DrivenShape - a data-driven approach for shape deformation

Tae-Yong Kim
Rhythm and Hues Studios

Eugene Vendrovsky†
Rhythm and Hues Studios.

, which can be solved with the standard quadratic programming (QP) techniques [Goldfarb and Idnani 1983]. Equation 6 ensures that we don’t get artifacts from negative weights.

Database Construction

We provide a tool to automatically extract N most distinct shapes based on a greedy process. We expand the set of pose shapes by adding at each step a shape that’s the most different (by Euclidean measure) from all the shapes contained within the set. After N iterations, we have a set of N shapes that are sufficiently different from each other. We also allow users to directly pick the pair of shapes they want to add. Usually, users start with small number of automated picks and add to the database as they wish.

Collision Detection

For highly deformable characters, a simple linear blend would not match the target shape (in general, \( P_{\text{new}} \neq \sum w_i \cdot P_i \)). One noticeable artifact is a deformation that intersects with the pose shape. Let \( P_{\text{new}} = \sum w_i \cdot P_i \) and \( D_{\text{new}} = \sum w_i \cdot D_i \). After weights are computed, we apply additional mapping \( P_{\text{new}} \rightarrow P_{\text{new}} \rightarrow D_{\text{new}} \) as follows. For each point \( d \) in \( D_{\text{new}} \), we find the closest triangle from \( P_{\text{new}} \) and construct a coordinate system on the closest point with one axis being the normal, and another axis being one edge of the triangle. Local coordinate of \( d^{\text{local}} \) is then computed, and used to reconstruct final position \( d^{\text{final}} \) after we move points of the triangle to \( P_{\text{new}} \) and update the coordinate system.

When two geometries are close (e.g. legs crossing each other), the closest triangle can come from wrong side and cause popping artifacts. We let users supply additional mapping to exclude unnecessary binding (e.g. left pants maps to left leg only).

Secondary Motion and Layering

Since output is directly mapped from the pose, we lose secondary motion that was contained in the original animation. When secondary motions are desired, we turn back to regular simulations, but users still use DrivenShape to guide the simulation (e.g using spring constraints) or to rapidly provide the initial draping of cloth. It also seamlessly works with additional deformations such as noise-based wind effects. Often users partition the geometry and apply DrivenShape to more rigid part, and simulate more dynamic part (such as hood of the sweater) with cloth simulator.

Conclusion

For 20-30 shape pairs with 5000 vertices, the system runs in real-time, providing rapid feedback for animators. This technique was extensively used in our recent production of Alvin and the Chipmunks, and we could eliminate the need for expensive cloth simulations for about 70 percent of the shots. Although it was originally developed to speed up the cloth simulation pipeline, users have expanded its use to anything that deforms such as muscles, facial structure, etc.

References