

effects

COLLABORATORS

	<i>TITLE :</i> effects		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		April 12, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	effects	1
1.1	main	1
1.2	AddAlpha	2
1.3	AddConstant	3
1.4	AutoCrop	3
1.5	Bitfield	4
1.6	Brightness	5
1.7	Composite	5
1.8	Contrast	7
1.9	ColorChange	7
1.10	ColorMix	8
1.11	Convolute	9
1.12	Crop	11
1.13	EmbedFile	11
1.14	Extract	14
1.15	ExtractFile	14
1.16	Flip	15
1.17	Gamma Correction	15
1.18	Greyscale	16
1.19	Histogram Equalization	16
1.20	Median	17
1.21	Negative	18
1.22	Noise	18
1.23	Resize	19
1.24	Remove Isolated Pixels	19
1.25	Roll	20
1.26	Rotate	20
1.27	Scale	21
1.28	Shift	22
1.29	Text	22

1.30 Threshold	23
1.31 Transparency	24
1.32 TrueColor	25
1.33 unnamed.1	25

Chapter 1

effects

1.1 main

MAIN INDEX OF EFFECTS

AddAlpha

AddConstant

AutoCrop

Bitfield

Brightness

ColorChange

ColorMix

Composite

Contrast

Convolute

Crop

EmbedFile

Extract

ExtractFile

Flip

Gamma

Greyscale

HistEq
Median
Negative
Noise
Resize
RIP
Roll
Rotate
Scale
Shift
Text
Threshold
Transparency
TrueColor
(* = Documentation not yet complete
or out of date.

1.2 AddAlpha

NAME

AddAlpha -- adds an alpha channel to the image or replaces old.

REXX TEMPLATE

AddAlpha ALPHA/A/N

GUI

None.

DESCRIPTION

AddAlpha takes the target image and makes it into an ARGB image. The alpha channel values are taken from the alpha image specified by the parameter.

If the target image already is an ARGB image (i.e. it has an alpha channel), the existing alpha channel values are replaced by the new values.

Note that you cannot start this effect from the Effects list. You can start it from REXX or by dragging a frame from the master frame list into the target image window.

BUGS

SEE ALSO

1.3 AddConstant

NAME

AddConstant -- adds a constant over the area.

REXX TEMPLATE

AddConstant RED/N, GREEN/N, BLUE/N, ALPHA/N, GREY/N

Each of the values should be between -255 and 255. Only the relevant attributes are used: for example, if you set the GREY value for an ARGB image, the value is ignored.

GUI

Use the sliders.

DESCRIPTION

This simple effect takes the image and adds the arguments in each pixel. You can use this to make a brightness correction, for example.

BUGS

SEE ALSO

Brightness

1.4 AutoCrop

NAME

AutoCrop -- Remove borders from an image

REXX TEMPLATE

AutoCrop XMARGIN=X/N, YMARGIN=Y/N

GUI

Use the sliders to set the amount of borders you wish to retain.

DESCRIPTION

You can use this effect to remove any 'borders' the image may have. It scans the image for a constant color border and then removes the border.

You may also set the amount of border pixels retained around the image. A value of zero means that no border remains. Note that you can use this to resize the image in the same way as the Resize - effect does. However, there is no guarantee of the final image size, but you might find this side effect useful, when you want several

images to have an equal amount of border.

BUGS

SEE ALSO

Crop

1.5 Bitfield

NAME

Bitfield -- Extract bitfield information.

REXX TEMPLATE

Bitfield LOW/N,HIGH/N

GUI

Use the sliders to set the low and high bits.

DESCRIPTION

You can use this effect to remove the high or low bits of an image. This may not seem to be very useful, but you can for example check for the amount of noise in your images by displaying the low pixels only.

Bits are numbered so that zero (0) is the least significant bit and seven (7) is the highest significant.

If you didn't understand this explanation, you probably should read some books about computer graphics in general, as well as about binary numbers.

EXAMPLES

Shows the least significant bit of the image:

Bitfield 0 0

Shows the two least significant bits of the image:

Bitfield 0 1

Shows the most significant bit of the image:

Bitfield 7 7

BUGS

Probably many, since I was in a fever while I wrote this. You know, the end user almost never realizes what the author has gone through when writing software.

SEE ALSO

EmbedFile

.

1.6 Brightness

NAME

Brightness -- Brightness adjustment

REXX TEMPLATE

Brightness AMOUNT

AMOUNT must be between -1.0 and 1.0.

GUI

Use the slider to select the change in brightness, or enter the floating point value into the gadget.

DESCRIPTION

This changes the general brightness of the image. The smaller the value is the darker the image becomes. 0.0 means no change.

BUGS

NOTES

Since v1.7 this effect accepts floating point values instead of direct values.

SEE ALSO

Contrast

1.7 Composite

NAME

Composite -- composite images.

REXX TEMPLATE

Composite WITH/A/N, TOP/N, LEFT/N, METHOD/K, RATIO/N, TILE/S

WITH - the frame id you wish to composite onto this image, ie. the source image.

TOP - the top co-ordinate of the image's top left corner onto this image.

LEFT - the left co-ordinate of the image's top left corner onto this image.

METHOD - This may be any of the following:

DIRECT
MINIMUM
MAXIMUM

ADD
SUBTRACT
MIX
TRANSPARENT BLACK
MULTIPLY
DIVIDE

RATIO - If METHOD is MIX, then you can use this to select the mixing ratio between two images.

TILE - causes the source image (WITH image) to be tiled onto the destination image. (v1.5)

GUI

You will see a box with a cross in the image. Use the mouse to move it around.

Choose the methods with the cycle gadgets.

DESCRIPTION

There are several compositing methods available:

Direct - the source image values directly replace the values in the destination image.

Minimum - the smaller value from either image is used.

Maximum - the bigger value from either image is used.

Add - the values from the images are averaged.

Subtract - the value from the source image is subtracted from the value in the destination image.

Mix - The resulting pixel value is calculated as a weighted average of the source and destination images so that if Mix ratio is 100%, the resulting pixel will be taken from the source image only and if the ratio is 0 %, the resulting pixel will be taken from the destination image only. A value of 50 % would take an equal amount from each image.

Transparent Black - If the source image pixel is black, then we'll use the destination image value, otherwise the source image.

Multiply - The source and destination images will be multiplied together and the resulting image is scaled to the pixel range.

Note that you cannot use this effect directly. You can use only either a REXX script or drag-and-drop an image from the main list into an another image.

Of course, if the source image has an alpha channel, it will determine which parts of the image will be transparent. In this case I suggest you use the Direct mode.

Tiling means that you can tile the source image over the destination image, i.e. repeat the source image throughout the entire selected

area.

BUGS

SEE ALSO

Transparency

1.8 Contrast

NAME

Contrast -- Contrast correction

REXX TEMPLATE

Contrast AMOUNT/N

GUI

Select the amount of correction you wish using the slider gadget.

DESCRIPTION

The minimum value of correction is -100 and the maximum is +100. At the minimum value all contrast is lost and all of the image is turned dull grey. At the maximum value the image becomes a thresholded image for maximum contrast.

BUGS

SEE ALSO

Brightness

1.9 ColorChange

NAME

ColorChange -- Changes a color within a given tolerance.

REXX TEMPLATE

SRCRED/N/A, SRCGREEN/N/A, SRCBLUE/N/A, DESTRED/N/A, DESTGREEN/N/A,
DESTBLUE/N/A, TOLERANCE/N

Default tolerance is zero.

GUI

On the left side, you see a color chooser for the source color and on the right side, you see a color chooser for the destination color. On the bottom you can set the tolerance.

If you click on the source image, the color of the pixel that is currently under the mouse pointer will be copied to the source color chooser.

DESCRIPTION

This little effect will change any color to an another color. The decision whether a pixel will be changed or not is determined by the color-cube distance between the image pixel and the source color. If this distance is smaller than or equal to the tolerance, then the color is changed.

If you set the tolerance level to zero, only exact colors are changed.

BUGS

SEE ALSO

1.10 ColorMix

NAME

ColorMix -- Mixes up several colors.

REXX TEMPLATE

```
COLORMIX R=RED, G=GREEN, B=BLUE, A=ALPHA
```

Each of the arguments RED, GREEN, BLUE and ALPHA should contain a string, which has the factors (in percents) separated by commas in the RGBA order, for example:

```
PROCESS frame COLORMIX RED "0,1.0,0.5" GREEN "0,0,0"
```

which would calculate the new values (marked with R', G' and B') as follows:

```
R' = 1.0 * G + 0.5 B
G' = 0
B' = B
```

GUI

Use the sliders to control the factors. Each of the groups represents the new color, and each of the sliders represents how much of the particular color should be included in this.

DESCRIPTION

This is quite a powerful effect. You can use this to manipulate the colors inside an image, based on other primary colors of the same pixel.

If this explanation sounds strange, then look at the below example. I have marked the new colors with R', G', and B' and the old colors in the image with R,G and B.

For example, if you would like to achieve a false color effect, ie. to make red colors into blue, blue into green and green into red, you could use the following REXX command to do this:

```
PROCESS frame COLORMIX RED "0,1.0,0" GREEN "0,0,1.0" BLUE "1.0,0,0"
```

This would calculate the new color values like this (1.0 = 100%):

$$\begin{aligned} R' &= 0.0 * R + 1.0 * G + 0.0 * B &=> R' = G \\ G' &= 0.0 * R + 0.0 * G + 1.0 * B &=> G' = B \\ B' &= 1.0 * R + 0.0 * G + 0.0 * B &=> B' = R \end{aligned}$$

which was exactly what you wanted.

You can also use the alpha channel of an image (if the image is an ARGB image) by specifying it last. This will change the alpha channel of an image to visible by making it red.

```
PROCESS frame COLORMIX RED "0,0,0,1.0" GREEN "0,0,0,0" BLUE "0,0,0,0" ALPHA ↔
    "0,0,0,0"
```

Note: The resulting values are clamped to a range of [0,255] and thus if you use a value of 1.0 for all colors, this will have the effect of brightening the image somewhat.

BUGS

SEE ALSO

AntiqueTint.prx script in the PROGDIR:Rexx subdirectory.
FalseColour.prx script in the PROGDIR:Rexx subdirectory.

1.11 Convolute

NAME

Convolute -- Do a convolution between a mask and the image.

REXX TEMPLATE

```
Convolute FILENAME/A
```

FILENAME - the convolution to apply. This file is first searched in "PROGDIR:" and then in the "PROGDIR:modules/convolutions/" subdirectory.

GUI

On the left side, you'll see a 7x7 matrix, that contains the matrix you wish to convolute the image with. Below it, you can enter the bias and the divisor.

The right side of the window contains several buttons. You can load or save the current convolution matrix by clicking on the respective buttons. "Clear" will reset the matrix. Click on "Convolute!", when you are done setting your convolution matrix.

There's also a menu that contains also the commands that are available as buttons. There are also some nice presets.

DESCRIPTION

Convolutions are a bit tricky subject. You can probably find a great deal of information from any good book that discusses image processing in detail. I'll try to run a short explanation here:

A convolution matrix is a matrix that is laid on top of a pixel and each value around it is multiplied by the corresponding weight in the matrix. Then they're all summed together. For example if the (3x3) convolution matrix is:

```
0 1 0
1 2 1
0 1 0
```

and the pixel values in the same neighbourhood are (we're processing pixel valued '8'):

```
5 5 9
4 8 3
0 3 0
```

The resulting image would be:

```
5 * 0 + 5 * 1 + 9 * 0
      +
4 * 1 + 8 * 2 + 3 * 1
      +
0 * 0 + 3 * 1 + 0 * 0 = 0+5+0+4+16+3+0+3+0 = 31. So, the result
pixel is 31 and so we change the 8 to 31 and move on to the next
pixel.
```

What's the point? Well, the theory is a bit complicated to get into here, but basically you can do low- and high-pass filtering, gradient calculation and lots of other cool stuff with this.

Oh, one more thing: Since you can use only whole numbers in the matrices (floating point is such a hassle) you need to provide a divisor. The resulting pixel is divided by the value of the divisor, so that the resulting pixel value is better scaled within the limits of the pixel (that is, between 0 and 255). Normally, you should use the value of 1. Aw hell, look at the examples provided.

Bias: The bias value is added to the pixel value after applying the divisor. The point is to make some operations (like some high-pass filters) to give a bit brighter image. You can also use a negative value.

BTW, take a look at the menus - you'll get some nice presets for different types of filters.

Loading and saving: Well, not much to say. You can make, save and load your own convolution matrices. This module defaults to using the modules/convolutions -directory for it's files.

BUGS

SEE ALSO

1.12 Crop

NAME

Crop -- crops to the selected area.

REXX TEMPLATE

CROP

GUI

None.

DESCRIPTION

Crops to the selected area.

BUGS

SEE ALSO

Autocrop

1.13 EmbedFile

NAME

EmbedFile -- Embeds a file in the image, aka. steganography.

REXX TEMPLATE

EmbedFile FILE/A,PASSPHRASE/K

FILE = file to be embedded

PASSPHRASE = Passphrase that can be up to 255 characters.

GUI

Choose the file to be embedded using the ASL file requester, then enter the passphrase you wish to use.

DESCRIPTION

This implements a technique called "steganography". The idea is to embed information into an image, sound, or another file in general so that it would be invisible to the human observer. If the information can be made to look like noise, then the information can be made virtually undetectable.

Images are a good place to hide things, since they often contain noise in their lower bits, that is, it is pretty much the same if the image has an intensity value of 40 or 41. The human eye cannot see the difference. We can use that to our advantage and make the least significant bits mean something. Further on, I will use the word 'cover' to mean the image in which the information is to be hidden.

The file (hereforth called 'message') is broken into bits and each

bit is embedded into the image file. If we used the first few bytes of the image, it would be plainly detectable by using statistical analyses, etc. For this reason, the bits are spread randomly over the image, using a hash function. The PPT implementation uses the Secure Hash Algorithm from NIST to make a pseudorandom distribution that has good randomness.

To increase security, PPT will use triple-DES encryption on the data it hides. Note that this may make using the EmbedFile effect illegal in your country!

PPT saves the filename with the image.

PASSPHRASES AND COVERS

The key to the message is calculated using both the passphrase you've supplied and the cover, thus making sure no two cover files with the message have the same distribution. Otherwise, it would be easy to deduce the changed bits and possibly reconstruct the message.

Selecting a good passphrase is important. Use these following rules to select your passphrase:

- Use at least 8 characters, preferably more. Since each character provides approximately 6 bits of variation, this will result in 48 bit keys. You should use at least 64 bits (~11 characters) for anything serious communication. Shorter keys are vulnerable to brute force methods (go see <http://www.distributed.net/> if you don't believe me!)
- Do not use words that have meaning in any language, or use long phrases, such as "Nobody listens to poor Zathras." Or better, mix up words from songs ("Jingle Tannebaum, Jingle Tannebaum"). A good passphrase would also have special characters and different capitalization ("HyPERACTIVE???? Who ARE yOu CallInG HyPerActiVe????")

You have 255 characters, use them... =)

- NEVER EVER USE commonly known personal things, like your name, your cat's name, your spouse's name, your license plate number, etc. They're way too easy to guess.

The cover files should also be chosen carefully. While that computer generated image of the cute reflective teapot with the black border may be nice to watch, it will probably make a bad cover, since computer-generated images tend to have too little variation. They have long runs of the same color and they are very predictable. A good cover image has lots of variation, lots of details and most of all - lots of noise. We want to hide the message as noise; if the image is not noisy to start with, the sudden presence of noise would arouse suspicions, yes?

Thus: Use the following guidelines:

- Use scanned images. They usually have their lowest bits as pure noise.

- Never use GIF files. They are converted from 256 colors which makes any variation easy to spot. Same goes for bitmapped IFF files (not the 24-bit ones).
- NEVER, EVER use the same cover image twice. It would be trivial to subtract the images from each other to find the secret messages.
- Destroy the original cover image after embedding the file. If someone gets hold of the original, it's too easy to find the message.
- Use the
 Bitfield
 effect to check if the lowest bits of the image
are noise or not. You'll see easily if that is the case.

MORE NOTES ON SECURITY

A couple more security things:

- Always use encrypted data for the message. It makes them more difficult to spot by statistical methods. Even though PPT crypts the data itself, DES cannot be considered very secure anymore. Use PGP or IDEA. Also, it might be a good idea to compress the data using XPK, GZIP, LHA or what-have-you.
- Do not process the image in any way after embedding a file. You'll lose the message.
- Use only truecolor packing formats such as PNG. Any colormapped format will destroy the message. Don't use JPEG, since it will lose information on the image.

BUGS

This can grow quite inefficient at times, due to the large number of random seeks within the image. You can speed it up considerably by using large VM buffers!

The algorithm is not compatible with anything else. I will probably release a standalone client that can be compiled on any platform.

Passphrase is visible when typing. It is also not checked in case of typing errors.

It may be possible that using this software is illegal in your country. In that case I pity you. Why don't you do something about it?

AVAILABILITY

If I included this effect with the main archive, it would be illegal to download the main archive, if you happen to be French, for example. This is why I've made a separate archive, available at http://www.iki.fi/%7ejalkanen/PPT/PPT_security.lha that contains the EmbedFile as well as the ExtractFile effects.

SEE ALSO

Article by Tuomas Aura at <http://www.hut.fi/%7Eaura/>, called "Practical invisibility in digital communication".

```
ExtractFile
',
Bitfield
```

1.14 Extract

NAME

Extract -- extract color information.

REXX TEMPLATE

Extract RED/S, GREEN/S, BLUE/S, ALPHA/S

GUI

Use the checkboxes to select the color components.

DESCRIPTION

This simple effect will keep those color components you specify in the image and discard all the rest.

Naturally, the resulting image will be black if you discard all of the components.

BUGS

SEE ALSO

1.15 ExtractFile

NAME

ExtractFile -- extracts the file embedded with EmbedFile.

REXX TEMPLATE

ExtractFile DIR/A, PASSPHRASE/K

GUI

First, enter the pass phrase. Then choose the directory to which the file is to be saved.

DESCRIPTION

Extracts the file embedded using
EmbedFile
. For more information,
see
EmbedFile
docs.

AVAILABILITY

If I included this effect with the main archive, it would be illegal to download the main archive, if you happen to be French, for example. This is why I've made a separate archive, available at

http://www.iki.fi/%7ejalkanen/PPT/PPT_security.lha
that contains the EmbedFile as well as the ExtractFile effects.

BUGS

SEE ALSO

EmbedFile

1.16 Flip

NAME

Flip -- flip an image.

REXX TEMPLATE

Flip X/S,Y/S

GUI

You may select which way to flip the image. X - as usual - means horizontal and Y vertical direction.

DESCRIPTION

Flip the image.

BUGS

SEE ALSO

1.17 Gamma Correction

NAME

Gamma -- Provide a gamma correction feature.

REXX TEMPLATE

Gamma AMOUNT/A

AMOUNT should be the gamma correction (must be > 0).

GUI

Enter the gamma value using the slider or the string gadget.
A value of 1.0 means no correction.

DESCRIPTION

Due to the unlinear nature of display devices, a gamma correction must be used. Most of the time this is done automatically by the display device, but occasionally you'll meet images that could benefit from additional gamma correction.

Gamma is usually defined as

g

F := A

where F = new value, A = old value and g = gamma correction. Obviously, gamma is a real (floating point) number and a value of 1.0 means no correction at all.

Basically, if you wish your image to be brighter, give a gamma correction over 1. If you wish to darken it, give a gamma correction of <1.

BUGS

NOTES

Since v1.4 this effect accepts floating point values.

SEE ALSO

Brightness
,
Contrast
.

1.18 Greyscale

NAME

Greyscale -- strip color information from the image.

REXX TEMPLATE

Greyscale

GUI

None.

DESCRIPTION

This will strip the color information from the given area.

If the whole image is selected, then the image format is changed into greyscale as well.

BUGS

SEE ALSO

TrueColor

1.19 Histogram Equalization

NAME

HistEq -- Perform histogram equalization

REXX TEMPLATE

```
HistEq LOCALRADIUS/N
```

GUI

Use the cycle gadget to select whether you wish to have a local or a global equalization. If you choose local, then you can use the slider to select the radius.

DESCRIPTION

This equalizes the histograms in the given area of the image, which usually improves the image contrast. However, the colors will suffer from this.

For a description of the algorithm, see any good textbook on digital image processing.

Note that the local histogram equalization is quite slow, since for it to be effective, you will need to set the radius to a quite large number. Radius in this case refers to a $\text{Radius} \times 2 + 1$ neighbourhood area, ie selecting a radius of 16 means that the local equalization will be done using a 33×33 local neighbourhood.

BUGS

SEE ALSO

1.20 Median

NAME

Median -- Do a median filter on the selected area.

REXX TEMPLATE

```
Median RADIUS/N/A
```

GUI

Use the slider to select the radius you wish to use for the median filtering.

DESCRIPTION

The idea of median filtering is to take an environment around a pixel and then sort the values. The middle value (also known as the median) is then selected to represent this pixel. For example, if we use a 3×3 matrix (ie radius = 1) and we look at the pixel surroundings we see:

```

3   5   78
7   8   9
0   2   1
```

We'll now sort these values and get : 0, 1, 2, 3, 5, 7, 8, 9, 78. It can be easily seen that the middle value is 5. As you can see, the anomalous value of 78 was filtered out. Median filtering removes strong 'spikes' quite well.

This method is quite good at removing noise and speckles from an

image, but it will cause some blurring of edges.

Currently, this method can use a maximum radius of 5 pixels to each direction (which makes the filtering matrix 11x11 pixels in size).

BUGS

SEE ALSO

Noise

1.21 Negative

NAME

Negative -- make a negative out of the current area

REXX TEMPLATE

Negative

GUI

None

DESCRIPTION

Make a negative out of currently selected area.

BUGS

SEE ALSO

1.22 Noise

NAME

Noise -- Add noise to the image.

REXX TEMPLATE

Noise AMOUNT/N/A,FREQUENCY=FREQ/N,TYPE/K

Values are given in percents.

Type may be either "ADDITIVE" or "MULTIPLIKATIVE".

GUI

Use the sliders to determine the amount of noise in percent and the frequency in percents.

DESCRIPTION

Additive noise is generated by adding a random value to the pixel value and then clamping it to the pixel range. This produces a flat distribution of noise to the image.

Multiplicative noise is generated by multiplying the random value with the pixel value and then scaling it to the pixel value range.

The noise amounts are given in percents of the full dynamic range of the pixel.

Frequency gives the probability for a given pixel that it gets noise applied. For example, a value of 50 means that every second pixel (on the average) will get noise added/multiplied to it. A value of 100 gives noise on every pixel.

BUGS

Should really support also Gaussian noise.
The user should be able to set if he/she wants to clamp or rescale the values...

SEE ALSO

Median

1.23 Resize

NAME

Resize -- Enlarges an image by adding black borders around it.

REXX TEMPLATE

Resize NEWHEIGHT=NH/N/A,NEWWIDTH=NW/N/A FILL/K

Arguments are in pixels.

FILL can be any of the four allowed types: black, white, border or background. The name can be abbreviated to the first few letters that uniquely define it.

GUI

Use the sliders to select the size, and the cycle gadget to select the method.

DESCRIPTION

Resizes the image canvas by adding extra border to the image. There are four different methods to choose from:

Black - the added border is black.

White - the added border is white.

Background - the background color of the image is used to fill the extra space.

Border - the border pixels of the image are extended to fill the extra space.

BUGS

SEE ALSO

1.24 Remove Isolated Pixels

NAME

RIP - Removes pixels with no neighbours.

REXX TEMPLATE

```
"Remove Isolated Pixels"
```

GUI

None.

DESCRIPTION

This removes all those pixels from an image whose surroundings consist of a single color only.

BUGS

Should really support a tolerance limit for the color.

SEE ALSO

1.25 Roll

NAME

Roll - Rolls the image horizontally and vertically.

REXX TEMPLATE

```
Roll X=XROLL/N, Y=YROLL/N
```

GUI

Quite self-explanatory.

DESCRIPTION

Rolls the image, i.e. it's very much like
Shift
, but the image
wraps around the border. Just try it out, it is not very
difficult to grasp.

Negative X direction means left, positive is right. Likewise,
negative Y direction is towards the top and positive is towards
the bottom.

BUGS

None known.

SEE ALSO

Shift

1.26 Rotate

NAME

Rotate -- rotates an image freely.

REXX TEMPLATE

```
Rotate ANGLE/N/A,INTERPOLATION=INT/K
```

ANGLE must be between -180 and 180 degrees.

INTERPOLATION defines the method of interpolation to be used.

Currently available values are:

"NearestNeighbour" - Use NN mapping. Fast but not very accurate.

"Bilinear" - Use bilinear interpolation. Close enough for government work. Slower than Nearestneighbour, but more accurate.

GUI

Select the angle using the slider and the interpolation method from the cycle gadget.

DESCRIPTION

You can use this to rotate the image from -180 to 180 degrees. The remaining area is filled with black.

Positive angle values mean clockwise and negative values mean counterclockwise.

NOTES

This was rewritten from scratch during PPT release A9. It should work now a lot better and it supports ARGB images too.

Note that this effect uses floating point hardware very much, so if you do not have a FPU, this is gonna be ssslllloowwww...

BUGS

Bicubic interpolation does not yet work, but I'll add it as soon as I can understand the example code.

SEE ALSO

1.27 Scale

NAME

Scale -- Change the size of the image

REXX TEMPLATE

```
Scale NW=NEWWIDTH/N,NH=NEWHEIGHT/N,MODE/K,PERCENT/S
```

Arguments are in pixels, unless the PERCENT switch is given, in which case they're interpreted as a percentage value of the old size.

GUI

You may control the new size of the image by using the slider gadgets. You can choose the units you desire with the cycle gadgets to the right of the dialog box.

If you wish that the aspect ratio of the image is not changed when

you fiddle with the sliders, check the 'Lock Aspect Ratio' checkbox and the height and width will be automatically updated.

The cycle gadget below tells the method you are going to use (see below).

DESCRIPTION

This scales an image using either a Quick method (fast, but blocky result) or a Color Average method, which is slower but produces better results.

The MODE argument from REXX can be either "QUICK" or "AVERAGE" to signify which method you wish to use.

BUGS

SEE ALSO

1.28 Shift

NAME

Shift -- Shifts pixels around in the image.

REXX TEMPLATE

```
Shift X/N,Y/N
```

X and Y should be signed integers.

GUI

Use the sliders to select the amount you wish to shift the image.

DESCRIPTION

Basically moves image to the direction you wish. The space that is revealed is filled with black.

Negative X values mean left, positive mean right. Negative Y values mean up and positive mean down.

BUGS

SEE ALSO

Roll

1.29 Text

NAME

Text -- Prints text onto the image.

REXX TEMPLATE

```
Text STRING/A, FONT, SIZE/N, FLAGS, TOP/N, LEFT/N, COLOR
```

STRING - The string you wish to use.
 FONT - The font name you wish to use.
 SIZE - The font size.
 FLAGS - A comma-separated list of any of the following:
 BOLD, UNDERLINED, or ITALIC. For example:
 TOP,LEFT - The top and left coordinates of the top left
 corner of the text
 COLOR - An RGB value for the text color.

For example, use the following to render a yellow text at
 25 pixels from the top and 25 pixels from the left in bold,
 underlined font:

```
PROCESS image TEXT "Hello!" FONT "topaz.font" SIZE 11 TOP 25 LEFT 25 FLAGS ←
      BOLD,UNDERLINED COLOR=0,255,255
```

GUI

You may set the placement of the text by moving the rectangle around in
 the image. Note that if the 'cross' from the center of the image
 disappears, this means that the entire text won't fit in the image.

Other than that, you may use the selectors to choose the color.
 In OS3.0 systems you will also see a colorwheel.

DESCRIPTION

Use this effect to write text to your images.

BUGS

SEE ALSO

1.30 Threshold

NAME

Threshold -- threshold an image

REXX TEMPLATE

```
Threshold LIMIT/N/A,INTENSITY/S
```

LIMIT must be between 0..255, if the image is a standard RGB or
 Graylevel image.

INTENSITY, if specified, sets the threshold to intensity instead
 of using separate channels.

GUI

Use the slider to set the thresholding limit. Whether you wish to
 use intensity or per-channel thresholding can be controlled with
 the checkbox.

DESCRIPTION

This will turn an image into a bicolor image by setting each value
 that is lower than the threshold to black and each value that is
 higher than the threshold to white. Values that are equal to the

threshold are set to white.

However, note that the thresholding is done on a channel-per-channel basis, unless you set intensity to true, so if you apply this effect to an RGB image, then you'll get an 8-color image!

If INTENSITY has been set, then this will always produce a bicolor image.

BUGS

SEE ALSO

1.31 Transparency

NAME

Transparency -- Manage transparency on an image.

REXX TEMPLATE

```
Transparency R=RED/N/A,G=GREEN/N/A,B=BLUE/N/A,TRANS/N/A,BACKGROUND/S,  
ALL/S,TOLERANCE=TOL/N
```

RED, GREEN and BLUE determine the color that should be transparent.

TRANS is an unsigned integer between 0 and 255 that will be the new transparency value for this image and color.

BACKGROUND overrides R, G and B, and selects the color based on a guess of the image's background color.

ALL changes ALL colors within the selected area to have the same transparency values.

TOLERANCE determines the distance from the RGB values which should also be turned transparent. The distance is the euclidian distance in the RGB color cube. (v1.3)

GUI

Use the slider gadgets to select a color that will be made transparent. The transparency value (0 = solid, 255 = fully transparent) can be determined with the slider below.

If you have colorwheel.gadget and gradientslider.gadget installed you'll also get one of each to help you in determining the color value. These gadgets are standard in OS 3.0 (V39) onwards.

Use the settings in the Presets - menu to select a suitable color default (White, black or background color).

You can also click on the image and the color under the mouse pointer is then copied into the sliders and colorwheel.

DESCRIPTION

This tool will enable you to set a transparent color in the image. It will convert the image into ARGB - colorspace, if needed.

If you ask for a background color, you should realize that it is calculated currently as an average of all corner pixels in the image. If your image does not have a clear background, this is likely to be wrong...

Note that if you change the image after this, your selections may not be correct anymore: for example, a rotate effect may add more black pixels to the image, but these will not necessarily be marked as transparent.

BUGS

SEE ALSO

1.32 TrueColor

NAME

TrueColor -- Change the image into TrueColor

REXX TEMPLATE

```
TrueColor
```

GUI

None.

DESCRIPTION

This is the opposite of the Greyscale effect. It transforms an image from the Grayscale format (8 bit) to full RGB format (24 bit). It does not really change any image information.

BUGS

SEE ALSO

@{ "Greyscale" link Greyscale}

1.33 unnamed.1

NAME

REXX TEMPLATE

GUI

DESCRIPTION

BUGS

SEE ALSO
