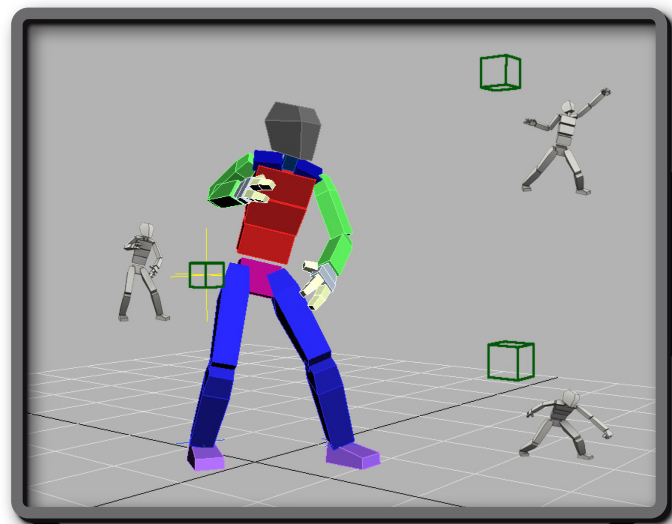
Interface Techniques for 3D Control of Spatial Keyframing

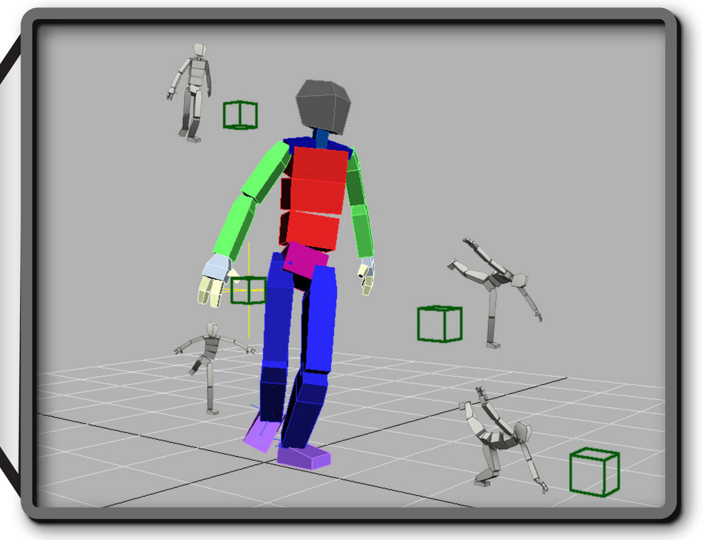
Igor Mordatch Patrick Coleman
Karan Singh Ravin Balakrishnan

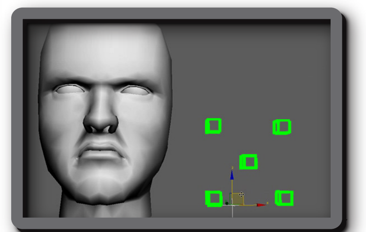
University of Toronto

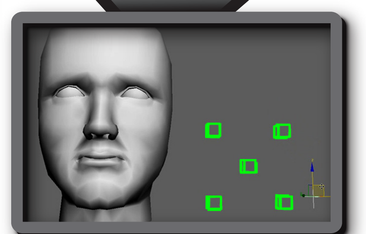
igor.mordatch@utoronto.ca, patrick@dgp.toronto.edu,
karan@dgp.toronto.edu, ravin@dgp.toronto.edu

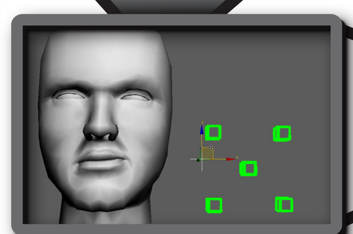
Spatial Keyframing is an approach to interactive character animation in which users lay out character poses in an abstract, low-dimensional control space. Cursor position in that low-dimensional space is interpreted as a weighted interpolation among poses, and cursor motion then defines a pose-to-pose style animation that the user can interactively create. As originally presented, spatial keyframing uses mouse motion in a two dimensional control space, although the underlying algorithms apply to control spaces of arbitrary dimension. As part of an ongoing project, we are investigating how support for three dimensional pose layouts can be made easier with software tools to assist with spatial navigation, as well as with true 3D interfaces for cursor navigation and visualization. Extending the layout to the third dimension allows users to better organize larger collections of poses, which simplifies the creation of complex motions with multiple actions and stylistic variation. With our 3D interfaces, the user acts as a virtual puppeteer. Hand motion in the space surrounding a 3D volumetrically-rendered character creates character motion among the target poses, and a particular pose layout acts as a motion control rig for a specific performance.

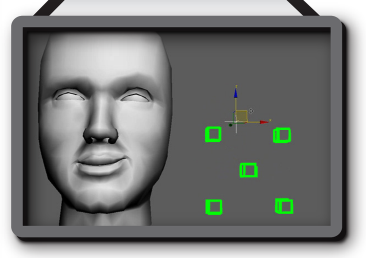












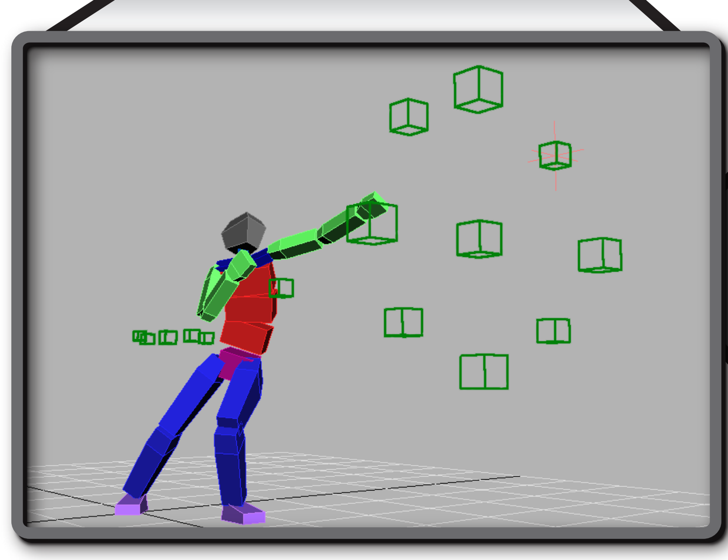
What is Spatial Keyframing?

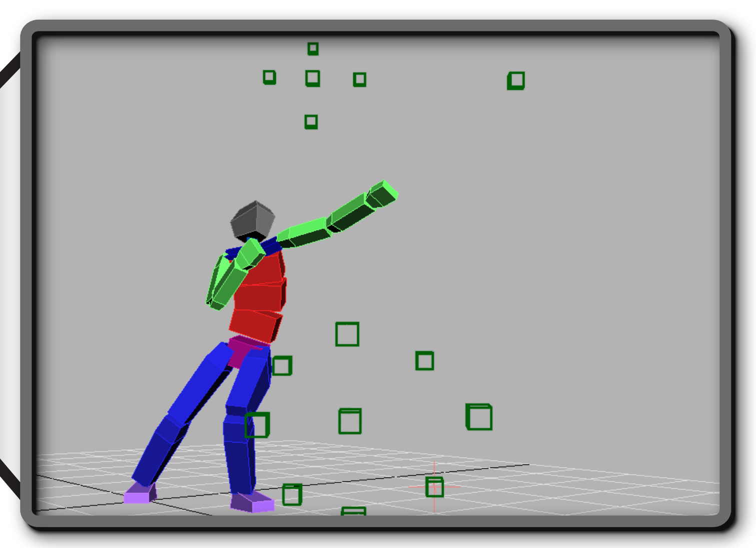
In Spatial Keyframing, developed by Takeo Igarashi and colleagues, user manipulation of a cursor with the mouse specifies a changing interpolation among a set of poses. Each pose has an associated target icon; users lay these out in an abstract control space. The cursor's position in the control space defines a continuous multi-target interpolation among the poses, which is implemented with radial basis function sparse data interpolation. In our work, the user positions the target icons in a 3D control space, which we overlay with the 3D character display. We display these target icons as green wireframe cubes in the images shown here, which include an articulated character as well as a human face model. Using a mouse to navigate among target icons laid out in 3D is fairly difficult, however, and we have developed tools to assist users with arranging targets in 3D with traditional interfaces, described in the box below. Further, our 3D interfaces, described to the right, allow for full navigation among targets laid out in 3D. This allows users to create complex pose collections that can model multiple actions or stylistic variation.

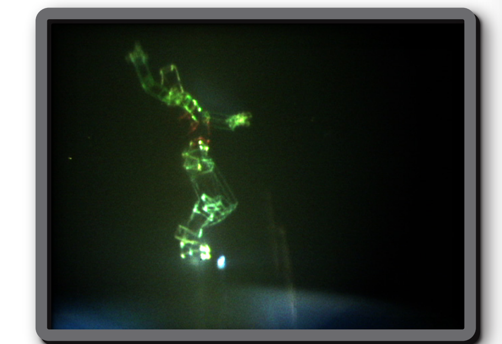
For more information, on the algorithms behind Spatial Keyframing, see the original work: "Spatial Keyframing for Performance-Driven Animation." Takeo Igarashi, Tomer Moscovich, and John Hughes. *Proceedings of SCA 2005*.

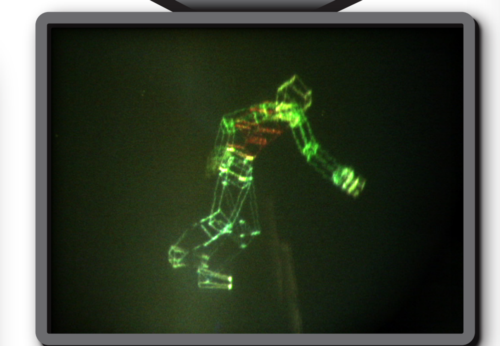
Traditional Interfaces with 3D Spatial Keyframing

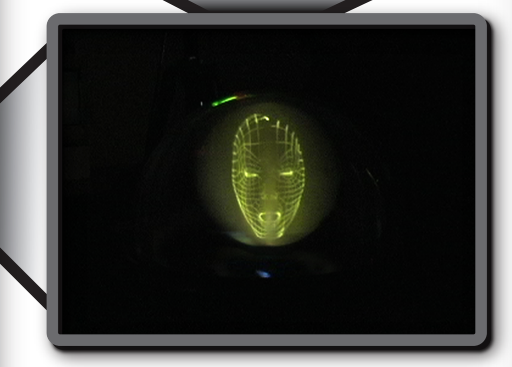
A mouse or other traditional manipulation device cannot fully navigate a 3D control space, as the device does not have sufficient degrees of freedom to fully specify 3D cursor location. When using such devices, we resolve the ambiguity by constraining cursor navigation to a constant depth relative to the current view. We have built our system as a plugin to Autodesk's 3ds Max animation system, which allows the user to retain full camera navigability to choose a desired view. However, 3D target layouts can be ambiguous to navigate among from a given view with the 2D constraint, as targets can overlap. We allow the user to reposition the targets as a group, or as a collection of groups, to increase the overall navigability. For any given collection of targets, we align the two eigenvectors of a Principal Components Analysis representation that have the two largest associated eigenvalues to the screen space coordinate system. This has the effect of providing the view of the targets that maximizes navigability. When creating complex motions, we have found it advantageous to cluster targets into action-specific groups as we create poses. Navigating from one logical cluster to another corresponds to a change in character action, and navigation among the poses in that group leads to the performance of that action. To retain this clustering in the new layout, we optionally cluster the targets using K-Means, align each group independently, and spread the groups in screen space. An example is shown in the images below. Three groups are obtained from the clustering (one has the single rest pose), and they are repositioned independently, maximizing navigability for each cluster while retaining the global cluster-like organization.











3D Control and Visualization

To fully exploit the 3D dimensional control space of our system, we have incorporated 3D navigation and visualization interfaces, allowing the user to act as a virtual puppeteer. We display the character using a 3D volumetric display from Actuality Systems, described below. Photographs of displayed characters are shown in the images at right. We optically track physical manipulators held by the user using a Vicon motion capture system, and we set up a control space that the user can navigate by positioning the manipulators in the space surrounding the dome, shown above. Targets are arranged in real world locations, and the user can view this layout relative to the display volume from the 3ds Max interface. Hand motion around the display, among the different target locations, allows the user to perform with the character as a virtual puppet. As we can track two manipulators, we can assign each to a separate control space with its own set of targets. We can separate the character's degrees of freedom to allow for independent user control of different portions of a character. We are currently investigating the use of this approach to independently control speech and expression for facial animation.

Our volumetric display creates a real-world 3D rendering by rotating a semi-transparent 2D screen around a vertical axis through the center of a glass dome. At each of 198 orientations, the screen displays a 768 x 768 image, changing as it rotates to maintain a volumetric frame rate of 24 frames per second. An OpenGL-like interface allows us to produce volumetric wireframe drawings, and we have developed an interface to render 3D mesh data in 3ds Max to the display. For more information, visit http://www.actuality-systems.com.

Ongoing Work

We are currently investigating the creation of a variety of complex character motions with stylistic variation, such as dance, locomotion, and jumping. We also plan to add support for direct manipulation of the character with the 3D interface, which would provide more immediate control to the user . A major theme looking forward is the representation of motions in the control space to support editing for interactive playback. If we consider each frame as a pose, we can algorithmically position these poses to embed a motion into the control space. The images at right show embeddings based on PCA and motion trails. Any curve in control space represents a motion, and the original motion is a curve connecting all the poses. We can edit a motion into multiple variations by choosing important frames to change. We position a target for the variation frames close to, but not on, the original path, and edit the poses to specify the variation. Displacement mapping can be used to fully form the path, and this then creates an interpolatable space of motions. We can use this for constraint-based editing, an example of which is shown below. The character's motion has been edited by adding a new pose in control space that causes him to make contact with the new bulls-eye.

Acknowledegments

We would like to thank Autodesk for the donation of 3ds Max. We are grateful to John Hancock for his photography and video assistance.