



Current state and 2012-2013 retrospective

David Tschumperlé, Jérome Boulanger and Patrick David

Libre Graphics Meeting, Madrid / Leipzig, April 2014



Project Overview

The G'MIC project : Overview





http://gmic.sourceforge.net

- A free software which aims at providing user interfaces to perform complex image processing operations.
- Technical means : G'MIC defines its own script language, specifically designed to build image processing pipelines. The G'MIC language interpreter is then embedded in all proposed user interfaces.

G'MIC script language



- Full-featured: More than 800 commands available for image visualization, filtering, geometry / color management, features extraction, 3d rendering, matrix computations, graphical plots, ...
- **Conciseness:** The G'MIC language has been designed specifically for being concise. This is an interpreted language, which can be extended by custom user-defined functions.
- \rightarrow Technical documentation (.pdf) has more than 400 pages. \rightarrow 83k lines for the whole source code (*Clmg included*).









- G'MIC provides an open-source implementation of the language interpreter (as a C++ library).
 - Integrations: Third-party softwares can easily get all G'MIC features (interesting for image retouching or painting softwares, ...).
 - Free software: The G'MIC interpreter is distributed under the CeCILL license (GPL-compatible).
 - \longrightarrow Examples of integrations:
 - * Krita (plug-in), painting software, integration started in 2013.
 - * EKD, video editing software, integration started in 2010.
 - * Planned: Delaboratory, RAW photograph postprocessing application.



 gmic: Tool to manipulate the G'MIC interpreter from the command line (CLI). Competitor to the CLI tools of the ImageMagick / GraphicsMagick projects.

**\$ gmic ~/work/img/lena.bmp -blur 3 -mirror x dtschump® [gmic]-0./ Start G'MIC parser. [gmic]-0./ Input custom commands file '/home/dtschump/work/src/resources.gmic' (adde<u>d 14 commands, total 1121).</u> [qmic]-0./ Input custom commands file '/home/dtschump/work/src/qmic/src/qmic def.qmic' (added 1107 commands, total 222 [gmic]-0./ Set dunamic 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*'. ena.beg> (512/512/dx3) : 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@ :~\$

G'MIC interfaces : GIMP plug-in



• gmic_gimp: Plug-in for GIMP, provides more than 600 image filters.

-00	G'MIC for GIMP 1.5.8.5 (be	ta) - Linux 64 bits _ = = *			
K. P. H.	Available filters (646) : Black & White Bwisteampunk-pen	Threshold etch : Replace tones with noise generated lines			
	Charcoal	Input Image Settings			
I have a	Colorize [comics] Colorize [photographs]	Threshold low 125 \$			
	Colorize [with colormap]	Threshold mid 153 🗘			
	Dithering Hard sketch	Threshold high 171 2			
	Hough sketch	Threshold max 🔋 185 🗘			
	Ink wash Multi-laver etch	Blur amount 0.10 \$			
	Pencil	Output Etch Settings			
	Sketch	Horizontal amount 50 🛟			
Preview 😑 🛨	Stamp	Cross-hatch amount 80 🛟			
Input / Output :	Threshold etch Colors	Vertical 1 amount 0 50 2			
Input layers ‡	Abstraction	Vertical 2 amount 10 💈			
Output mode ‡	Boost-Fade Channel processing	Horizontal length 15 🗘			
Output messages ‡	CMYK tone p	Vertical 1 length			
Output preview ‡	Colorhalaasa	Vertical 2 length 20 🗘			
	+ C 🗹 Internet 🖃	Flip cross-hatch			
		X Cancel Reset Maximize Apply VK			

G'MIC interfaces : Web service



• G'MIC Online: Web service for manipulating images online (like the GIMP plug-in, with less filters and running on a web browser). https://gmicol.greyc.fr

		Reportery Gougier Four Ba
Preview	283 titers	Local normalization
	Sefermations Selection Second S	Anpthude Anpthude Anpthude Angthude Angthude
Local view:	Simple local contrast	
Upload your image Upload your image Choose File No file chosen Upload (Upload (Upload) Upload) Upload Upload Upload	Texture enhance Tone mapping Overview	

G'MIC interfaces : Webcam effects



• **ZArt**: A QT-based interface for manipulating images acquired from the webcam (used as a demonstration plateform).









Filter Showcase:

Polygonize



- **Goal:** Transform an image into a polygonized rendering with triangles having uniform flat colors.
- Made by : David, to test the stability of 3d flat objects rendering in G'MIC. Then, we realized it was a cool filter, so we kept it ©.
- How is this done? Starting from an uniform grid, we move the grid points iteratively towards the nearest contour points.
- \Rightarrow **35** lines of G'MIC code.

(all included: GUI description + algorithm).





Open input image.



-æ G'i	MIC for GIMP 1.5.8.5 (beta) - Li	inux 64 bits	- • ×
	Available filters (646) : Polygonize Posterized dithering Poster edges Psychedelic glasswork tile	Polygonize : Amplitude Smoothness Minimal area C	169 ¢
	Rodilius Shapeism Simple noise canvas Warhol Watercolor Whirl drawing :	X-resolution Y-resolution Outline color Preview type Full	5.00 \$
Preview Input / Output : Input layers Output mode Output messages Output preview Output preview	Wild cartoonizer • Black & white • Colors • Contours • Deformations • Degradations • Details • Film emulation • Frames + C V Internet	Author: David Tschumperlé. Latest update: 2013/12/02.	
		X Cancel Reset Maximize Apply	✓ ОК

Invoke G'MIC plug-in and select Artistic / Polygonize.





Get your polygonized result.





Another filter variation created by Samj on GimpChat (folded paper?).



Filter Showcase:

Rodilius



- Goal: Try to mimic the famous Fractalius effect from Redfield (39.90\$ plug-in for Photoshop).
- Made for : Rod on GimpChat has designed a first version of the filter as a FilterForge pipeline. David has translated it into G'MIC code.
- How is this done? We compute linear blurs with various orientations then mix them all using Lighten only or Darken only blending modes. Additional (aggressive) anisotropic smoothing and sharpening are added on each orientation layer.

⇒ 28 lines of G'MIC code. (all included: GUI description + algorithm).





Open input image.



	G'MIC for GIMP 1.5.8.5 (beta	a) - I	Linux 64 bits		-	• ×
	Available filters (646) :		Rodilius : —			
	Cubism		Amplitude		7.30	-
	Dream smoothing		Thickness		19.03	1
• •	Ellipsionism		Sharppoor		- 401.15	
	Graphic boost 2	1	Sharphess		491.13	-
(PND)	Graphic novel		Orientations	(=)	16	÷
	Hope poster		Offset		30.00	-
	Kuwahara		Smoothness		2	÷
	Lylejk's painting		Color mode	Lighter		
		2	color mode			-
	: Painting	:	Channel(s)	All		-
	Pastell art		Preview type	Full		
	Pen drawing					-
Preview	D Phoenix steam-pencil		Authors: Rod/Gir Latest update: 2	mpChat and David Tschumperlé. 013/03/05.		
Input / Output :	Photoniustration					
Input layers ‡	Posterized dithering					
Output mode	Poster edges					
	Psychedelic glasswork tile					
Output messages ‡	Rodilius					
Output preview ‡	Shapeism					
	+ C Internet F					
Ensicaen						
			× Cancel 🗧	Reset Maximize <a>Apply	√ ок	<

Invoke G'MIC plug-in and select Artistic / Rodilius.





Wait a little bit, then enjoy ! (recently parallelized for speeding up FFTs).





Two other examples, works quite well on fur.





Another example : with Darken only blending mode used.



Filter Showcase: Colorize [comics]



- Goal: Help coloring black and white sketches by allowing the artist to drawn only small color spots inside the regions to fill-in.
- Made by : David for David Revoy and Thimothé Giet, two artists (famous Krita users), after they have seen the Lazy Brush plug-in for TVPaint.
- How is this done? Color spots are extrapolated considering edge-based priority maps, with a watershed-like algorithm.
- \implies 44 lines of G'MIC code.

(all included: GUI description + algorithm).





Open input image (here, two layers : dark lineart + white background).





Add top layer with color spots on it.



-9	G'MIC for GIMP 1.5.8.5 (be	eta) - Linux 64 l	bits _ = ×				
	Available filters (646) : About	ble filters (646) : Layers ordering:					
	 Arrays & tiles Artistic 	Input layers	Color spots + lineart ‡				
	 Black & white B&W stencil Black & white 	Output layers Smoothness	Lineart + extrapolated colors 0.51 ‡				
	Black & White Bw steampunk-pen Charcoal Colorize [photographs] Colorize [with colormap] Dithering : Hard sketch	Note: You probably need to select <i>All</i> for the <i>input layers</i> option on the left. <i>color spots</i> = your layer with color indications. <i>lineart</i> = your layer with line-art (b&w or transparent). <i>extrapolated colors</i> = the G'MIC generated layer with flat colors. Warning: Do not rely too much on the preview, it is probably not accurate !					
Preview Preview C D Input / Output : All C	Hough sketch Ink wash Multi-layer etch Pencil Pencil portrait	Authors: David Ts David Revoy. Li	schumperlé, Timothée Ciet, atest update: 2013/06/19. <u>More infos on this filter</u>				
Output mode \$ Output messages \$	Stamp Threshold etch • Colors						
	<pre> Contours + C ♥ Internet ●</pre>	× Cancel 🗐	Reset Maximize 🗸 Apply 🗸 OK				
		a concet	Apply VOK				

Invoke G'MIC plug-in and select Black & White / Colorize [comics].





Wait a little bit, and enjoy !





Another example from Thimothé Giet: Original lineart + color strokes.





Result of the G'MIC Colorize [comics] filter.



Filter Showcase: Colorize [photographs]

Black & White : Colorize [photographs] GREYC

- Goal: Same goal as before but for more classical photographs.
- Made by : David, to test the extension of the previous colorization algorithm to usual photographs.
- How is this done? Same kind of color extrapolation but only on the chrominance channels CbCr of the input image, so that luminance is preserved.
- \implies **24** lines of G'MIC code.

(all included: GUI description + algorithm).

Black & white : Colorize [photographs] GREYC



Open input image (single-layer B&W photograph).

Black & white : Colorize [photographs] GREYC



Add top layer with color strokes on it.

Black & white : Colorize [photographs] GREYC

.@	G'MIC for GIMP 1.5.8.5 (beta) - Li	nux 64 bits			
✓ Preview Input / Output : All Output mode Output preview Output preview	Available filters (646) : Available filters (646) : Arrays & tiles Arrays & tiles Black & white Bw steampunk-pen Charcoal Colorize [photograph5] Colorize [with colormap] Dithering Hard sketch Hough sketch Ink wash Multi-layer etch Pencil Pencil portrait + C @ Internet 1		Colorize [pho Smoothness Anisotropy Output mode Note: This filt The bottom la top layer cont extrapolated i the entire imar recolored imar Author: David Tsc	tographs] : Merge brightness/colors er needs two layers to work proper yer must be a B&W image, while the ins color packets that will be n a smart way (edge-directed) to fil ge. At the end, you get a completely ge. chumperlé. Latest update: 2013/01/16.	= 2 = 0.20	
			× Cancel 🛛	Reset Maximize ✓ Apply	/ ~ o	ж

Invoke G'MIC plug-in and select Black & White / Colorize [photographs].
Black & white : Colorize [photographs] GREYC



Result of the filter (courtesy of pogogogo / GimpChat).



Filter Showcase: Split details



- Goal: Allow the decomposition of an image into several scales of details, so one can work on these different scales separately.
- Made by : Jérome and David for having something similar to the Wavelet decompose feature in G'MIC.
- How is this done? Images are decomposed/recomposed using a stack of gaussian-filtered image pyramids + residuals.
- \implies **74** lines of G'MIC code.

(all included: GUI description + algorithm).





Open input image.



.ee G	'MIC for GIMP 1.5.8.5 (beta) - Li	inux 64 bits		×
	MIC for GIMP 1.5.8.5 (beta) - LI Available filters (646) : Sharpen [richardson-lucy] Sharpen [shock filters] Sharpen [usharp mask] Sharpen [usharp mask] Sharpen [whiten] Simple local contrast Split details Spotify Texture enhance p Tone mapping [fast] Yag effect ;	In ux 64 bits Split details : Action Decompose options: Number of scales Base scale Details scale Output mode Preview type	Decompose/update	
Preview Preview All Output mode Output messages Output preview	Yag effect Yag effect Film emulation Frames Lights & shadows Patterns Rendering Repair Sequences Spectral filters + C V Internet C	Preview type Instructions of use: 1. Set parameter ACC parameters. Then clic 2. Process your image set use using the neutral set use using the neutral resulting recomposed canvas. 4. Repeat steps 2. and 5. When you are happ again, set parameter - 6. Get your final mod Author: Jérome Boula Latest update: 2013/1	All scales ion to Decompose/update and choose decompose K.C. For or katance, remove details in the appropriate a control (728, 728, 728), the fill of the same of the same of the same of the same image updated and shown at the top left of you 33. as many times as you want. with the result, open the GMIC plug-in once ktion to Recompose, and click OX. Fied image alone. anger and David Tschumperlé. 2/14.	the r
		× Cancel	et Maximize Apply	ж

Invoke G'MIC plug-in and select Details / Split details.





You get your input (top-left) + the decomposition into scales (here 3 scales).





Do what you want on the scales (here, we simply erase the skin defects on the middle scale).



-10	G'MIC for GIMP 1.5.8.5 (beta) - Lir	nux 64 bits		- • ×
	Available filters (646) :	Split details :		
	Local normalization	Action	Recompose	:
	Local variance normalizatio	Decompose options:	3	
	Mask creator	Number of scales		÷
No preview available	Portrait retouching	Base scale	5.00	÷
	Sharpen [gold-meinel]	Details scale		-
e e a	Sharpen [inverse diffusion] Sharpen [octave sharpening	Output mode	Tiling	:
	Sharpen [richardson-lucy]	Preview type	All scales	-
Preview Input / Output : All Output mode Output messages Output preview	 Sharpen [shock filters] Sharpen [unsharp mask] Sharpen [whiten] Simple local contrast Split details Spotify Texture enhance p Tone mapping [fast] Yag effect Yag effect 	Instructions of use: 1. Set parameter Action parameters. Then clicit 2. Process your image scales using the neutr. 3. Use CTRL+FCo apply resulting recomposed canvas. 4. Repeat steps 2. and 5. When you are happ again, set parameter / 6. Get your final mod Author: Jkrome Boula Latest update: 2013/1 X Cancel	ion to Decompose/update and choose decompose k KC. For instance, remove details in the appropriate al color (128, 128, 128). y the filter again (update mode). The preview of image updated and shown at the top left of you d 3. as many times as you want. y with the result, open the GYMIC plug-in once <i>Action to Recomposes</i> , and click OK. fired image alone. ngper and David Tschumperlé. 2/14.	e the ir

Invoke G'MIC plug-in again, to recompose the final image.





Result of the recomposition, with cleaner skin (5mn work !).





Comparison with initial image.



Filter Showcase: Extract objects



- Goal: Extract independent objects located on a flat colored background.
- Made by : David, to ease the use of the next filter Pack Sprite.
- How is this done? Background pixels are extracted (by their color), then the residual pixels are grouped into several connected regions corresponding to the objects to extract.
- \implies **72** lines of G'MIC code.

(all included: GUI description + algorithm).





Open input image (single-layer).



·••	'MIC for GIMP 1.5.8.5 (beta)) - Li	nux 64 bits		-	• *
19 dijala	Available filters (646) :		Extract objects :			
	 About 	11	X-background	0	0.00	*
	 Arrays & tiles 		Y-background		0.00	
	Array [faded]			~		
	Array [mirrored]		Color tolerance		20	÷
	Array [random]		Opacity threshold	0	50	* *
	Array [random colors]		Minimal area	Сгор		
	Array [regular]			Segmentation		
	Dices		Connectivity	Located crop		
	Extract objects	40	Output type	Located segmentation		
	. Grid	١.			3	=1
	Puzzle		Author: David Tschump	perlé. Latest update: 2014/02/14.		
	Taquin			Filter explained here		
	Tileable rotation					
Input / Output :	Tiled isolation					
Input layers ‡	Tiled normalization					
	Tiled parameterization					
Output mode ‡	Tiled random shifts					
Output messages ‡	Tiled rotation					
	Black & white					
Output preview ‡	Colors					
	+ C 🗹 Internet 🕑					
Tensicaen?						
			× Cancel 2 Rese	et Maximize 🗸 Apply	~0	к

Invoke G'MIC plug-in and select Arrays & tiles / Extract object.





Output looks similar as input, but is divided into several layers.





Managing each object independently is now possible (here, position change).



Filter Showcase:

Pack sprites



- **Goal:** Render an image where several small images have been packed together (scaled and rotated) without intersecting.
- Made by : David, for Lyle Kroll on GimpChat who has asked this for a long time.
- How is this done? Pseudo random positions (random + heuristic) are iteratively tried to pack images, with decreasing scales.
- \implies **122** lines of G'MIC code.

(all included: GUI description + algorithm).





Select your objects to pack (multi-layer image).



-9	G'MIC for GIMP 1.5	5.8.5 (beta) - Linux 64 bits _ = *
No preview	Available filters (646) : · Lignts & snadows · Patterns Baver filter	Pack sprites : Number of scales Minimal scale [%] 10.90
available	Box fitting Canvas Canvas texture	Allowangle Any C
Preview	Clouds Cracks Halftone	Precision 7 C
:	Hartone snapes Hearts Lava Marble	; Width (2048 2) Height (2048 2)
Input / Output : All ‡	Maze Mineral mosaic Mosaic	voces: - Parameters Width and Height are considered only when No masking mode is selected. - Set different spirts on different layers to pack multiple sprites at the same time.
Output mode \$ Output messages \$	Op art Pack sprites Paper texture	Author: David Tschumperlé. Latest update: 2013/06/24.
Output preview	+ C Internet	Þ
		X Cancel Reset Maximize Apply VOK

Invoke G'MIC plug-in and select Patterns / Pack sprites.





Get your image with randomly packed sprites (after a while).





Now, you can add a bottom layer to restrict packing on transparent regions.



-100	G'MIC for GIMP 1.5	.8.5 (beta) - Linux	64 bits	-	• ×
	C'MIC for GIMP 1.5 Available filters (646) :	 B.5 (beta) - Linux Pack sprites : Number of scales Minimal scale [%] Allow angle Spacing Precision Masking Width Height Notes: - Param selected, - Set different sprites the same time. Author: David Tschumg 	64 bits	- 10 10.90 -1 7 2048 1280	
Output preview	Plaid Plaima + C Internet	>	Cancel 20Reset Maximize Appl	у 🗸 ок	:

Invoke G'MIC again, and select Mask : Mask as bottom layer.





Go for a coffee, and you get this.





Detail of the result.





Example of rendering, by Chris Fiedler, on GimpChat.



Filter Showcase:

Shapeism



- **Goal:** Try to get close to the Circlism effect from artist Ben Heine (who do this manually, takes days), i.e. render an image with non-intersecting colored circles (or other shapes).
- Made by : David, for Lyle Kroll on GimpChat who has asked this for a long time.
- How is this done? Multi-scale monochrome shapes are packed together with a priority to put smaller shapes on image contours, then each shape is colored separately according to the corresponding image color behind.

⇒ 75 lines of G'MIC code. (all included: GUI description + algorithm).





Open input image.



. . .	G'MIC for GIMP 1.5.8.5	(beta) - Liı	nux 64 bits			- • ×
	Available filters (646) Make squiggly	:	Shapeism : Shape	Circles		;
	Painting	ng	Branches	-0-	3	•
	Pastell art Pen drawing		Thickness		1.00	¢
	Phoenix steam-per	ncil	Angle		29.14	÷
	Photoillustration Polygonize		Note: Parameter are used only for	s Branches, Thickness and Angle Custom shapes.		
	Posterized ditherir	ng	🗹 Antialiasir	ng		
	Poster edges	vork tile	Scales		8	:
Preview	□		Maximal size		64	÷
Input / Output :	Shapeism		Minimal size	-	6	1
Input layers ‡	Simple noise canva Warhol	12	Allow angle	0 deg.		\$
Output mode ‡	Watercolor Whirl drawing		Spacing		-1	:
Output messages ‡	Wild cartoonizer		Precision		9	\$
Output preview ‡	Black & white	_	Edges		0.50	:
	+ C 🗹 Internet	•	Smoothness		1.00	•
		:	× Cancel 🚦	Reset Maximize Appl	y 🔽	ок

Invoke G'MIC plug-in and select Artistic / Shapeism.





Go drink a (big) coffee, and enjoy the result ! (can be slow to compute).





Result with another shape selected (a star).



Filter Showcase: Inpainting [patch-based]



- Goal: Automatically heal missing image regions with a texture-aware algorithm. Similar to what the Resynthetizer plug-in does, but directly in G'MIC.
- Made by : David and Maxime, to have an alternate healing method in an active project (Resynthetizer looks stagnant).
- How is this done? It implements the Criminisi-Perez-etal's patch-based inpainting algorithm + a patch blending technique we've specifically designed.
- ⇒ 427 lines of C++ code (native command) + 35 lines of G'MIC code. (all included: GUI description + algorithm).

Repair : Inpainting





Open input image.

Repair : Inpainting





Draw an inpainting mask directly on it (with a constant known color).


ح – ح م الم G'MIC for GIMP 1.5.8.5 (beta) - Linux 64 bits –				
	Available filters (646) :	Inpaint [patch-based] :		
	Banding denoise P Bayer reconstruction L Deinterlace L Despeckle L	Patch size		
Vienner Ca		Lookup size 16.00 🗘		
No preview		Lookup factor		
avaliable				
	lain's rast denoise p	Blend size		
	Inpaint [diffusion]	Blend threshold 0.00 🗘		
	Inpaint [patch-based]	Blend decay 0.05 ‡		
Preview 📃 🕀 🗄	Inpaint [solidify]	Blend scales 10		
logut (Output)	Local similarity mask			
Input / Output :	Pixel denoise	Allow outer blending		
Input layers ‡	Recursive median	Mask type Mask by color *		
Output mode	Red-eye attenuation	If Mask by color back color back color		
Couple inc com	Remove hot pixels	IF Mask by color has been selected :		
Output messages ‡	Repair scanned document	Mask color		
Output preview ‡	Smooth [hilstors]	A quick tutorial on how to use this filter can be found here:		
	+ C 🗹 Internet	<u>G'MIC Inpainting tutorial on Patrick David's blog</u>		
		X Cancel Reset Maximize Apply OK		

Invoke G'MIC plug-in and select Repair / Inpaint [patch-based].





If you choose carefully the parameters, this is what you get.





Example from Patrick David: Input image.





Example from Patrick David: Inpainted image.





Example from Patrick David: Input image.





Example from Patrick David: Inpainted image.





Comparison with Resynthetizer (Extreme case!): Input image (boat to be removed).





Result by the Resynthetizer heal selection algorithm.





Result by the G'MIC inpainting algorithm.



Filter Showcase: Denoising filters



- **Goal:** Provide a lot of algorithms to smooth images without losing (too much) the details and the textures, e.g. to remove shot noise.
- Made by : David, Jérome, lain and others.
- Made for : A lot of people need this. This is the logical sequel of our previous plug-in called GREYCstoration (now discontinued).
- How is this done? Lot of different smoothing algorithms have been implemented in G'MIC: Diffusion PDE's, NL-means, Wavelets-based, etc... In 2013, we have parallelized most of them.





Open input (noisy) image.





Open input (noiy) image (detail).



د م G'MiC for GIMP 1.5.8.5 (beta) - Linux 64 bits ـ ٥ ×						
Before	Available filters (646) :	Smooth [anisotropic] :				
	r Renair	Amplitude	-0[200.00	\$	
	Banding denoise	Sharpness		0.70	2	
	Bayer reconstruction	Anicotrony		0.20		
	Deinterlace	Anisocropy	-	0.30	-	
	Despeckle	Gradient smoothness	-Q	0.60	2	
	lain's fast denoise p	Tensor smoothness	-0[1.10	2	
and the second	lain's noise reduction	Spatial precision		0.80	0	
	Inpaint [patch-based]	Angular precision		30.00		
	Inpaint [solidify]	Anguar precision		50.00	-	
	Local similarity mask	Value precision	(2.00		
	Pixel denoise	Interpolation	Nearest neighbor		:	
	Recursive median	Fast approximation				
	Red-eye attenuation	Iterations		2		
After	Remove not pixels	iceracions	Q (2	-	
	Smooth [anisotropic]	Channel(s)	All		\$	
Serview	Smooth [bilateral]	Parallal processing	Auto			
	Smooth [block pca]	Faratterprocessing	Auto		-	
	Smooth [diffusion]	Spatial overlap	-0	24	2	
	Smooth [iuwt]	Preview type	Duplicate vertical		-	
	Smooth [mean-curvature]	Freview type	Dopirate vertical		-	
	Smooth [natch-based]	Author: David Tschumperle	Latest update: 201	3/08/27.		
Input / Output :	Smooth [selective gaussian]					
Input lavers	Smooth [skin]	2				
	Smooth [thin brush]					
Output mode 🗘	Smooth [total variation]					
Output messages 0	Smooth [wavelets]					
Output provinu	Smooth [wiener]					
Output preview	onscrip					
	+ C 🗹 Internet 💿					
	X Cancel	Decot Maximi				
	× cancer	Maximi	e Abbia	* UK		

Invoke G'MIC plug-in, and select one of the denoising filters (more than 20 methods available).





Denoised result (with heavy parameters for making the effect more clear).





Comparison between original / denoised image (equalized images for clarity).



Filter Showcase:

Dream smoothing



- **Goal:** Apply exaggerated edge-directed smoothing and boost colors to create a kind of painting effect.
- Made by : Arto Huotari (aka Naggobot), artist and coder at the same time, who uses it on his own photographic workflow.
- How is this done? It intensively uses anisotropic smoothing (native G'MIC feature) as well as aggressive color mixing.
- \implies **76** lines of G'MIC code.

(all included: GUI description + algorithm).





Open input image.



	or GIMP 1.5.8.5 (beta) - Linux 64 i	bits _
✓ Preview Input / Output : Input layers Output mode Output preview Output preview	Available filters (646) : About Arrays & tiles Arrays & tiles Aurora Black crayon graffiti Blockism Bokeh Caltoon Colored pencils Color abstraction paint Color sketch Color sketch Color sketch Color stamp Cubism Dream smoothing Ellipsionism Felt pen Graphic boost 2 Graphic boost 2 Graphic novel Hope poster Kuwahara Lylejk's painting + © @ Internet @	softer image as does lower opacity values. Practical modes for merging are Alpha and Average. Note that results are resolution dependent. Ceneral settings Iterations

Invoke G'MIC plug-in and select Artistic / Dream Smoothing.





Enjoy your result ! (takes some time to render, recently parallelized).





How artists use it for real: Processing done by Zarir Madon.





How artists use it for real: Processing done by Arto Huotari.



Filter Showcase:

Film emulation



- Goal: Provide free film emulation filters, similar to what proprietary DXO FilmPack proposes.
- Made by : Patrick requested David to make his color profiles easily available for everyone.
- How is this done? Color transformations are encoded as RGB CLUT files, stored on the G'MIC server. Each color profile is downloaded on demand.
- ⇒ 476 lines of G'MIC code (mostly for GUI). (all included: GUI description + algorithm).





Open input image.



.@	G'MIC for GIMP 1.5.8.5 (beta)	ı) - Linux 64 bits	- • ×		
Before Contraction	G'MIC for GIMP 1.5.8.5 (beta) - Available filters (646) : About Arrays & tiles Artistic Black & white Colors Contours	- Linux 64 bits Negative [new]: Preset Kodak Portra 400 Effect High Gamma Contrast 1.00			
Preview □ ♥ Input / Output : Input layers ♥	 Deformations Degradations Details Film emulation B&W films Grain Instant [consumer] Instant [pro] Negative [color] Negative [new] Other effects 	Brightness Hue Saturation Post-normalize Preview type Forward vertical Authors: Patrick David and David Tschumperlé. Latest update: 2013/08/29.	- 0.00 \$ 0.00 \$ - 1.00 \$		
Output mode \$ Output messages \$ Output preview \$	Silde [color] • [collages] [download all presets] • Frames + @ @ Internet @	X Cancel €Reset Maximize ✓ Apply ✓ OF			

Invoke G'MIC plug-in, and choose one filter in folder Film emulation/.





Comparison: Before (left) / After (right).





Two other examples: TMAX-3200 (left) and Kodak Kodachrome 64 (right).





Patrick David has indeed done a lot of presets (here, a sample of them).





Technically speaking:

- Each preset defines a mapping function from RGB to RGB (CLUT).
- The values of these functions are explicitly stored for all RGB colors.
- To avoid huge datasets, we consider 64x64x64 downsampled versions of the CLUTs and interpolate intermediate colors.
- → 77Mb of data for 271 film emulation presets.
- As the original color mappings are smooth functions, interpolation has almost no incidence on the quality.



.gnic_film_presets _ = = x					
File Edit Go	Bookma	rks View Tools Help			
,+, < ▼ >	^ /ho	me/dtschump/.gmic_film_presets			\$
Places	*	gmic_rgb.cimgz	gmic_fuji_provia_400f.cimgz	gmic_kodak_portra_400_vc.cimgz	gmic_fuj
dtschump 国 Desktop		gmic_polaroid_polachrome.cimgz	gmic_fuji_provia_100f.cimgz	gmic_kodak_portra_400_vccimgz	gmic_fuj
🛗 Trash Can		gmic_lomography_x-pro_slide_200.cimgz	gmic_fuji_fp_100c.cimgz	gmic_kodak_portra_400_uc_++.cimgz	gmic_fuj
Applications		gmic_kodak_kodachrome_64.cimgz	gmic_fuji_astia_100f.cimgz	gmic_kodak_portra_400_uc_+.cimgz	gmic_fuj
dtschump donnees	 ≜	gmic_kodak_kodachrome_25.cimgz	gmic_agfa_precisa_100.cimgz	gmic_kodak_portra_400_uc.cimgz	gmic_fuj
gmic_frs		gmic_kodak_kodachrome_200.cimgz	gmic_warm_yellow.cimgz	gmic_kodak_portra_400_uccimgz	gmic_fuj
gmic gmicol	≜	gmic_kodak_elite_extracolor_100.cimgz	gmic_warm_spring.cimgz	gmic_kodak_portra_400_nc_++.cimgz	gmic_fuj
i work	_	gmic_kodak_elite_chrome_400.cimgz	gmic_warm.cimgz	gmic_kodak_portra_400_nc_+.cimgz	gmic_fuj
🕅 src		gmic_kodak_elite_chrome_200.cimgz	gmic_vintage_2.cimgz	gmic_kodak_portra_400_nc.cimgz	gmic_fuj
img Filatov		gmic_kodak_ektachrome_100_vs.cimgz	gmic_vintage.cimgz	gmic_kodak_portra_400_nccimgz	gmic_fuj
Pictures	:	gmic_kodak_e-100_gx_ektachrome_100.cimgz	gmic_sutrofx.cimgz	gmic_kodak_portra_160_vc_++.cimgz	gmic_fuj
		gmic_generic_kodak_ektachrome_100_vs.cimgz	gmic_retro.cimgz	gmic_kodak_portra_160_vc_+.cimgz	gmic_fuj
		gmic_generic_kodachrome_64.cimgz	gmic_nostalgic.cimgz	gmic_kodak_portra_160_vccimgz	gmic_fuj
		gmic_generic_fuji_velvia_100.cimgz	gmic_lomo.cimgz	gmic_kodak_portra_160_nc_++.cimgz	gmic_fuj
		gmic_generic_fuji_provia_100.cimgz	gmic_hkfilm.cimgz	gmic_kodak_portra_160_nc_+.cimgz	gmic_fuj
		gmic_generic_fuji_astia_100.cimgz	gmic_60s_3.cimgz	gmic_kodak_portra_160_nccimgz	gmic_fuj
		gmic_fuji_velvia_50.cimgz	gmic_60s_2.cimgz	gmic_fuji_superia_1600_++.cimgz	gmic_fuj
		gmic_fuji_superia_200_xpro.cimgz	gmic_60s.cimgz	gmic_fuji_superia_1600_+.cimgz	gmic_fuj
		gmic_fuji_sensia_100.cimgz	gmic_kodak_portra_400_vc_++.cimgz	gmic_fuji_superia_1600.cimgz	gmic_fuj
		gmic_fuji_provia_400x.cimgz	gmic_kodak_portra_400_vc_+.cimgz	gmic_fuji_superia_1600cimgz	gmic_ko
272 items				Free space: 384,2 GiB (To	ital: 426,9 GiB)

Once downloaded, presets are stored locally on your drive for off-line use.



Conclusions

Conclusions & The end



- G'MIC is really meant to be a generic image processing framework.
- All filters we regularly can be potentially available for all interfaces or open-source projects that integrates the G'MIC library.
- Since the beginning, lot of filters have been done in collaboration with artists. Lot of good ideas come from users.

Thanks for your attention!

Any questions ?

