Dynamic Animation and Control Environment (DANCE)

Victor Ng Thow Hing
Petros Faloutsos
University of Toronto
victorng@dgp.toronto.edu

Physics-based animation is increasingly being used to create realistic, autonomous animations. Although commercial systems have been developed that produce passive dynamics, lifelike creatures must have their internal forces and torques for locomotion actively controlled. Other simulation systems are applicable to only one type of object representation like articulated figures or deformable cloth.[1]

Control schemes have been developed in different customized simulation environments.[2] To remedy the current fragmentation of simulation systems and control schemes, we have developed a software framework called DANCE (Dynamic Animation and Control Environment). DANCE provides a mechanism to easily integrate different controller schemes and simulate diverse object representations together. With this environment, new practitioners can quickly focus on new techniques without investing large amounts of time to build support infrastructure.

Base Classes

We identify a minimal set of object-oriented abstract base classes that capture many features common in existing simulation systems. The small number of different classes offers a clear perspective of the components of a physical system. Specialized features are added by creating subclasses of each base class through dynamically loaded plug-ins. A central driver orchestrates the communication and management of the plug-ins.

A system describes an entity that can undergo physical change of state. In general, the system class can represent any physical system that can be described with a set of generalized coordinates. We have implemented articulated figures, deformable parametric surfaces, and particle systems. The geometry class provides graphical representations for systems.

Actuators are entities that exert forces or torques on systems. Gravitational fields, rigid body collision, and biomechanical muscles can be modeled as actuators. Actuators can manage other actuators to form complex hierarchical controllers.

Simulators update the state of systems over time, using either physics-based or kinematic processes. In DANCE, the degrees of freedom of all systems can be handled by different simulators. Conversely, simulators can be assigned to individual systems or groups of systems.

DANCE employs application programming interfaces (APIs) that are available on all major systems: OpenGL for 3D graphics, Tcl/Tk for the scripting and graphical user interface, and GLUTfor window management and input event handling. It has been ported to Windows NT, Linux, and Irix. The GUI is completely external to the DANCE system. Therefore, different applications can have customized interfaces.

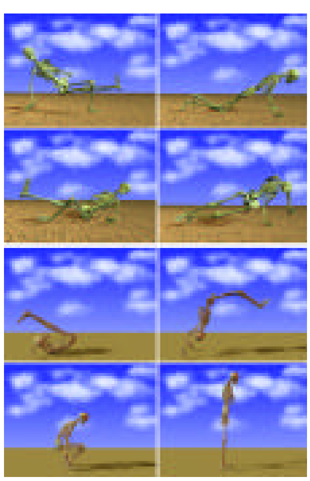
Applications

We have used DANCE to implement two significantly different research projects: a biomechanical, musculoskeletal simulation system featuring deformable musculotendons and ligaments as actuators and a software control system for composing a diverse array of different motion controllers for autonomous and versatile virtual stuntmen. We believe that DANCE provides a useful environment for prototyping physical systems and controllers, and sharing results.

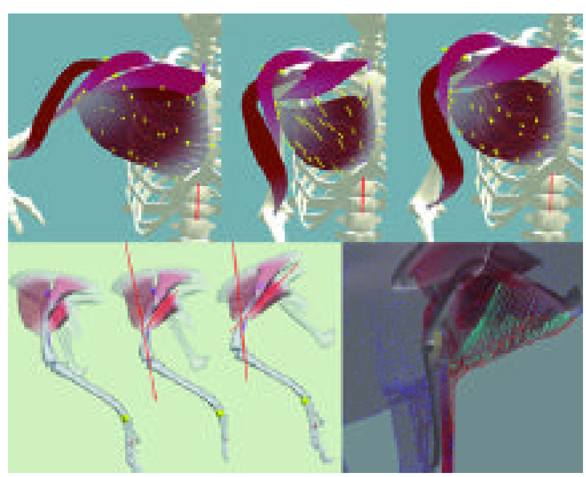
DANCE is being actively enhanced and is available at:
www.dgp.toronto.edu/DGP/software/dance/dance.html

References

1. Baraff, D. & Witkin. (1998). A. Large steps in cloth simulation. In *Computer Graphics (Proc. SIGGRAPH 98)*, 43-54.
2. Hodgins, J. K., Wooten, W. L., Brogan, D. C., & O'Brien, J. F. (1995). Animating human athletics. In *Computer Graphics (Proc. SIGGRAPH 95)*, 71-78.



A physics-based human responding to pushes and performing a stunt.



An anatomically based modeler that allows muscle, ligament, and skin simulations.