

# Different approaches to offset paths

## AFTER EFFECTS

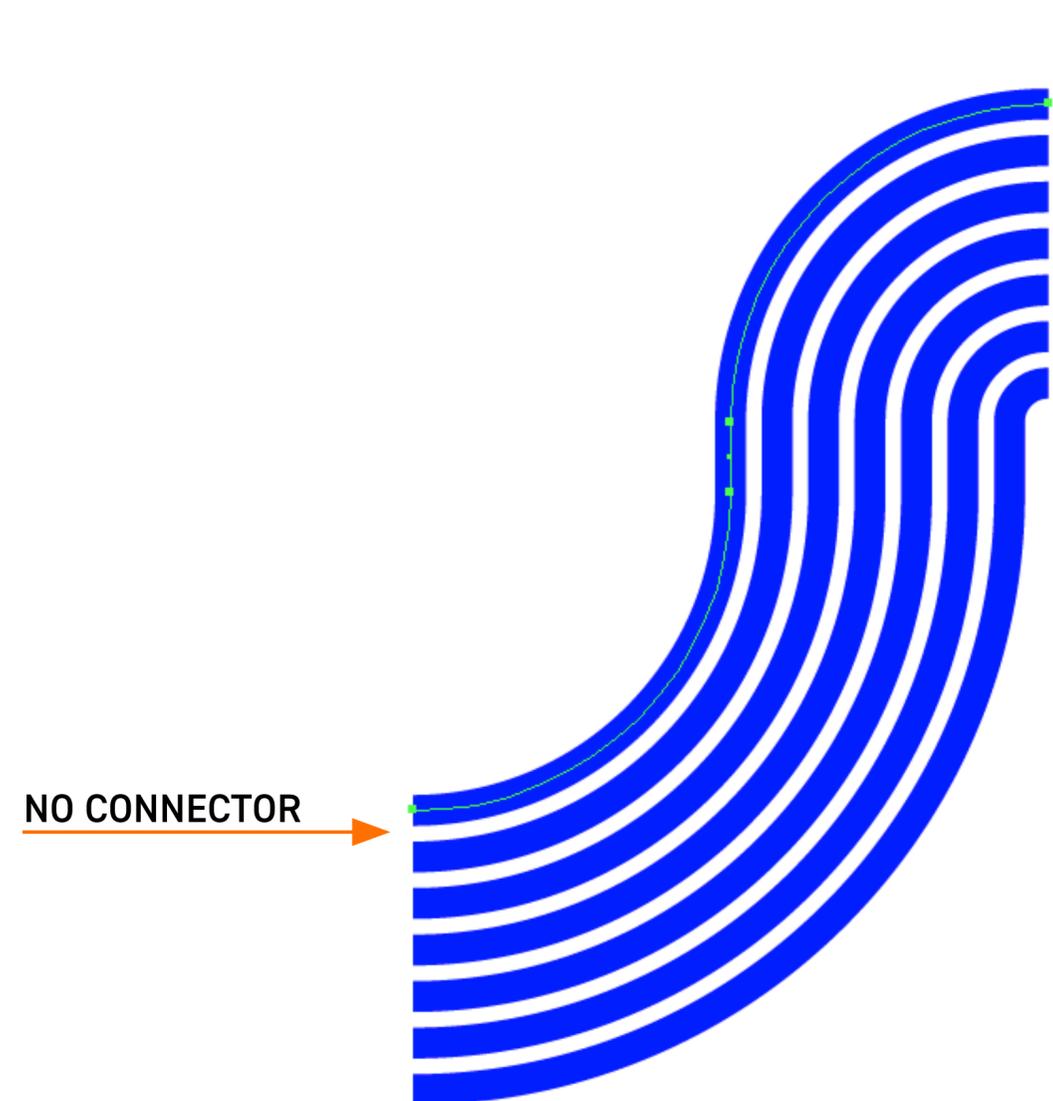


I noticed that AE is following Illustrators' idea of how an offset should behave when it comes to joining the paths. And maybe it should, however, I feel that the example to the right is much closer to what people would like to achieve when using such an effect. At a minimum I do :)

Note also how AE expands from the middle and always leads to an even number of copies, uneven if you'd bring back the original. I have no preference, yet. Playing with both effects I noticed both approaches have it's uses.

At the back of this document I'll include a screenshot of the AG offset extension. It has a few options that I'd for one would find very handy in After Effects.

## ILLUSTRATOR - AG OFFSET



# My Issues with offset paths .01

## Animating trim paths



## OPTION TO KEEP ORIGINAL

I played around and found that all existing path operators work very well with the new offset abilities.

When I wanted to really make something the first thing I thought of was animating some evenly spaced lines in the shape of an S.

I tried but found there was no (easy) way. Since the paths are offset from the middle, there's always a gap. Easy enough you'd think, use the original path and duplicate it and its stroke.

But then If you'd want to trim and animate it you're out of luck. The path lengths are different and there's a lot of tweaking involved to get hem to line up.

This gap is not some I'd want much and it is probably mostly an issue with open paths. Fix imho would be an option to maintain original.

# My Issues with offset paths .02

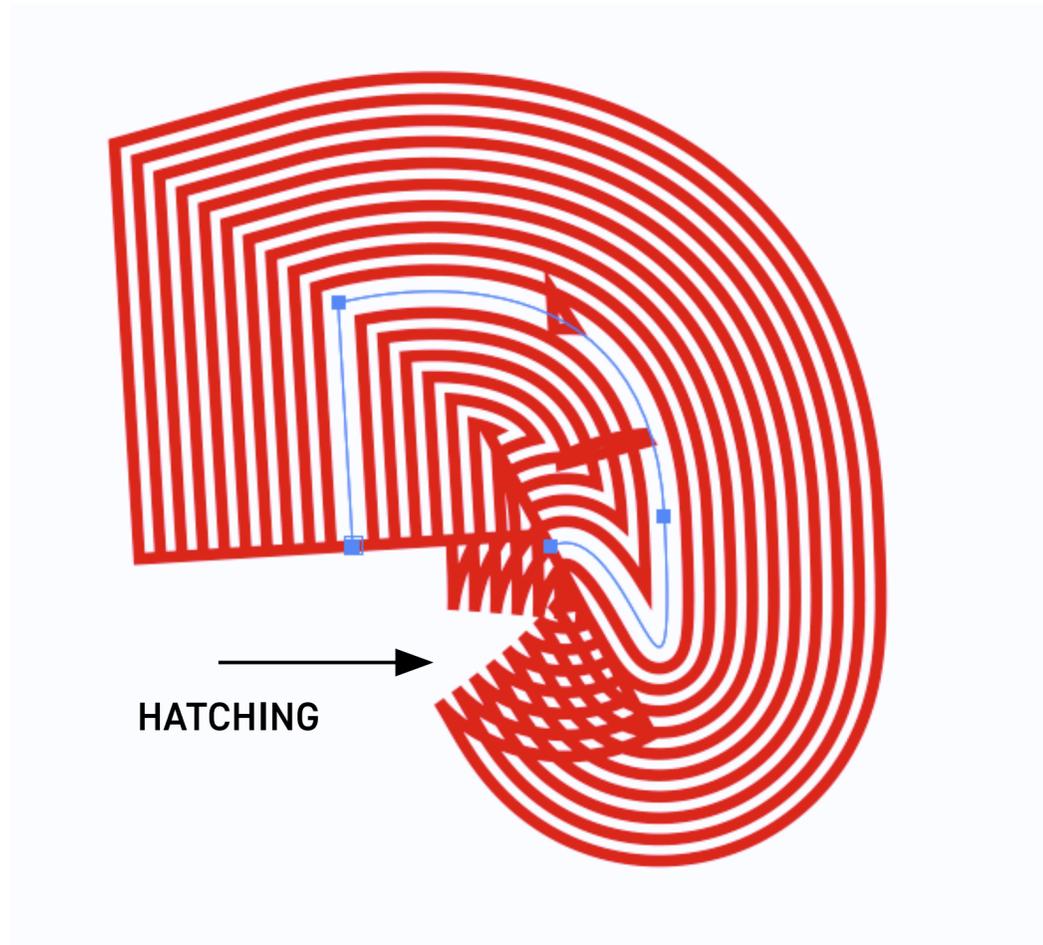
## Artefacting (?)

### HATCH EFFECT

I notice that when endpoints on an open path have tangents there's this effect of extending the copy but it returns to some line in a circular fashion. There is this hatching effect that should be optional if it were up to me. If at all possible

You can also see some glitches happening on the path itself, that is not persistent when changing the amount number.

I just noticed that this is not happening with the round join.



MITER JOIN



ROUND JOIN, SAME PATH

# Cloners in AE

## CLONES IN AE

AE in it's native capabilities feels lacking in cloning options. The shape offset path helps a little. There's the repeater and text animators but if you want some real power you'd have to buy plugins, extensions and scripts. Or be very good at expressions.

Seeing the offset paths added to the shape layer tool-set made me curious as to where the future of cloning in ae would lie.

I can tell you what I would want having worked with c4d for a long time, and having a global idea of what the cavalry-beta brings to the table.

On your right, first column are the duplicator functions from cavalry, there are some obvious ones like the grid but also interesting ones like clone on shape points.

You asked me what I considered advance cloning options. Well it starts with having multiple distribution options.

And once they are distributed you like to be able to manipulate the clones with something like the effectors in the second column do (c4d).

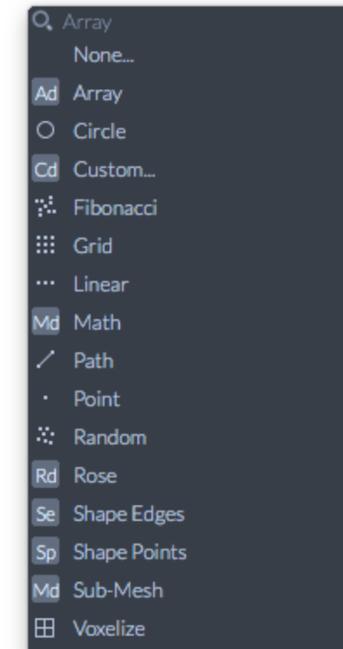
For instance the shader effector lets you manipulate position, scale and rotation (amongst others) based on a texture or shader input. (not unlike card dance). Spline distributes along a path. Step in/ decreases something following direction or index/object numbers. Delay evens the clones animation or makes it springy.

All these elements can be combined in limit-less way.

To top it off you can limit the effector on all sort of ways with falloff. Which acts as a kind of masking. On Steroids.

Hit me up if you want more info on this. I realize is pretty abstract.

## CAVALRY



## CINEMA 4D



# AG Offset

## LAST ITEM :)

I figured AG Offset has a few nice features that might you might be interested in. Since I'm at it and the world around me came to a standstill I might as well include that here.

As you can see you can alter color, the weight of the stroke of the copies and it morphs between them.

Thanks,

Coen

