Interactive Script Based Dynamics in Big Hero 6

Dong Joo Byun* Zubin Wadia Michael Kaschalk
Walt Disney Animation Studios

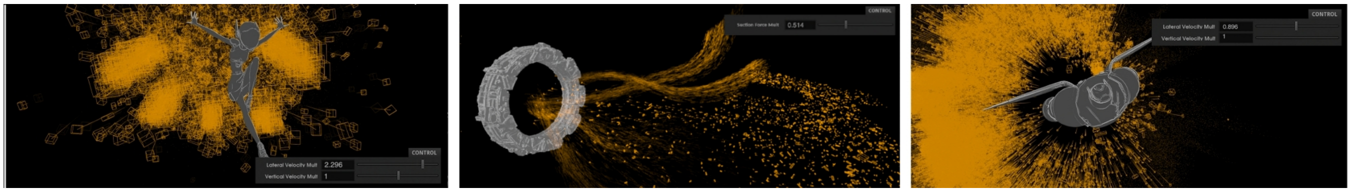

Figure 1: *Interactive Script Based Dynamics*

1 Introduction

In production environments, dynamic effects have been commonly done by rigid body simulations or particle simulations. In Big Hero 6, we set up a directable and very efficient rig for dynamic effects. We used motion equations in VEX script to calculate and control the dynamic terms of each geometry piece without using solvers.
This rig was implemented in a history independent procedural animation context and it provided artists more room for art direction and fine tuning. In most cases, we could control the motion in real time. If necessary, we used parts of the geometry pieces for motion design, in order to maintain the interactivity.
This rig could be used mostly for explosive motion where there are not collisions. But for shots where collisions are needed, we connected our rig into the rigid body solver, first to design the motion in the rig, and next to calculate accurate collisions in the simulation.
Since all physical properties are pre defined in a continuous time space, a huge benefit of our rig was that we could get data of any float number of frame, unlike simulation where more expensive sub step solving to get the sub frame data is needed.

2 Usage Analysis

When running particle or rigid body simulations, even though solvers provide accurate motion and collisions, in order to get the proper outputs, it often requires many iterations of expensive simulations. This makes the work flow slow and less interactive, and inhibits artists working intuitively.
For fast explosively dispersing motion, especially at the early stage of it, we can't clearly see the collisions among debris pieces since they would have extremely stretched motion blur. Ironically, the most time taking process in this type of dynamic simulation is the collision detection where there are a lot of packed geometry pieces at this stage. The initial position and velocity of the explosive motion determines the overall composition and shape of dynamic effects.
From this analysis, we set up a rig with which we can focus more on designing the shape of the explosive motion than on struggling to get accurate, yet unnecessary time consuming collision calculation. We proposed a procedural animation rig which satisfies this requirement.

* e-mail: dong.joo.byun@disneyanimation.com

SIGGRAPH 2015 Talks, August 09 – 13, 2015, Los Angeles, CA.
ACM 978-1-4503-3636-9/15/08.
http://dx.doi.org/10.1145/2775280.2792534

3 Motion Equations and Rotation Setup

We adopted motion equations in order to calculate position and velocity from initial velocity and uniform acceleration.

$$r = v_0 t + \frac{1}{2}at^2 \quad (1)$$
$$v = v_0 + at \quad (2)$$

The rotational motion was derived from the initial rotational velocity. It is defined by customized rotation axis and angle which is linealy increased by time.

4 Rig Setup

Input into the system is fractured geometry or points which would be instanced by geometry at the render time. Not being a point type input, geometry is packed per piece since the dynamics works per piece and we need very light representitive geometry type. Attributes are applied per geometry piece. Release frame attribute is the frame where the dynamic motion is triggered. The difference between current frame and the release frame is the age of the piece. Initial velocity could be customized and designed by different methods. Rotation set up is used to generate tumbling motion of the pieces. With all assigned attributes, motion equations calculate position and velocity. Velocity and angular velocity are calculated either by motion equations or by seperate post process.
We could either directly render the animated geometry after unpacking, or instance geometry pieces onto points and render them.

Figure 2: *Script Based Dynamics Rig Setup*

5 Expansion to Rigid Body Simulation

We connected this rig to rigid body simulations when necessary. Each of the geometry pieces were added into a rigid body simulation, which contained position, velocity, and angular velocity derived from our rig. Since the geometry pieces would have been already seperated out from other pieces and moving directions have been shaped, collision detection was not very slow and overall composition could be very predictable. This combination process allowed us to get very art directed and physically correct output with a lot less and faster iterations.

6 Conclusions

The interactive script based dynamics is a very efficient substitue approach for dynamic effects. It provided artists a lot more intuitive, refined and interactive work flow. We could dramatically increase the performance of dynamic effect tasks in the ***Big Hero 6*** production. It was very beneficial that we could design and modify dynamic effects by handling millions of geometry pieces, interactively in the viewport.