

Traitement du signal et des images

Projet de filtre - Rapport

Organisation du projet :

Le projet est composé de trois fichiers app.sce / fonctions.sci / filtres.sci ainsi qu'un dossier Images contenant une image : blackPanther.jpg, sur laquelle nous allons appliquer les différents filtres.

Prérequis : Scilab 6.0.2 + SVIP

app.sce

Ce fichier représente le « main » du projet, c'est celui qui se chargera de lancer notre application de filtres. De plus, c'est également ici, que nous mettrons en place toute l'interface graphique.

Voici sa représentation :

```
//Configurations
formats=["JPG/JPEG-format" ".jpg" ".png" "PNG-format"]
window=figure()
window.BackgroundColor=[50 50 50]
window.figure_name=gettext("MyPhotoshop-Cc")

handle_Axe1=newaxes()
handle_Axe1.margins=[0.0 0.0]
handle_Axe1.axes_bounds=[0.05 0.2 0.66 0.5]
handle_Axe1.background_color=[30 30 30]
handle_Axe1.auto_clear='on'

//Création d'un module pour charger l'image via chargement_image
hl=uiMENU("parent" window "label" gettext("Charger une image") "callback" "chargement_image")

//Cacher les menus
delmenu(window figure_id gettext("File"))
delmenu(window figure_id gettext("Tools"))
delmenu(window figure_id gettext("Edit"))
delmenu(window figure_id gettext("Help"))

//Textes

//Photoshop-CC
photoshopCc=uiCONTROL(window "style" "text")
photoshopCc.string="Photoshop-CC"
photoshopCc.fontweight="bold"
photoshopCc.position=[160 370 150 25]
photoshopCc.BackgroundColor=[0.20 0.20 0.20]
photoshopCc.ForegroundColor=[1.0 0.4]
photoshopCc.HorizontalAlignment="center"

//Filtres
filtres=uiCONTROL(window "style" "text")
filtres.string="Filtres"
filtres.fontweight="bold"
filtres.position=[448 320 130 25]
filtres.BackgroundColor=[0.12 0.12 0.12]
filtres.ForegroundColor=[1.0 0.4]
filtres.HorizontalAlignment="center"

//La-Filtre-Box
filtres_box=uiCONTROL(window "style" "listbox")
filtres_box.BackgroundColor=[0.12 0.12 0.12]
filtres_box.ForegroundColor=[0.25 0.25 0.25]
filtres_box.Position=[448 170 130 150]
filtres_box.String=["Canny" "Bonus" "Horizontal" "Vertical" "Horizontale" "Gaussien" "N/B" "Negatif" "Pratt" "Sobel" "Rouge" "Vert" "Bleu" "Cyan" "Jaune" "Violet"]
filtres_box.Value=-1

//Les.boutons

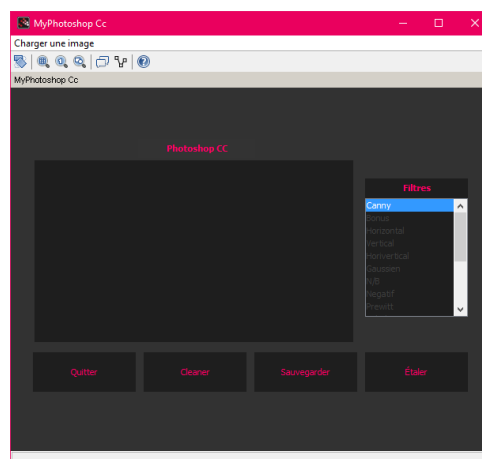
//quitter
quitter=uiCONTROL(window "style" "pushbutton")
quitter.String="Quitter"
quitter.BackgroundColor=[0.12 0.12 0.12]
quitter.ForegroundColor=[1.0 0.4]
quitter.Position=[28 75 130 50]
quitter.Callback="delete(window)"

//cleaner
cleaner=uiCONTROL(window "style" "pushbutton")
cleaner.String="Cleaner"
cleaner.BackgroundColor=[0.12 0.12 0.12]
cleaner.ForegroundColor=[1.0 0.4]
cleaner.Position=[168 75 130 50]
cleaner.Callback="[current_ns_image].=-cleaner_picture(handle);"

//sauvegarder
sauvegarder=uiCONTROL(window "style" "pushbutton")
sauvegarder.String="Sauvegarder"
sauvegarder.BackgroundColor=[0.12 0.12 0.12]
sauvegarder.ForegroundColor=[1.0 0.4]
sauvegarder.Position=[308 75 130 50]
sauvegarder.Callback="sauvegarder_image(formats,handle)"

//etaler
etaler=uiCONTROL(window "style" "pushbutton")
etaler.String="Étaler"
etaler.BackgroundColor=[0.12 0.12 0.12]
etaler.ForegroundColor=[1.0 0.4]
etaler.Position=[448 75 130 50]
etaler.Callback="[current_s_image].=-etaler_filtre(filtres_box,handle)";
```

Son exécution



fonctions.sci

Dans ce fichier, nous trouverons les définitions de quatre fonctions importantes dans la gestion de l'application.

Il s'agit d'étaler_filtre, qui, comme son nom l'indique, se chargera d'appliquer les filtres. Pour ce faire, on récupère l'indice sélectionné et on le compare avec chaque valeur allant de 1 à 16 (pour les 16 filtres) puis on appelle la fonction correspondante à l'indice sélectionné, (l'implémentation est similaire à celle d'un switch-case en c).

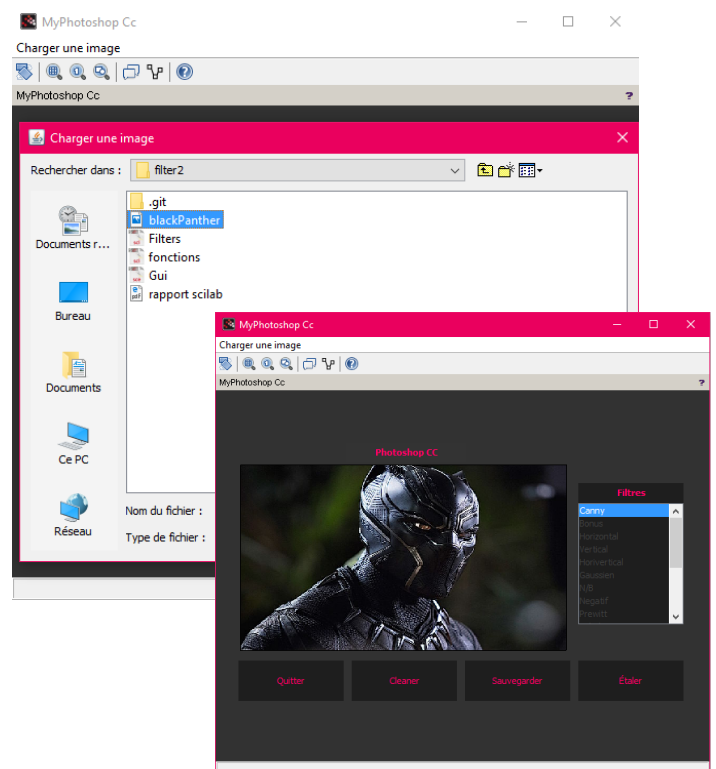
Voici sa représentation :

```
//etaler_filtre
function [N] = etaler_filtre(filterlist handle)
...if filterlist.Value == 1 then
>> N = canny(handle.S);
>> imshow(N)
.....handle.N = N
.....handle.resume(handle)
>> end
...if filterlist.Value == 2 then
>> N = bonus(handle.S);
>> imshow(N)
.....handle.N = N
.....handle.resume(handle)
>> end
...if filterlist.Value == 3 then
>> N = horizontal(handle.S);
>> imshow(N)
.....handle.N = N
.....handle.resume(handle)
>> end
...if filterlist.Value == 4 then
>> N = vertical(handle.S);
>> imshow(N)
.....handle.N = N
.....handle.resume(handle)
>> end
...if filterlist.Value == 5 then
>> N = horivertical(handle.S);
>> imshow(N)
.....handle.N = N
.....handle.resume(handle)
>> end
...if filterlist.Value == 6 then
>> N = gaussien(handle.S);
>> imshow(N)
.....handle.N = N
.....handle.resume(handle)
>> end
...if filterlist.Value == 7 then
>> N = no(handle.S);
>> imshow(N)
.....handle.N = N
.....handle.resume(handle)
>> end
...if filterlist.Value == 8 then
>> N = negatif(handle.S);
>> imshow(N)
.....handle.N = N
.....handle.resume(handle)
>> end
...if filterlist.Value == 9 then
>> N = prewit(handle.S);
>> imshow(N)
.....handle.N = N
.....handle.resume(handle)
>> end
...if filterlist.Value == 10 then
>> N = sobel(handle.S);
>> imshow(N)
.....handle.N = N
.....handle.resume(handle)
>> end
...if filterlist.Value == 11 then
>> N = end(handle.S);
>> imshow(N)
.....handle.N = N
.....handle.resume(handle)
>> end
...if filterlist.Value == 12 then
>> N = robert(handle.S);
>> imshow(N)
.....handle.N = N
.....handle.resume(handle)
>> end
...if filterlist.Value == 13 then
>> N = sobel(handle.S);
>> imshow(N)
.....handle.N = N
.....handle.resume(handle)
>> end
...if filterlist.Value == 14 then
>> N = sobel(handle.S);
>> imshow(N)
.....handle.N = N
.....handle.resume(handle)
>> end
...if filterlist.Value == 15 then
>> N = yellow(handle.S);
>> imshow(N)
.....handle.N = N
.....handle.resume(handle)
>> end
...if filterlist.Value == 16 then
>> N = purple(handle.S);
>> imshow(N)
.....handle.N = N
.....handle.resume(handle)
>> end
end
```

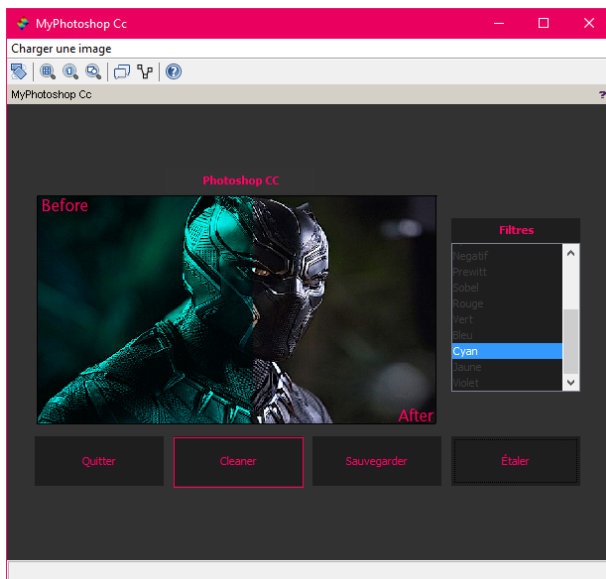
Quant aux trois autres fonctions :

chargement_image

permettra de définir la fonctionnalité du bouton « Charger une image » créé plus haut dans app.sce, (il s'agit tout simplement d'ouvrir le gestionnaire de fichiers sur le dossier images)

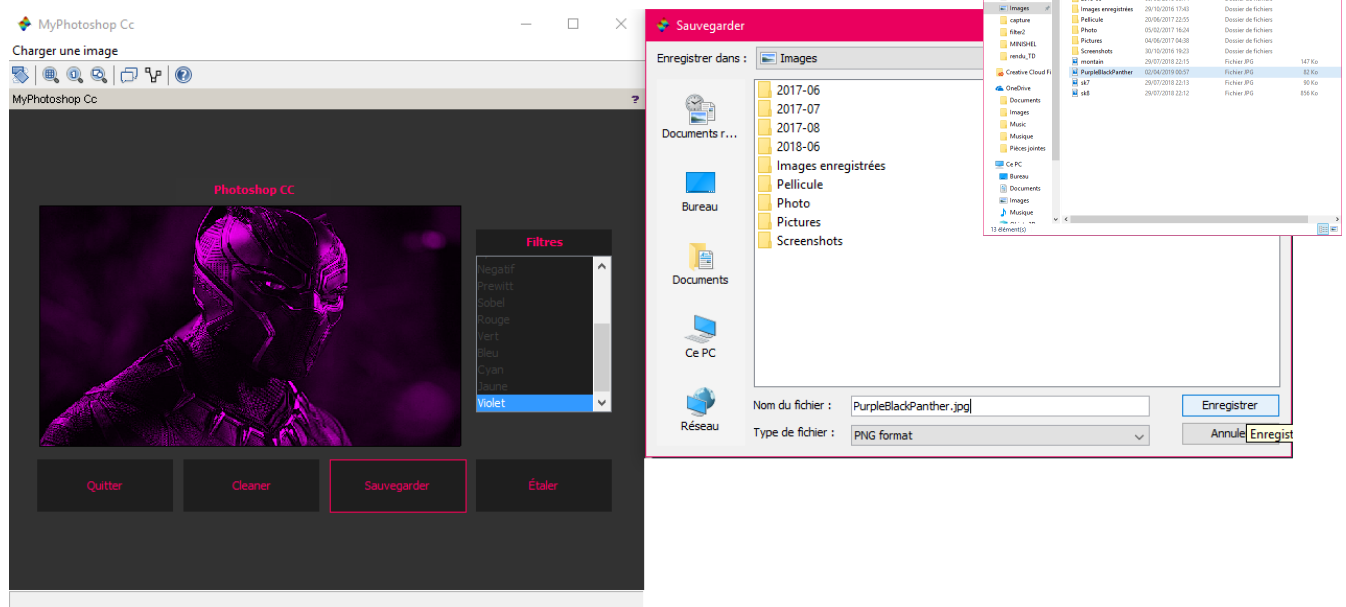


cleaner_picture
permet d'afficher l'image d'origine



sauvegarder_image

enregistre l'image sur le dossier « C:\Users\hedda\Pictures » (il est à noter que, pour que cette fonctionnalité soit remplie, il faut adapter ce chemin de répertoire en fonction du dossier dans lequel l'utilisateur souhaite enregistrer ses images. De plus, il faut toujours écrire manuellement l'extension du fichier lors de la sauvegarde comme ci-dessous, cela évite la fermeture brutale du logiciel).



Voici la représentation du fichier fonctions.sci :

```
//Des-Fonctions

//chargement_image
function.[image S].--chargement_image(formats handle)
> image.--imread(uigetfile(formats "%HOME%" "Charger-une-image"))
> S.--image
> imshow(S)
...handle.S.--S
...handle.--resume(handle)
endfunction

//cleaner_picture
function.[image].--cleaner_picture(handle)
...image.--handle.S
...imshow(image)
endfunction

//sauvegarder_image
function.[].--sauvegarder_image(formats handle)
> imwrite(handle.N.--uioutputfile(formats "C:\Users\hedda\Pictures", "Sauvegarder"))
endfunction
```

C'est ici, que nous déclarerons les 16 fonctions filtres. Pour certaines, nous appliquerons les formules en utilisant les coordonnées de l'images, tandis que pour d'autres nous nous contenterons d'appeler simplement les fonctions prédéfinies.

```
function.cny.-canny(image)
...image.-nb(image)
...cny.-edge(image 'canny')
endfunction

function.y.-bonus(x)
....[n1,n2] size(x)
....y.-[]
....for i=2:n1-1
....for j=2:n2-1
....y(i,j)=(x(i,j)+x(i+1,j+1)+x(i,j+1)+x(i+1,j+1)+x(i+1,j))
....end
....end
endfunction

function.flou.-gaussien(image)
..G.-fspecial('gaussian',27,3);
..flou.-imfilter(image,G)
endfunction

function.hrstl.-horizontal(image)
....[x,y,c].-size(image)
....for i=1:-x/2+1
....for j=-1:-y
....hrstl(i,j).-matrix(image(x-i+1,j):-1,c)
....hrstl(x-i+1,j).-matrix(image(i,j):-1,c)
....end
....end
endfunction

function.hv.-horizontal(image)
....[x,y,c].-size(image)
....for i=1:x
....for j=1:y/2+1
....hv(i,j).-matrix(image(x-i+1,-y-j+1):-1,3)
....hv(x-i+1,-y-j+1).-matrix(image(i,j):-1,3)
....end
....end
endfunction

function.vrtcl.-vertical(image)
....[x,y,c].-size(image)
....for i=1:-x
....for j=-1:-y/2+1
....vrtcl(i,j).-matrix(image(i,y-j+1):-1,c)
....vrtcl(x,y-j+1).-matrix(image(i,j):-1,c)
....end
....end
endfunction

function.img.-nb(image)
> r.-image(: : 1)
> g.-image(: : 2)
> b.-image(: : 3)
> img.-imlincomb(0.299 r 0.597 g 0.114 b)
endfunction

function.ngtff.-negatif(image)
..ngtff.-255.-image
endfunction

function.prwtt.-prewitt(image)
..image.-nb(image)
..prwtt.-edge(image 'prewitt')
endfunction

function.sbl.-sobel(image)
..S.-fspecial('sobel');
..sbl.-imfilter(image,S)
endfunction

function.rouge.-red(image)
..Blanc.-uint8(zeros(image(: : 1)))
> rouge.-image(: : 1)
> rouge(: : 2).-Blanc
> rouge(: : 3).-Blanc
endfunction

function.vert.-green(image)
..Blanc.-uint8(zeros(image(: : 1)))
> vert(: : 1).-Blanc
> vert(: : 2).-image(: : 2)
> vert(: : 3).-Blanc
endfunction

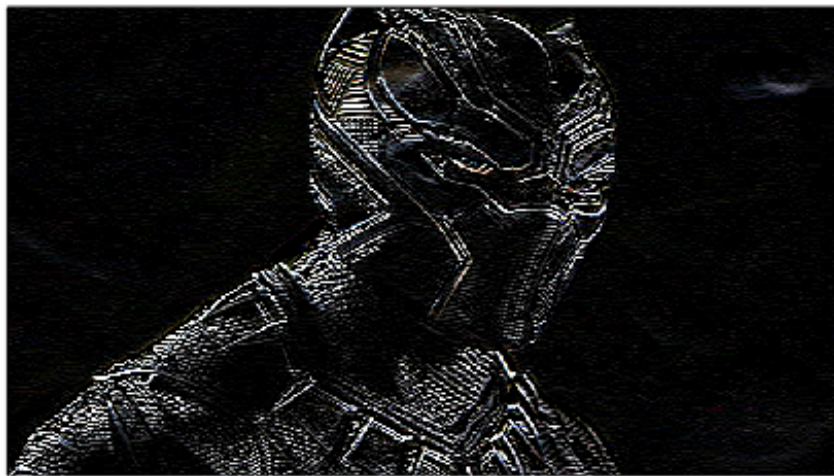
function.bleu.-blue(image)
..Blanc.-uint8(zeros(image(: : 1)))
> bleu(: : 1).-Blanc
> bleu(: : 2).-Blanc
> bleu(: : 3).-image(: : 3)
endfunction

function.cyn.-cyan(image)
..Blanc.-uint8(zeros(image(: : 1)))
> cyn(: : 1).-Blanc
> cyn(: : 2).-image(: : 2)
> cyn(: : 3).-image(: : 1)
endfunction

function.violet.-purple(image)
..Blanc.-uint8(zeros(image(: : 1)))
> violet(: : 1).-image(: : 1)
> violet(: : 2).-Blanc
> violet(: : 3).-image(: : 1)
endfunction

function.jaune.-yellow(image)
..Blanc.-uint8(zeros(image(: : 1)))
> jaune(: : 1).-image(: : 1)
> jaune(: : 2).-image(: : 1)
> jaune(: : 3).-Blanc
endfunction
```

The screenshot shows the Adobe Photoshop CC interface. The top menu bar is visible with the 'Filtres' (Filters) menu open. The 'N/B' (Noise & Blur) filter is selected. The main canvas displays a portrait of Iron Man. To the right, a grid of 10 small images shows the results of applying various filters: Canny, Bonus, Horizontal, Vertical, Horizontal, Vertical, Horizontal, Vertical, Gaussien, and Prewitt. The 'N/B' filter is highlighted in the menu, and its result is shown in the bottom right of the grid.



Sobel

Bibliographie :

<https://www.scilab.org/news/tutorials>

<http://www.ai.univ-paris8.fr/~boubchir/TSI/>