Retime track in nuke

Retiming a curve

There are many retiming options in Nuke but how do you retime tracking data or animation?
The answer is actually very simple and can be used wherever you have an animation curve in your script. If you are using an OFlow node, called ‘OFlow1’, and are using ‘Source Frame’ timing, rather than ‘Speed’ to retime a plate, you can access the retime curve by referring to ‘***OFlow1.timingFrame***‘.

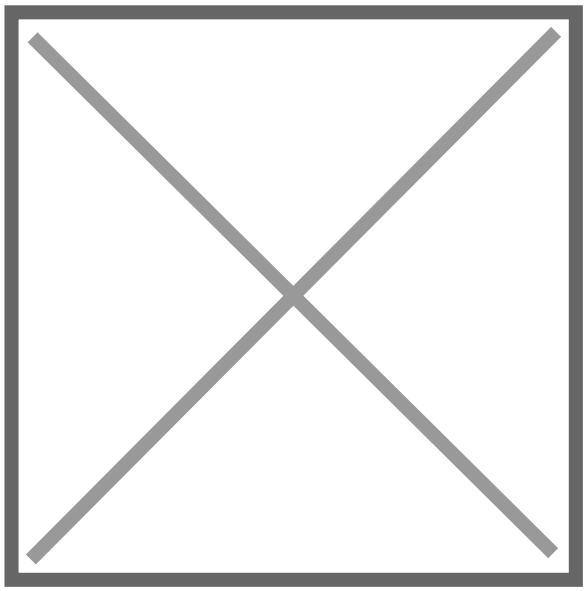
So if you apply the following expression to any other curve in your script, that animation curve will be retimed to match OFlow1:
curve(OFlow1.timingFrame)



Translated: “*For the frame I am currently on, instead of using the current value, jump to the frame the OFlow is currently looking at and use the value from this curve at that frame*“.

Offsetting a curve

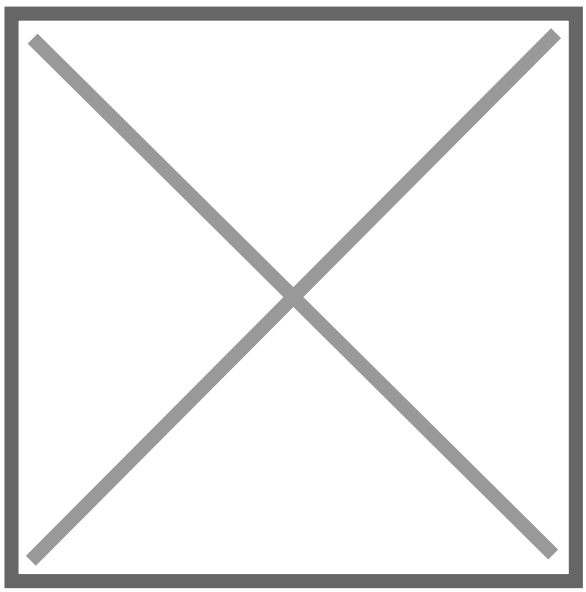
Any control that has keyframe animation, right click the curve icon and choose ‘***Edit Expressions…***‘ and you should see it has one expression in there already: ‘***curve***‘. This is just the curve created from your keyframes. You can treat that curve like a variable, for example ‘***curve*2***‘ to double the amplitude of the curve. Putting a value in brackets after it allows you to slip the curve back and forth in time. ‘***curve(frame-5)***‘ would offset the curve by 5 frames. “*At this frame, don’t use this value, use the one 5 frames earlier*“. ‘***frame***‘ is a variable that always contains the current frame number. The resulting output would look something like this:



An example script

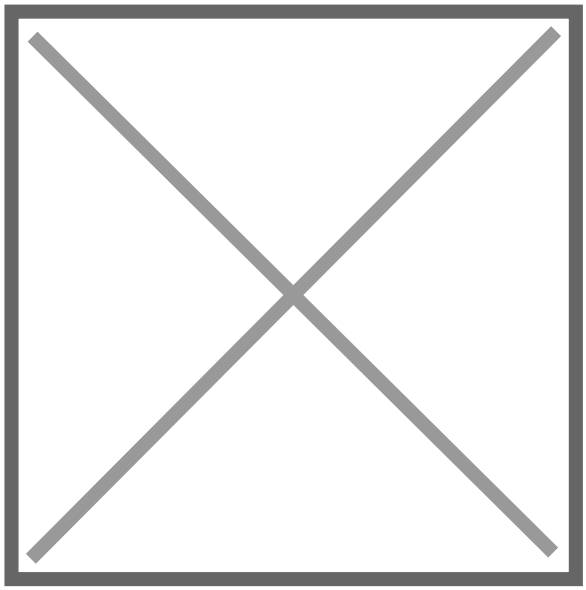
Let’s say you have a shot of a building and you want to patch in some extra windows. You don’t want to use a still because there are shadows of people moving around inside the building so you want to copy another window, translate it to the new position and then retime the input so the people moving round look less repetitive. If it was a locked off shot, you could just retime it, copy and translate it. But if there is a camera move and camera wobble it will need to be stabilized, translated then match-moved. But where do you put the retime node? It makes sense to put it after the stabilize, but then you will break concatentation with the translate and match-move nodes, potentially introducing softening. Using the above method of retiming the tracking data, you can keep the transform nodes together and keep concatentation.

In this example I’m using a CornerPin node as my stablize/match-move, exported from Mocha Pro.

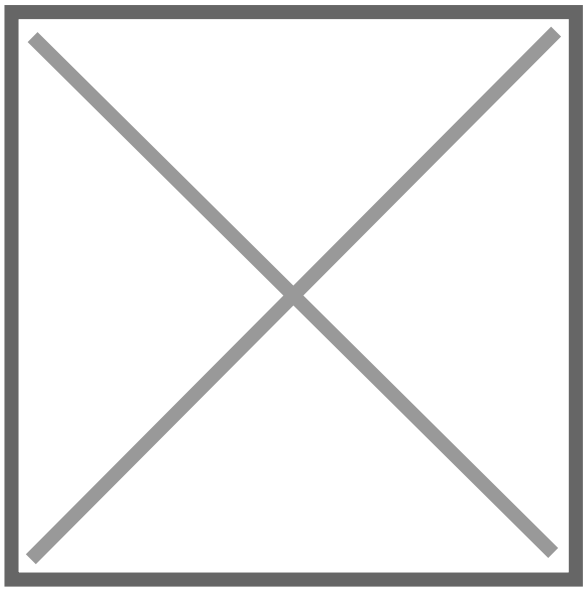


One of the patches in this example doesn’t use any retiming, and the other does. They use an identical setup but one has expressions linked to the OFlow. I have put a dot node in there so you can see the expression line more clearly. The CornerPin nodes labelled ‘stabilize’ and ‘match’ are identical but the stabilize ones have the ‘invert’ check box ticked. I could have cloned several of them but the tracking data is unlikely to change so a straight copy/paste is better here.

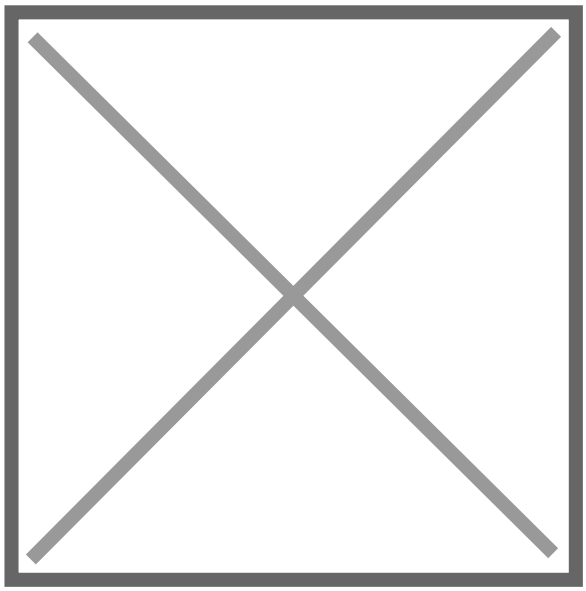
When I right click on any parameter in cornerPinTrack4 and choose ‘***Edit Expressions…***‘ I enter the above expression into each corner:



No matter how I animate my OFlow1 node, the cornerPinTrack4 node will adjust it’s tracking data to match the newly retimed plate, keeping it stabilized. Of course, if you retime it to extremes, the plate and/or tracking data will run out. In this example I have just slowed it down and sped it up again. The retime curve looks like this:



When I look at the curve for one of the values in my corner pin (the x value of corner 1) it shows the original keyframes with the newly calculated curve on top:



I drew the roto shape on my reference frame around the window I wanted to copy, whilst looking at the result of the cornerPinTrack4. I then cloned the offset transform node and put it beneath the roto shape. I could have just viewed the result of the offset transform node when I drew my rotoshape to achieve the same thing, but this way I can change my mind about the offset later if I want, moving it to a different position and the rotoshape will always be in the correct place.

To avoid grain issues you will probably want to degrain the plate and regrain the patches afterwards. Even if it doesn't get softened, you don't want grain running at different speeds.

Using a TimeWarp node instead

If you are retiming with a timewarp node, you can use the 'lookup' value instead to achieve the same thing:
curve(TimeWarp1.lookup)

Summary

This trick can be used anywhere in your nuke script; anywhere there's a keyframed knob, you can adjust the curve by adding an expression to it. In most cases retiming is done to the plate beforehand or at the end of a comp, so your matchmove will have been done on the retimed plate already, or if at the end, then it doesn't matter. But occasionally a decision will be made later on, after you have your matchmove, to retime something and it means throwing off all your animation.

But using this expression you can keep the existing animation, whether that be tracking, a 3D camera, switches, fades or transforms, and avoid precomps and other methods that might introduce softness by breaking concatenation.

And yes, I did say 3D camera… if you have a tracked camera but the plate has to be retimed, you can put this expression on every animated value in that camera (once you untick '***read from file***') and the camera will now match the retimed plate. No need to send it back to the matchmove department.

Revision #1
Created 2022-02-24 14:01:59 UTC by Anthony
Updated 2023-12-01 08:56:57 UTC by Anthony