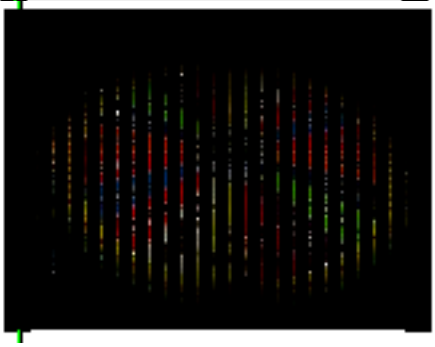
Video Process Gallery.

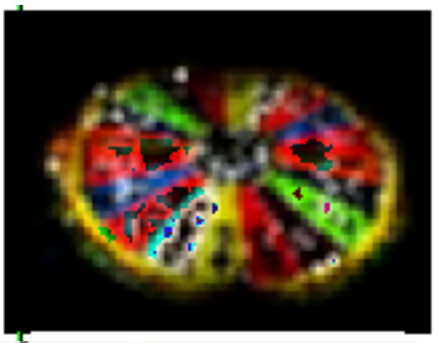
Jitter has many objects that act as video filters of various types. Unfortunately, the names of the objects do not always make it clear what the object does. So here's a list of objects with screen shots that show samples of the output. The best way to learn these is to look at the help files.

The unprocessed images from wheel.mov and chilis.jpg used in these examples.

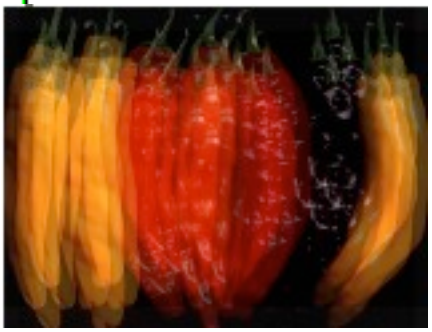
Jit.alphablend
Combine two images with the alpha channel (plane 0) of the left inlet in control. The help files shows how to use a movie color channel for alpha, but for simple crossfade, insert jit.scalebias before the left inlet and adjust abias.



jit.altern
Alternate groups of pixels are erased. You set x and y spacing.



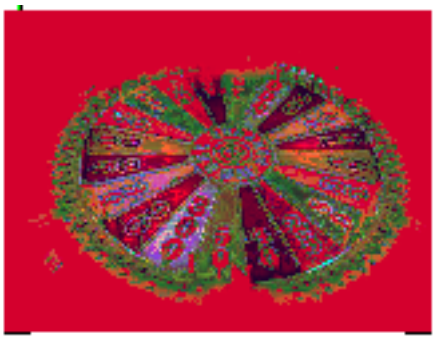
Jit.ameba
This is a pixilation (resampling) trick. The help file admits this was supposed to be something else, but didn't work. Needs a spell check too, but pretty interesting.



Jit.avg4
Replaces each pixel with an average of 4 other cells around it.

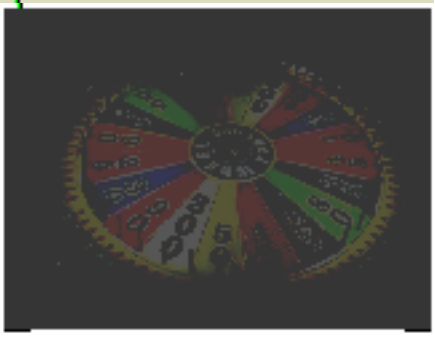
Jit.brass
Uses convolution to get an embossed effect. You set depth and color.

Jit.brcosa
Controls brightness, contrast, saturation. Brightness is a simple multiplication of the RGB values. Contrast adjusts the difference between RG and B. (Contrast can be negative with interesting effects as shown.) Saturation of 0 is a grayscale (same brightness as colors) image, increasing values morph toward the normal image at 1 and max out beyond.



Jit.charmap
Arbitrarily replaces selected values with others of your choice. You control this with a 4 plane matrix that's 1x256. Any 0 value is replaced with what is found in location 0 and so on.

Jit.chromakey
Combines two images by replacing pixels of a specified range of color with the same pixel (by address) from the second image. Jit.chromakey is generalized and very powerful but not as efficient as the other keying objects.



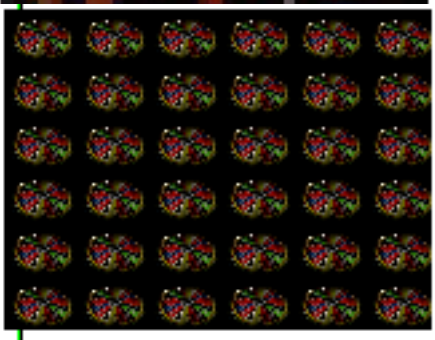
Jit.clip
Limits values to within specified range. This is most useful for preprocessing before another effect.

Jit.concat
Stuffs two images into the same matrix. They can be side by side or one above the other.

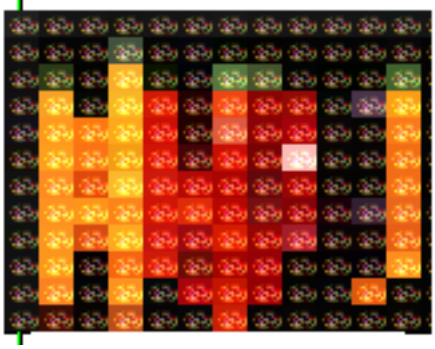
Jit.convolve
Convolution is multiplying every cell in the image matrix by the first cell in another matrix (called the " kernel"), then by the second cell, third, etc. and combining the results. It primarily affects the edges of the image, blurring or possibly sharpening it. This is a very slow process.

Jit.dimmap
Changes the orientation of the data in the matrix, flipping left to right or vertical for horizontal.

Jit.dimop
Breaks a matrix into small sections and applies an operation to all of the values in each section. Ops are average, sum, multiply, min ans max.



Jit.eclipse
Copies the image into a grid of little images.

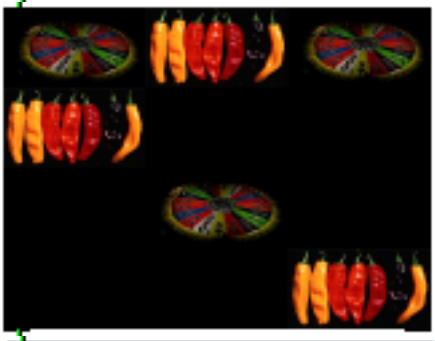


Jit.eclipse (with control image)
A second image into the right inlet tints the sub images to give a hint of the control image.

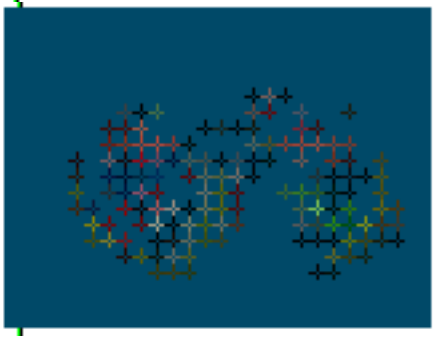
Jit.fastblur
An optimized convolution. Limited to a 3x3 symmetrical kernel, it produces the most useful blurring effects.

Jit.fluoride
Replaces the pixels of a certain luminance with a color of your choice.

Jit.glop
A simple feed back effect, basically increases the persistence of the image (notice how the edge light is wrapped all around the wheel).



Jit.glue
Pastes matrices into an array of images.



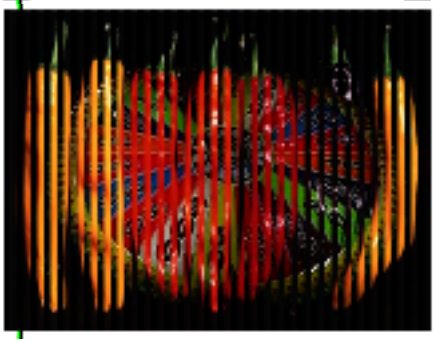
Jit.hatch
Turns pixels into little crosses, with thresholding and background color.

Jit.hue
Rotates all of the colors by a designated angle in HSL colorspace.

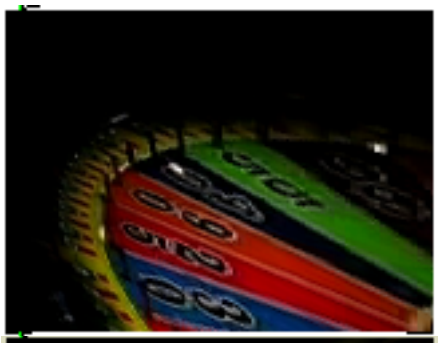
Jit.keyscreen
Combine two images by keying: pixels of a specified color in one image (here it's black) is replaced by the same pixel from another image. This can be restricted to a region of the image by a mask. This object requires much less CPU than chromakey.

Jit.lumakey
Combine two images by keying based on luminance. Unlike jit.keyscreen, the images can fade into each other. Jit.lumakey is also useful for making a mask for jit.keyscreen.

Jit.map
Pixel values within a specified range are converted to fit within a new range. Pixel values outside the acceptance range are clipped to the min or max (or not if you want a lot of bizarre char wrapping effects.)



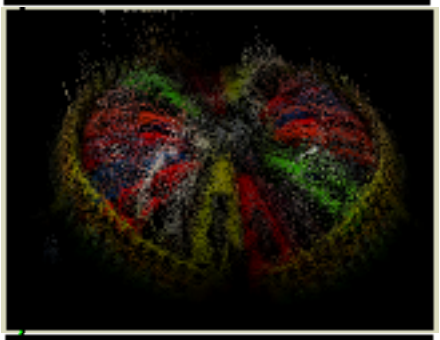
Jit.multiplex
Interleaves two images, slicing and alternating them. You can set the starting intervals of the slices and the width of the slices for various Venetian blind effects. There is a demultiplex object that can get the images back.



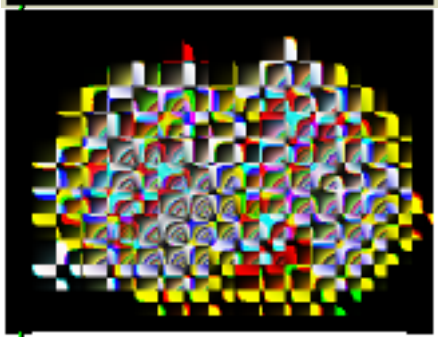
Jit.mxform2d
Performs 2 dimensional matrix transformations on images. With a bit of experimentation, you can get all manner of rotations and perspective angles.

Not to mention distortions. A CPU hog, but worth it.

Jit.normalize
Normalizes an image- i.e. makes the highest value white. (Note how bright this has become.)

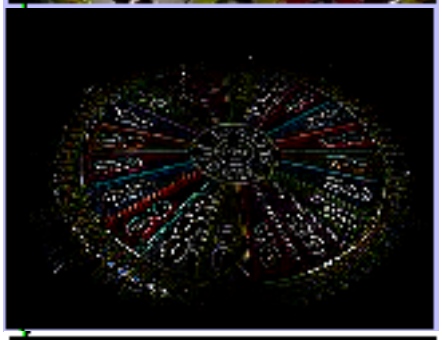


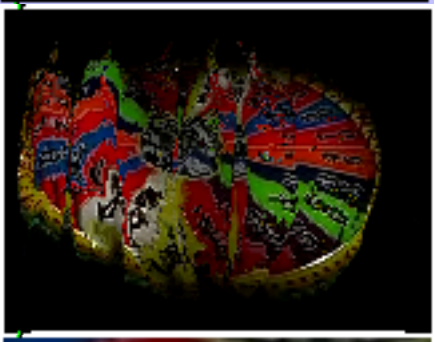
Jit.plume
Moves pixels based on their luminance. It's a neat way to blow things up.



Jit.plur
Pixelation on steroids. The image is downsampled, which converts it to blocks, then the blocks are upsampled in various ways.

Jit.qt.effects
This is a whole bag of effects taken from the Apple Quicktime library. Many duplicate jitter objects, some are simple but useful wipes and crossfades and some are pointless. Use the help file to discover what's available and how the parameters work. Many of the effects have a type parameter- try numbers that aren't listed, as new effects are added with every version of Quicktime. "Tweening" is an automation of the effect. If an effect can be tweened, param_a sets the start point and param_b set the end point. The speed of the effect is set by the steps message. (See the tutorial on Effect Spotting)



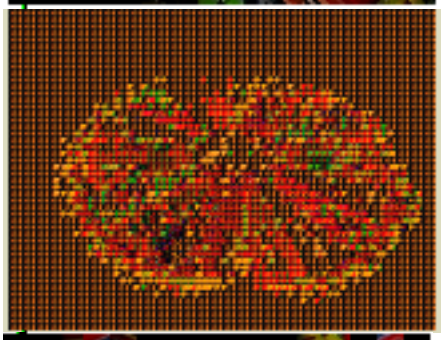


Jit.repos
Repositions pixels according to a control matrix. You can compute a control matrix using any matrix filling technique, or use a movie with a little tweaking.

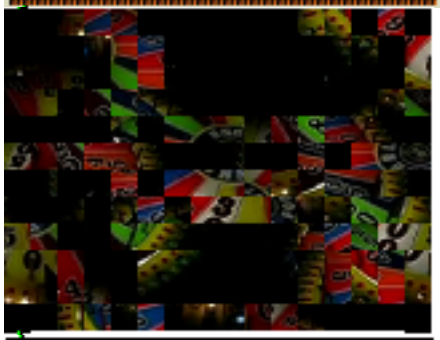
Jit.resamp
Resamples the image, converting it to a new scale. Very efficient for a cheap zoom effect.

Jit.robcross
Edge detection.

Jit.rota
A Swiss Army zoomer and rotator. Set anchor_x and anchor_y to the center of the image size to get symmetrical effects. Rotation is specified with theta, in radians. CPU expensive, but does the job of four other objects.

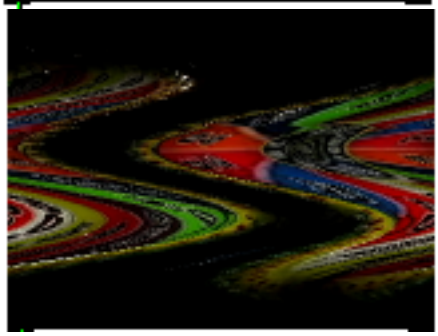


Jit.roy
Halftone screen effect a la Roy Lichtenstein. There are special control matrices to correctly manganize your portrait, but movies work too, with a bit of fiddling.



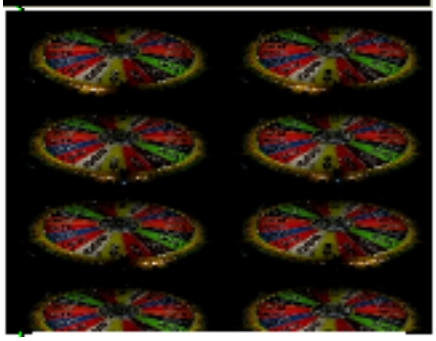
Jit.rubic
Chop the image into squares and rearrange them.

Jit.scalebias
Lets you individually tweak the scale and offset (bias) of each color value.

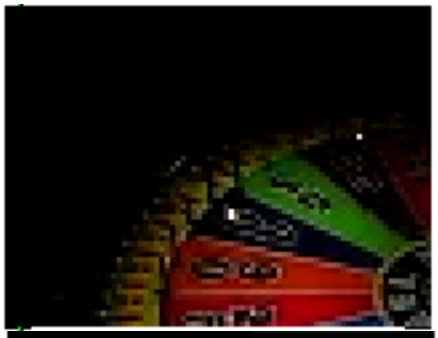


Jit.scanoffset
Slide horizontal or vertical lines around as defined by a control matrix.

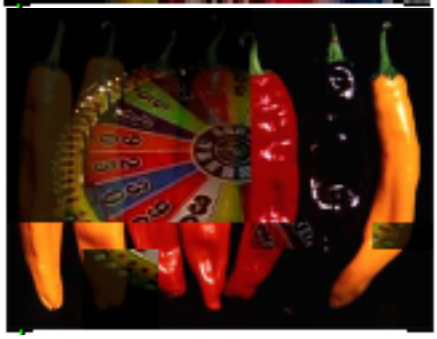
Jit.scanslide
A smear effect. Slide_up smears toward white, slide_down smears toward black. To get symmetrical smearing, set the offset to the middle of the width and use mode 2.



Jit.scanwrap
If you look closely, you will see that this is not one image repeated, but 8 frames of the movie. Usually when you transfer data from a small matrix to a larger one, the data is interpolated to fill the space. With jit.scanwrap, it takes several tries to fill the matrix and get an output. The frame rate goes way down of course. Jit.scanwrap will also work the other way, cutting big matrices into little ones.



Jit.scissors
Jit.scissors will cut up images into neat square pieces. The jit.glue helpfile shows how to combine the two in an interesting way.

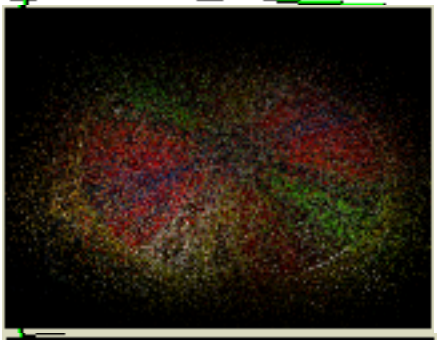


Jit.shade
Jit.shade does an automatic crossfade between two images, with a control matrix to set the rate of fade (the number of frames the fade will cover.) The control matrix here was taken from the colorbar.jpg

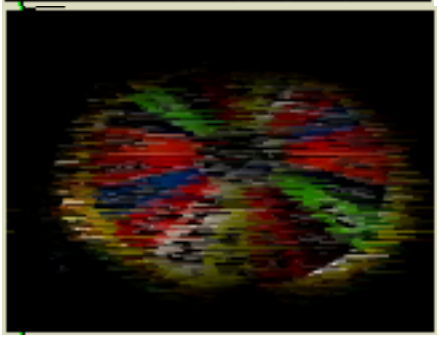
Jit.slide
Jit.slide forces an envelope onto any changes in pixel value. That means instead of flashing on and off, pixels fade in and fade out. With a moving image this gives echoes of the image. The wheel does not show this well, so I made the example by spinning the chillis with jit.jota and processing that.

Jit.sobel
Another edge detector.

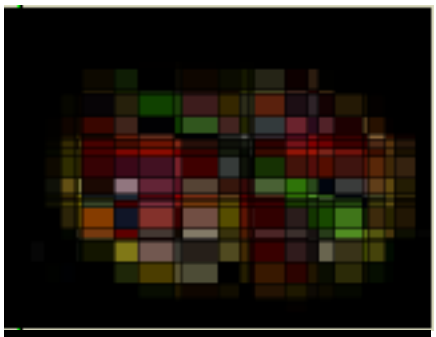
Jit.split
Splits matrices in two.



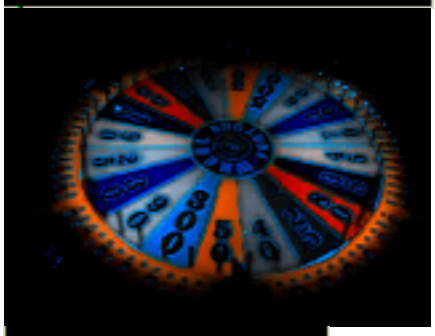
Jit.sprinkle
Randomly moves pixels about.



Jit.streak
Randomly adds streaks.



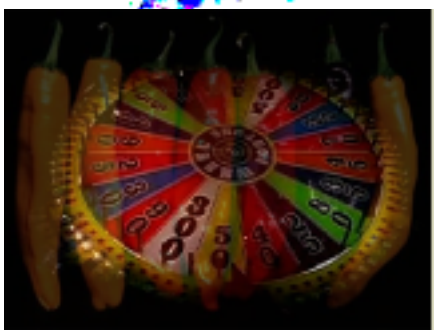
Jit.tiffany
Pixelation, but with more class.



Jit.traffic
Wicked colorspace converter. The help file allows you to build custom control matrices. If you want some understanding of the math in the help file, I suggest the wikipedia article on color.

Jit.transpose
Turns a matrix on its side (transposing the x and y cells). Not as flexible as jit.dimmap, and no more efficient, apparently.

Jit.wake
This is feedback with a built in convolution. Interesting effects happen when you modulate the parameters.



Jit.xfade
An essential part of any video toolbox. Aside from the usual image swapping, it can be used to interpolate between control matrices in objects like jit.repos.