Transparency Suite: Example Filters made with Filter Factory

by Ky Hopwood 8/31/94

The Transparency Suite are a set of example filters made with Filter Factory that edit a layer's transparency data. This set of filters are not so much special effect filters but rather utility filters. The filters are as follows:

Filter Name	Settings File Name
Add Remove Transparency	EditTrans.afs
Set Transparency Level	SetTrans.afs
Luminosity->Opacity	LumOpac.afs
Opacity->Luminosity	OpacLum.afs

This document discusses what these filters are used for, and how they work (they are simple). Hopefully you will find these useful and will create other related plug-ins.

Why Transparency filters are needed.

Photoshop 3.0 introduced the concept of layers. Each layer is in essence a sheet of acetate that can be stacked and composited (If there is any confusion in this matter, please refer to the Layers section in your manual). All layers with the exception of the Background Layer, is a transparent layer with varying levels of transparency from 0 (completely transparent), to 255 (completely opaque).

Each transparent layer has 2 different types of transparency: layer transparency and pixel transparency. Layer transparency is how much that layer as a whole will affect other layers in the merged view of the document. For instance, you can fill a new layer with 100% opaque black, and set the Opacity of the layer via the Layers Palette to 10% and if the layer viewed over a white layer, the merged view of the document will be a light gray, even though the pixels are 100% black. Layer transparency affects all the pixels in the layer equally. Photoshop's Layers Palette gives you adequate control over this type of transparency, and these filters will not effect this type of transparency.

Pixel transparency is the opacity of each pixel. This is a separate entity from the layers transparency. To get a feeling for what this does, imagine a single empty layer that has a white background layer beneath it. If you were to set the foreground to black and pull a Foreground to Transparent gradient, what you would get is a layer that looks grayscale. However viewing the Info Palette, you would find that the document is 100% black! The difference is it is also in various levels of opacity. There are few tools inside of Photoshop to edit this information.

While there are many tools to reduce the opacity (such as Fill in clear mode, soft selection delete, eraser tool etc.), there are few tools to increase the opacity. Paint tools or fills will increase opacity, but they have other disadvantages such as brush strokes, and permanently altering the data. It is easiest to describe the lack of control by giving an example.

Let's say I were to sketch a head with black as my foreground color using a table with opacity set, as if I were doing a sketch with a pencil on paper. The result would look grayscale, but if I were to examine it with the Info Palette, I would discover that the file is completely black. This may make it hard to

edit the file for other uses, for example how can I increase/decrease the contrast if all the pixels are the same color. These filters are made to deal with these issues. They are not filters to amaze & woo your friends, but they can be quite helpful while working with Photoshop.

Note: some things to keep in mind when working with these filters.
- To use these filters, Preserve Transparency must be off. However Preserve transparency is often useful in editing these files at other times, so be aware of it's state.
- Sometimes using these filters results in garbage from very transparent data suddenly rearing it's ugly head when it is made completely opaque. Consider editing the following filters to have a threshold setting.

The Filters Examples:

Set Transparency:
This filter uniformly sets all pixels in a layer to a uniform opacity. This can be useful for creating mask for another layer. The formula is as follows:

```
R: r
G: g
B: b
a: a>0 ? ctl(1):0
```

This function is really simple. We keep the color channels the same. The a channel is used to describe the transparency of that pixel. 0 denotes completely transparent, and 255 means 100% opaque.

Thus this means
a>0   - evaluated to TRUE if pixel has some opacity (in other words, greater than 0 value)
?   - marks division between conditional expression and choices 1 & 2
ctl(1)   - if expression was true, set the opacity value to ctl(1)(that is slider 1). (Make certain to check slider 1
            in the make dialog!)
:   - marks division between choices
0   - If it was transparent, keep it transparent

That's all.

Add Remove Transparency:
This filter allows you to either increase or decrease each pixel relative to it's current opacity. A useful aspect of using this is that data is not lost. (Warning however, you can reduce the range of your opacity by setting them all to 0 or 255.) You are editing the invisible transparency mask, not the actual data on your layer. This means that even if you reduce the Opacity to 0 by using this, you could still get your original data by running the Set Transparency filter or modifying this one to recognize 0 transparency pixels. The formula is as follows:

```
R: r
G: g
B: b
a>0?ctl(1)<128?a - (256 - (2 * (ctl(1) + 1))):a + (2 * (ctl(1) -128)) : 0
```

This means
keep r,g,b channels the same
a>0   - evaluated to TRUE if pixel has some opacity (in other words, greater than 0 value)
?   - marks division between conditional expression and choices 1 & 2
ctl(1)<128   - evaluate to TRUE if slider < 128. Left side of the slider should take away opacity, right side
                        should add opacity.

NOTE: In tis example, the middle of the value should really be 0 instead of 128, and range -128 to 128. However this is a limitation of this version of Filter Factory. It looks strange, but it works.

?   - marks division between conditional expression and choices 1 & 2

```
a - (256 - (2 * (ctl(1) + 1)))
```
                       - This allows to reduce to the opacity. It works as follows from right to left. (ctl(1) + 1) changes the value from ctl from 0 - 127 to 1 - 128 . (2 * (ctl(1) + 1)) This scales our value to 256 so that we can subtract the full amount from the a channel value. `a - (256 - (2 * (ctl(1) + 1)))` subtracts our new slider derived value from the current a channel value. Or the opacity value of that pixel has the slider value subtracted from it.
:   - marks division between choices of left/right slider
```
a + (2 * (ctl(1) -128))
```
                       - Essentially the same as the above, but it is rescaling the values so that 128 - 255 map to 1 to 256. The logic is about the same.
:   - marks the division between whether the original a channel pixel was completely opaque
0   - If it was transparent, keep it so.

Luminosity->Opacity & Opacity->Luminosity
These two are a bit stranger, but can be quite interesting at times. The first will take a layer and fill it with pure black in various layers of opacity. Opacity->Luminosity does just the opposite in that it takes the transparency information of the layer and changes it to grayscale at 100% opacity. The reason for this filter is that sometimes I find it useful to have a file with many layers of transparency, and at other times I want color values. As an example: If I drew a sketch in black with various levels of opacity, I cannot alter contrast with Levels because it is 100% black. However I can use this filter to convert the transparent data to flat grayscale, use levels, the convert back to 100% black image with variable opacity. Play with it.

Luminosity->Opacity

```
R:  0
G:  0
B:  0
a:  a>0 ?255 - (r+g+b)/3:0
```

This means:

```
R:  0
G:  0
B:  0          - Note how the R,G,B channels are converted to pure black
a>0 ?255 - (r+g+b)/3:0
```
                       - Again, we use an a>0 statement to find if the pixel is has any opacity, if not we ignore it. Then we calculate the approx luminosity by averaging the 3 channels ((r+g+b)/3) for each pixel. This appoximates the pixels luminosity and is good enough for our purposes. We then subtract that value from 255 so that a white value will now = clear(or transparent), and a black value will now bee 100% opaque.

Opacity->Luminosity

```
R:  a>0 ? 255 - a: r
G:  a>0 ? 255 - a: g
B:  a>0 ? 255 - a: b
a:  a>0 ? 255 : 0
```

This means:

255 - a   - If pixel is transaprent, we ignore it. Then 255 - current a channel value gives the proper gray value
a: a>0 ? 255 : 0   - pixel is 0, we ignore it. Else we make it 100% opaque

Conclusion:

Adobe Photoshop gives you a lot of power with transparency. Filter Factory gives you much more flexibility on how to alter/edit your pixel transparency values. These four filters are meant to give some idea how to do this. These were done very quickly just to help me do what I wanted to do so pardon if they are a bit sloppy. For the most part however they work...enjoy.

Ky (no I am not Kai Krause) Hopwood