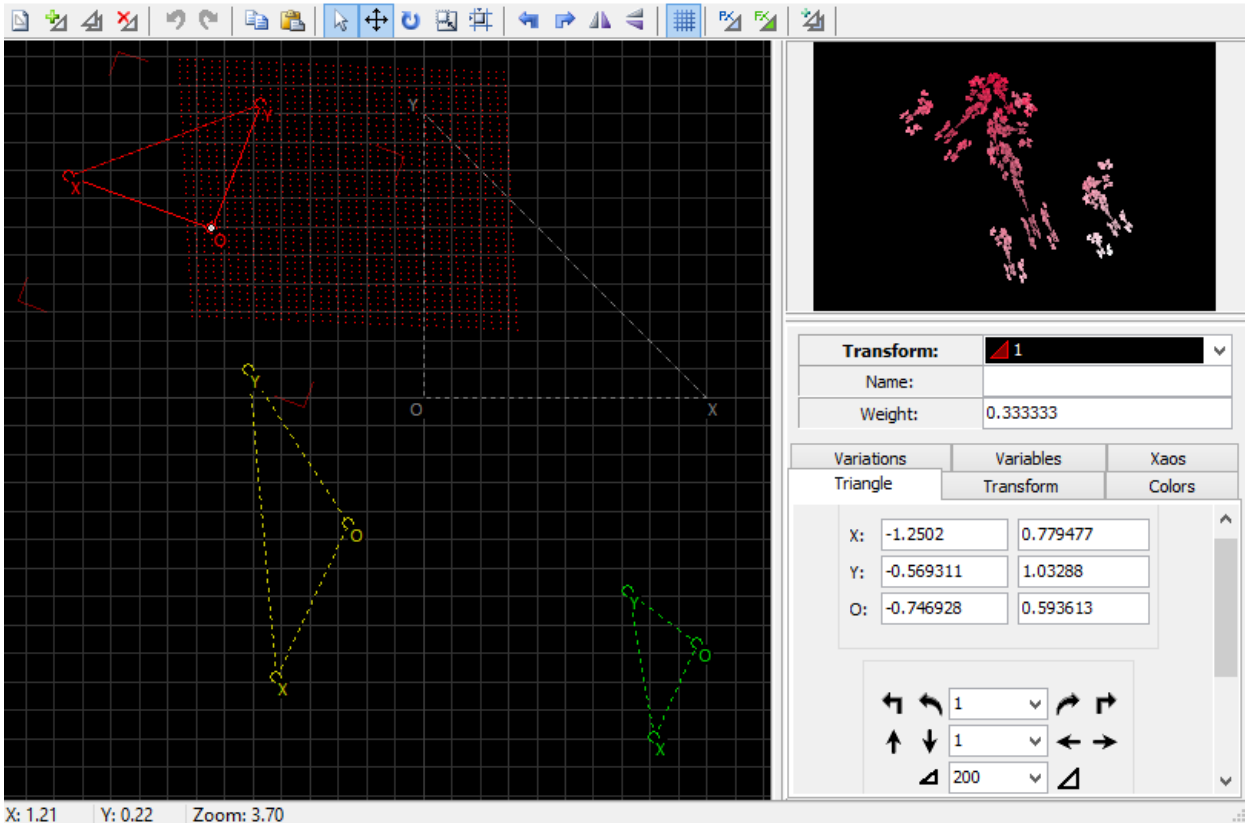
ALL ABOUT GNARLS - PART 1

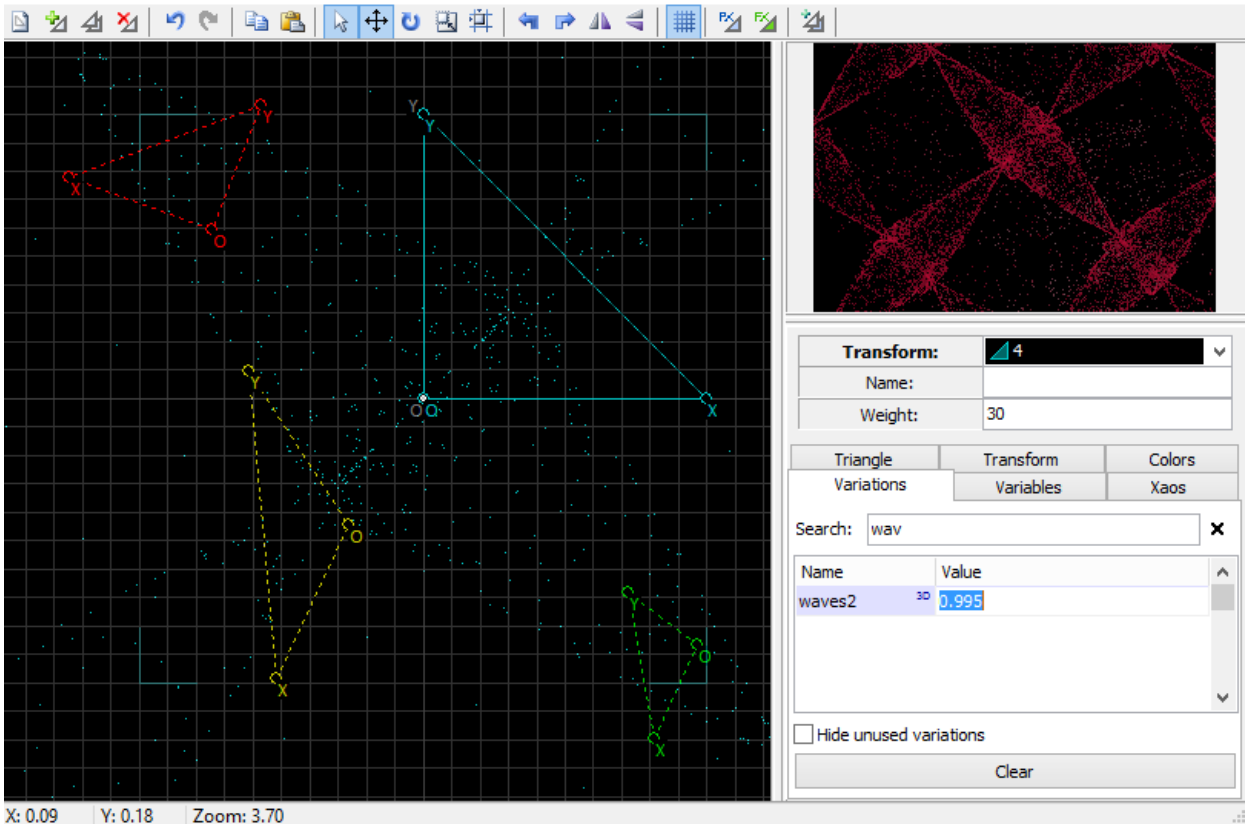
All I know about gnarls, I mean :dummy: We will go through some basic tricks, transforms that can create gnarly shapes and colouring.

Basic Gnarl

Lets first make some basic gnarly structure to evolve it into something crazy later on. To make it easier, we will turn a random fractal into gnarl.

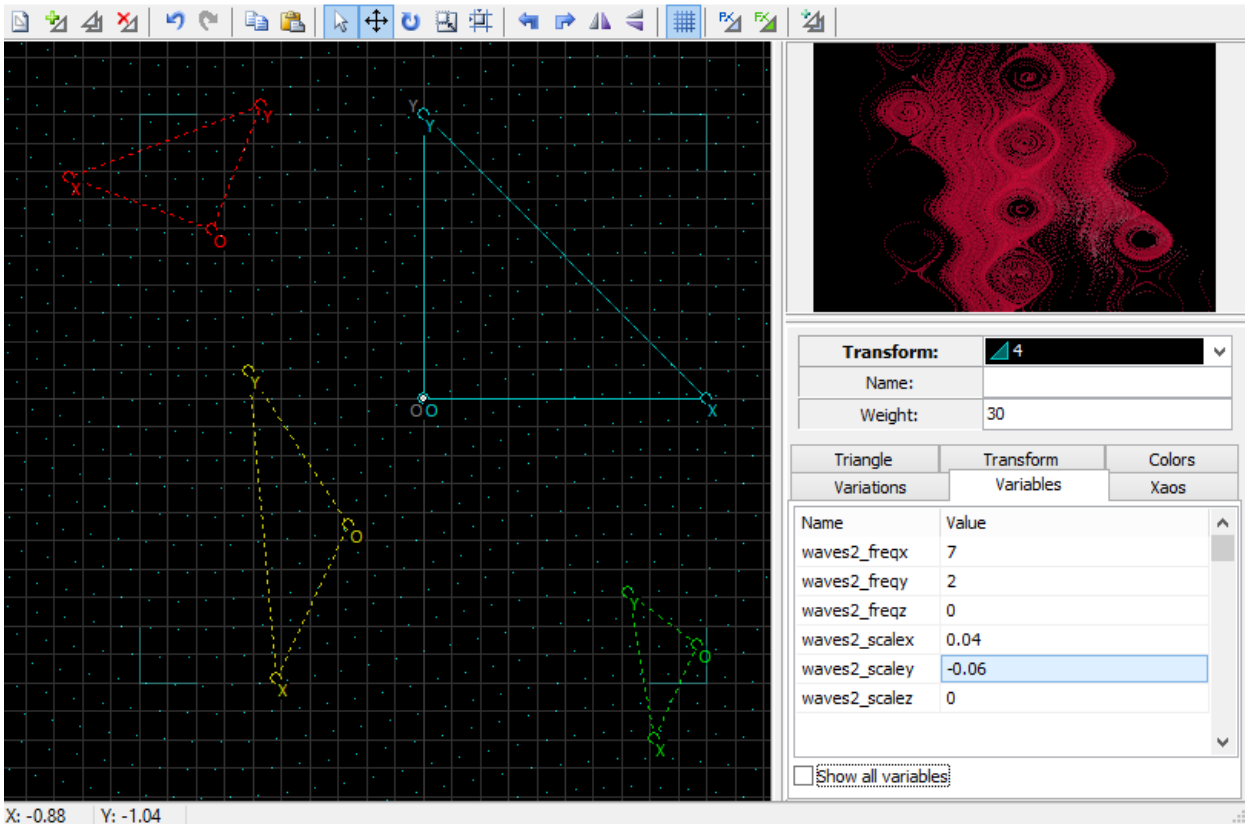


Now, add a new transform, with waves2 (set amount to something like 0.98 - 0.995 instead of 1). Set the Weight to a high value, lets say 20 - 50.



Now, you just need right parameters. Set frequencies to some nice values - I usually keep them between 5 and 20. The scales, now, should be something between -0.1 and 0.1.

Later, we will have a more detailed look of what each of those values does.



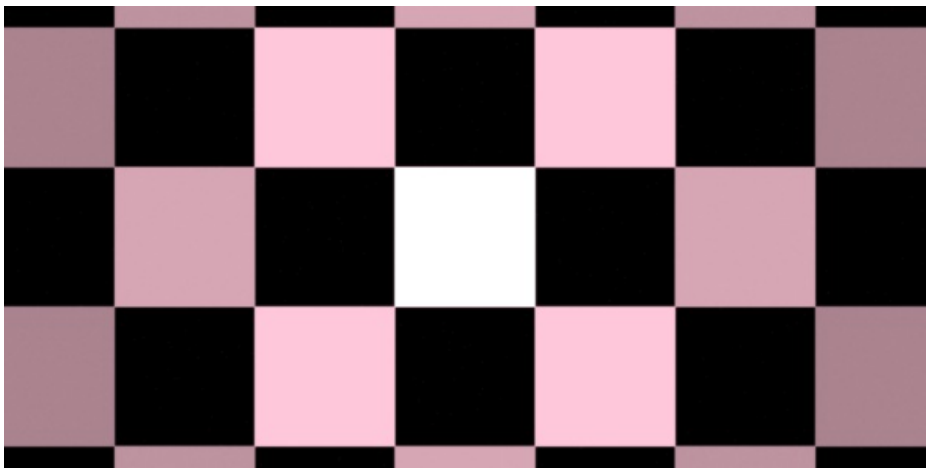
Tip: move around the affine for the gnarl transform (waves2 here).

Here we go, a very very basic gnarly structure. Lets now learn how to improve it.

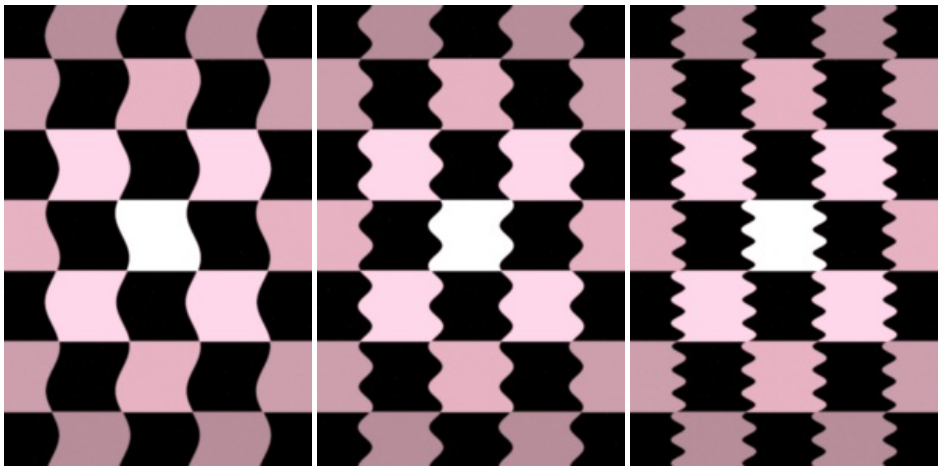
Waves2 Parameters

This section is an overview of what each of the waves2 parameters does, and its effect on the gnarl.

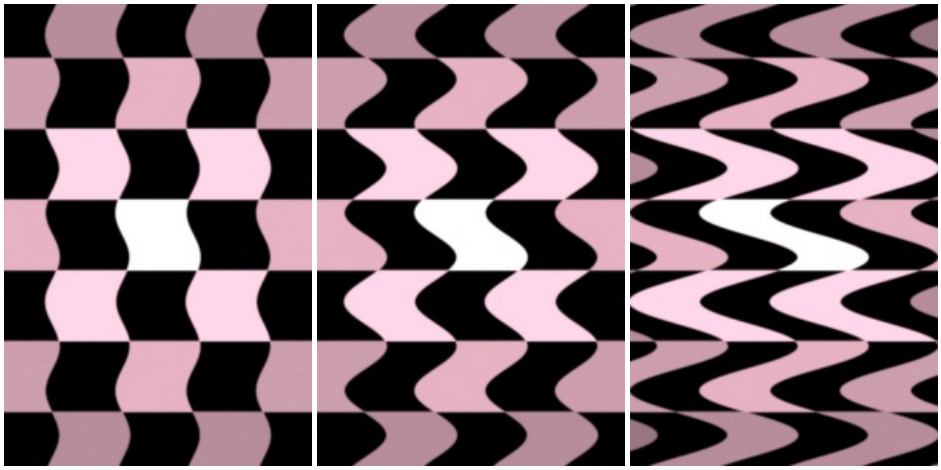
We will first look at a basic checkboard pattern, and apply waves2 as final (camera) transform, to separate its effect from everything else.



Now, we apply waves2. Lets set all variables to 0, and then: scale-x = 0.1 and freq-x = 5 (left), 11 (middle) and 19 (right):



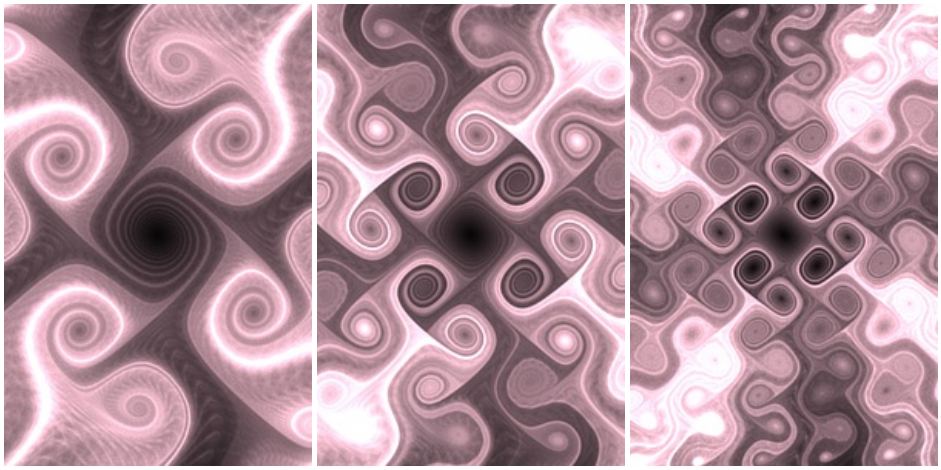
Below, freq-x fixed at 5, and change scale: scale-x = 0.1 (left), 0.3 (middle) and 0.7 (right).



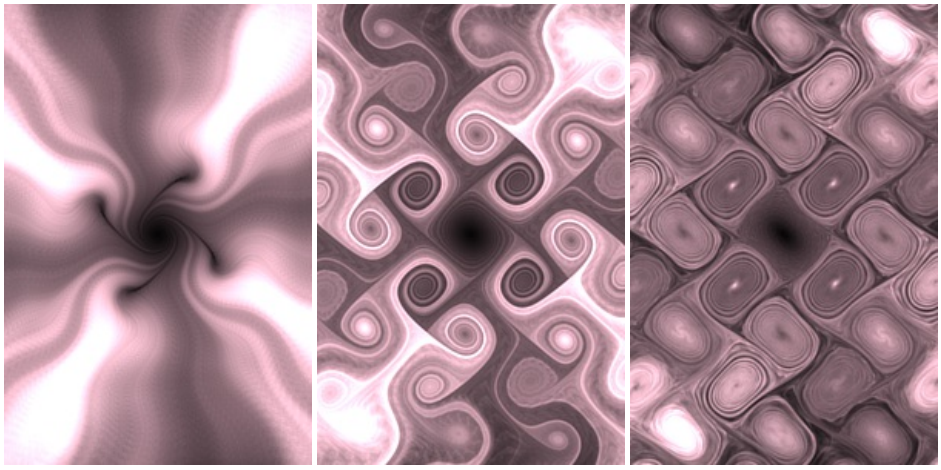
The freq-y and scale-y work in a similar way. Take a look at the preview below, will all -x variables set to 0, scale-y = 0.3 and freq-y = 12:

In a few words, freq parameter defines the frequency of waves, and scale defines their amplitude. Applied to gnarls, high frequency values will create many tiny swirls, while low frequency values will create few big swirls. The scale will define how "rippled" they are: if the scale (in absolute value) is too low, the gnarl will not swirl. If it is too high, it will look messy.

Lets see how it works on a "real" gnarl. Below, the only change I make is increasing the frequency: from 4 (left) to 7 (middle) and then to 11 (right):

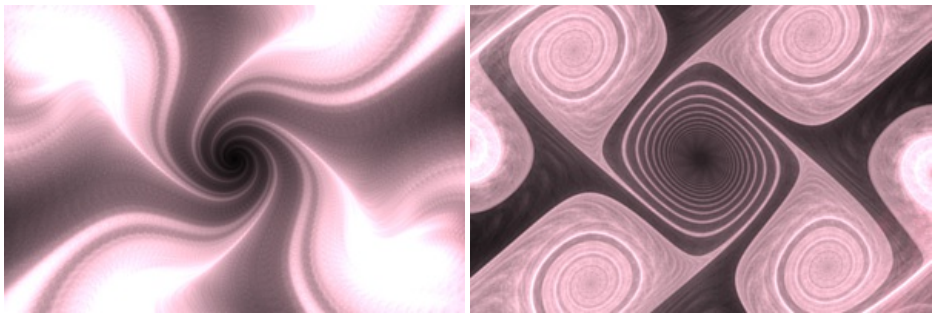


Notice how the gnarly shapes get smaller and their numbers increase. Now, lets do the same with the scale. I'm increasing it from 0.01 (left) to 0.05 (middle) to 0.15 (right)

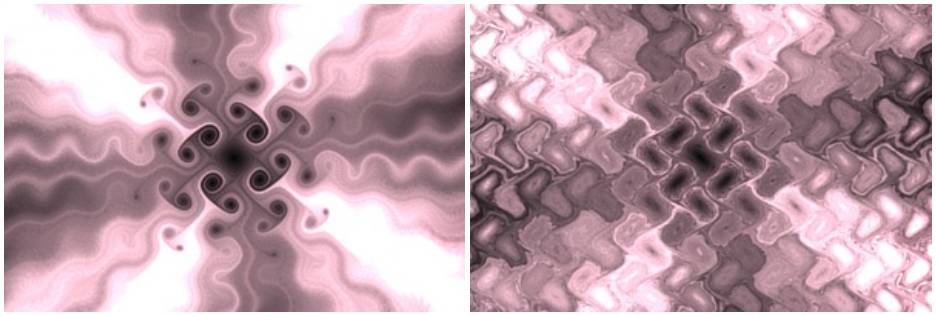


Observe that the right variant is maybe just a bit too swirly already, the shapes are clogged and not clear enough. To get nice clean shapes, you should reduce the scale as you increase the frequency.

Again, lets look at examples. First, we take freq 3. Below, we have scale = 0.02 (left) and 0.1 (right):



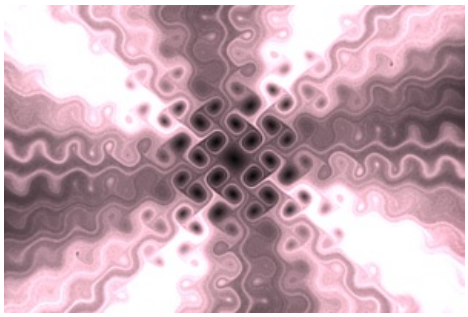
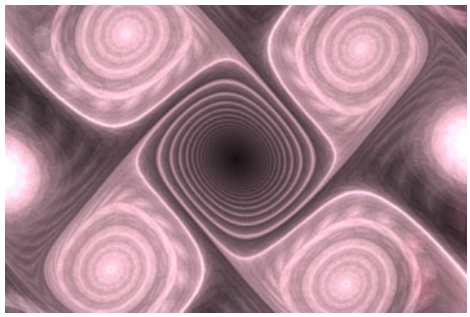
What happens with the same parameters if we increase frequency to 15 on both?



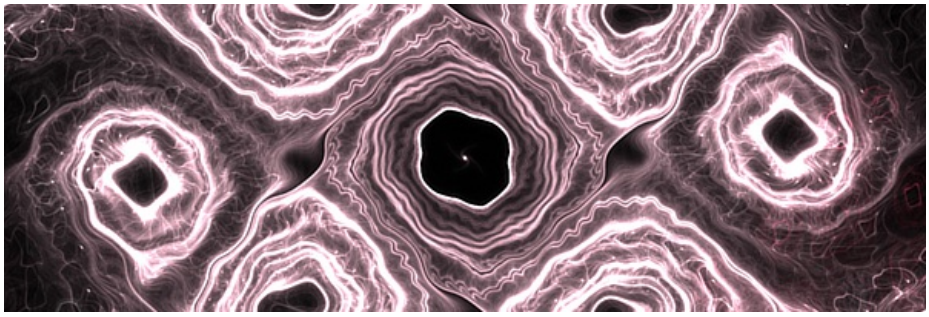
Observe that, for frequency 3, scale 0.02 looks too plain, while scale 0.1 is fine. Meanwhile, for frequency 15, scale 0.02 is fine, and scale 0.1 is way too messy.

Finally, until now, we only looked at how x freq and scale work, and how y scale work, but not how they work together. Well, basically, they add up their effects.

Lets see how it works. Look at two gnarls below. The left one has freq-x = 20 and scale-x = 0.03. The right one has freq-y = 3 and scale-y = 0.11.

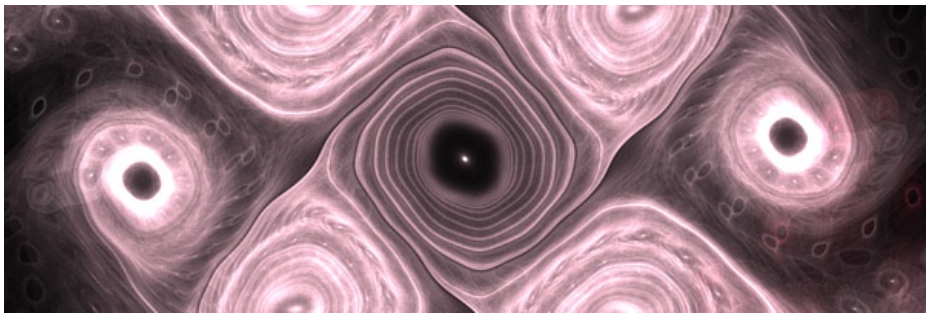



Now, we put those two together: freq-x = 20, freq-y = 3, scale-x = 0.03, scale-y = 0.11:

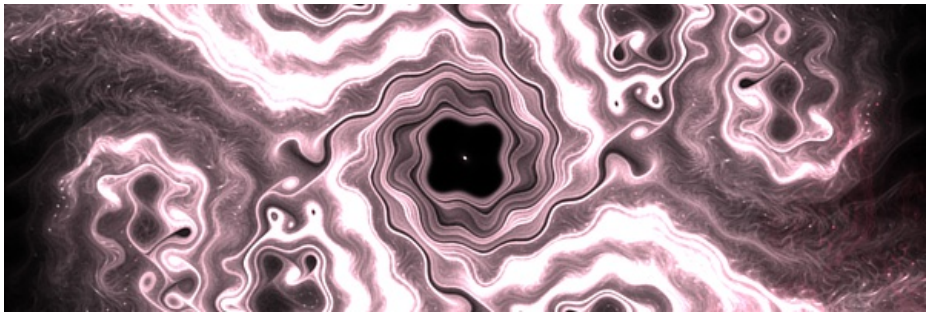


The result has shapes similar to the pattern with lower frequency, but some additional ripples caused by the high frequency pattern. I personally like it the way it is, but we want some cleaner patterns, we need to reduce one (or both) scales.

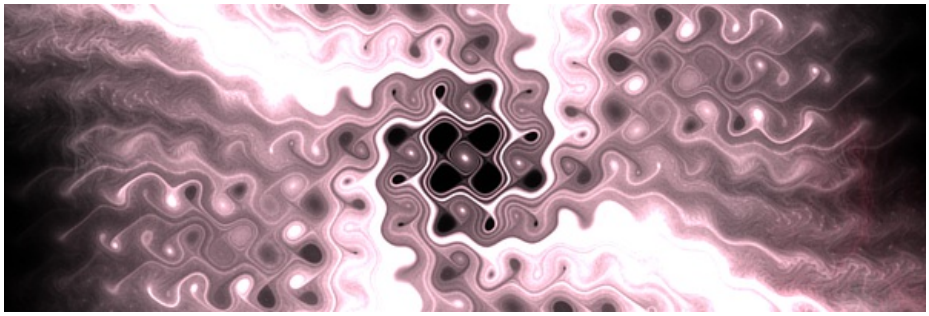
Case 1: freq-x = 20, freq-y = 3, scale-x = 0.01, scale-y = 0.11. This setup leaves, untouched, the big pattern and adds some minor irregularities to it.



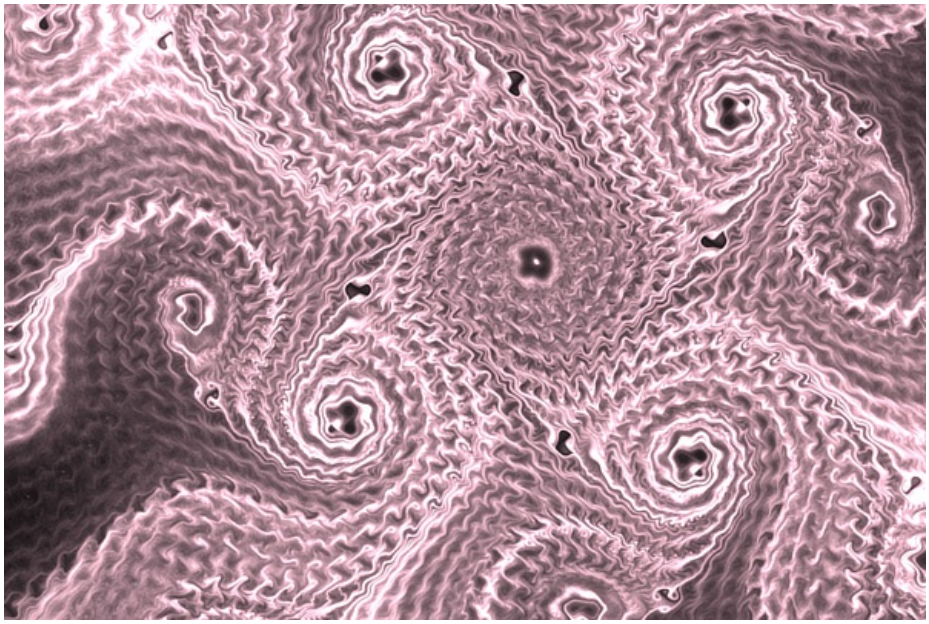
Case 2: freq-x = 20, freq-y = 3, scale-x = 0.03, scale-y = 0.05. Here, we leave the small ripples, and swirl them a little bit into a bigger gnarl.



Case 3: freq-x = 20, freq-y = 3, scale-x = 0.03, scale-y = 0.02. A variation on case 2.

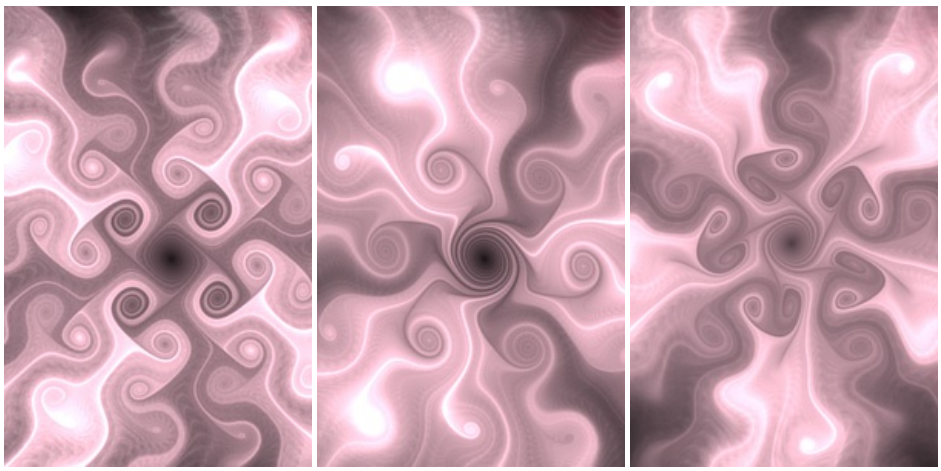


And, just to close this section, an example of what you can achieve by combining x and y effects:



Radial Symmetry

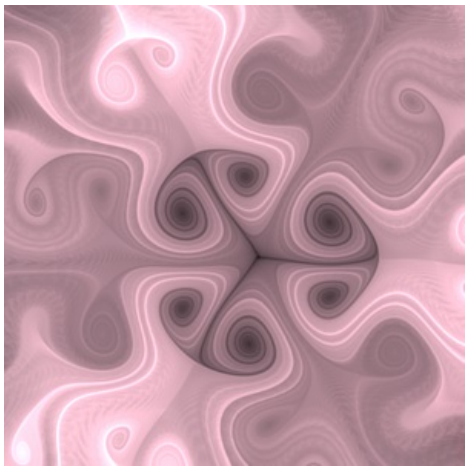
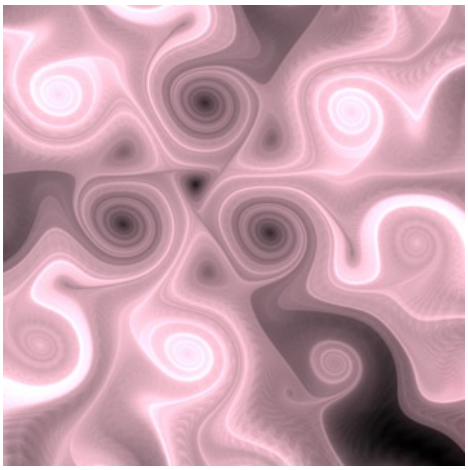
Now, how do people get things like below?



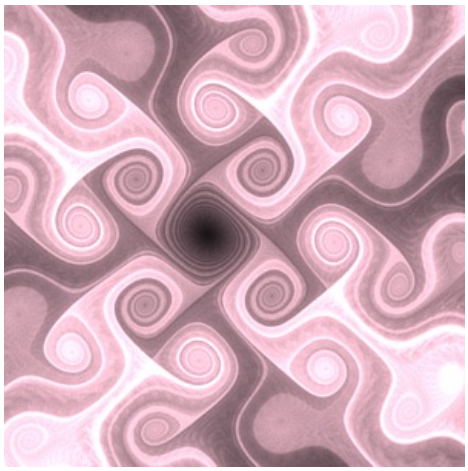
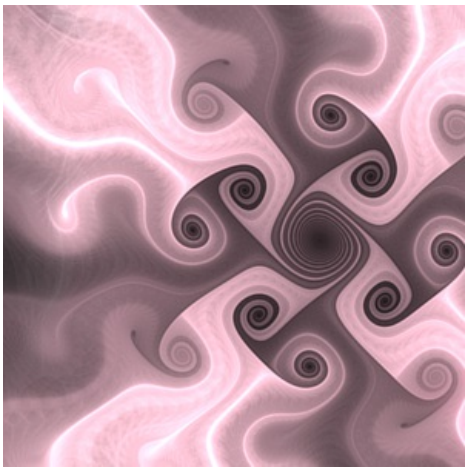
Well, first thing: gnarls are not perfectly symmetric. It upsets me big time because it makes very hard to put them into hypertile patterns, but I didn't find a way to overcome it yet.

To achieve this almost symmetry, rotate the gnarl transform by 360 / n degrees, where n is the desired order of symmetry. You can rotate either counter clockwise or clockwise - it doesn't really matter. I personally prefer the CCW for some reason, but CW works as well for all examples below.

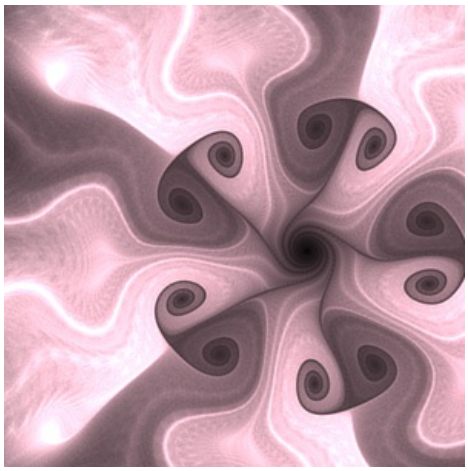
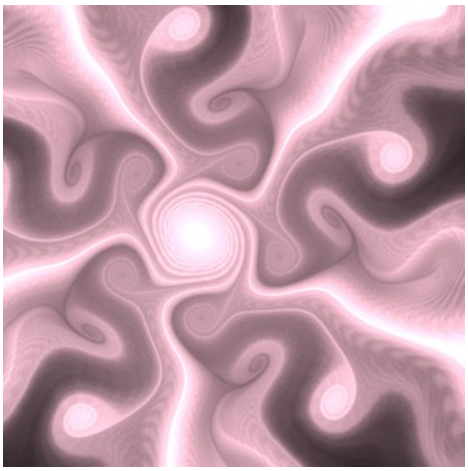
Case n = 3. We have that 360 / 3 = 120. Two examples of CCW rotation by 120 degrees:

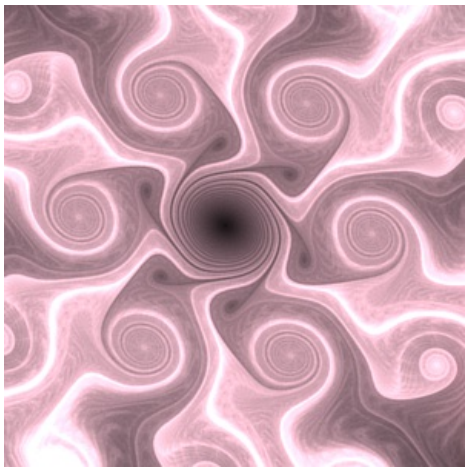
Case n = 4. We have that 360 / 4 = 90. Two examples of CCW rotation by 90 degrees:

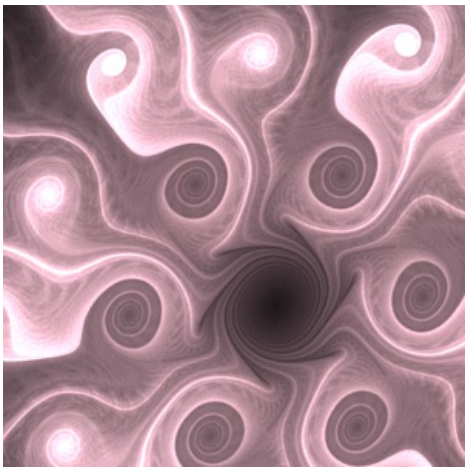



Case n = 5. We have that 360 / 5 = 72. Two examples of CCW rotation by 72 degrees:

Case n = 6. We have that 360 / 6 = 60. Two examples of CCW rotation by 60 degrees:





And so on. Between, I used the same waves2 setup for all examples above, only changing the rotation angle and the position of the gnarl transform affine. Those rotations work not only for waves2, but for other gnarl transforms too, and let you achieve some nice and sharp patterns

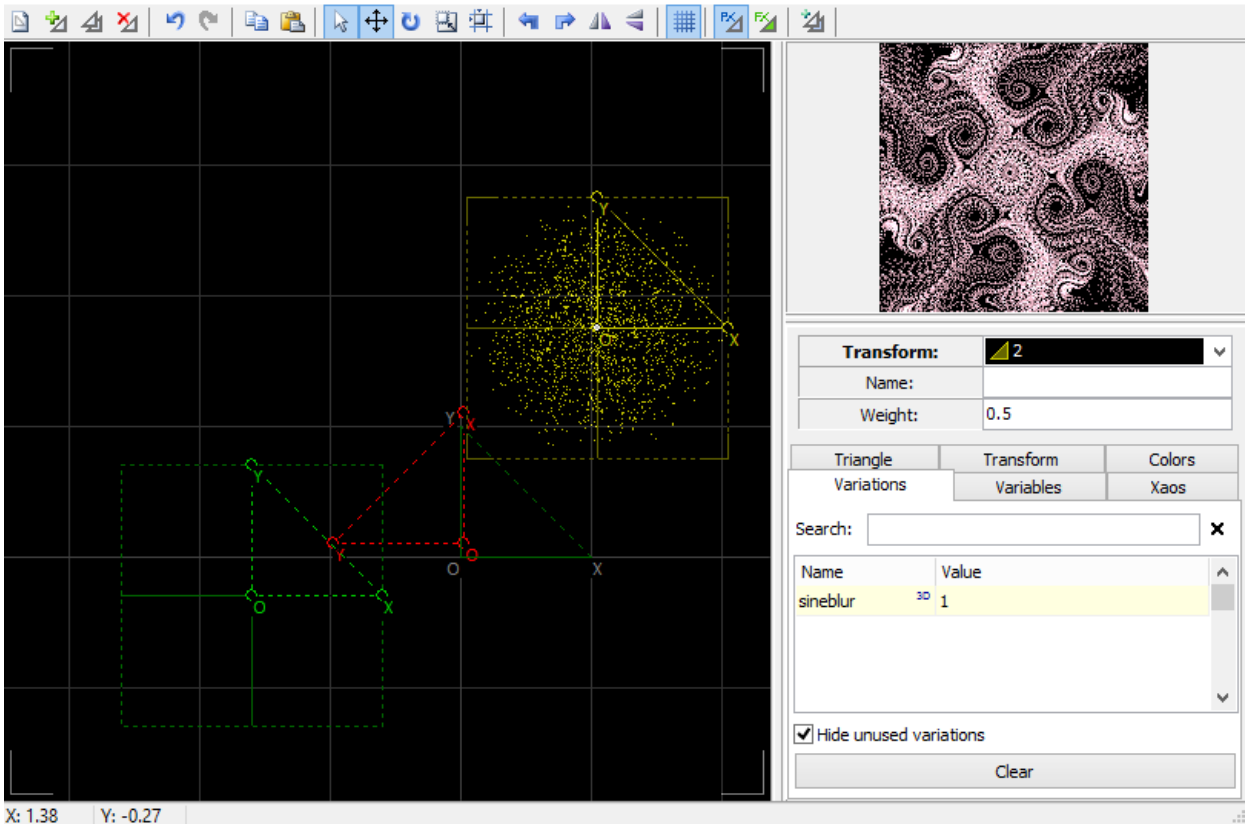
Texturing

In the beginning of this tutorial, we made a simple gnarl pattern starting from a random. As we could see, the gnarl is made of a gnarl transform (waves2 or something else) that creates the swirls, and other transforms - in that example, the transforms that created the random - that add some texture to it.

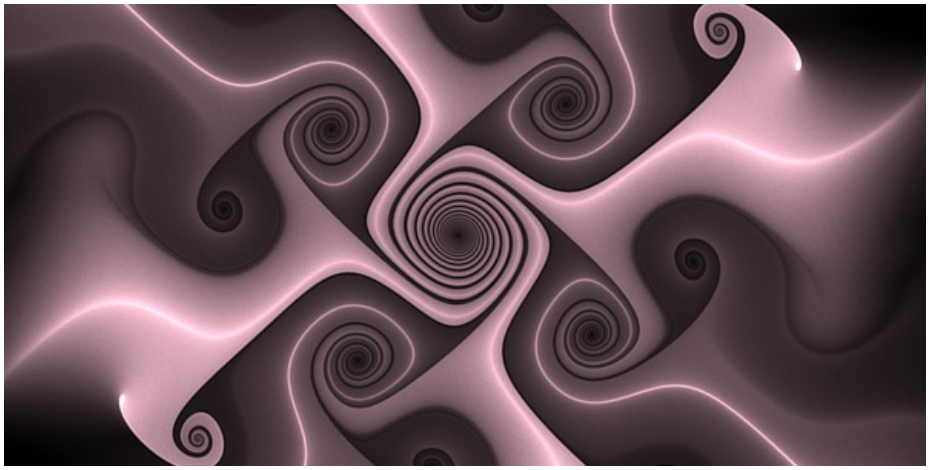
You can use any transforms to texture gnarls. I have 2 groups of textures in my head, basically - blurs and actual textures.

The blur setup is simple: start by creating a gnarl transform, and then add a few (1-3 but can be any number) of transforms with blur only (such as sineblur, gaussian or radial), and move them around using post affines.

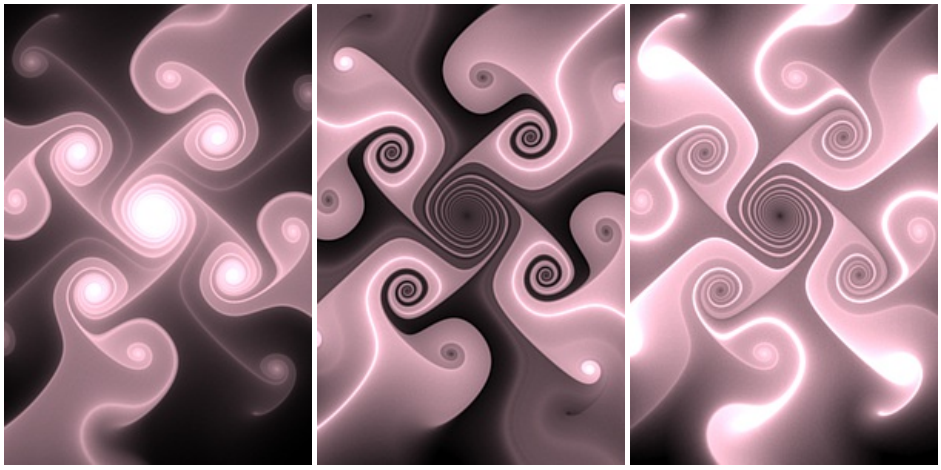
Take a look at the example below. We have a gnarl transform (red), and two blur transforms with sineblur (yellow and green):



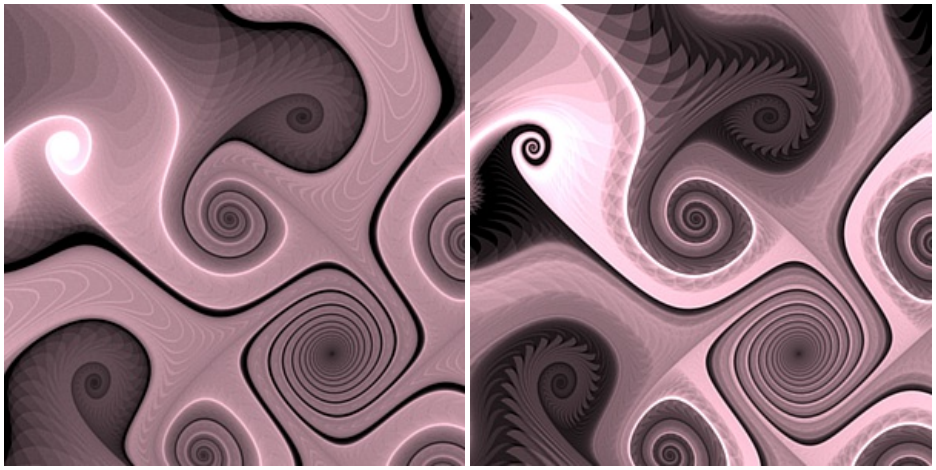
The actual fractal looks like this:



Moving around the blur transforms (this works for any texture, not just blur) will create different lightning effects. Below, I only moved around and scaled up and down the blurs:

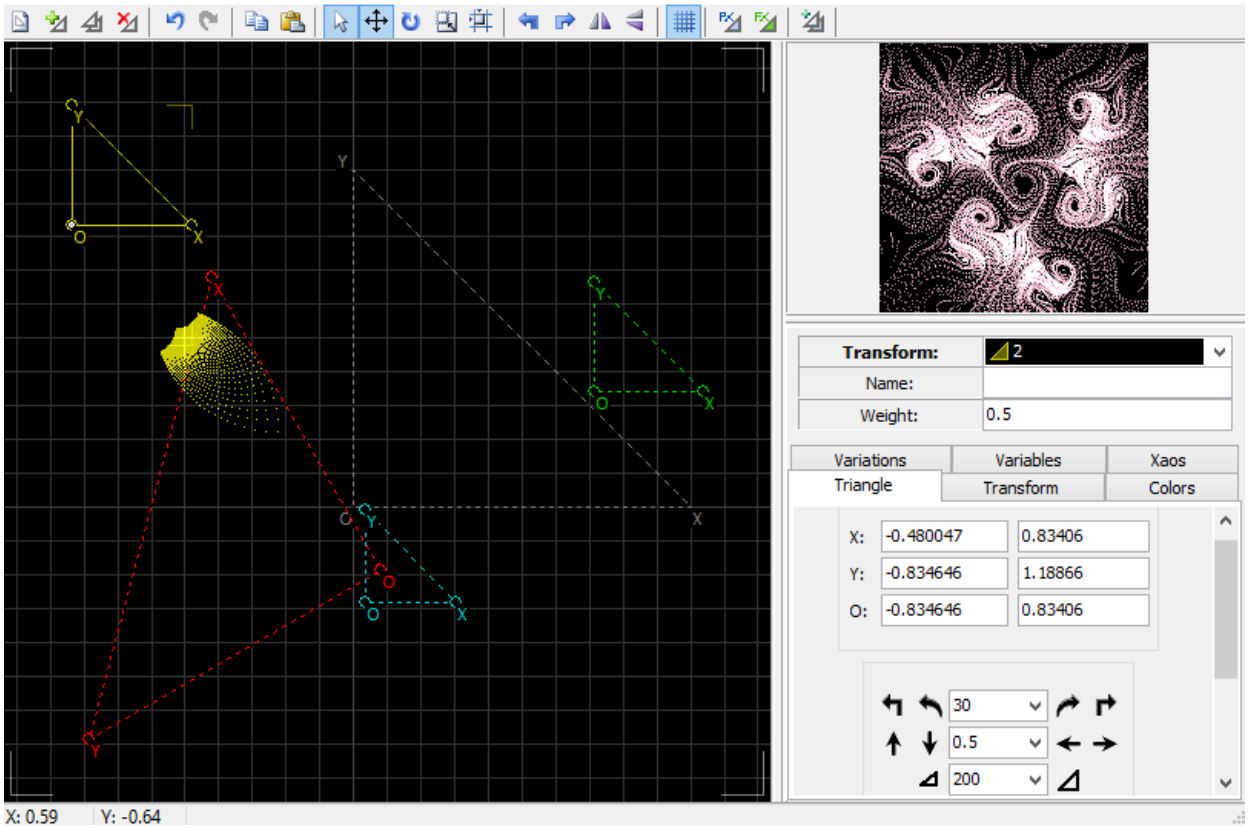


Sineblur is not the only blur in the world. For example, circleblur (left) and starblur (right) make things like those:

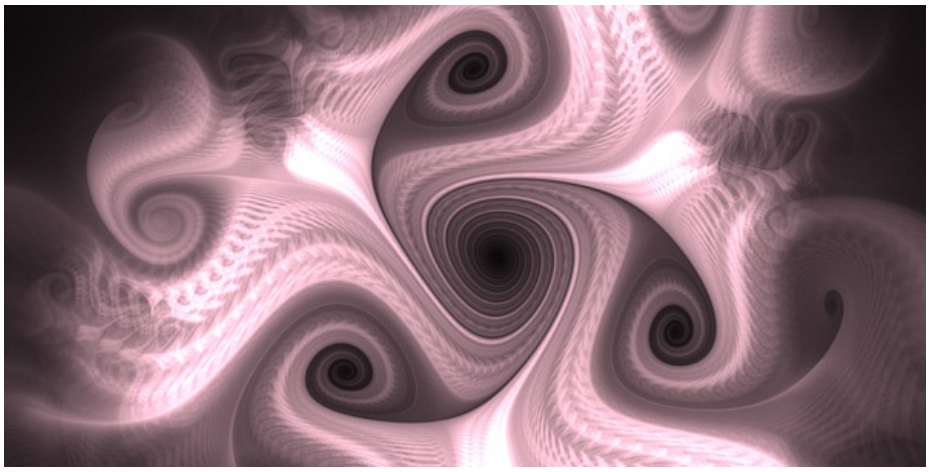


But, if you want some more interesting texture, you should use other transforms that are not blur. The method is the same: add a bunch of transforms and move them around until it looks good. A small difference though: you may need to scale down the transforms to get a distinctive texture instead of noisy blurry messy thing.

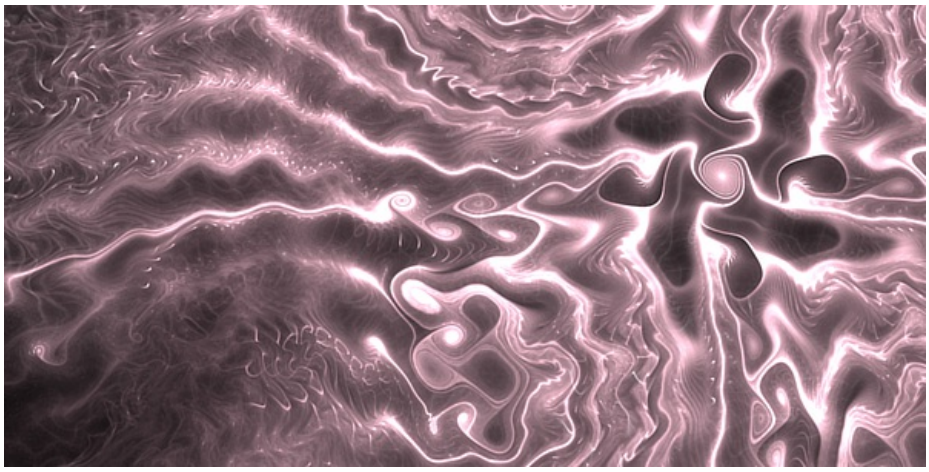
Also, you may need to do way more moving and re-positioning to make the texture stand out, depending on transforms you used.



As an example, below, a texture based on sphericals:



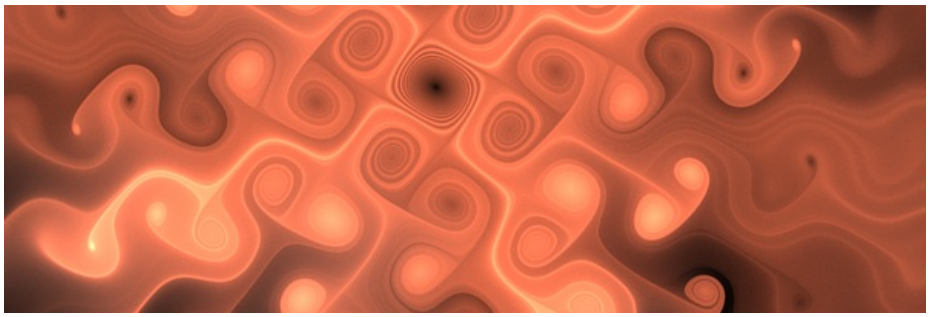
And another one, with linears:



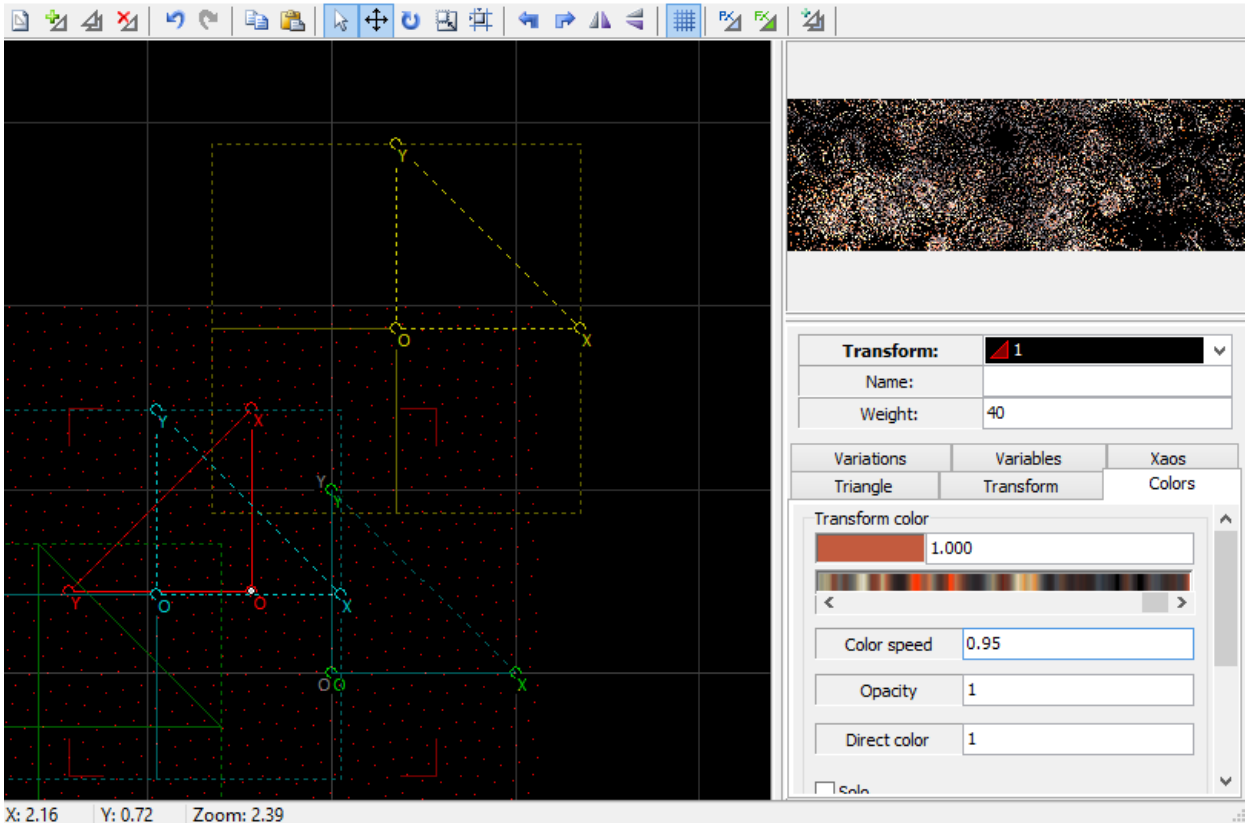
Colouring

Last but not least, colouring. After all, it looks kinda boring still.

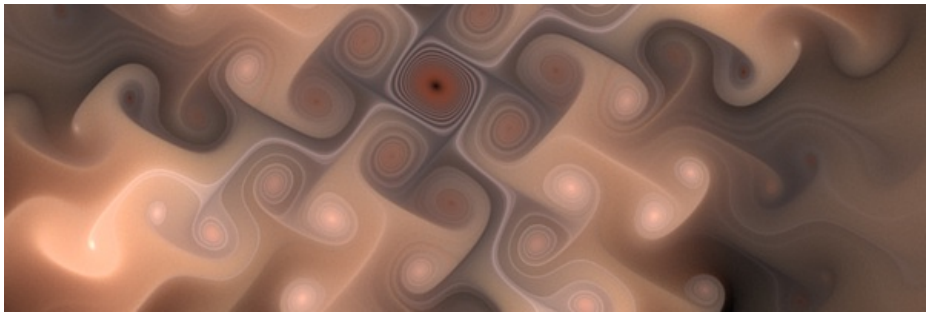
At this point, you probably have something like this:



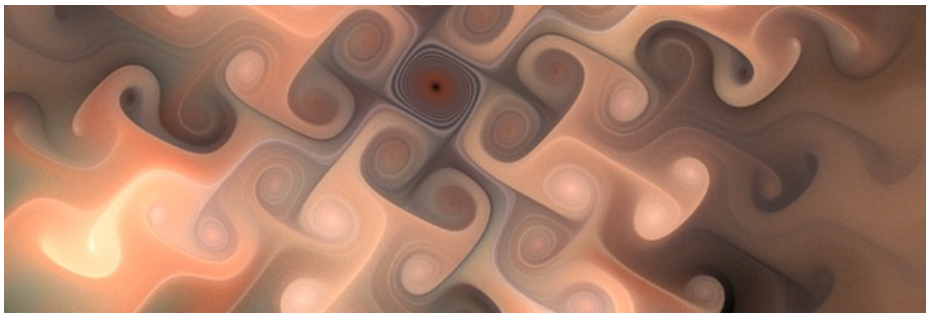
A quick thing, go to the colour tab and set: transform colour = 1 and colour speed = 0.95 or so (in Chaotica, the equivalent is palette location = 1 and blend speed = 0.025).



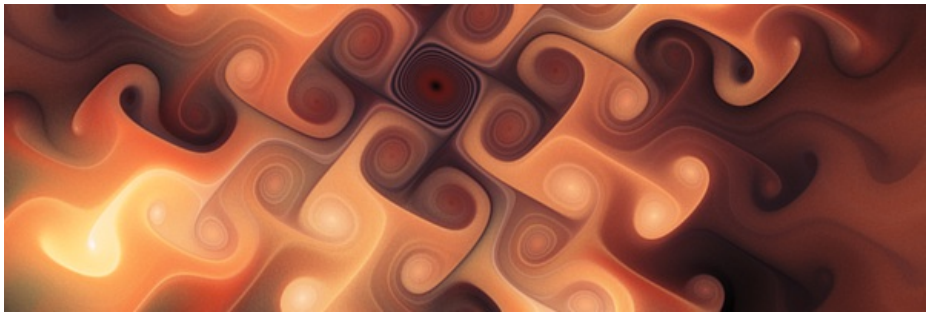
Already a bit of an improvement:



If you are working with blur texture, there is a trick to improve it further. Go through all blur transforms and set the colour speed to -1 (or, in chaotica, blend speed to 1) - which means solid colour.



(or, just to show off, with curves it would end up looking like this)

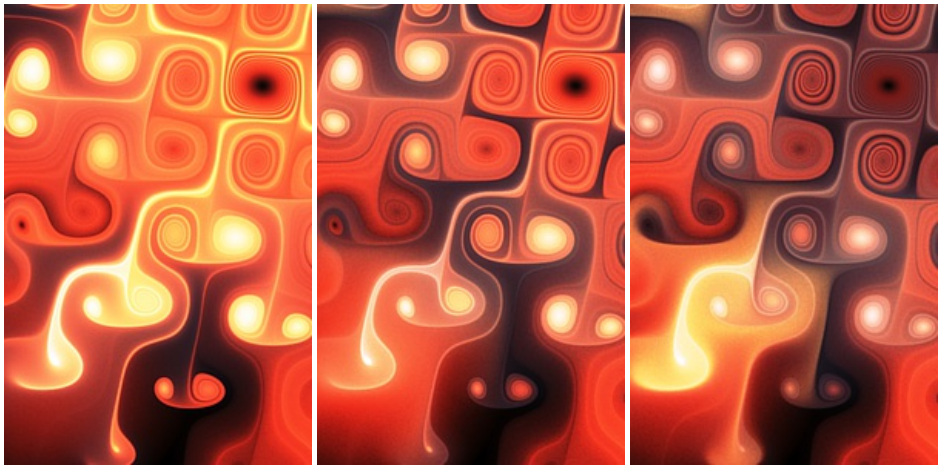


This second trick may not work well when you have actual texture, as it may leave the texture a bit too plain / boring, as it would kill all texture colour variations.

A bit on how colour speed works. It basically sets up the amount of colour that will be retained from previous transforms. -1 means no colour from previous transforms will be taken, and only the transform colour will be used - which makes it solid colour. 1 means that transform colour will not be used at all.

As the gnarl transform is iterated many times to create those shapes, we must add transform colour in really small amounts, setting a value close to 1 - or transform colour will cover up most colour variations.

Take a look at the previews below. Here, the transform colour of the gnarl transform is orange. See how the result looks like for colour speed = 0.7 (left), 0.9 (middle) and 0.95 (right):

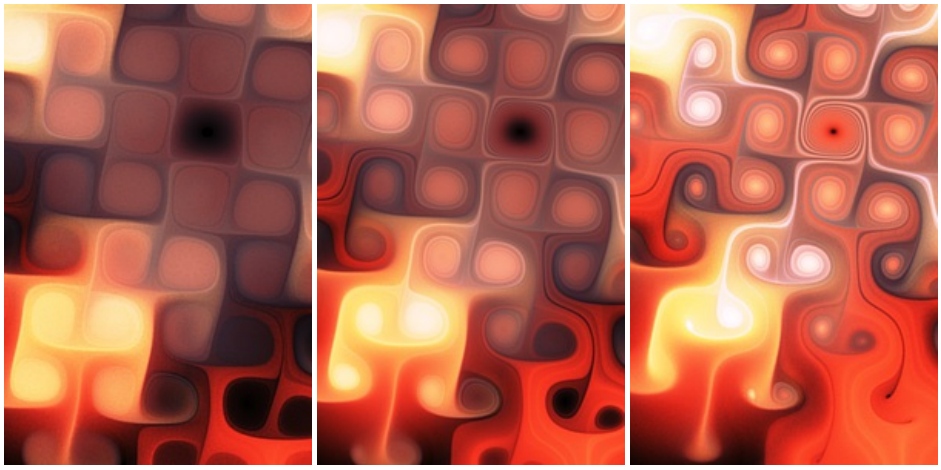


Common Issues And How To Fix Them

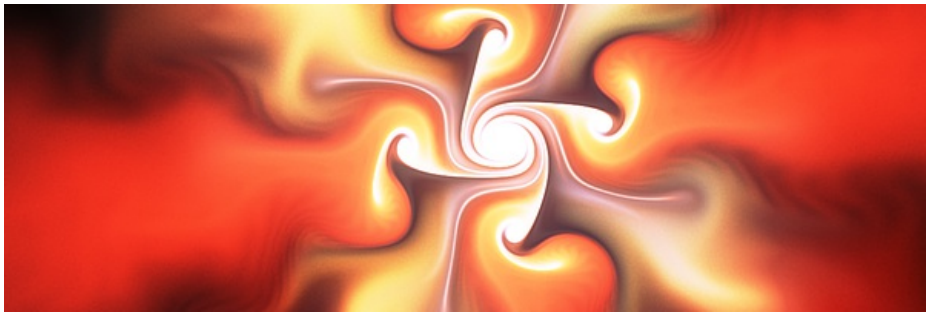
Last but not least, lets go through a few issues that commonly arise in gnarling, and ways to fix them. Most of those issues can be used as artistic effect too and not always require fixing.

Issue 1: we did everything right, but the gnarls somehow don't make clear swirls, like on the picture below.

To make swirls visible, reduce the amount of waves2. See below: waves2 = 0.997 (left), 0.994 (middle) and 0.99 (right):



Issue 2: the gnarl is too condensed in the center:

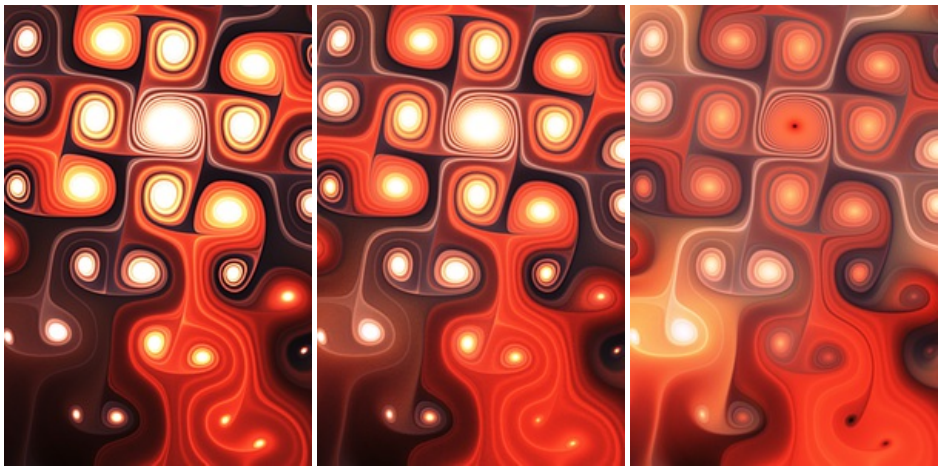


This is related to previous one. Try increasing the amount of waves2 and see what happens.

Also, instead of changing the amount of waves2, you may try changing the transform scale, as it leads to a similar result.

Issue 3: the swirls are too bright:

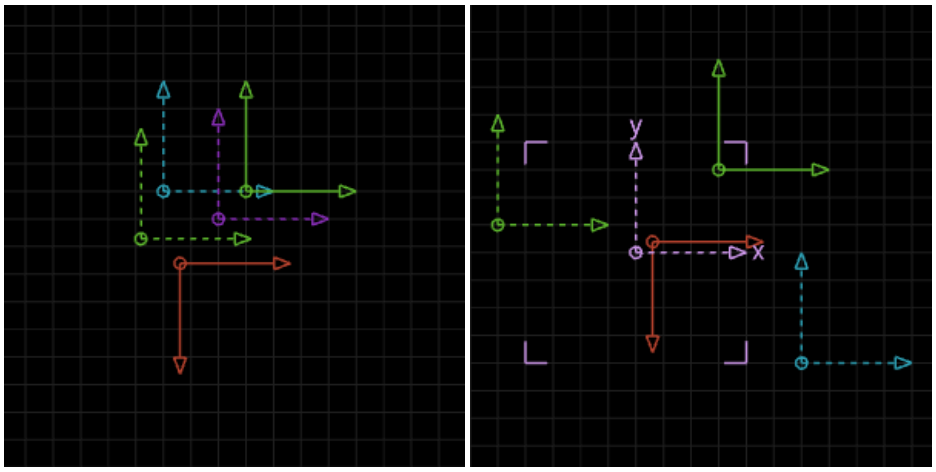
This means that the weight of gnarl transform may be too high. Try decreasing it. Compare below, weight = 200 (left), 100 (middle) and 50 (right)



Issue 4: there area with gnarly structures is too small and there is no nice differentiation of colours:

There are 2 possible solutions. The first one envolves increasing frequency (and reducing scale correspondingly). This will not fix the coloring though.

So, instead of doing anything with the gnarl transform, you need to move the post_affines of blur transforms. Below, the red transform is the gnarl one, and the other are blur ones. The setup on the left corresponds to the picture above...



...and the right setup, to the picture below. See how moving the blurs around can change your fractal's appearence. A lot.