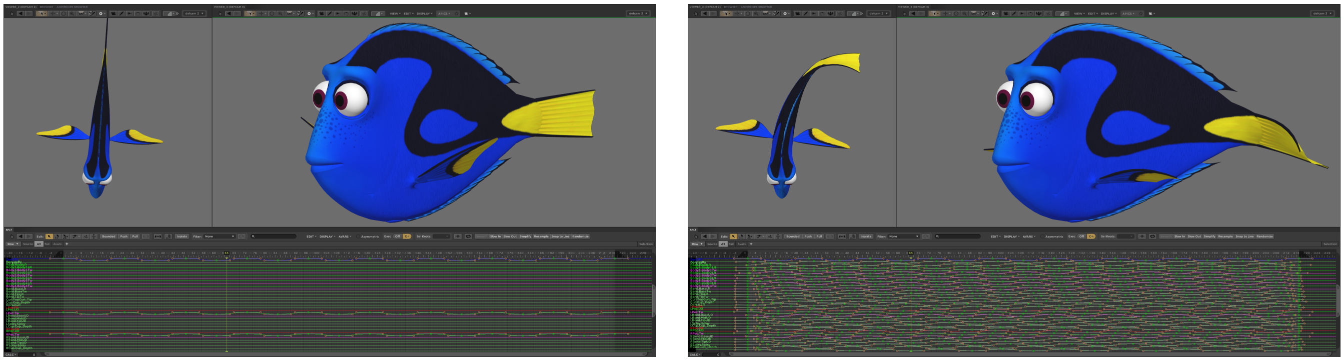
Animation Recipes

Turning an Animator's Trick into an Automatic Animation System

Chen Shen *　Tom Hahn　Bret Parker　Sarah Shen
Pixar Animation Studios



(a)　(b)

Figure 1: *Apply a swim recipe to Dory. (a) Rough blocking before the recipe is applied with only three animation variables (avars) hand animated. (b) After the recipe is applied, all detailed avars are filled in and Dory is fully animated. ©Disney / Pixar. All rights reserved.*

With a traditional key frame animation system, animators author a tremendous number of key frames across many animation variables (avars) manually. In particular, with cycles or ambient body motion, animators have a simple but powerful trick to avoid having to do this. They hand animate the avar that is driving the motion, then duplicate and transform the motion on that avar to other detailed avars throughout the body. We have generalized this trick into a system that turns rough blocking, done with a minimum number of avars, into fully detailed animation with the click of a button.

Animate Using a Recipe

When animating long articulated objects like fish bodies, dinosaur tails, tentacles, and even human spines, animators try to only animate the root joint and duplicate it into other joints along the chain with value and timing adjustments.

We have generalized this trick into a concept called an *animation recipe*, which defines how the root animation should be duplicated and transformed into other joints. A recipe can contain multiple components, called *ingredients*. Each ingredient contains one source avar and a list of detailed avars with their transformation parameters including value scale, value offset, and time shift. For example, a fish swim recipe has three ingredients, one for the tail, two for the left and right fins. The tail ingredient has the root R_z (the rotation around Z axis) as the source avar and a list of detailed avars that control each body segment including the dorsal fin to turn left and right. Recipes allow animators to encode a consistent style of motion for a particular character.

We further extended the concept by providing a way for animators to procedurally apply recipes to their shots. Applying a recipe authors the avars that can be modified interactively in two ways: either by altering the recipe or adjusting the source avars in the shot. We provide an intuitive interface to allow animators to tweak the recipe and get real-time feedback. Since the source avars and the detailed avars are internally linked, any editing to the source avars will be automatically propagated to all the detailed avars. For example, in the case of fish swimming, animators can make adjustments to the root orientation and speed, then the details on the tail and fin animation will automatically follow, without doing any extra manual work. Our animators have embraced recipes, unlike other procedural systems they have experienced, because recipes result in regular animation splines, like what they are used to working with.

In the example of an animated fish making turns, naively duplicating R_z would cause the tail and fins curl up to become undesired spiral shapes. Animators would either give up using the trick, in this case, or manually fix the spirals. The root problem is that we only want to duplicate the swim motion but not the turns. Signal processing techniques allow us separate motions with different frequencies. We implemented a low pass filter with FFT to filter out undesired motion elements before they are duplicated.

Extract Recipes from Examples

Creating a new recipe is an artistic process that requires careful design of all parameters. Each recipe is for a particular motion of a character, manually designing recipes for hundreds of characters would require an unrealistic animation budget. We have developed a tool to extract recipes from existing animations. With a simple interface, animators can easily specify source and detailed avars, then all the transformation parameters are automatically extracted by a two-step algorithm. For a pair of source and detailed avars, we first measure the time shift to best align them in time with another signal processing technique to compute their cross-correlation using FFT. Then, in the second step, we solve a 2x2 least squares fitting problem to recover value scales and offsets. When a recipe is extracted, the style of that animation is encapsulated in the recipe and can be easily applied to other shots for maximum efficiency in reaching polished animation.

[*] e-mail: csh@pixar.com

SIGGRAPH 2015 Talks, August 09 – 13, 2015, Los Angeles, CA.
ACM 978-1-4503-3636-9/15/08.
http://dx.doi.org/10.1145/2775280.2792531