G`MIC ma już 10 lat!

David Tschumperlé
https://discuss.pixls.us/t/help-writing-an-article-for-the-10-years-of-gmic/8698
Niedawno napisałem artykuł (po francusku), aby świętować 10. urodziny G'MIC, podsumowując główne nowe materiały dodane do projektu przez ostatnie 6 miesięcy. Ten artykuł został opublikowany, na LinuxFR:
https://linuxfr.org/news/g-mic-2-3-4-traiter-ses-images-en-se-disant-deja-10-ans
bardzo dobrze znanej francuskiej stronie z wiadomościami o świecie wolnego oprogramowania (nie koncentrującym się szczególnie na fotografii).

29 sierpnia 2018 ukazało się Tłumaczenie na język ang.

G'MIC 2.3.6:
10 years of open-source image processing
https://pixls.us/blog/2018/08/g-mic-2-3-6/

Poniżej treść artykułu tłumaczona na j. polski:

Zespół IMAGE https://linuxfr.org/users/dtschump/journaux/inrcast-un-autre-outil-de-manipulation-dimages z GREYC https://www.greyc.fr/ chętnie świętuje z wami dziesięć lat oprogramowania *G'MIC* w ramach wolnych oprogramowań (na licencji CeCILL), rodzajowe i rozszerzalny do przetwarzania obrazu https://fr.wikipedia.org/wiki/Traitement_d%27images.
GREYC jest publicznym laboratorium badawczym w dziedzinie nauk cyfrowych, zlokalizowanym w Caen w Normandii, nadzorowanym przez trzech powierników:
CNRS http://www.cnrs.fr/ UMR 6072),
University of Caen http://www.unicaen.fr/ i
ENSICAEN http://www.ensicaen.fr/

G'MIC-Qt, główny interfejs użytkownika darmowego projektu G'MIC.

Ta uroczystość daje nam doskonałą okazję, aby ogłosić wydanie nowej wersji (**2.3.6**) tego darmowego oprogramowania i podzielić się z *Państwem* podsumowaniem najnowszych zauważalnych zmian od czasu naszego ostatniego raportu *G'MIC*, opublikowanego na stronie *PIXLS.US* w Lutym 2018 https://pixls.us/blog/2018/02/g-mic-2-2/.

- Projekt G'MIC https://gmic.eu/ (467 kliknięcia)
- Kanał na Twitterze https://twitter.com/gmic_ip (30 kliknięć)
- Ogłoszenie pierwszej wersji G'MIC na LinuxFr.org https://linuxfr.org/news/gmic-un-nouvel-outil-libre-de-manipulation-dimages (67 kliknięć)
- Poprzedni artykuł na temat G'MIC na PIXLS.US https://pixls.us/blog/2018/02/g-mic-2-2/ (70 kliknięć)

1. Patrząc wstecz na 10 lat rozwoju

G'MIC to program wieloplatformowy (GNU / Linux, MacOS, Windows ...) zapewniając różne interfejsy użytkownika do manipulowania *ogólnymi* danymi obrazu, takimi jak obrazy hiperspektralne 2D lub 3D lub sekwencje obrazów z wartościami zmiennoprzecinkowymi (włączając w to "normalne" kolorowe obrazy).
Dostępnych jest ponad 1000 różnych operatorów http://gmic.eu/reference.shtml do przetwarzania obrazu, których liczba jest nieograniczana, ponieważ użytkownicy mogą dodawać własne funkcje za pomocą wbudowanego języka skryptowego.
Pod koniec lipca 2008 r. Powstały pierwsze wiersze kodu *G'MIC* (w *C* ++).
W tym czasie byłem głównym programistą zajmującym się w *CiMG*, biblioteką *C* ++ o *otwartym kodzie źródłowym* do przetwarzania obrazów, kiedy zrobiłem następujące obserwację:

- Początkowy cel *CImg* , który miał zapewnić "minimalną" bibliotekę funkcji, aby pomóc programistom *C* ++ *w* opracowaniu algorytmów przetwarzania obrazu, został szeroko osiągnięty: Większość algorytmów uważałem za "niezbędne" w przetwarzanie obrazu zostało zintegrowane. *Początkowo CImg* miał pozostać lekki, więc nie chciałem *dodawać* nowych algorytmów *ad vitam ameutam*, które byłyby zbyt ciężkie lub zbyt szczegółowe i zdradzałyby początkową koncepcję biblioteki.

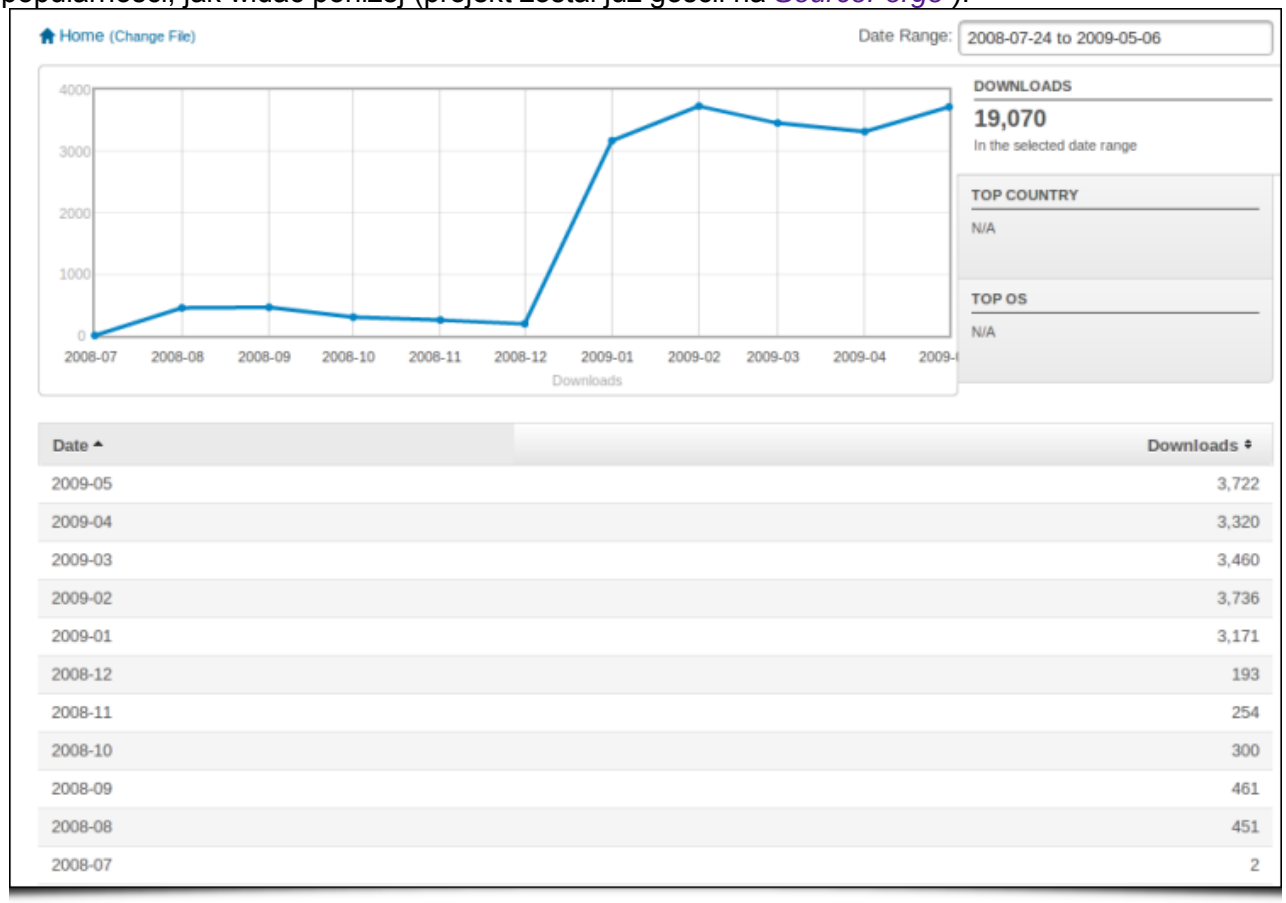
- To jednak zaspokajałoby raczej małą społeczność ludzi posiadających wiedzę w zakresie *C* ++ **i** przetwarzania obrazów! Jednym z naturalnych ewolucji projektu, tworząc *powiązania* z *CiMG* do innych języków programowania, nie przemawiał do mnie zbytnio, biorąc pod uwagę brak zainteresowania pisaniem tego kodu.
  Te potencjalne *powiązania* nadal dotyczyły jedynie odbiorców posiadających pewne doświadczenie programistyczne.

Moje pomysły stopniowo *nabrały* kształtu: konieczne było zaproponowanie sposobu wykorzystania *funkcji* przetwarzania obrazu *CImg* dla **nie-programistów**. A dlaczego nie na początku, poprzez opracowanie narzędzia użytkowej wiersza komend (w słynny sposób konwersji https://www.imagemagick.org/script/convert.php z *ImageMagick*)? Pierwsza próba w czerwcu 2008 roku (*inrcast* , który został przedstawiony w czasopiśmie *LinuxFr* https://linuxfr.org/users/dtschump/journaux/inrcast-un-autre-outil-de-manipulation-dimages), okazały się nieskuteczne, ale pozwolił mi lepiej zrozumieć specyfikę, co jest wymagane dla tego typu narzędzia, aby łatwo przetwarzać obrazy z wiersza poleceń.
W szczególności przyszło mi do głowy, że **zwięzłość** i **spójność** składni polecenia są dwiema najważniejszymi rzeczami, na których można się oprzeć. Były to aspekty, które wymagały największego wysiłku w zakresie badań i rozwoju (*rzeczywiste* funkcje przetwarzania obrazu są już zaimplementowane w *CImg*). Ostatecznie skupienie się na zwięzłości i spójności zabrało mi znacznie więcej niż pierwotnie planowano, ponieważ *G'MIC* ma interpreter https://fr.wikipedia.org/wiki/Interpr%C3%A8te_(informatique) z własnym językiem skryptowym https://gmic.eu/tutorial/basics.shtml
i kompilator *JIT* https://fr.wikipedia.org/wiki/Compilation_%C3%A0_la_vol%C3%A9e dla oceny wyrażeń matematycznych i algorytmy przetwarzania obrazu działające na poziomie pikseli.
Dzięki tym pomysłom, pod koniec lipca 2008 r. Z przyjemnością ogłosiłem pierwszą wersję G'MIC https://linuxfr.org/news/gmic-un-nouvel-outil-libre-de-manipulation-dimages. Projekt został oficjalnie uruchomiony!

Rys. 1.1: Logo projektu G'MIC, Free Frameworks do obróbki obrazu i jego urocza maskotka "Gmicky" (ilustrowana przez Davida Revoy'a).

Kilka miesięcy później, w styczniu 2009 roku, wzbogacony przez moje poprzednie doświadczenia w rozwoju oprogramowania *GREYCstoration* http://cimg.eu/greycstoration (darmowe narzędzie do usuwania szumu i nieliniowej interpolacji obrazu, z którego stworzono wtyczkę *GIMP*), w nadziei, aby dotrzeć do jeszcze większej publiczności, wydałem wersję *G'MIC, która* ma postać wtyczki *GTK* dla *GIMP* https://linuxfr.org/news/traitement-dimages-quand-gmic-130-sinvite-dans-gimp.

Ten krok okazał się być definiującym momentem dla projektu G'MIC, dając mu znaczący wzrost popularności, jak widać poniżej (projekt został już gościł na *SourceForge*).



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

Rys.1.2: Miesięczne statystyki pobierania G'MIC, od lipca 2008 r. Do maja 2009 r. (Nadejście wtyczki do GIMP w styczniu 2009 r.).

Nagłe zainteresowanie wtyczką różnych użytkowników GIMP (fotografów, ilustratorów i innych twórców) było rzeczywiście prawdziwym starterem dla projektu, z szybkim pojawieniem się różnych wkładów i zewnętrznych sugestii (dla kodu, zarządzania forami, strony internetowe, pisanie samouczków i realizacja filmów itp.). Często wyidealizowany efekt społecznościowy wolnego oprogramowania wreszcie zaczął startować! Użytkownicy i programiści zaczęli bliżej przyglądać się działaniu oryginalnego interfejsu wiersza poleceń i związanego z nim języka skryptowego (co do tej pory nie zainteresowało wielu osób!). Stamtąd wielu z nich podjęło decyzję

o wprowadzeniu nowych filtrów przetwarzania obrazu w języku G'MIC https://github.com/dtschump/gmic-community, stale integrując je z wtyczką GIMP. Dziś te składki stanowią prawie połowę filtrów dostępnych w wtyczce.
Równocześnie znaczne i wielokrotne wkłady *Sébastien Fourey* https://foureys.users.greyc.fr/Fr/index.php, współpracownika zespołu *IMAGE* z *GREYC* (i doświadczonego programisty *C* ++) znacznie poprawiły łatwość użycia *G'MIC*. *Sébastien* jest rzeczywiście podstawą rozwoju głównych graficznych interfejsów projektu, a mianowicie:

- Serwis *internetowy G'MIC Online* https://gmicol.greyc.fr/ (który został później ponownie zorganizowany przez *dział* rozwoju *GREYC*).
- Wolne oprogramowanie *ZArt* https://github.com/c-koi/zart, interfejs graficzny, oparty na bibliotece *Qt* https://www.qt.io/, do zastosowania filtrów *G'MIC* w sekwencjach wideo (z plików lub strumieni kamer cyfrowych).
- A przede wszystkim *Sébastien* pod koniec 2016 roku podjął się kompletnego przepisania wtyczki *G'MIC* dla *GIMP* w bardziej **ogólnej** formie, zwanej *G'MIC-Qt* https://github.com/c-koi/gmic-qt. Ten komponent, oparty również na bibliotece *Qt* (jak sugeruje jej nazwa), jest pojedynczą wtyczką, która działa równie dobrze w *GIMP* i *Kricie* , dwóch wiodących bezpłatnych programach do retuszowania i edycji zdjęć. malarstwo cyfrowe. *G'MIC-Qt* całkowicie wyparł wtyczkę *GTK*
  Oryginalny dzięki wielu funkcjom: wbudowanej wyszukiwarce filtrów, lepszemu podglądowi, wyższej interaktywności itp. Dziś jest to najbardziej zaawansowany interfejs i najczęściej wykorzystywany projekt *G'MIC*, i mamy nadzieję, że będziemy mogli podrzucić go w przyszłości dla innych hostów oprogramowania (skontaktuj się z nami, jeśli jesteś zainteresowany tym temat!).

Rys.1.3: Różne interfejsy graficzne projektu G'MIC, opracowane przez Sébastiena Fourey: G'MIC-Qt, G'MIC Online i ZArt.

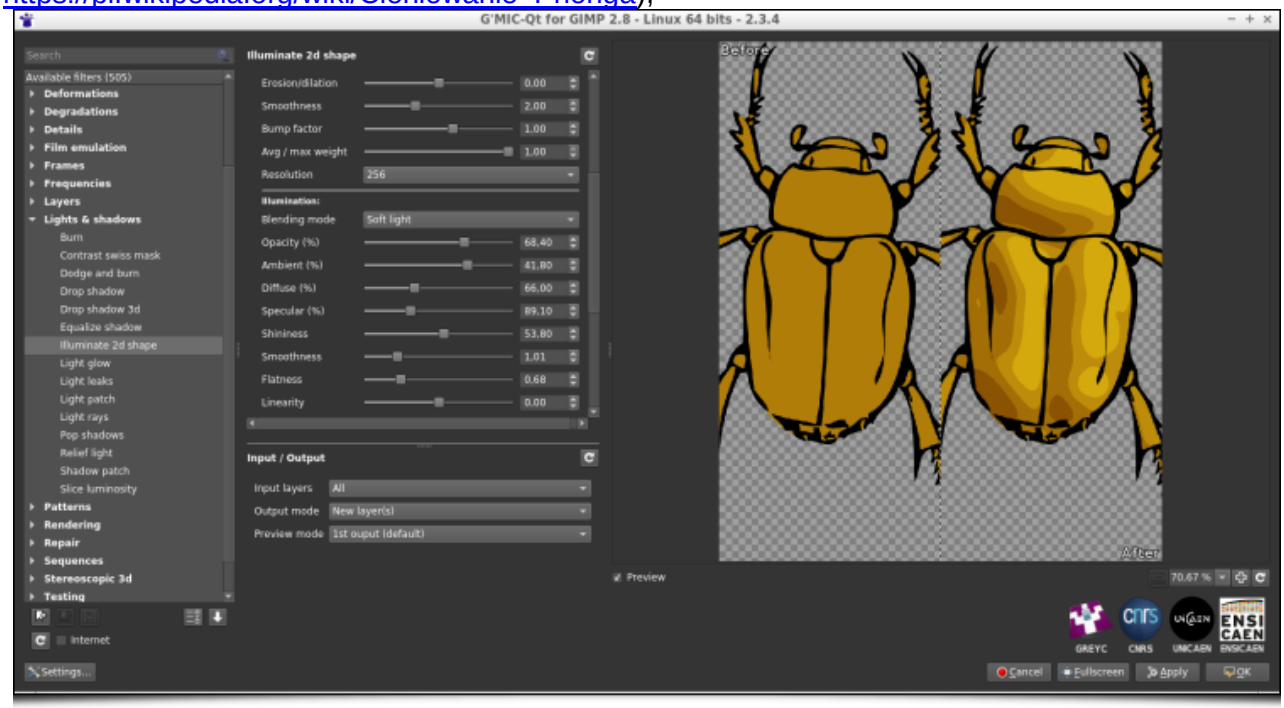
Ideą tej ekspedycji nie jest zbyt szczegółowe zapoznawanie się z historią projektu, wystarczy powiedzieć, że nie mieliśmy naprawdę czasu na nudę w ciągu ostatnich dziesięciu lat!
Dzisiaj, *Sébastien* i ja jesteśmy dwoma głównymi opiekunami projektu *G'MIC* (*Sébastien* , głównie z punktu widzenia interfejsu i ja sam dla rozwoju i ulepszania filtrów i serca obliczenia), oprócz naszej głównej działalności zawodowej (badania i nauczanie / coaching).
Spójrzmy prawdzie w oczy, zarządzanie wolnym projektem, takim jak *G'MIC,* zajmuje sporo czasu, pomimo jego skromnego rozmiaru (*kl 120klocs*).
Ale pierwotny cel został osiągnięty: Tysiące nie-programistów ma możliwość swobodnego i łatwego korzystania z naszych algorytmów przetwarzania obrazu w wielu różnych obszarach: retuszu zdjęć https://fr.wikipedia.org/wiki/Retouche_d%27image, ilustracji i malarstwo cyfrowe https://fr.wikipedia.org/wiki/Peinture_num%C3%A9rique, przetwarzanie wideo https://fr.wikipedia.org/wiki/Traitement_de_la_vid%C3%A9o, ilustracja naukowa, generowanie proceduralne https://fr.wikipedia.org/wiki/G%C3%A9n%C3%A9ration_proc%C3%A9durale, *glitch art* https://fr.wikipedia.org/wiki/Glitch_art ....
Pasek z *3,5 mln Liczba pobrań* przekroczył w ubiegłym roku, z obecnej średniej około *400* codzienne pobieranie z oficjalnej strony internetowej (liczby stale spadały w ostatnich latach, ponieważ *G'MIC* jest coraz częściej pobierane i instalowane za pośrednictwem zewnętrznych źródeł).
Czasami trudno jest utrzymać stałe tempo rozwoju i motywację, która mu towarzyszy, ale trzymamy się, myśląc z powrotem do szczęśliwych użytkowników, którzy od czasu do czasu dzielą się swoim entuzjazmem dla tego projektu!

My oczywiście nie potrafi wymienić wszystkie osoby, autorami *G'MIC* chcielibyśmy podziękować, iz którymi ucztował wymianę podczas tych dziesięciu lat, ale serce jest! Również podziękować laboratorium *GREYC* oraz Instytut *INS2I CNRS* http://www.cnrs.fr/ins2i/, które wykazują silne poparcie dla tego bezpłatnego projektu. Wielkie podziękowania także dla zespołu *LinuxFr.org,* który nie miał nic przeciwko *ponownemu* przeczytaniu naszych regularnych propozycji wysyłek na *G'MIC* https://linuxfr.org/tags/gmic/public ;).
Ale przestańmy przywracać stare wspomnienia, a teraz przejdźmy do rzeczy: co nowego od ostatniej wersji głównej *2.2* !

2. Automatyczne podświetlanie płaskich rysunków

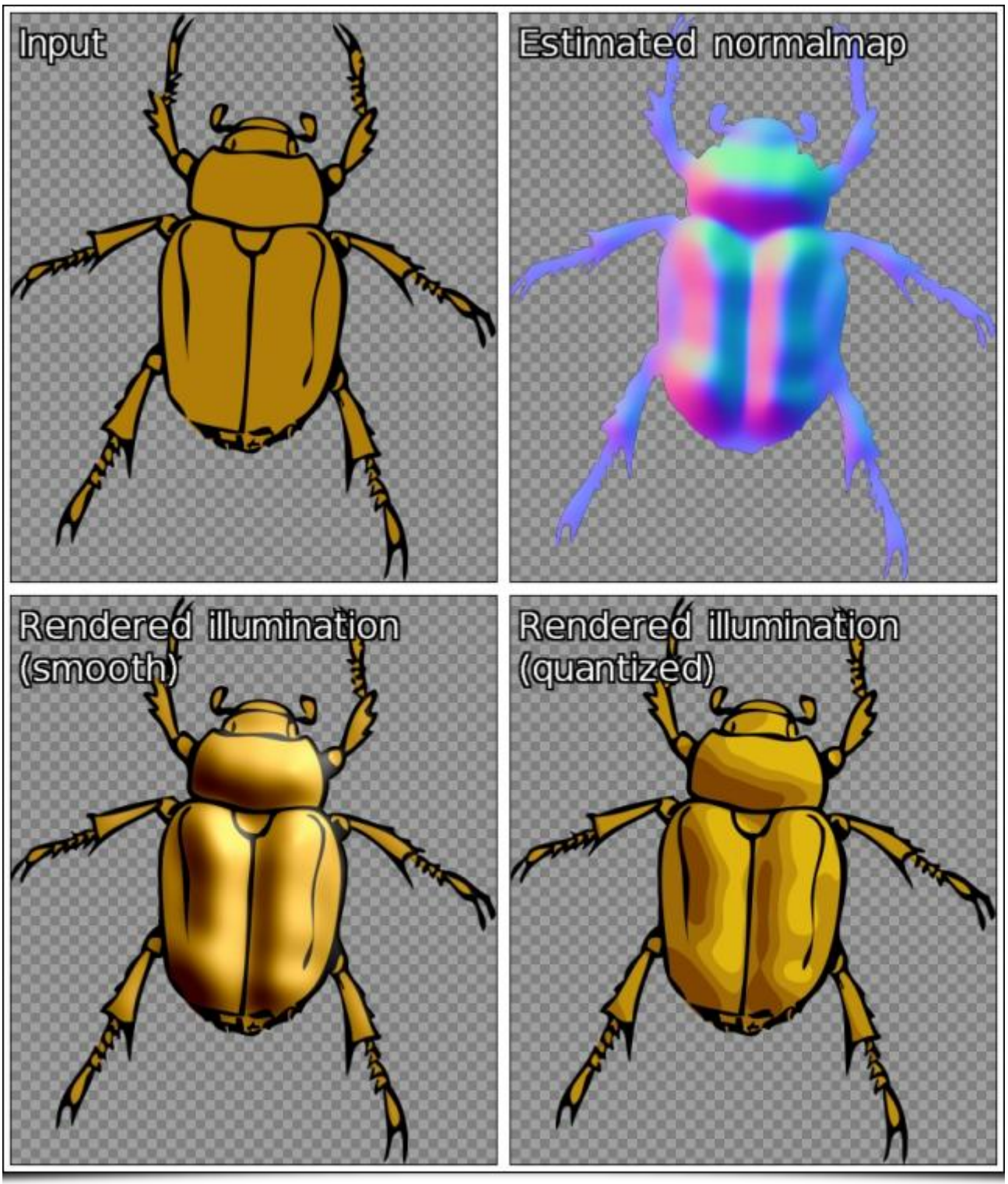
G'MIC niedawno zyskał całkiem nowy niesamowity filtr o nazwie "**Illuminate 2D Shape**", którego celem jest automatyczne dodawanie stref oświetlonych i czystych cieni na rysunkach 2D pokolorowanych w jednolitych kolorach https://fr.wikipedia.org/wiki/Aplat, aby nadać im wygląd 3D.

Początkowo użytkownik dostarcza obiekt do oświetlania, w postaci obrazu na przezroczystym tle (zazwyczaj rysunek postaci lub zwierzęcia). Analizując kształt i zawartość obrazu, G*'MIC* następnie próbuje wydedukować zgodną mapę wypukłości 3D ("*bumpmap*"). Uzyskana mapa wypukłości nie jest oczywiście dokładna, ponieważ rysunek 2D pokolorowany na obszarach bryłowych nie zawiera informacji wprost na temat powiązanej struktury 3D!
Na podstawie szacowanych rzędnych 3D łatwo jest wydedukować *normalną* mapę ("*normalmap*"), która jest używana z kolei do wygenerowania warstwy iluminacji związanej z rysunkiem (wg. modelu cieniowania Phonga https://fr.wikipedia.org/wiki/Ombrage_de_Phong https://pl.wikipedia.org/wiki/Cieniowanie_Phonga),



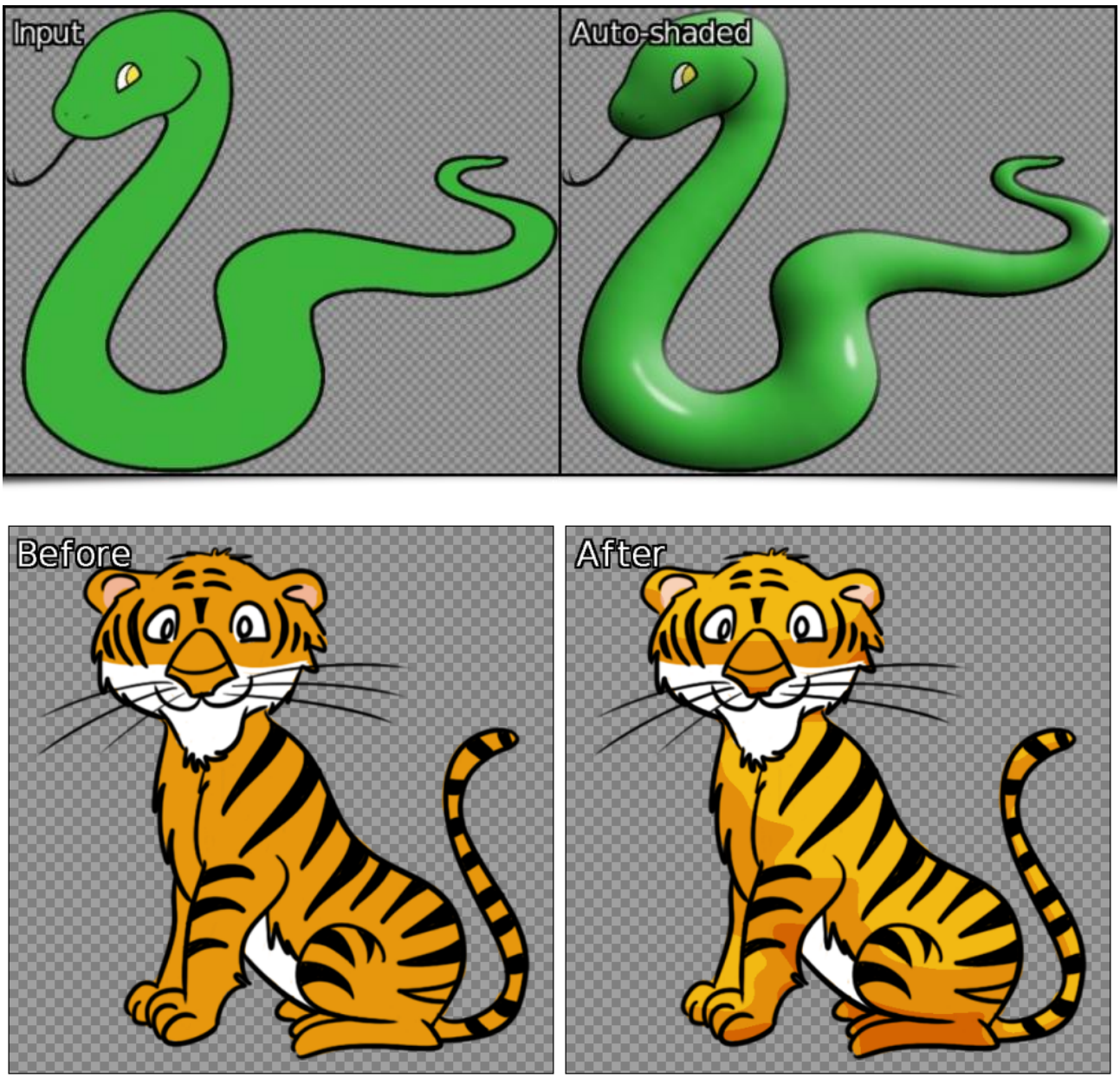
Rys. 2.1: Filtr G'MIC **'Illuminate 2D shape'** w akcji, demonstrujący automatyczne cieniowanie rysunku chrząszcza (zacieniowany wynik po prawej).

Ten nowy filtr jest bardzo elastyczny i pozwala użytkownikowi na dość precyzyjną kontrolę nad parametrami oświetlenia (typem renderowania pozycji i źródła światła), oraz oszacowanie elewacjach 3D. Dodatkowo filtr daje artystce możliwość przerobienia wygenerowanej warstwy iluminacji, a nawet bezpośrednio mapy wypukłości i szacunkowych wartości 3D. Poniższy rysunek ilustruje ten proces jako całość; za pomocą stałego kolorowego obrazu chrząszcza (u góry po lewej) filtr w pełni automatycznie szacuje powiązaną mapę normalną 3D (u góry po prawej). Umożliwia to generowanie interpretacji na podstawie rysunku (*dolny wiersz*, z dwoma różnymi stylami renderowania: gładkim i skwantyzowanym.



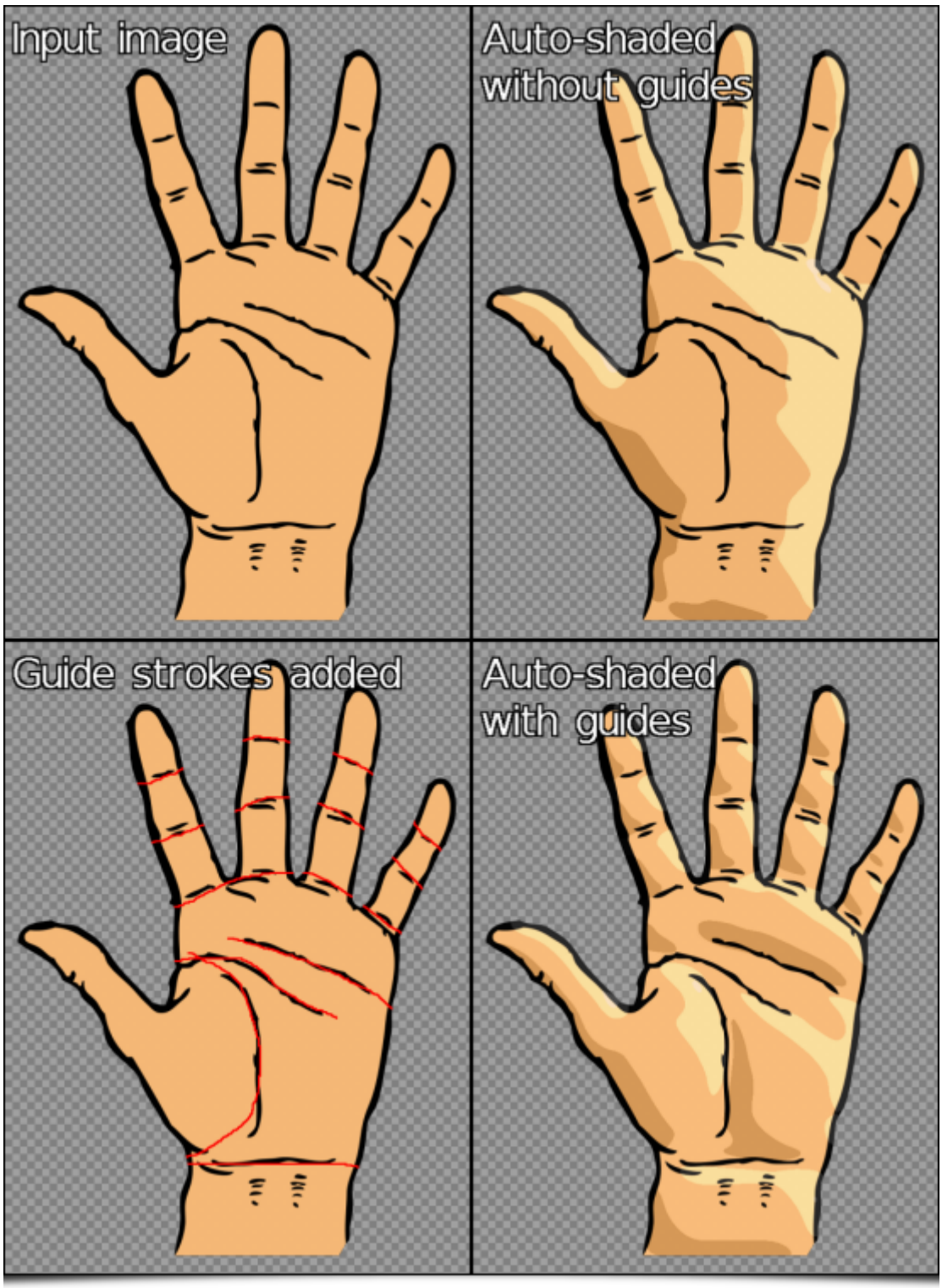
Rys. 2.2: Potok procesowy filtra G'MIC " ***Illuminate 2D Shape"*** obejmuje oszacowanie normalnej mapy 3D w celu wygenerowania automatycznego podświetlenia rysunku.

Pomimo trudnej sytuacji związanej z przekształcaniem obrazu 2D w informacje o wypukłości 3D, zastosowany algorytm jest zadziwiająco skuteczny w wielu przypadkach, przy czym oszacowana mapa wypukłości 3D jest wystarczająco spójna dla automatycznego generowania wiarygodnej iluminacji rysunku 2D, co ilustrują dwa poniższe przykłady, uzyskane za pomocą zaledwie kilku kliknięć!



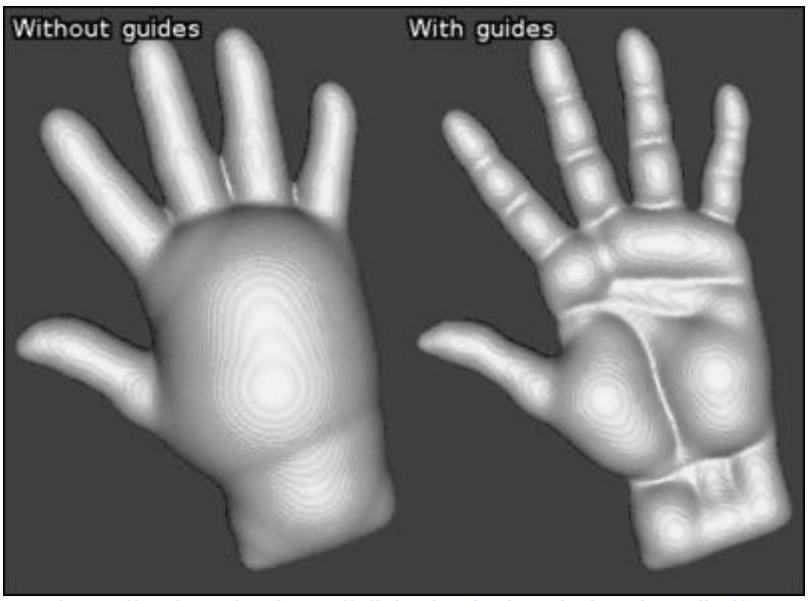
Rys. 2.3: Dwa przykłady całkowicie automatycznego cieniowania rysunków 2D generowanych przez G'MIC.

Zdarza się oczywiście, że szacowana mapa wypukłości 3D nie zawsze pasuje do tego, czego można chcieć. Nieważne, filtr umożliwia użytkownikowi dostarczenie "prowadnic", w postaci dodatkowej warstwy złożonej z kolorowych linii, dostarczających dokładniejszych informacji do algorytmu o strukturze rysunku, który ma być analizowany. Poniższy rysunek ilustruje przydatność tych prowadnic dla przykładu podświetlenia rysunku dłoni (u *góry po lewej*); oświetlenie uzyskane całkowicie automatycznie (u *góry po prawej*) nie uwzględnia informacji o liniach ręki. Włączenie tych kilka linii w dodatkowej warstwie "prowadnic" (na czerwono, u *dołu po lewej)*) pomaga algorytmowi w bardziej zadowalającym oświetleniu rysunku.



Rys. 2.4: Używanie warstwy "prowadnic" w celu poprawy automatycznego renderowania oświetlenia generowanego przez G'MIC.

Jeśli przeanalizujemy dokładniej różnice uzyskane między szacunkowymi mapami wypukłości 3D z i bez "prowadnic" (zilustrowanych poniżej jako symetryczne obiekty 3D), nie ma porównania: przechodzimy od bardzo okrągłej rękawicy bokserskiej do znacznie bardziej szczegółowej oceny ręki 3D!



https://gmic.eu/gmic234/fullsize/gmic_hand3d_anim_all.gif

Rys. 2.5: Oszacowane wzniesień rzędne 3D dla poprzedniego rysunku ręki, z i bez użycia "prowadnic".

Na koniec zauważmy, że ten filtr ma również **interaktywny** tryb podglądu, pozwalający użytkownikowi przesuwać źródło światła (za pomocą myszy) i wyświetlać podgląd rysunku w czasie rzeczywistym. Modyfikując parametry pozycji źródła światła, można w ten sposób uzyskać poniższy typ animacji w bardzo krótkim czasie, co daje dość dokładny obraz struktury 3D oszacowanej przez algorytm z oryginalnego rysunku.

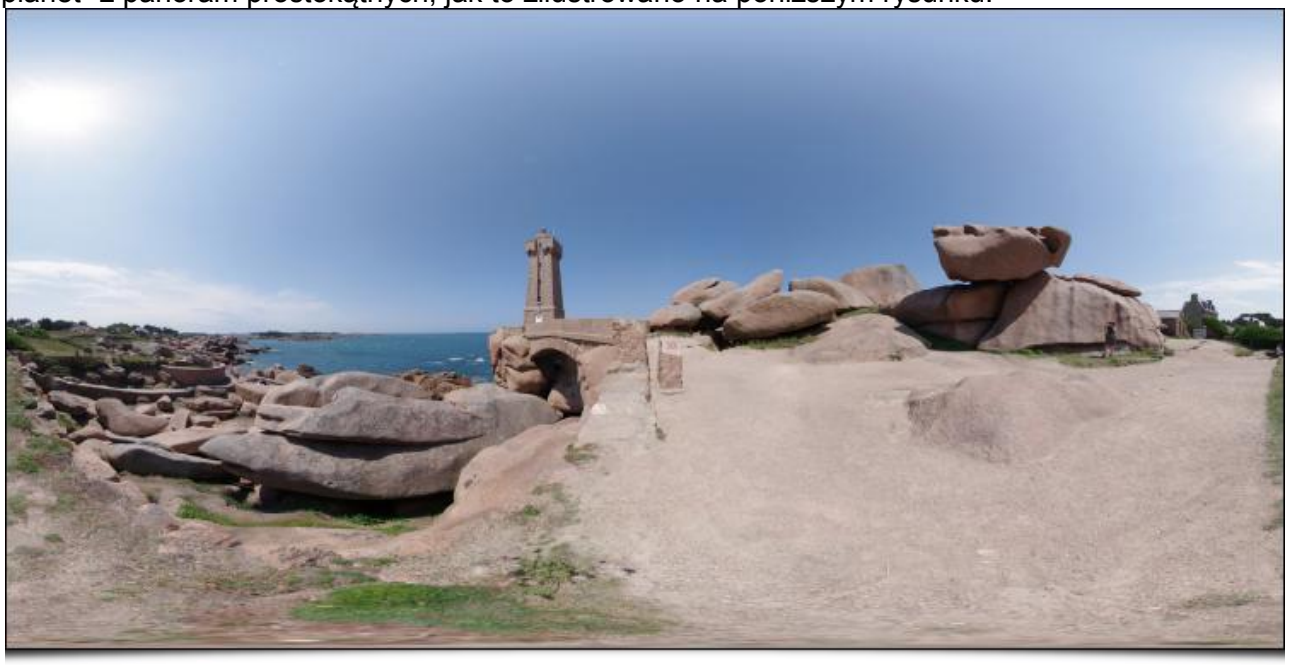
https://gmic.eu/gmic234/fullsize/gmic_hand.gif

Rys. 2.6: Modyfikacja położenia źródła światła i związanych z nim renderingów iluminacji, obliczana automatycznie przez G'MIC.

Film przedstawiający różne możliwości edycji oświetlenia dozwolonym przez ten filtr jest tu widoczne https://www.youtube.com/watch?v=G1wYSJTsVtI. Mam nadzieję, że ta nowa funkcja *G'MIC* pozwoli artystom przyspieszyć etap iluminacji i cieniowania ich przyszłych rysunków!

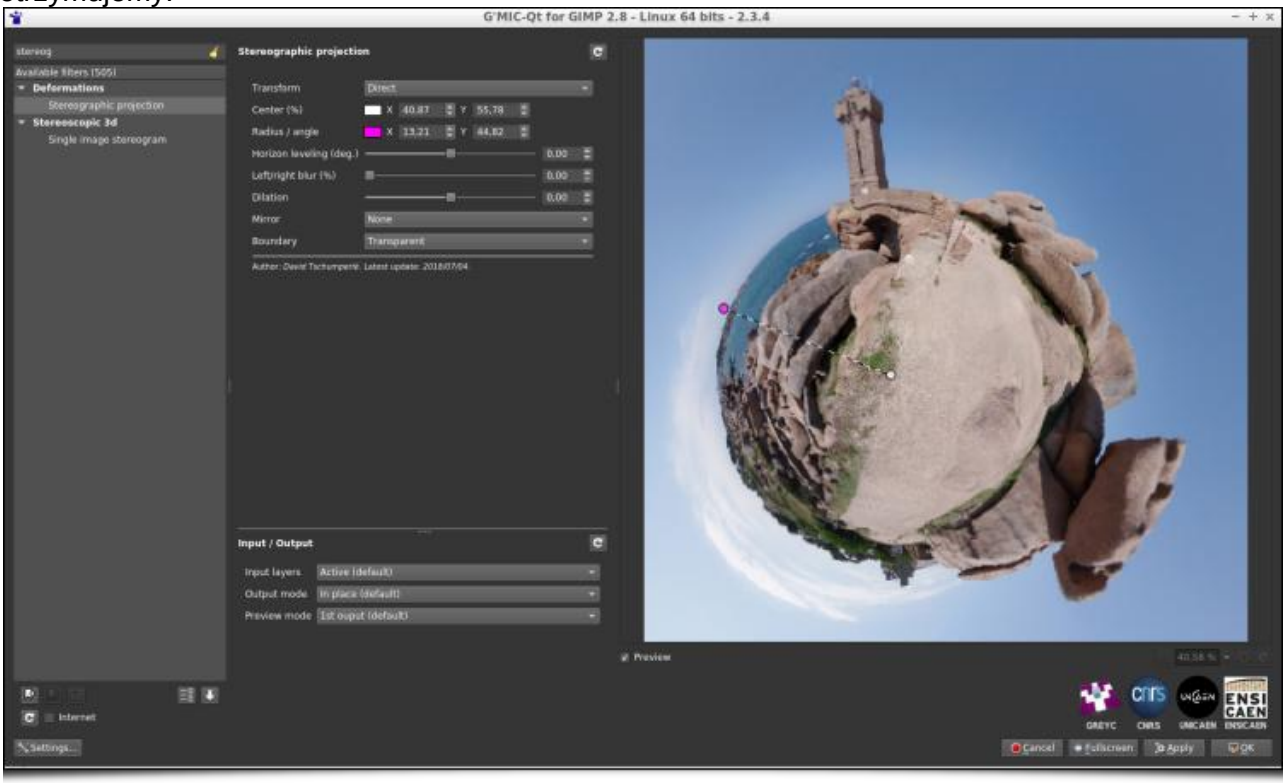
3. Projekcja stereograficzna

W zupełnie innym gatunku dodaliśmy również do *G'MIC* filtr realizujący projekcję stereograficzną https://pl.wikipedia.org/wiki/Rzut_stereograficzny, o nazwie “**Stereographic projection**“.
Ten rodzaj projekcji kartograficznej umożliwia rzutowanie planarnie zdefiniowanych danych obrazu na kulę. Należy zauważyć, że jest to zwykła projekcja używana do generowania obrazów "mini-planet" z panoram prostokątnych, jak to zilustrowano na poniższym rysunku.



Rys. 3.1: Przykład prostokątnej panoramy (autor: https://www.flickr.com/photos/gadl).

Jeśli na tej panoramie uruchamiamy wtyczką *G'MIC* i wybierzemy filtr "**Stereographic projection**", otrzymujemy:

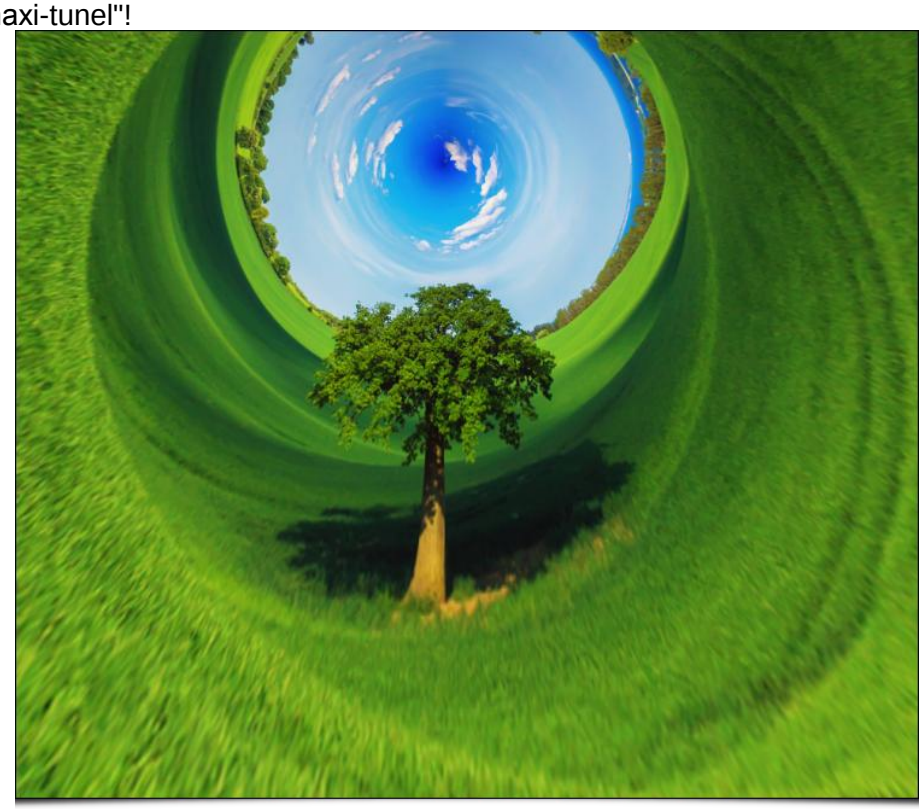


Rys. 3.2: Filtr G'MIC "Stereographic projection" w akcji w wtyczce GIMP lub Krita.

Filtr umożliwia precyzyjną regulację środka projekcji, kąta obrotu i promienia kuli, wszystkie interaktywnie wyświetlane bezpośrednio w oknie podglądu (do tego wrócimy później). Za pomocą kilku kliknięć i po zastosowaniu filtru otrzymujemy pożądaną "mini planetę":

Rys. 3.3: "Mini-planeta" otrzymana po projekcji stereograficznej.

Zabawne jest również to, że po prostu odwracając pionową oś obrazów, przekształcamy "mini planetę" w "maxi-tunel"!



Rys. 3.4: "Maxi-tunel" uzyskany przez odwrócenie osi pionowej, a następnie projekcja stereograficzna.

Ponownie zrobiliśmy to małe wideo https://www.youtube.com/watch?v=5BYV1lwuF3w, które pokazuje ten filtr w rzeczywistych warunkach użytkowania. Zauważ, że *G'MIC* już miał podobny filtr

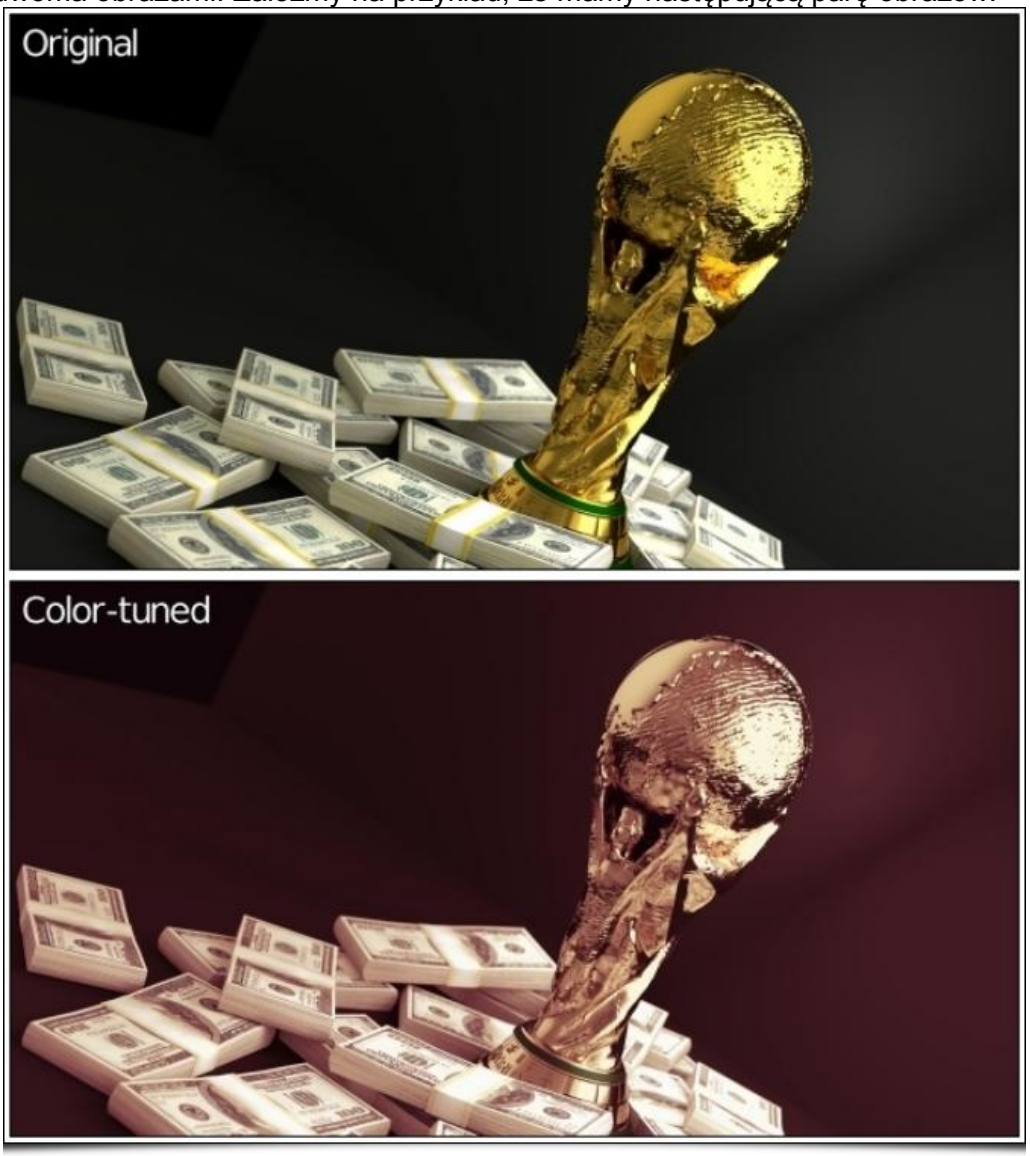
(o nazwie "*Sphere*"), który mogłyby zostać użyty do stworzenia „mini-planet”, ale z typem projekcji mniej odpowiednim niż obecnie dostępna projekcja stereograficzna

4. Coraz więcej możliwości manipulacji kolorami

Manipulowanie kolorami obrazów jest powtarzającym się zajęciem wśród fotografów i ilustratorów a *G'MIC* już dziesiątki filtrów dla tego konkretnego działania, pogrupowanych w specjalną kategorię (pierwotnie nazywaną kategorią "**Kolory**"!). Ta kategoria wciąż rośnie, a ostatnio pojawiły się dwa nowe filtry:

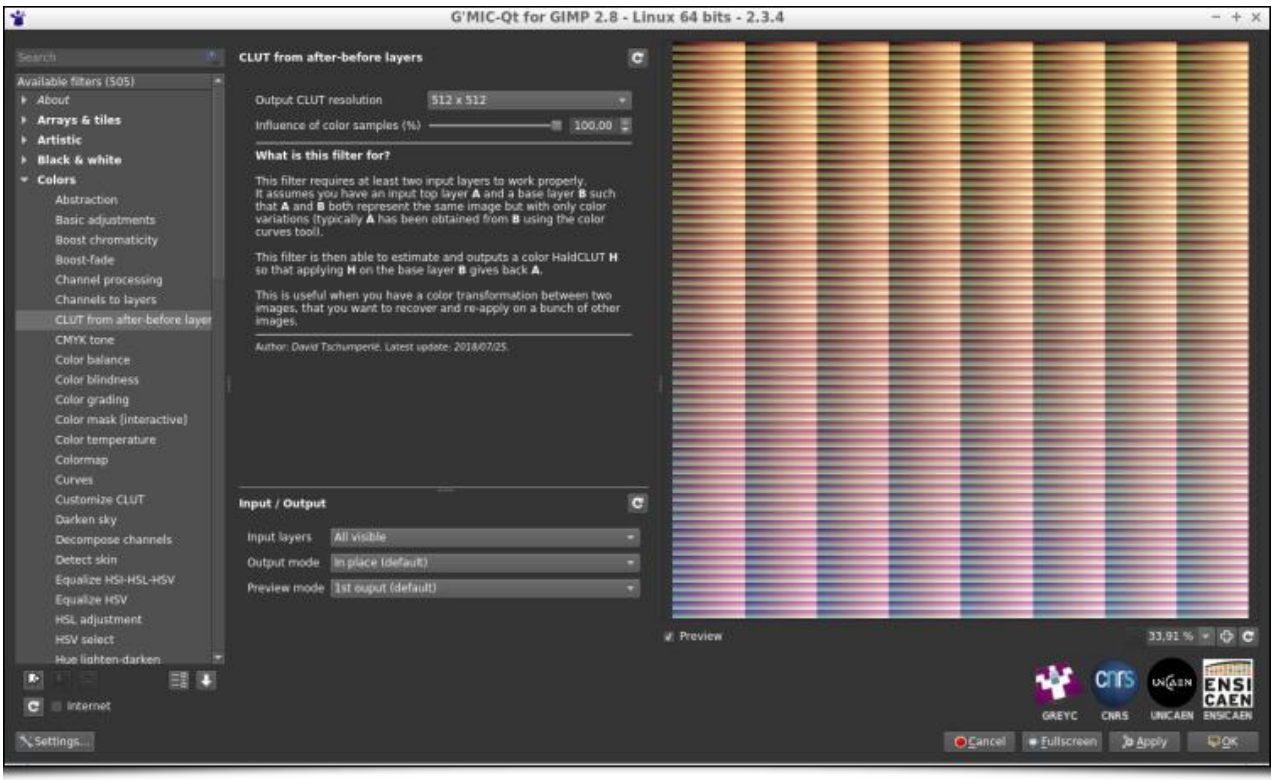
4.1. Filtr «CLUT from after-before layers»

Filtr "**CLUT from after-before layers**" próbuje modelować transformację kolorów wykonaną między dwoma obrazami. Załóżmy na przykład, że mamy następującą parę obrazów:



Rys. 4.1: Para obrazów, dla których nieznana transformacja kolorymetryczna została zastosowana z obrazu górnego, aby uzyskać dolną.

Problem : w ogóle nie pamiętamy, jak przeszliśmy od oryginalnego obrazu do zmodyfikowanego obrazu, ale chcielibyśmy zastosować ten sam proces do innego obrazu. Nie martw się, wywołaj *G'MIC* na ratunek! Filtr ten będzie dążył do lepszego modelowania modyfikacji kolorów w postaci *HaldCLUT* http://www.quelsolaar.com/technology/clut.html, który jest klasycznym sposobem reprezentowania dowolnej transformacji kolorymetrycznej.



Rys. 4.2: Filtr modeluje transformację kolorów między dwoma obrazami jako HaldCLUT.

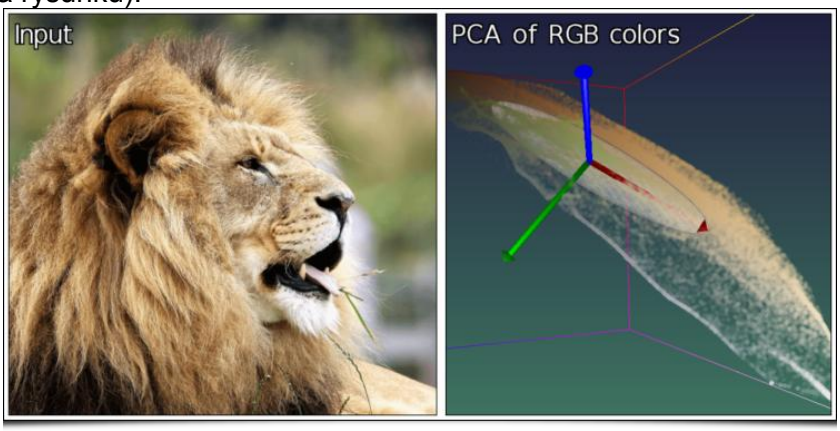
HaldCLUT wygenerowany przez filtr zostanie zapisany i ponownie zastosowany do innych obrazów z żądaną właściwością, że aplikacja *HaldCLUT* na oryginalnym obrazie tworzy docelowy obraz modelu, który został pierwotnie wykorzystany do nauki transformacji.
Stamtąd możemy zastosować równoważną zmianę koloru do dowolnego innego obrazu:

Rys. 4.3: Szacowana transformacja kolorów w postaci HaldCLUT jest ponownie stosowana do innego obrazu.

Ten Filtr pozwala *ostatecznie* stworzyć *HaldCLUT* "przez przykład", a tym samym może zainteresować wielu fotografów (w tym tych, którzy rozpowszechniają kompilacje plików *HaldCLUT*, wolne https://rawpedia.rawtherapee.com/Film_Simulation lub w inny sposób!).

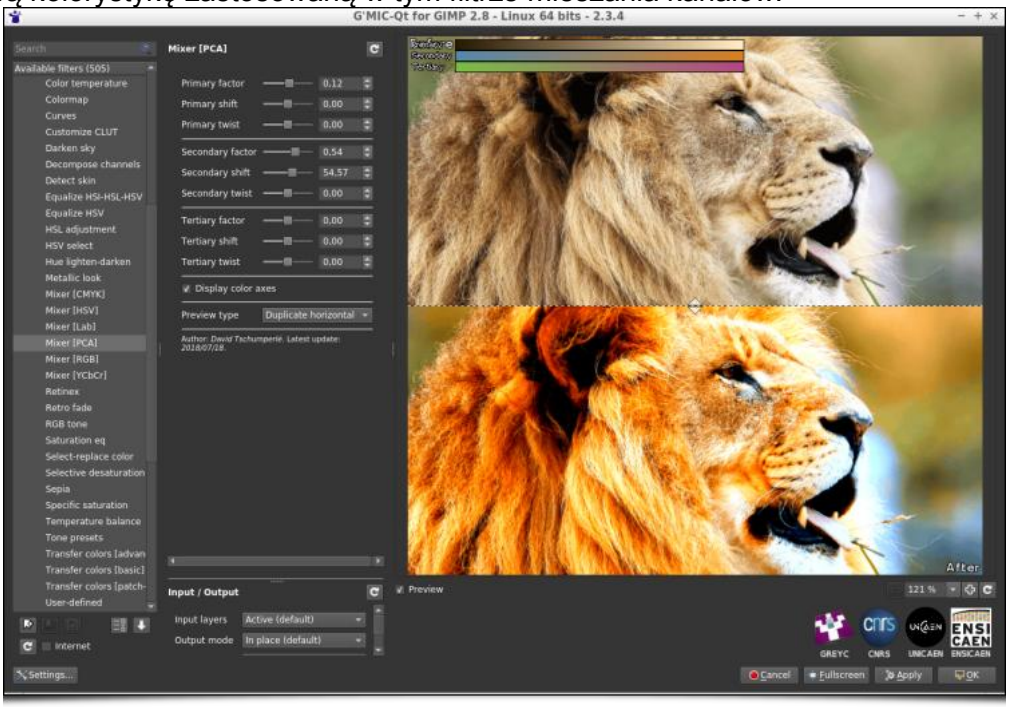
4.2. Filtr «Mixer [PCA]»

Drugi filtr manipulacji kolorem, nazwany "**Mixer [PCA]**", został niedawno zintegrowany z *G'MIC*. Działa jak klasyczny mikser kanałów https://docs.gimp.org/fr/plug-in-colors-channel-mixer.html, ale zamiast pracować z wcześniej zdefiniowaną przestrzenią kolorów (np. *SRGB, HSV, Lab ...*), działa on w "naturalnej" przestrzeni kolorów obrazu wejściowego, uzyskaną przez analizę głównych składników https://fr.wikipedia.org/wiki/Analyse_en_composantes_principales (*PCA*) jej kolorów *RGB* . W ten sposób każdy obraz zostanie powiązany z inną przestrzenią kolorów. Na przykład, jeśli weźmiemy obraz "*lwa"* poniżej i spojrzymy na rozkład jego kolorów w kostce *RGB* (*prawy obraz*), widzimy, że główna oś zmiany koloru jest określona przez linię prostą od ciemnego pomarańcza do jasnego beżu (oś oznaczona czerwoną strzałką na rysunku).



Rys. 4.4: Rozkład kolorów obrazu "lwa" w kostce RGB i powiązane osie główne (pokolorowane na czerwono, zielono i niebiesko).

Druga oś zmiany (*zielona strzałka*) przechodzi z niebieskiego na pomarańczowy, a trzecia oś (*niebieska strzałka*) z zielonego na różowy. To te osie zmienności (a nie osie *RGB*) definiują podstawową kolorystykę zastosowaną w tym filtrze mieszania kanałów.



*Rys. 4.5: Filtr "**Mikser [PCA]" jest mikserem kanałowym działającym na osiach "naturalnych" wariacji kolorów obrazu.*

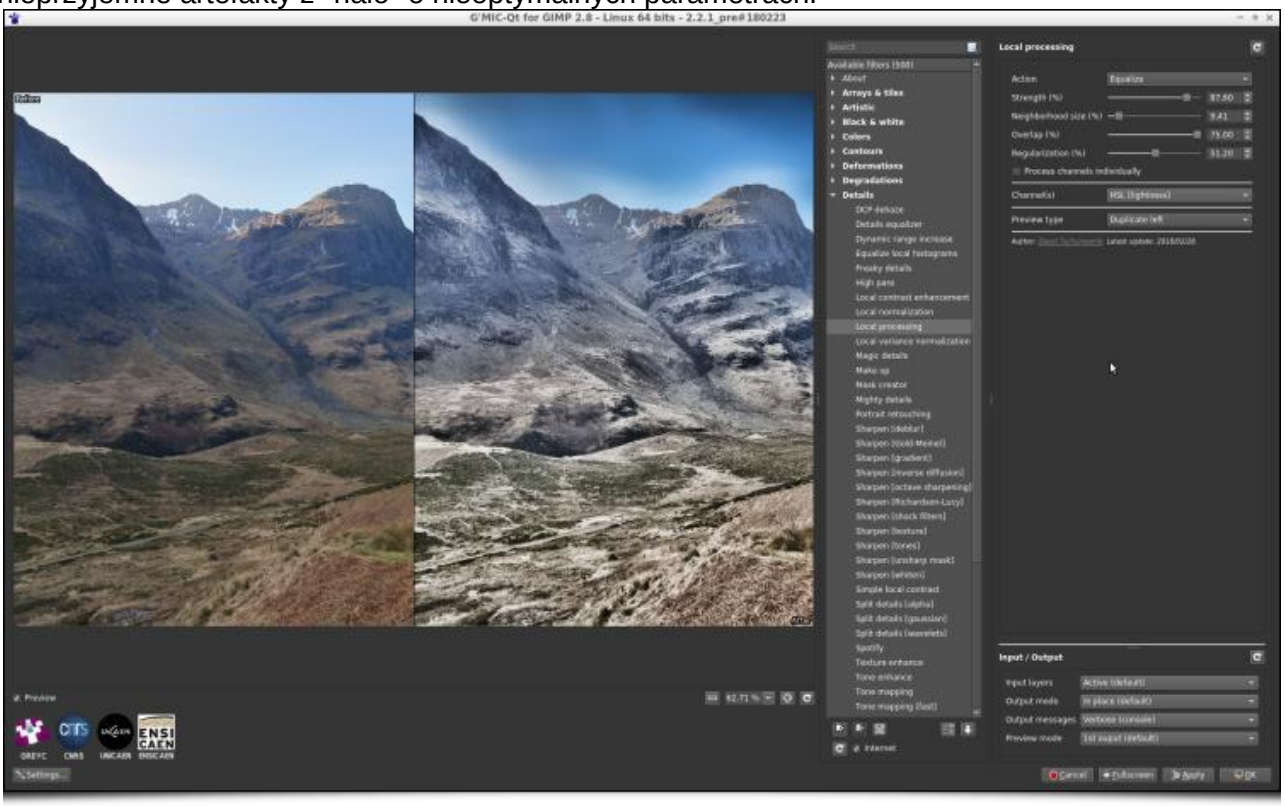
Błędem byłoby sugerować, że zawsze lepiej jest wziąć pod uwagę bazę kolorów uzyskaną przez *PCA* do mieszania kanałów, a ten nowy filtr oczywiście nie ma być "ostatecznym" mikserem, który zastąpiłby wszystkie inne. Po prostu istnieje jako alternatywa dla zwykłych narzędzi do mieszania kanałów kolorów, której wyniki okazały się dość interesujące w testach kilku obrazów wykorzystywanych podczas opracowywania tego filtra. W każdym razie nie zaszkodzi spróbować...

5. Mieszanina Filtrów

W tej sekcji opisano kilka innych ulepszonych lub uwzględnionych ostatnio w G'MIC filtrów, które zasługują na to, aby je omówić, bez zbytniego zajmowania się nimi.

5.1. Filtr "Local processing"

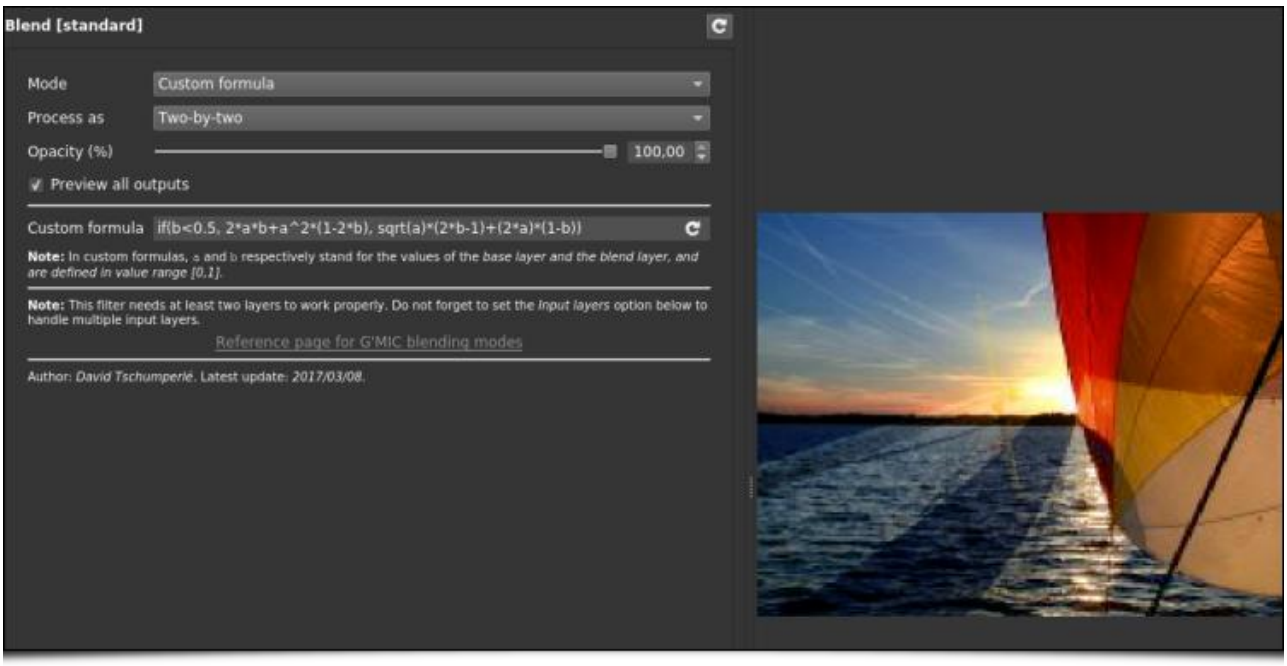
Filtr **"Local processing**" umożliwia zastosowanie procesu normalizacji lub wyrównania kolorów w lokalnych obszarach obrazów (z możliwym nakładaniem się). Jest to dodatkowy filtr, który sprawia że na zdjęciach pojawiają się detale niedoświetlone lub prześwietlone, ale może tworzyć silne i nieprzyjemne artefakty z "halo" o nieoptymalnych parametrach.



*Rys. 5.1: Filtr "***Local processing"** *poprawia szczegóły na zdjęciach o zbyt dużej lub zbyt małej ekspozycji.*

5.2. Filtr «Blend [standard]»

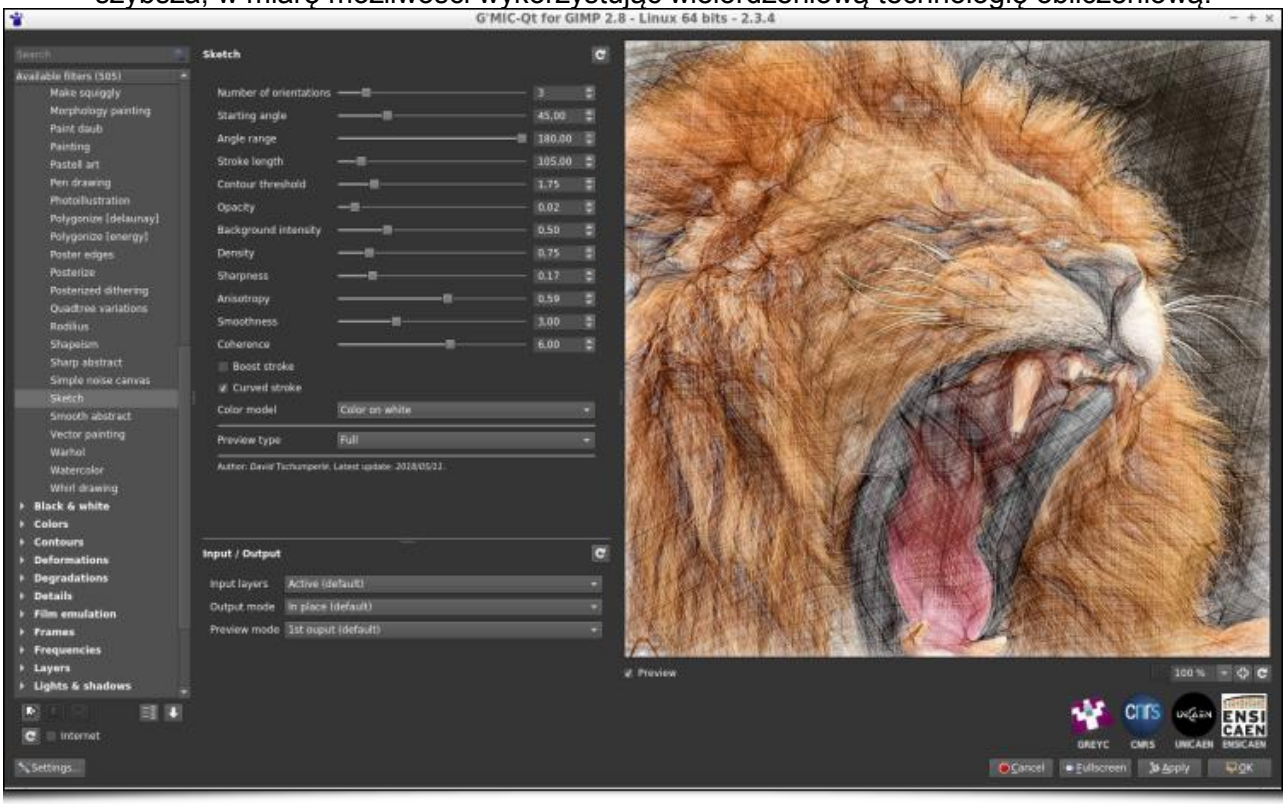
Jeśli uważasz, że liczba trybów mieszania warstw dostępnych w GIMP lub Krita jest niewystarczająca i marzysz o zdefiniowaniu własnej formuły mieszania, to ostatnia poprawa filtru G'MIC «Blend [standard]» sprawi Ci przyjemność!. Ten filtr, już istniejący wcześniej, został wzbogacony o funkcję " *Formuła niestandardowa* ", która pozwala użytkownikowi określić własną matematyczną formułę podczas łączenia dwóch warstw http://www.pegtop.net/delphi/articles/blendmodes/. Wszystkie Twoje życzenia mieszania stają się możliwe!



Rys. 5.2: Filtr " ***Blend [standard]*** *" pozwala teraz definiować własne formuły matematyczne do łączenia warstw.*

5.3. Filtr «Sketch»

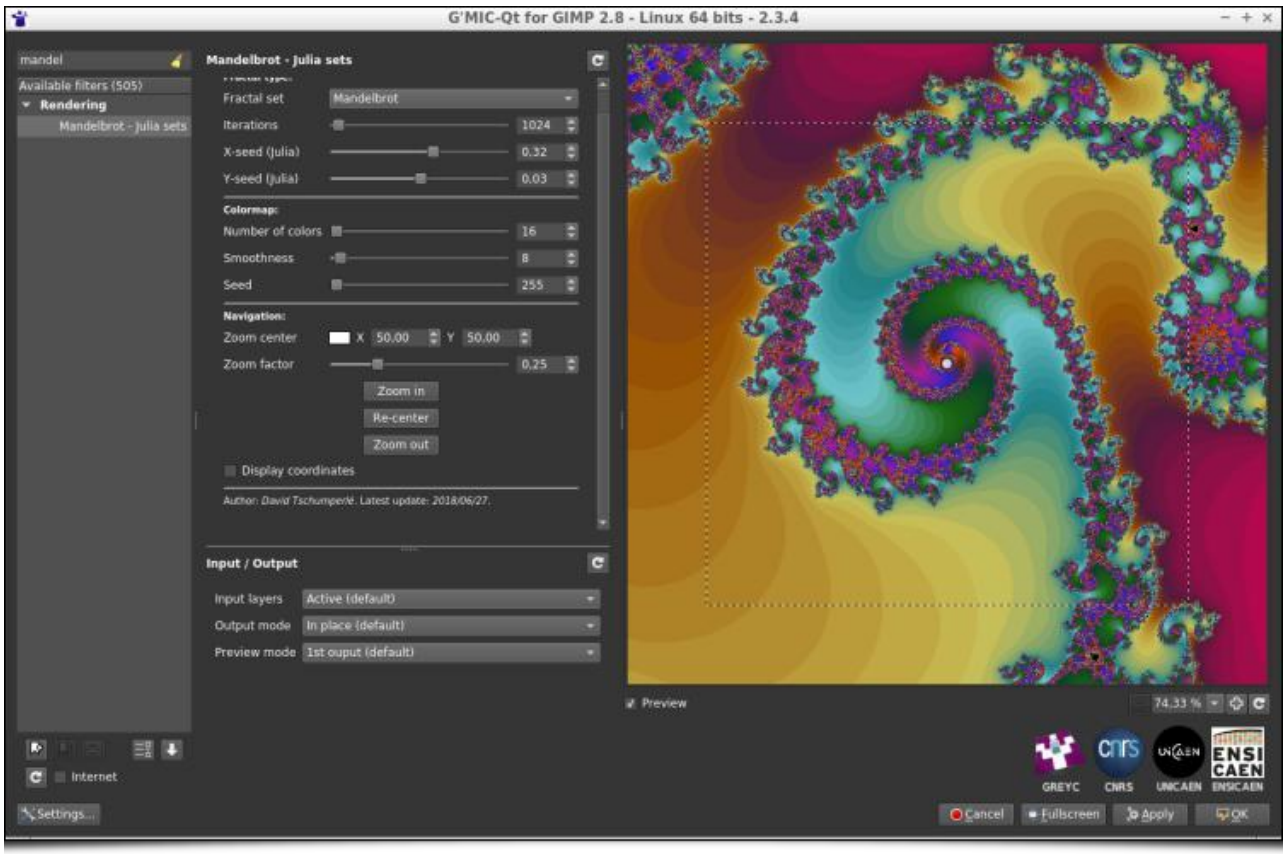
Zauważ także pełną ponowną implementacją ładnego filtru "**Szkic**", który istniał przez kilka lat, ale może być nieco wolny na dużych obrazach. Nowa implementacja jest znacznie szybsza, w miarę możliwości wykorzystując wielordzeniową technologię obliczeniową.



Rys. 5.3: Filtr " ***Szkic*** *" został ponownie wdrożony i teraz wykorzystuje wszystkie dostępne rdzenie obliczeniowe.*

5.4. Filtr «Mandelbrot - Julia sets»

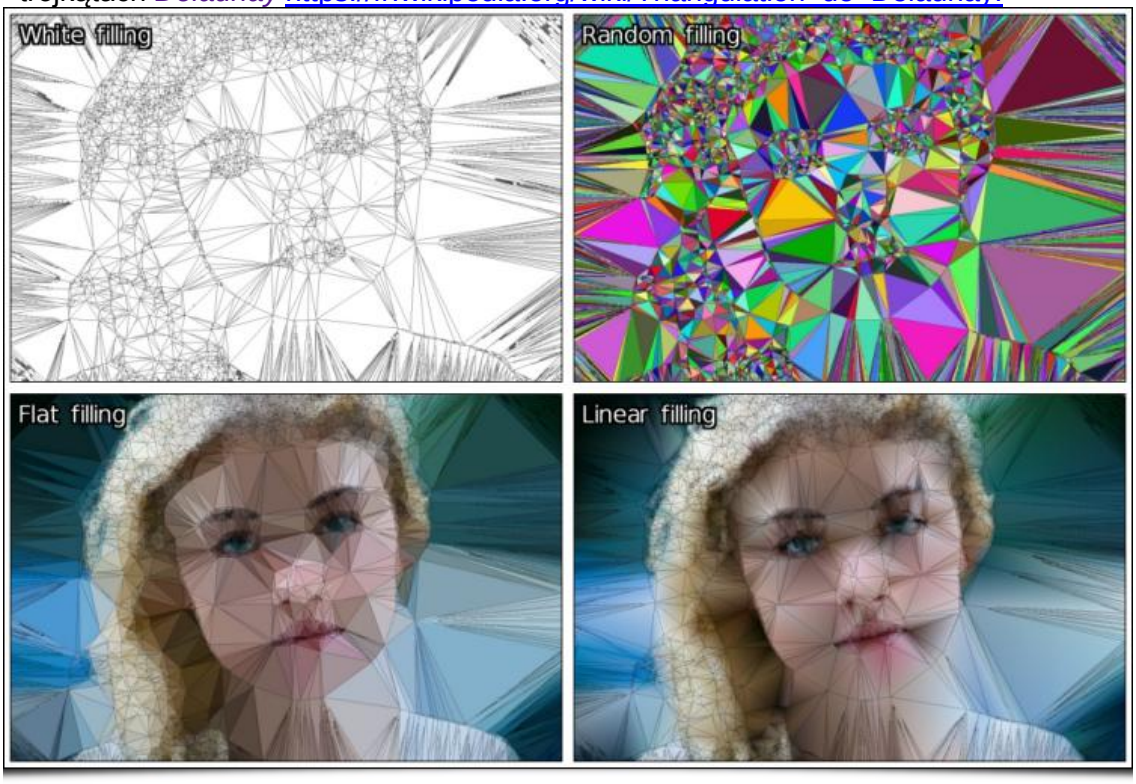
Dużą ponowną implementację pracy przeprowadzono również na filtrze "**Mandelbrot - Julia ses"**, ponieważ interfejs nawigacyjny został całkowicie przeprojektowany, dzięki eksploracji zbioru Mandelbrota (https://fr.wikipedia.org/wiki/Ensemble_de_Mandelbrot) bardziej komfortowe (jak pokazano w filmie wideo https://youtu.be/wZv3BQF00gA). Pojawiły się również nowe opcje wyboru kolorów.



*Rys. 5.4: Filtr "**Mandelbrot - Julia sets**" i jego nowy interfejs nawigacyjny w złożonej przestrzeni.*

5.5. Filtr «Polygonize [Delaunay]»

Filtr "**Polygonize [Delaunay]**", generuje wielokątne renderowania kolorowych obrazów, ma nowy tryb renderowania, wykorzystujący liniowo interpolowane kolory w wygenerowanych trójkątach *Delaunay* https://fr.wikipedia.org/wiki/Triangulation_de_Delaunay.



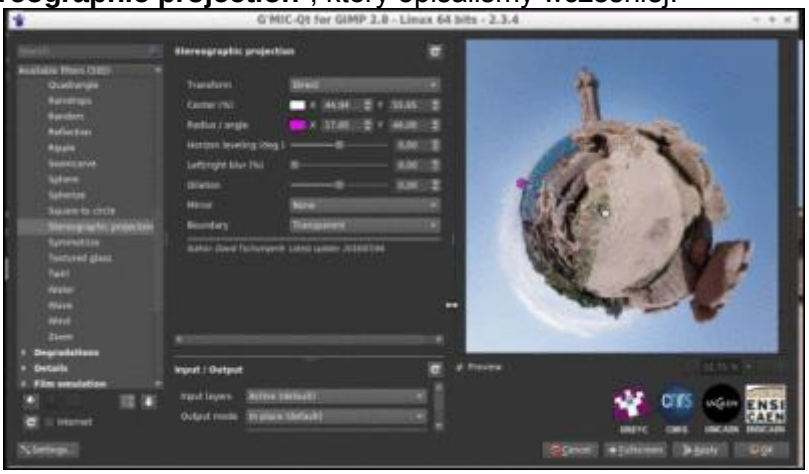
*Rys. 5.5: Różne tryby renderowania filtru "**Polygonize [Delaunay]**".*

6. Inne ważne elementy projektu

6.1. Ulepszenia interfejsu wtyczki

Oczywiście, co nowego w *G'MIC* to nie tylko filtry przetwarzania obrazu! *Dokonano znacznych* prac, na przykład na graficznym interfejsie wtyczki *G'MIC-Qt*, w szczególności:

- Filtry wtyczki mają teraz możliwość określenia nowego typu parametru `point()`, który wyświetla się jako małe kolorowe kółko na oknie podglądu, którym można manipulować bezpośrednio w oknie podglądu. Użytkownik może przeciągnąć ten okrąg i przesunąć go za pomocą myszy.
  W praktyce sprawia to, że okno podglądu jest interaktywne, to nie jest mała rzecz! Wiele filtrów korzysta teraz z tej możliwości, co czyni je znacznie przyjemniejszymi i intuicyjnymi w użyciu (zobacz film wideo https://www.youtube.com/watch?v=iQ0ZEmsDErY dla niektórych przykładów). Poniższa animacja pokazuje na przykład, w jaki sposób te interaktywne punkty są używane w nowym filtrze " **Stereographic projection**", który opisaliśmy wcześniej.



http://gmic.eu/gmic234/fullsize/gmic_point_anim.gif

Rys. 6.1: Okno podglądu wtyczki G'MIC-Qt zostało wzbogacone o nowe możliwości interakcji dla użytkownika.

- Ponadto wprowadzenie tych interaktywnych punktów umożliwiło ulepszenie trybów *podzielonego podglądu*, dostępnych w wielu filtrach, aby wyświetlać obok siebie widoki "before / after" podczas ustawiania parametrów filtru we wtyczce. Teraz możliwe jest przesunięcie tego separatora «przed / po», co ilustruje poniższa animacja. Ponadto dołączono dwa nowe tryby podziału ('Checkered' i 'Inverse checkered').

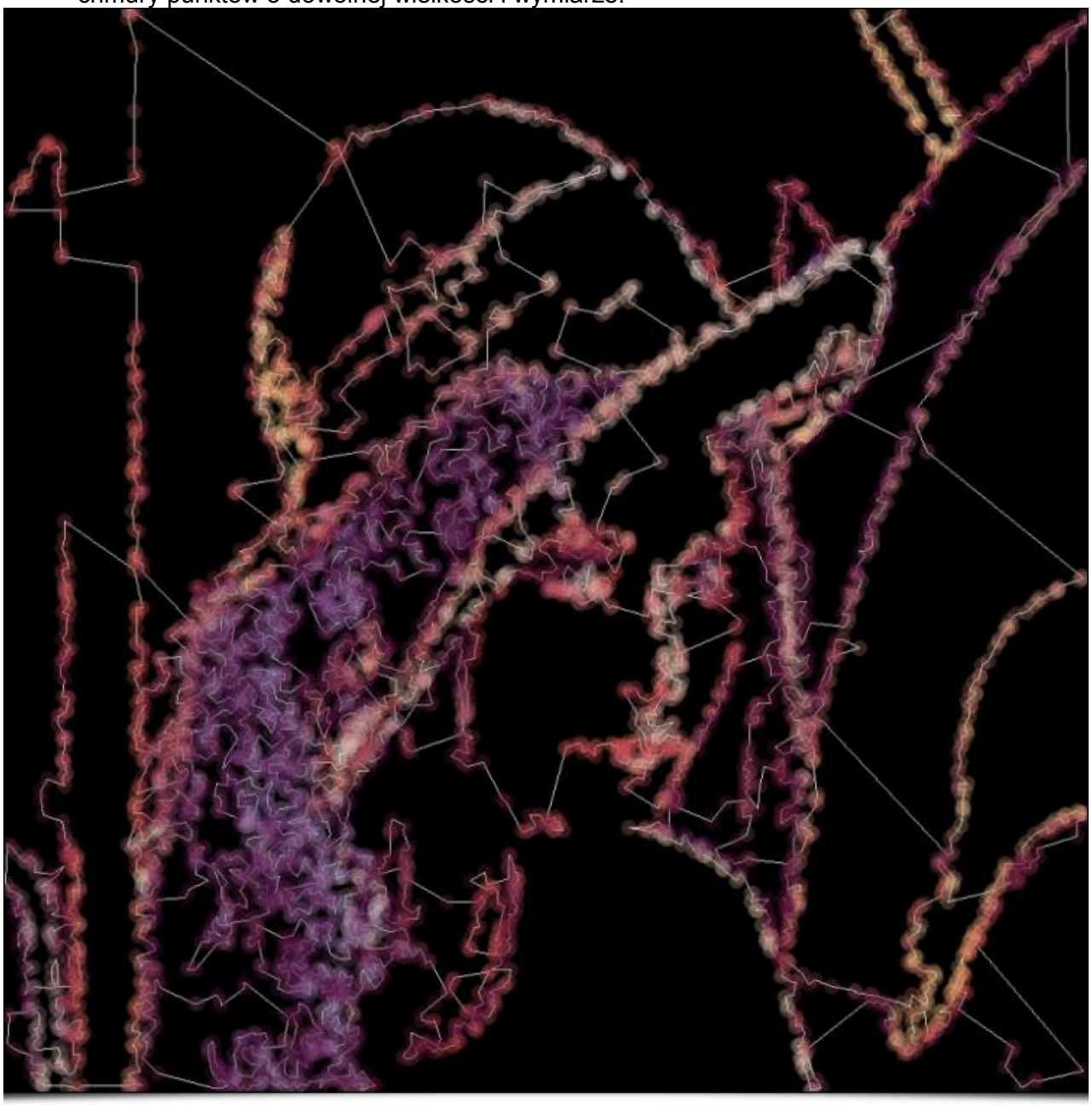
Rys. 6.2: Tryby podziału podglądu mają teraz ruchomą granicę " before / after*".*

W kodzie wtyczki wprowadzono wiele innych drobnych poprawek:
obsługa najnowszej wersji GIMP (2.10), Qt 5.11, poprawiona obsługa komunikatów o błędach wyświetlanych na widżecie podglądu, czystszy zaprojektowany interfejs i inne małe zmiany zostały wprowadzone pod maską, które niekoniecznie są widoczne, ale nieco poprawiają komfort użytkownika (np. mechanizm bufora obrazu dla widżetu podglądu). Krótko mówiąc, to całkiem nieźle!

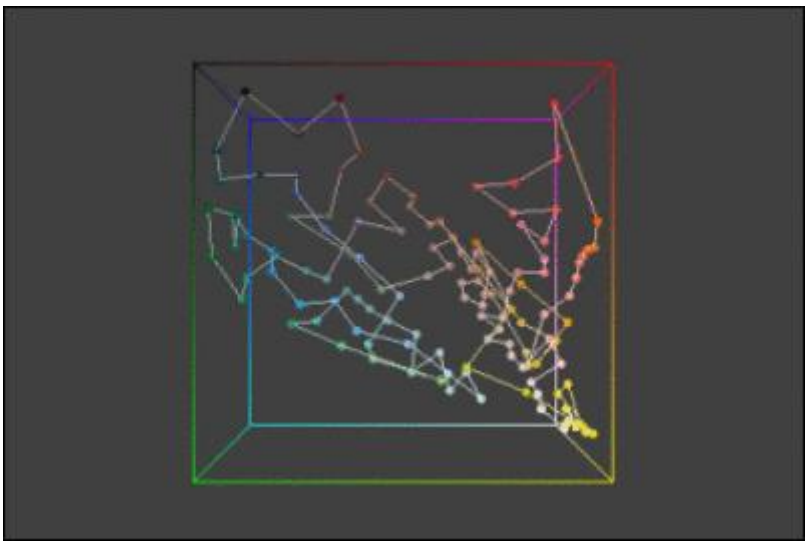
6.2 Doskonalenie rdzenia oprogramowania

Ostatnio wprowadzono nowe udoskonalenia rdzenia obliczeniowego G'MIC:

- "standard library" języka skryptowego *G'MIC* otrzymała nowe polecenia do obliczania odwrotnych funkcji hiperbolicznych (`acohs`, `asinh` i `atanh`) oraz sterowanie `tsp` (***t**ravelling **s**alesman **p**roblem* problem komiwojażera), które szacuje akceptowalne rozwiązanie dla dobrze znanego problemu komiwojażera https://fr.wikipedia.org/wiki/Probl%C3%A8me_du_voyageur_de_commerce , i to, dla chmury punktów o dowolnej wielkości i wymiarze.



Rys. 6.3: Szacowanie najkrótszego obwodu między kilkoma setkami punktów 2D za pomocą polecenia `tsp` G'MIC



https://gmic.eu/gmic234/fullsize/tsp3d.gif

Rys. 6.4: Oszacowanie najkrótszego obwodu między kilkoma kolorami kostki RGB (w 3D) dzięki poleceniu `tsp` G'MIC.

- Interfejs demonstracyjny, uruchamiany po wywołaniu `gmic` bez argumentów z wiersza poleceń, został również ponownie uruchomiony od zera.

https://gmic.eu/gmic234/fullsize/gmic_demo.gif

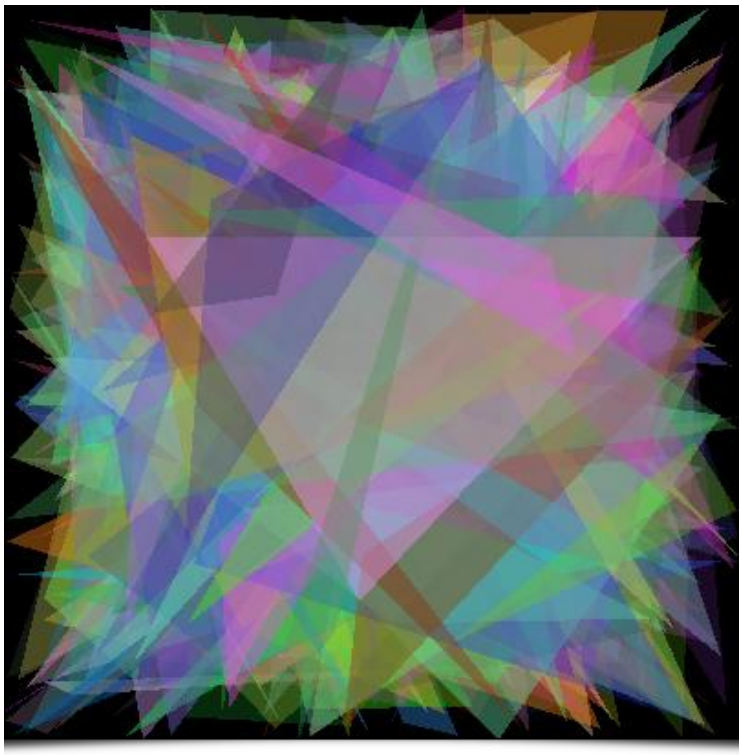
Rys. 6.5: Nowy interfejs demonstracyjny narzędzia gmic, wiersza poleceń G'MIC.

- Wbudowany Kompilator *JIT* używany do oceny wyrażeń matematycznych nie został pominięty i otrzymał nowe funkcje rysowania wielokątów (funkcja `polygon()`) lub elipsy (funkcja `ellipse()`) w obrazach. Te wyrażenia matematyczne mogą w rzeczywistości definiować małe programy (ze zmiennymi lokalnymi, funkcjami zdefiniowanymi przez użytkownika i przepływem sterowania). Można na przykład łatwo wygenerować obrazy syntetyczne z wiersza poleceń, jak pokazano w dwóch przykładach poniżej.

Przykład 1:

```
$ gmic 400,400,1,3 eval "for (k = 0, k<300, ++k,
polygon(3,u([vector10(0),[w,h,w,h,w,h,0.5,255,255,255])))"
```

Wynik :

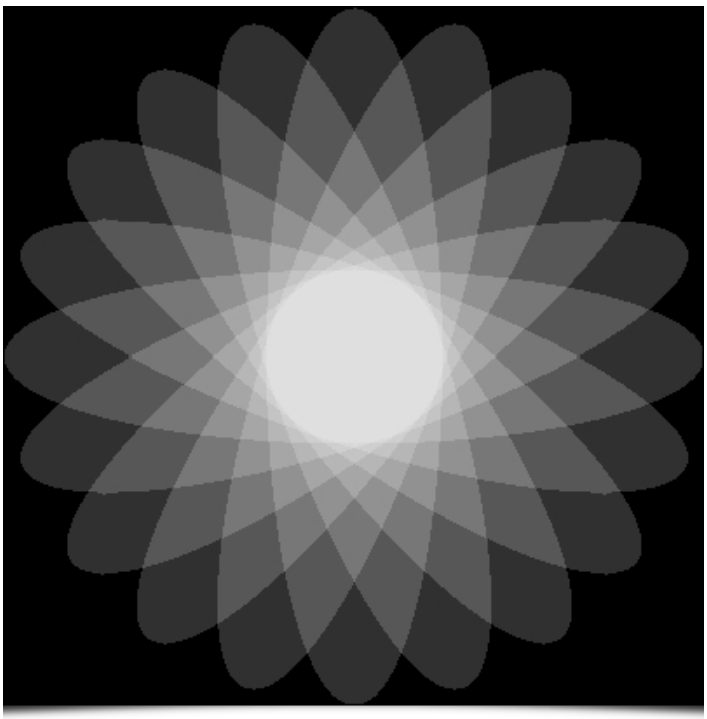


Ryc. 6.6: Używanie nowej funkcji kompilatora JIT G'MIC do generowania losowego obrazu syntetycznego.

Przykład 2:

```
$ gmic 400,400,1,3 eval "for (k=0, k<20, ++k,
ellipse(w/2,h/2,w/2,w/8,k*360/20,0.1,255))"
```

Wynik :



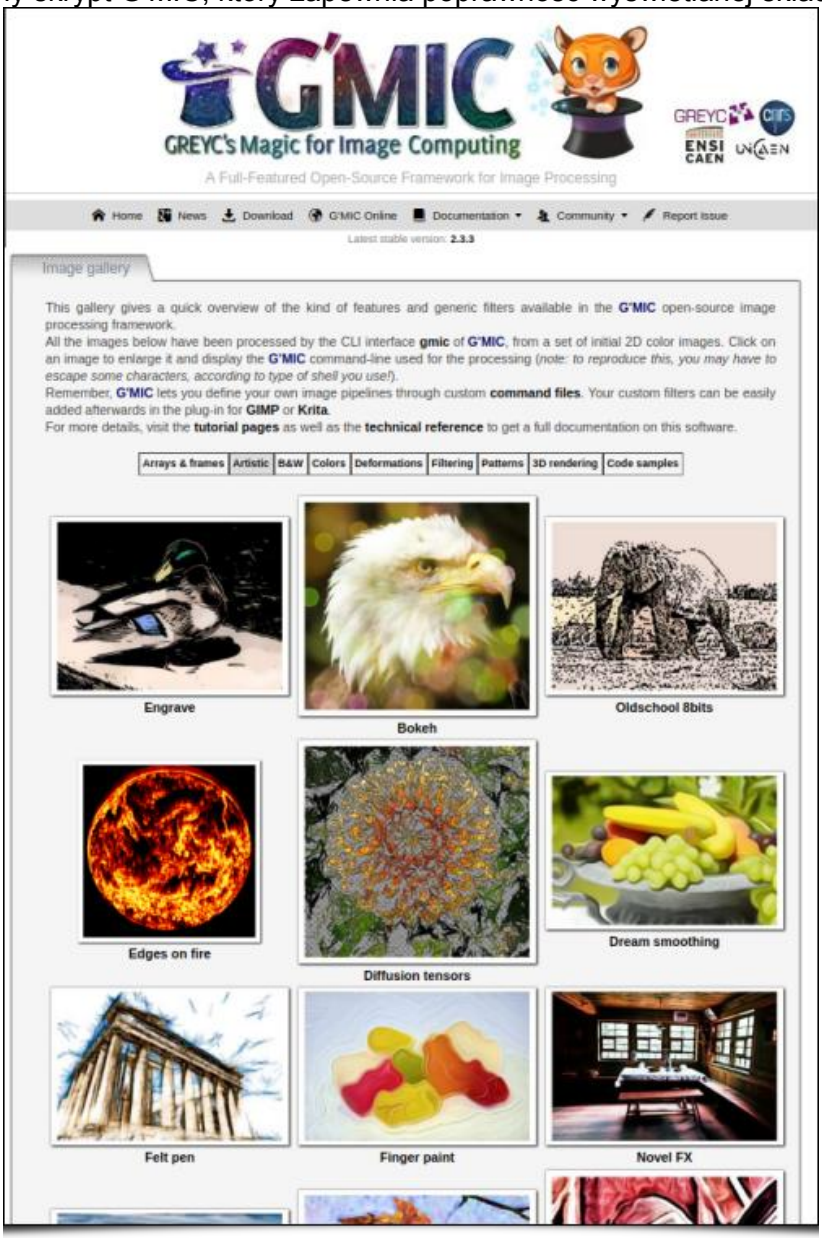
Ryc. 6.7: Używanie nowej funkcji kompilatora JIT G'MIC do generowania syntetycznego obrazu kwiatu.

- Zauważ również, że wartości `NaN` podczas obliczeń wykonywanych przez serce oprogramowania, co pozwala *G'MIC* zachować spójne zachowanie nawet po skompilowaniu z optymalizacją są teraz lepiej zarządzane przy wykonywaniu obliczeń w rdzeniu, co oznacza, że G'MIC zachowuje spójne zachowanie, nawet jeśli zostało skompilowane z optymalizacją `-ffast-math`. W ten sposób G'MIC może być bezbłędnie skompilowany teraz na maksymalny poziom optymalizacji `-Ofast` obsługiwany przez kompilator `g++`, podczas gdy wcześniej ograniczaliśmy się do używania `-O3`. Poprawa szybkości obliczeń jest wyraźnie widoczna w przypadku niektórych oferowanych filtrów!!

6.3. Media rozpowszechniające

Wprowadzono również wiele zmian w kanałach dystrybucji używanych przez projekt:

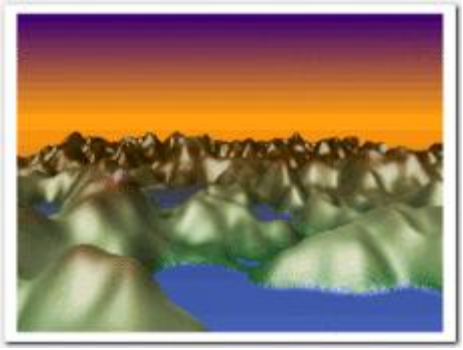
- Przede wszystkim strony projektu (które teraz domyślnie korzystają z zabezpieczonych połączeń `https`) mają nową galerię obrazów http://gmic.eu/gallery/. Ta galeria pokazuje zarówno wyniki filtrowanego obrazu z G'MIC, jak i sposób ich odtworzenia (z wiersza poleceń). Zwróć uwagę, że te strony galerii są generowane automatycznie przez dedykowany skrypt G'MIC, który zapewnia poprawność wyświetlanej składni polecenia.



Rys. 6.8: Nowa strona galerii obrazów na stronie G'MIC.

Ta galeria jest podzielona na kilka sekcji, w zależności od rodzaju wykonywanych zabiegów (*artystyczne, czarno-białe, wypaczenia, filtrowanie itp*.). Ostatnia sekcja **Próbka kodu** https://gmic.eu/gallery/codesamples.shtml jest osobiście tym, co uważam za najbardziej zabawne, ponieważ przedstawia małe sekwencje obrazów (w postaci animowanych *GIF-ów*), generowanych w całości przez krótkie skrypty w języku *G'MIC* . Nieco egzotyczny sposób użycia *G'MIC*, ale pokazujący jego potencjał dla sztuki generatywnej https://fr.wikipedia.org/wiki/Art_g%C3%A9n%C3%A9ratif.

https://gmic.eu/gallery/codesamples_full_3.gif



https://gmic.eu/gallery/codesamples_full_4.gif

Rys. 6.9: Dwa przykłady animacji GIF generowanych przez skrypty w języku G'MIC, widoczne w galerii obrazów.

- Przenieśliśmy również główne repozytorium źródłowe git projektu do Framagit https://framagit.org/dtschump/gmic wciąż zachowując synchronizowane zwierciadło na Githubie w tym samym miejscu co poprzednio (aby skorzystać z faktu, że wielu programistów ma już konto na Github, co ułatwia je do rozwidlenia projektu i napisania raportów o błędach).

7. Wnioski i perspektywy

To wszystko! Nasze zwiedzanie nowości (z ostatnich sześciu miesięcy działalności) projektu *G'MIC* wreszcie się kończy.
Cieszymy się, że mogliśmy przeżyć dziesięć pięknych lat komputerowych emocji wraz z narodzinami i ewolucją tego darmowego projektu oraz móc podzielić się na naszej drodze, ze wszystkimi użytkownikami, technikami zaawansowanego przetwarzania obrazu. Mamy nadzieję, że odejdziemy jeszcze przez wiele lat! Wsparcia są coraz bardziej obecne wokół nas, więc mówimy, że należy to zrobić (w tym względzie, jeśli chcesz w jakikolwiek sposób wnieść swój wkład do projektu, to zapraszamy!).
Zauważ, że w przyszłym roku będziemy również świętować *20-lecie* istnienia *CiMG* http://cimg.eu/, bibliotekę *C* ++ przetwarzania obrazu, który jest bezpośrednio za projekt *G'MIC*(i która urodziła się w listopadzie 1999 r., nie odmłodzi mojej dobrej pani ...). Dowód na to, że zainteresowanie wolnym oprogramowaniem leży w dłuższej perspektywie!
Czekając na kolejną wysyłkę na *G'MIC* , nie wahaj się przetestować tego oprogramowania, odtwarzaj i trituruj swoje obrazy w najbardziej darmowy i kreatywny sposób!