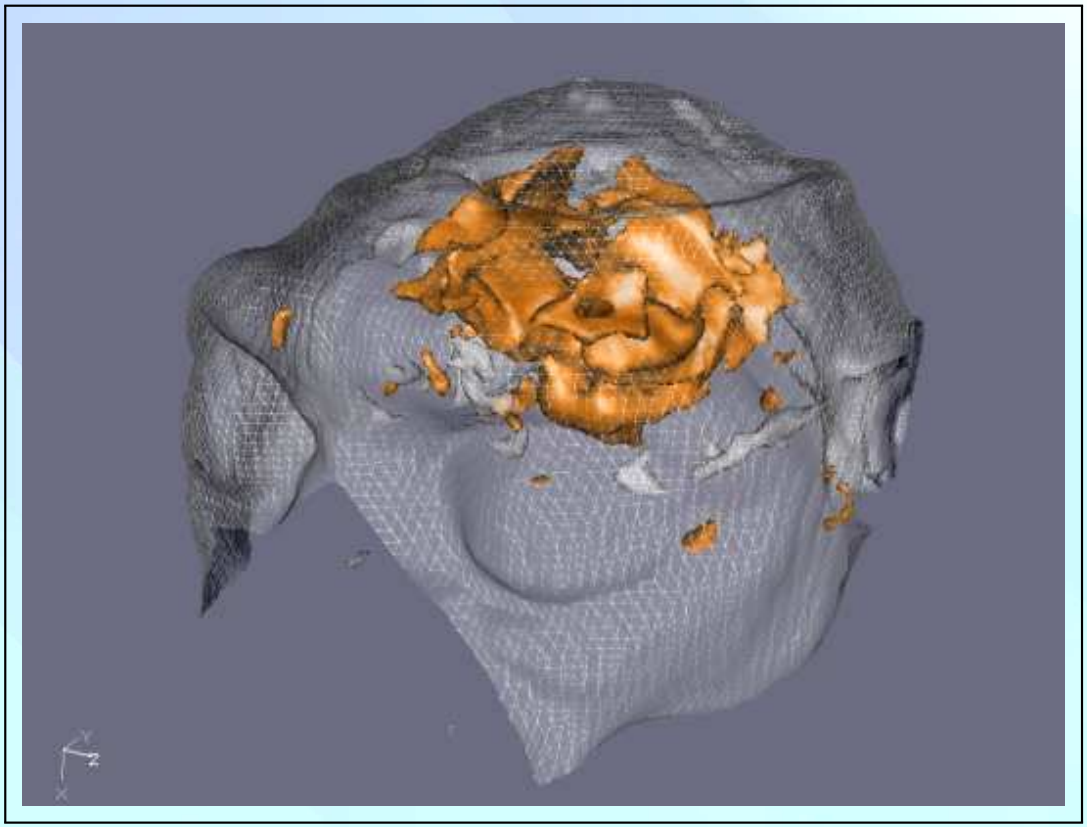
Exemple d'utilisation du langage G'MIC

```
gmic lena.bmp -blur 3 -sharpen 1000 -noise 30 -+
"'cos(x/3)*30'"
```

Exemple d'utilisation du langage G'MIC

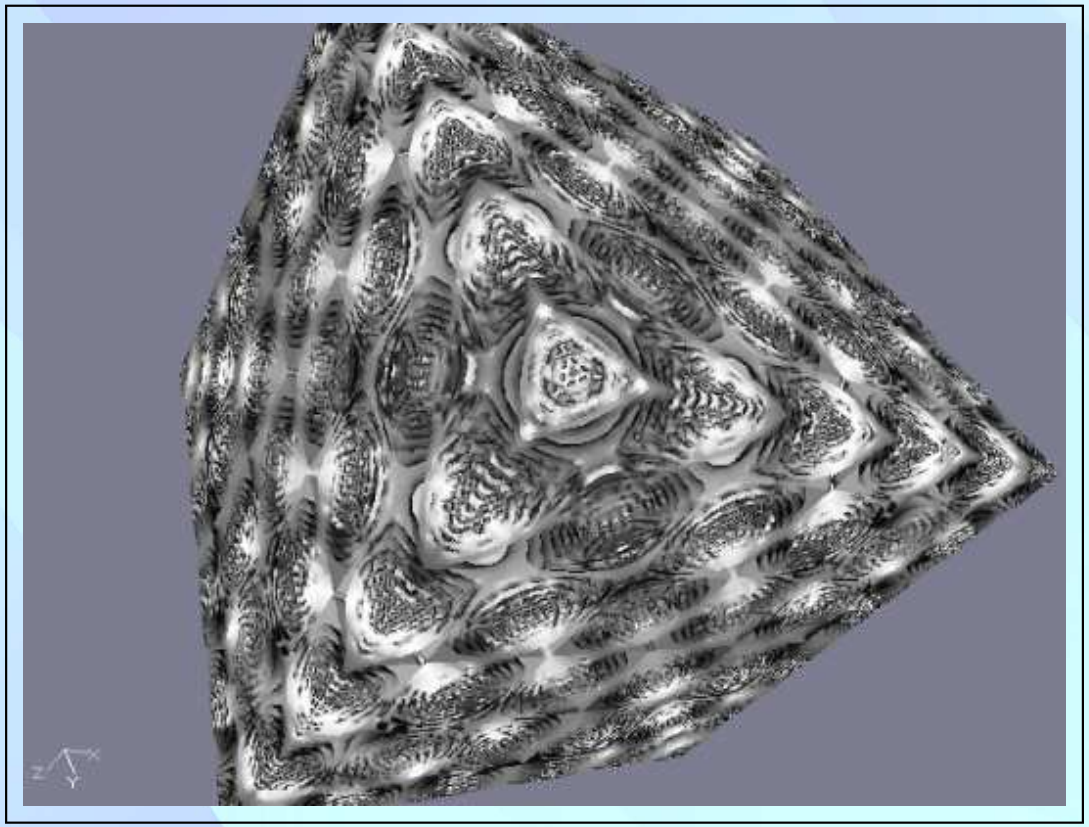
```
gmic reference.inr -flood 23,53,30,50,1,1,1000 -flood[-2]
0,0,0,30,1,1,1000 -blur 1 -isosurface3d 900 -opacity3d[-2] 0.2
-color3d[-1] 255,128,0 -+3d
```





Exemple d'utilisation du langage G'MIC

```
gmic -isosurface3d
"'sin(x*y*z)'",0,-10,-10,-10,10,10,10,128,128,64
```



Exemple d'utilisation du langage G'MIC

```
gmic milla.bmp -f '255*(i/255)^1.7' -histogram 128,0,255 -a c -plot
```

is the G'MIC equivalent to this C++ code (using CImg):

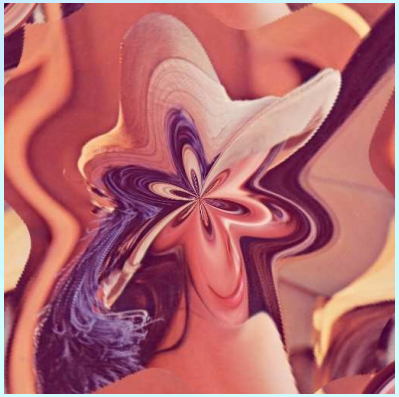
```
#include "CImg.h"
using namespace cimg_library;
int main(int argc,char **argv) {
const CImg<>
img("milla.bmp"),
hist = img.get_histogram(128,0,255),
img2 = img.get_fill("255*((i/255)^1.7)",true),
hist2 = img2.get_histogram(128,0,255);
(hist,hist2).get_append('c').display_graph("Histograms");
return 0;

}
```

Exemple d'utilisation du langage G'MIC

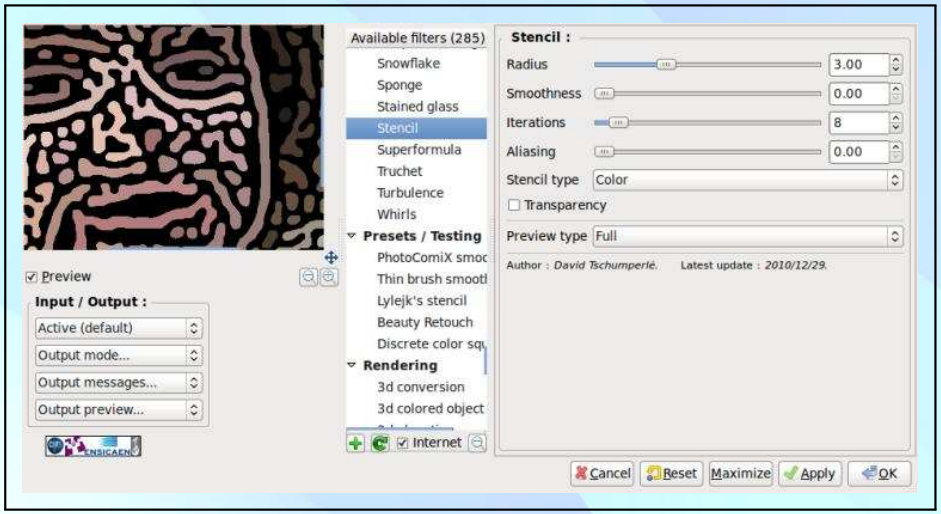
- Un pipeline écrit en G'MIC peut s'ajouter comme une nouvelle fonction G'MIC.
- Faire des pipelines permet aussi d'écrire des filtres d'effets artistiques !

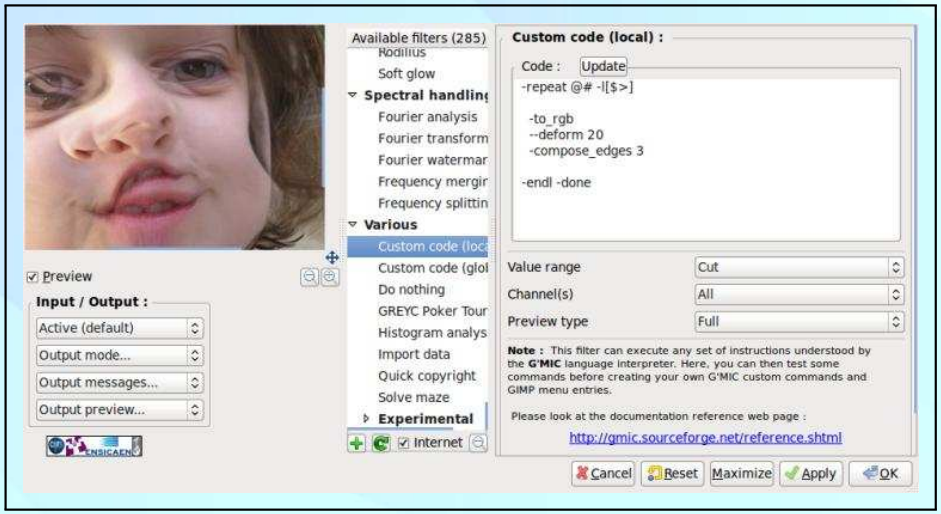
```
gmic lena.jpg -pencilbw 0.3 -o gmic_lena1.jpg; gmic lena.jpg
-cubism 160 -o gmic_lena3.jpg
gmic lena.jpg -flower 10 -o gmic_lena4.jpg; gmic lena.jpg
-stencibw 30 -o gmic_lena2.jpg
```



Buts du projet G'MIC

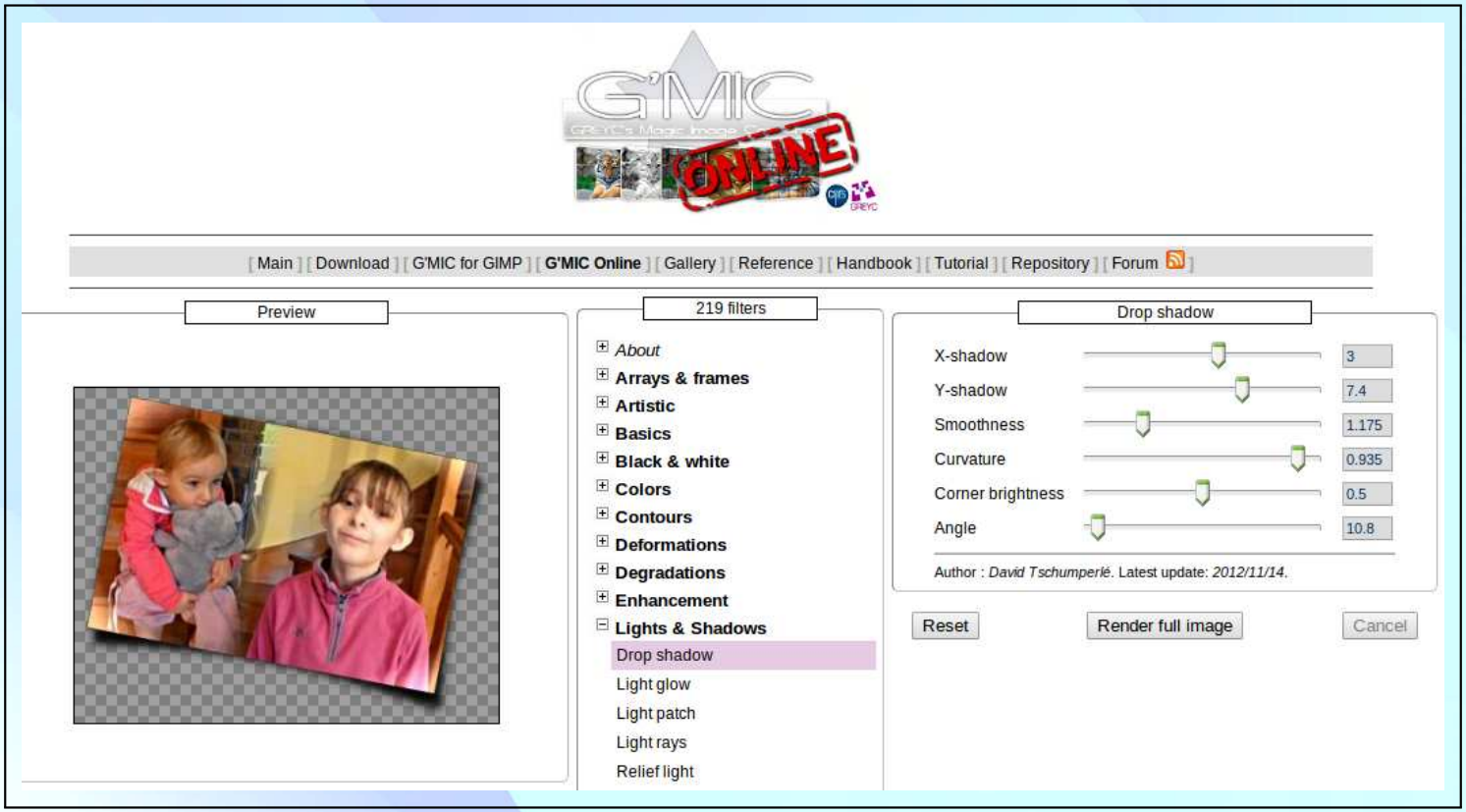
3. Fournir des interfaces **utilisables par tous**, embarquant l'interpréteur *G'MIC* (multi-plateforme).
   - **gmic_gimp :** Plug-in pour GIMP permettant d'appliquer des filtres créés avec *G'MIC* sur des images 2D.





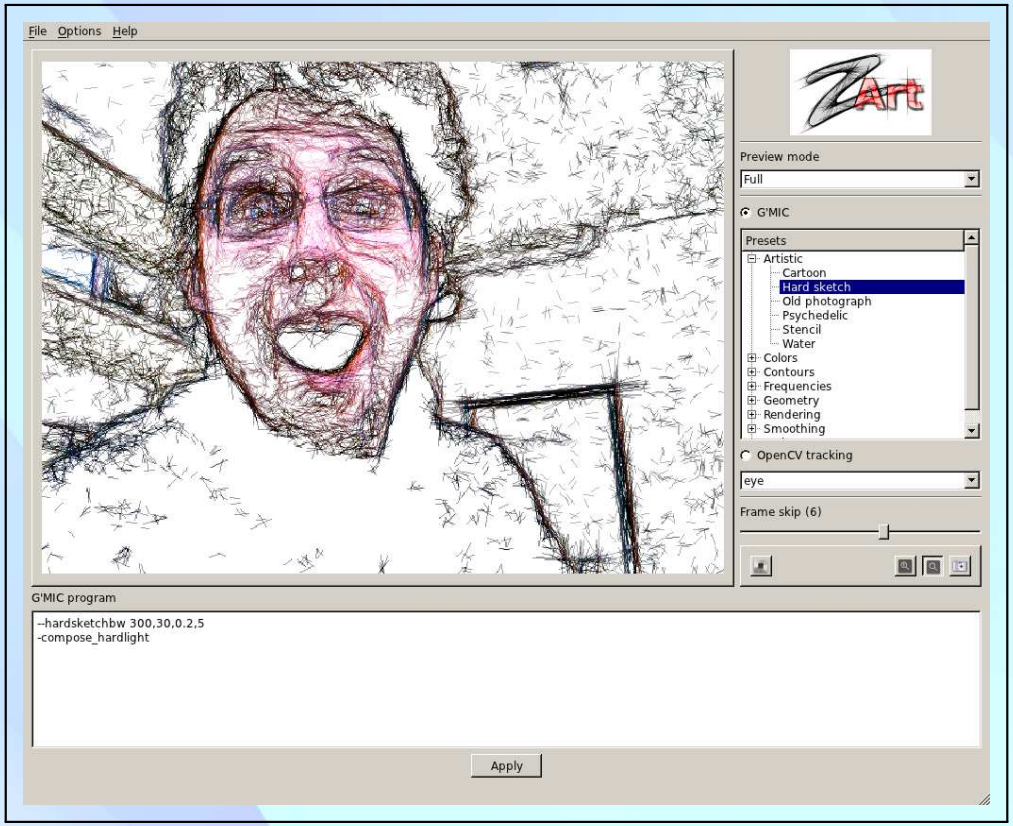
Buts du projet G'MIC

3. Fournir des interfaces **utilisables par tous**, embarquant l'interpréteur *G'MIC* (multi-plateforme).
   - **G'MIC Online :** Web service pour la manipulation d'images online (utilisable à partir d'un simple navigateur web).
     `https://gmicol.greyc.fr`

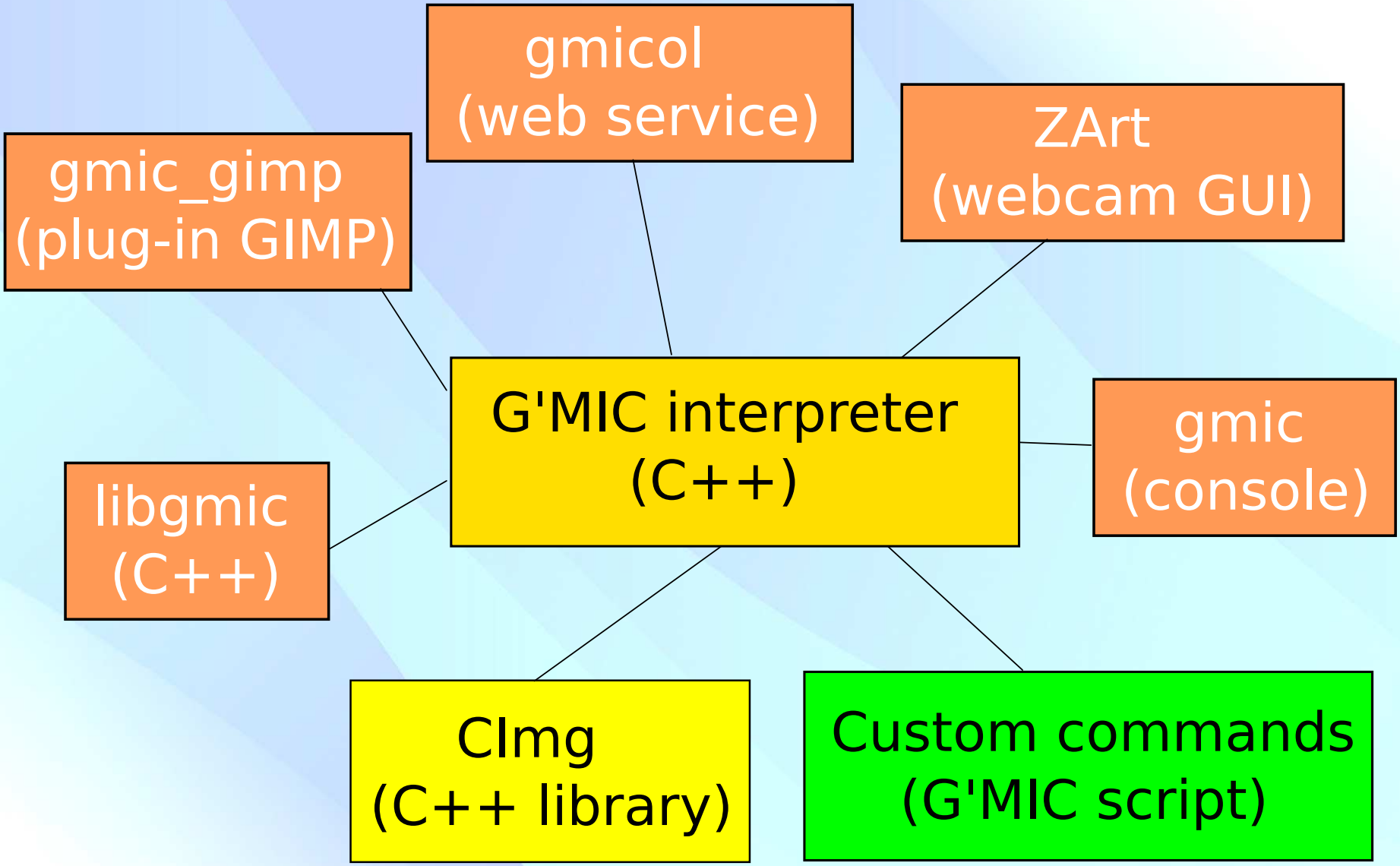


Buts du projet G'MIC

3. Fournir des interfaces **utilisables par tous**, embarquant l'interpréteur *G'MIC* (multi-plateforme).
   - **ZArt :** Interface QT pour la manipulation d'images provenant de la webcam (utilisée comme plateforme de démonstration).



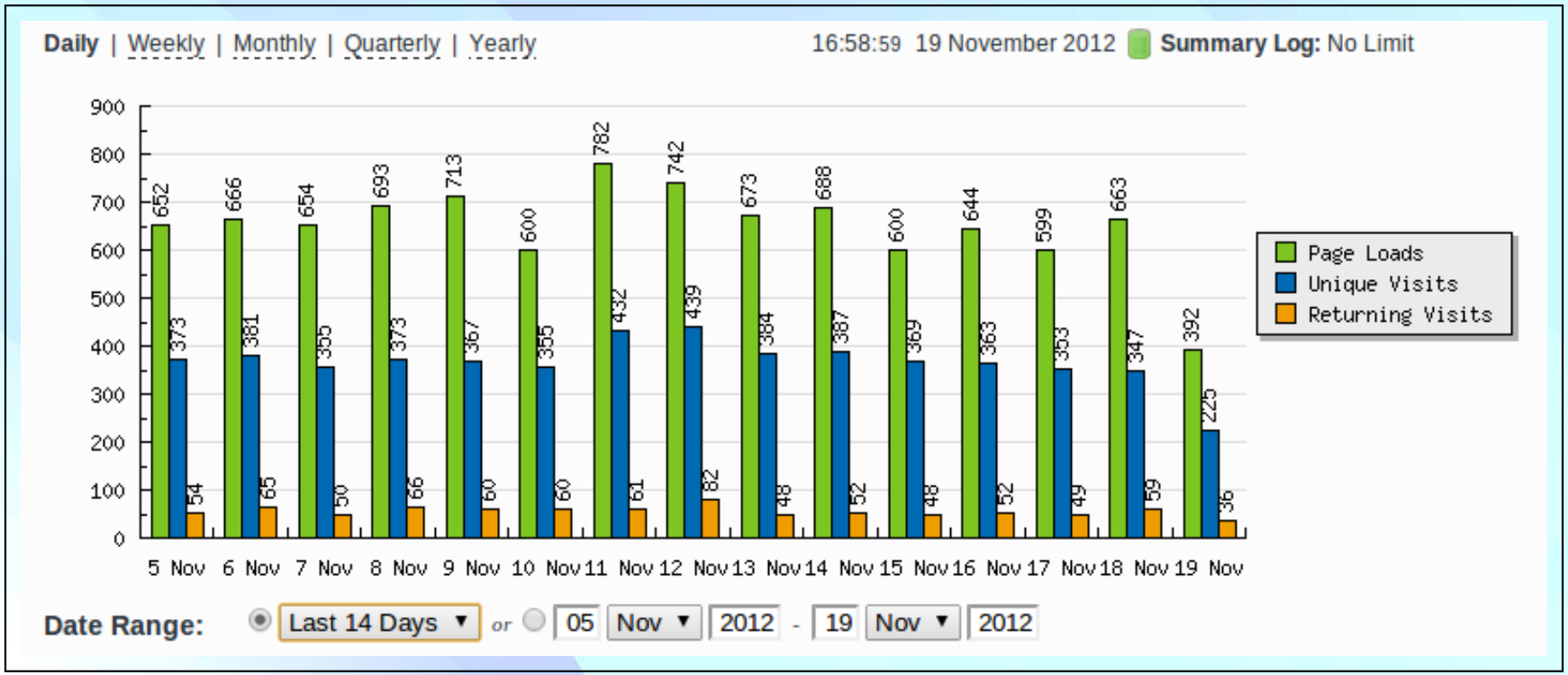
Vue d'ensemble de G'MIC



Etat actuel

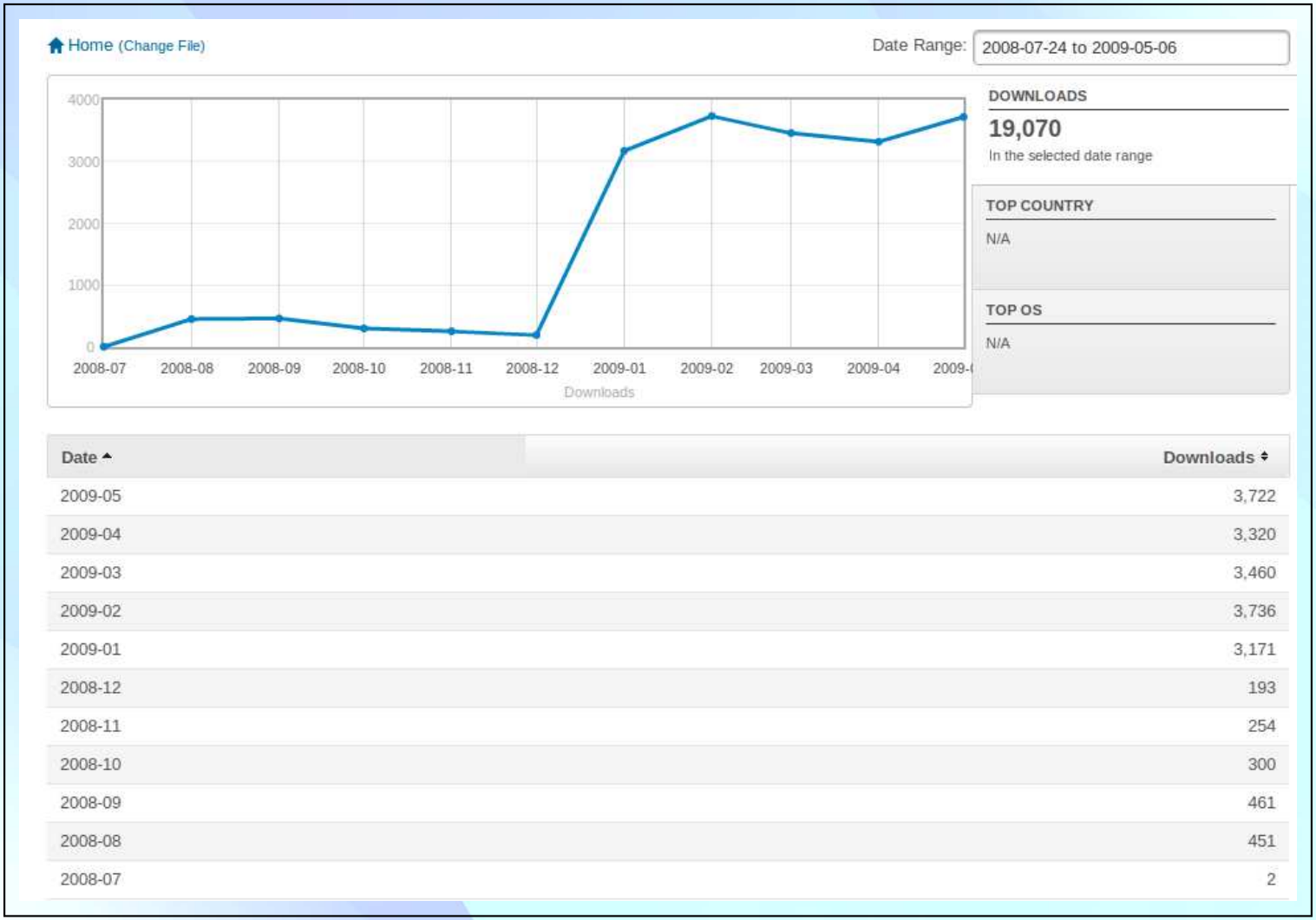
- Aujourd'hui, le projet *G'MIC* c'est :
  - Un peu moins de 100.000 lignes de code (en langages C++ et G'MIC essentiellement).
  - 250-350 téléchargements / jour (+ de 700.000 depuis Juillet 2008).
  - 350-400 visiteurs uniques / jour sur les pages web du projet.

  ⇒ Chiffres satisfaisants compte tenu du public visé initialement !



Etapes d'évolution

- Sources/binaires du plug-in GIMP, disponible en Janvier 2009.
  $\longrightarrow$ "Explosion" du nombre de visites et de téléchargements.



Date	Downloads
2009-05	3,722
2009-04	3,320
2009-03	3,460
2009-02	3,736
2009-01	3,171
2008-12	193
2008-11	254
2008-10	300
2008-09	461
2008-08	451
2008-07	2

Etapes d'évolution

- **Effet cascade :** De plus en plus d'utilisateurs. Référencement de *G'MIC* sur des forums, des blogs, des sites d'informations (parlant graphisme ou logiciels libres) : `framasoft, linuxfr, webupd8, libregraphicsworld, pcastuces, gimpfr, linuxgraphics, gimpusers, ...`).
- **Le développement devient collaboratif :** Aide de béta-testeurs, d'empaqueteurs (Debian, Arch, Gentoo,..), suggestions d'améliorations, traduction, design de mascottes, filtres.

Etapes d'évolution

- **Retour à la case départ :** Mes collègues de l'équipe s'aperçoivent que *G'MIC*, c'est quand même bien pratique !

→ Création de ZArt et G'MIC Online par Sébastien Fourey de l'équipe IMAGE du GREYC, et distribution avec *G'MIC*.

- Utilisation pour la fête de la science, et pour illustrer des concepts de TI aux étudiants de l'ENSICAEN.

<happy>J'ai réussi à convaincre des professionnels du traitement d'image de l'intérêt du projet ! ☺ **</happy>**

Plan de l'exposé

1. Contexte & motivations
2. Buts & structure du projet G'MIC
3. Démonstrations

Ce qui a intéressé

- **Utilité** : Quelques caractéristiques qui ont fait de la pub au projet.

1. Un des rares projets libres a proposer un algorithme de “débruitage” d’image performant :

Ce qui a intéressé

- **Utilité** : Quelques caractéristiques qui ont fait de la pub au projet.

1. Un des rares projets libres a proposer un algorithme de “débruitage” d’image performant :

Ce qui a intéressé

- **Utilité** : Quelques caractéristiques qui ont fait de la pub au projet.

2. Un des rares projets libres a proposer un algorithme d' “inpainting” d'image :

Ce qui a intéressé

- **Utilité** : Quelques caractéristiques qui ont fait de la pub au projet.

2. Un des rares projets libres a proposer un algorithme d' “inpainting” d'image :

Ce qui a intéressé

- **Utilité** : Quelques caractéristiques qui ont fait de la pub au projet.

2. Un des rares projets libres a proposer un algorithme d' “inpainting” d'image :

Ce qui a intéressé

- **Utilité** : Quelques caractéristiques qui ont fait de la pub au projet.

2. Un des rares projets libres a proposer un algorithme d' “inpainting” d'image :

Ce qui a intéressé

- **Utilité** : Quelques caractéristiques qui ont fait de la pub au projet.

3. Effet 'Fractalius' (49$ pour Photoshop) reproduit avec *G'MIC* 'Rodilius' ($\rightarrow$ 0$, 10 lignes de code G'MIC !) :

Redfield Fractalius

G'MIC Rodilius

Rodilius en G'MIC

```
rodilius :
  -v - -repeat @# -l[$>] -split_opacity -rv
    -if {!$6} -negative[-1] -endif
    --f[-1] 0 -nm[-1] @{-2,n}
    -repeat {round($4)}
      angle={$5+$>*180/round($4)}
      --blur_linear[-2] $1%,{$1*$2/100}%,$angle,1 -b[-1] 0.7 -sharpen[-1] $3 -max[-2,-1]
    -done -rm[-2]
    -if {!$6} -negative[-1] -endif
  -rv -a c -endl -done -v +
```

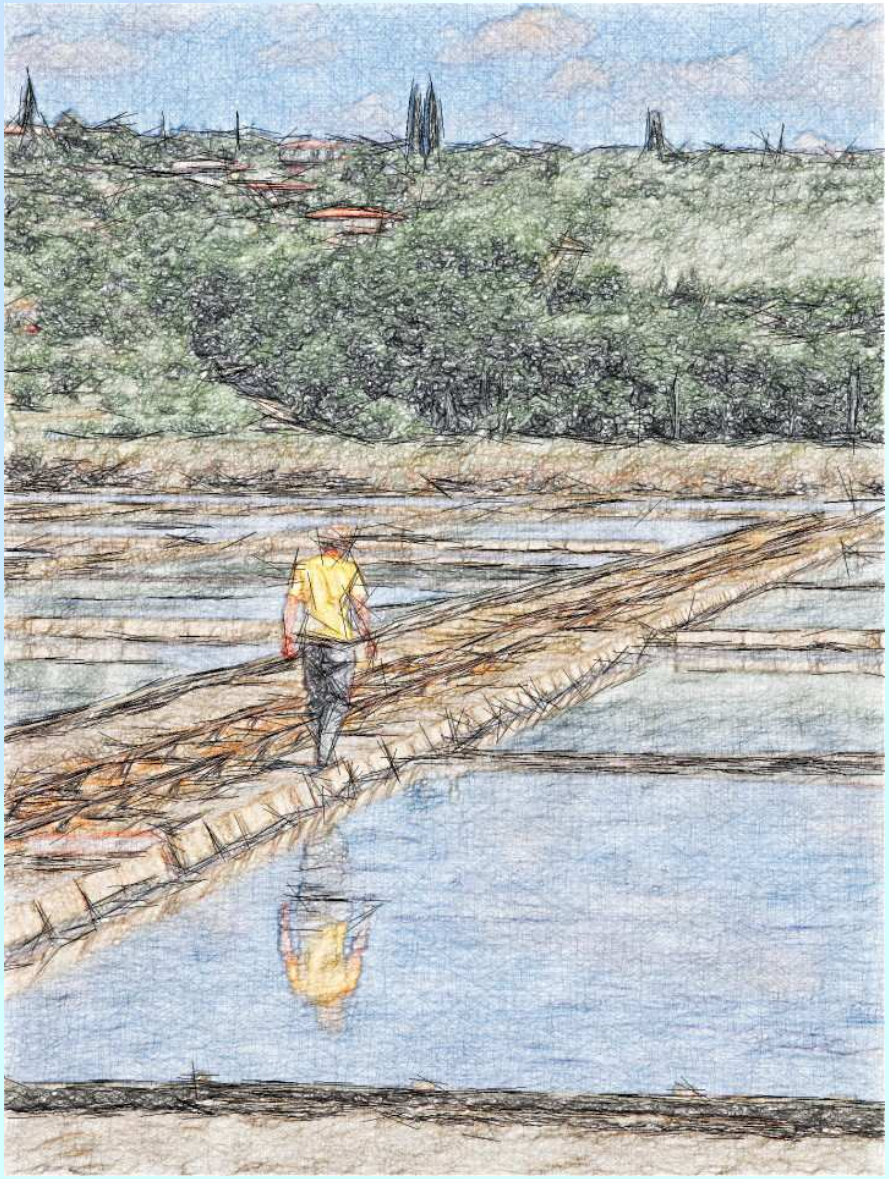
Ce qui a intéressé

- **Utilité** : Quelques caractéristiques qui ont fait de la pub au projet.

4. Effet 'Sketch' original disponible dans *G'MIC*.

→ Publication d'un article à la conférence **IEEE International Conference on Image Processing**, en 2011.

Résultats de sketch



Résultats de sketch

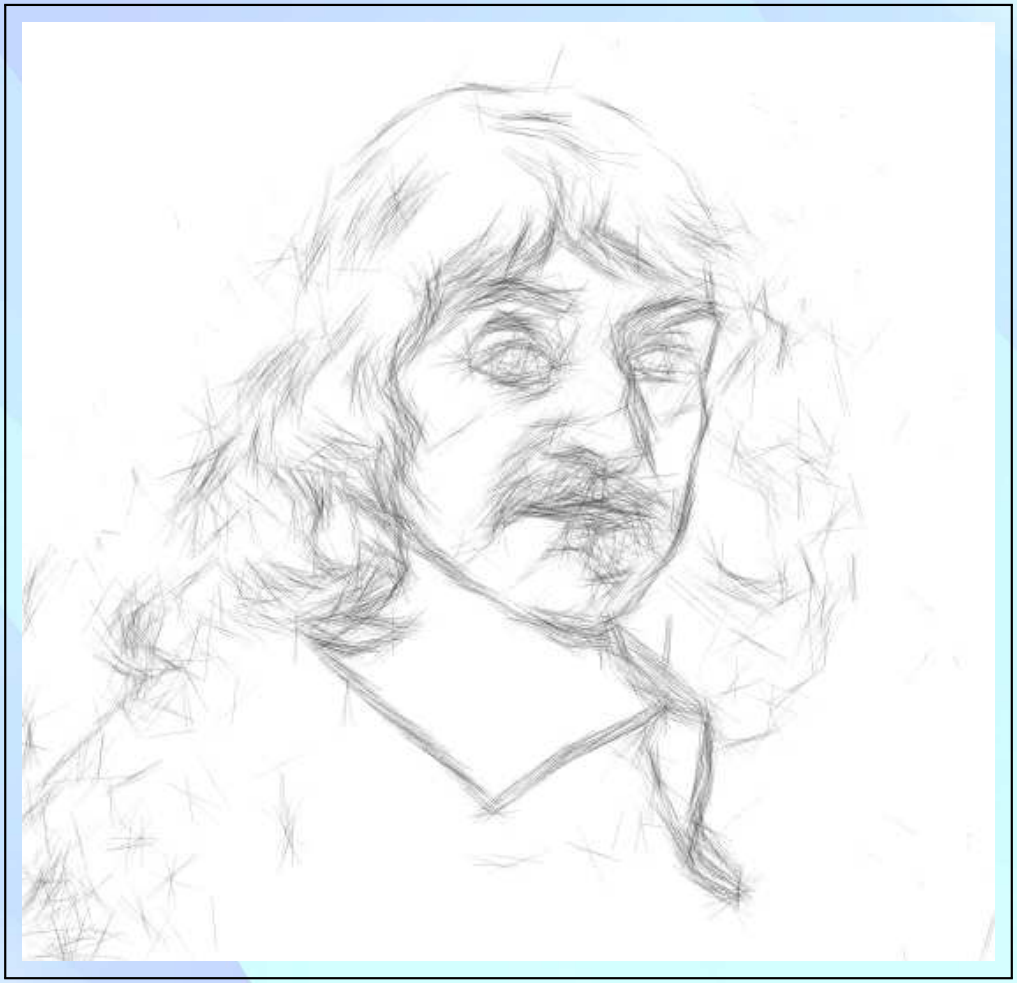
Résultats de sketch

Résultats de sketch

Résultats de sketch

Résultats de sketch

- Permet de simuler plusieurs étapes de croquis vers un dessin final.



Step 1 (straight strokes)

Résultats de sketch

Step 2 (curved strokes)

Résultats de sketch

Step 3 (curved strokes + colorization)

Résultats de sketch

Step 4 (curved strokes + colorization)

Résultats de sketch

Peinture finale (l'image d'origine en réalité !)

Live demo !

(Plug-in GIMP, CLI, G'MIC Online, ZArt)

The end

Merci de votre attention !

Si vous avez des questions...