Animation and Simulation of Octopus Arms in 'The Night at the Museum 2'

Matt Derksen*
Rhythm and Hues Studios

Tae-Yong Kim†
Rhythm and Hues Studios.

Overview

We present a set of techniques we used to animate flexible arms of the monstrous octopus character in 'The Night at the Museum 2'. For this production, we adopted layered approaches, where key motions of the arms were animated with conventional methods by animators, while details were *cleaned up* by multiple passes of simulations of skeleton, flesh and tentacles.

Smart Tentacles

For simulations, we used a modified version of our in-house cloth simulators. In this modification, each arm was simulated as a cloth strip and this cloth object was attached to the original animation with soft constraints (by springs). This way, the cloth simulation would roughly follow the arm animation scripted by animators.

The next pass was to give a bit of *brain* to the cloth control. In this 'smart tentacle' approach, the simulated cloth follows user-guided rest shapes. Especially, the bending energy term would constantly track the user-supplied rest shapes in terms of bending angles. This gave illusion of the cloth object behaving *actively* to the environment. For example, users could supply an abrupt bending motion in the guide geometry, and the resulting cloth simulation would react to the environment as usual (such as collision with ground, etc.), but also reflect the bending motion at the same time.

This way, the overall motion of arms could be entirely controlled by the artists (by key framed animation + the smart tentacle), but simulations would clean up any physically invalid state (such as penetration into collision object) of the original animations.

The next step was to use an 'envelope' geometry to represent the volume surrounding the cloth skeleton. This was made as another pass of volumetric softbody simulations, not very differently from the cloth system.

In layered simulations, users decided whether to simulate the whole set of arms in one step, or in multiple passes. For example, users would simulate one arm first and lock the motion. Then they would simulate another arm with the first arm as a pure collider. We found that users usually prefer the latter option. Figure 1 shows examples of simulated arms interacting with each other after multiple passes of such simulations.

Sticky Collision

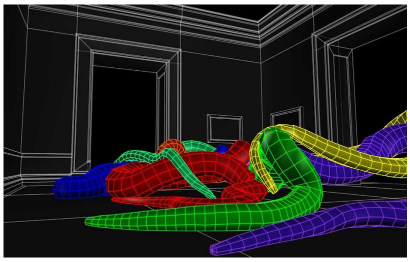

Figure 1: *Envelope geometry displayed in distinct colors for each arm.*

The final step was to attach 'suckers' sitting on the arms (Figure 2). We represented each object with simple triangles, and simulated many of those triangles at the final stage of simulations.

To mimic sticking and abrubt releasing behavior of the tentacle surface to/from the ground, we developed some unusual friction handling model in the collision handling process. Typically, friction is considered to exert continuous resistance to the tangential motion of the collision surface - we did not like it in our case. Instead, we modeled the friction as dynamically attached/detached springs between the collider and simulated object. Users would then specify some 'releasing conditions' such as build-up of the spring energy, distance from the original attachment point, etc. This could represent the initial sticking of the simulated object as well as abrupt releasing when simulated objects moved far away from the collider.

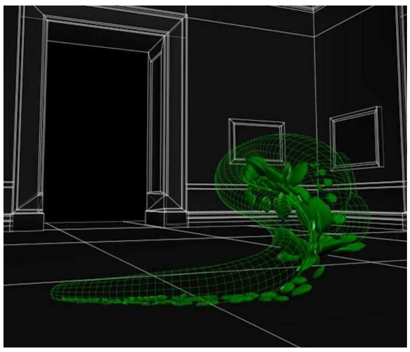

Figure 2: *Simulation of sticky tentacles.*

*e-mail: mderksen@rhythm.com
†e-mail:tae@rhythm.com