# Bulging Muscles and Sliding Skin: Deformation Systems for Hellboy

William Todd Stinson
Tippett Studio

Paul G. Thuriot
Tippett Studio

For the film, ***Hellboy***, we needed to create full ranges of realistic (and sometimes very unrealistic) motion and deformations. The demands for the show required a fast solving, semi-interactive, and physically-based deformation system that fit into our existing animation pipeline.

## 1. Background

The system needed to be a layered system, simulating physical properties of muscle, fat, and sliding skin. It also needed to be interactive enough to give the animators instant feedback for pose deformations and to have a considerably short, off-line solve time. This was realized by two separate **Maya** plug-ins that could be used individually or in combination: **tipMuscleBuilder** and **tipSkinBuilder**.

## 2. Building Muscle

For underlying structures, curve-based, NURBS muscle surfaces are built by stating where the surfaces attach to the underlying anatomy of a creature in a relaxed pose. The muscle is then shaped via attributes to fit the anatomical needs. Volumetric attributes are turned on, storing the base volume at a relaxed pose, and any deformation to the underlying structure drives the muscle. These deformations of muscles can be viewed in real-time inside of the animator's scene file for instantaneous playback for posing purposes.

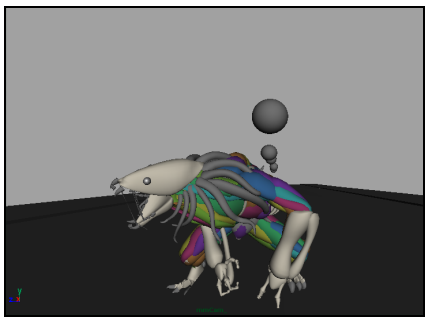

Fig 1. Muscle internal workings.

Muscle tension can also be simulated via artistic, *isometric contractions*, prior to muscle deformations, using animation information from skeletal structures.

Because of the plug-ins integration into **Alias' Maya** software, any of **Maya's** feature sets can be applied such as soft-body dynamics, or particle collisions for extra added jiggle.

## 3. Simulating Skin

The skin plug-in is applied to the rendered surface(s) model. Using any collection of NURBS objects as collision surface(s), the skin plug-in finds and fills in the empty voids as a **Fat Layer** between skin points and underlying structures by multiple connection processes. Numerous attribute values are then "painted" for each skin vertex, stating what dynamic forces are to be applied on and under the skins' surface during simulation.

Fig. 2. Transparent skin stretched over below surfaces.

Due to advanced facial animation or no need for dynamically solving certain areas of characters (i.e. hands and feet), any skin point can be turned into a **Bound Point**, meaning it is driven by another deformed object and not solved. These **Bound Points** do apply forces to the neighboring skin solved points however.

Two skin solvers were finalized for the project. The **Dynamic Solver** which adjusts skin vertices by each point's velocity, underlying masses, and other specified forces. And the **Static Solver** which relaxes skin points after first being moved by the connected, underlying structure. With small visual difference between the two, an animation sequence using the **Static Solver** could be run in *parallel* on the solve farm, solving any frame of an animation without needing to solve the previous one.