Rope Bridge Animation System in *Kung Fu Panda*

Amaury Aubel, DreamWorks Animation
Sven Pohle, PDI/DreamWorks
Mitch Cockerham, Matt Steele, DreamWorks Animation

In DreamWorks Animation's upcoming film, *Kung Fu Panda*, characters fight on a rope bridge. From a production standpoint, this presents a number of challenges: the interplay between the characters and the dynamic structure of the bridge creates an animation loop, the range of motion is extreme as characters fight kung fu style, and the bridge gets severely damaged along the way.

We handled the key issue of the interdependency in the animation by building a rig that enables animators to pose the bridge at the same time they are animating other characters. Secondary dynamics are then layered on top using a procedural FX system. Corrective feedback is automatically applied to the characters-bridge constraints, thus reducing the need for manual adjustments. As a result, the whole process can be iterated easily. In addition, as the system is fully layered, animation changes at any level will automatically trickle down to finer animation layers.

Animation Rig

In order to deal with the complexity and size of the bridge, we construct a hierarchical rig: controls are grouped in levels, starting with a superset of 3 controls to give the broad swinging motion and ending with numerous local controls to animate hand grabs and other such localized effects. This arrangement is very natural for animators as they can start by painting "broad strokes" and then move on to more specific points of the animation as needed.

One of the key components of the setup is that each level controls finer detail than the previous one, so one never "fights" the animation generated by higher-level controls. Controls are also placed automatically in a local space system defined by the cascade of operations of higher levels.

Controls can slide along the splines that represent the general shape of the bridge, so the animator can increase the level of detail in a specific area by moving controls closer to each other (without changing the general shape of the bridge). Each control used for a swinging motion comes with a movable pivot point. Additional rotational controls allow a twisting motion. Posable constraint points facilitate the animation of characters on the bridge.

Display optimizations include a spline-only mode to visualize the general shape of the bridge and the ability to turn off entire sections. Color-coding of the vertical support ropes also helps animators gauge their level of stretching or compression.

To show increasing levels of destruction during the fight sequence, we rely on a variation system that allows the automatic application of the rig to different sets of geometry.

Adding Dynamics

We use a Maya-based, two-stage procedural system to add secondary dynamics. From the animation rig, we derive a polygonal cage with a U-shaped cross-section (or a ribbon-shaped cage in the case of the damaged variant). We deform this proxy geometry and ultimately transfer the animation back to the actual model using standard wrap and wire deformers.

Dynamics are simulated using essentially a combination of damped harmonic equations. The animator controls impulses that trigger the propagation of exponentially decaying sine waves down the bridge. Each wave is defined by a set of parameters including the velocity, the maximum travel distance, the frequency of the induced oscillations, their amplitude and decay rate. Most controls are also augmented with multipliers, so one can overlap harmonic oscillations at various frequencies easily. Waves can get absorbed when they reach the bridge end, or get transmitted from one rope to the other via planks in the damaged bridge rig. Impulses can be positioned anywhere and at any time to create various types of oscillations: vertical or rolling oscillations of the planks, independent or constrained vertical oscillations of the hand ropes, rolling of the hand ropes around the bottom planks, etc. Various types of noise and the ability to freeze any given part of the bridge complete the system.

In a second stage, we procedurally apply some follow through animation to dangling broken planks. The resulting animation is embellished with additional oscillations created by special impulses that only affect the broken bits. Vertical support ropes are also animated at this stage using a dynamics cloth solver, constraining both ends to the top and bottom ropes.

Finally, we automatically adjust the original animation of the characters by registering and tracking on the FX bridge the joints used for constraining the characters. The quality of the adjustment mostly depends on the complexity of the animation and constraints.

Conclusion

We developed a fully hierarchical animation system suitable for an iterative process: if one moves the bridge globally, the animation created by finer levels, all the way down to the vertical support ropes simulation, automatically follows. Simulating the dynamics with mostly procedural methods also helped greatly in production: unconditional stability, real-time feedback, readily customizable procedural code, and ease of retiming were key elements.